

# **Data Offloading and Traffic Steering in Device-to-Device and Heterogeneous Wireless Networks**

Thrasyvoulos Spyropoulos

September 21, 2018



# Contents

<b>List of Tables</b>	<b>5</b>
<b>List of Figures</b>	<b>7</b>
<b>1 Extended Summary</b>	<b>10</b>
1.1 Introduction . . . . .	11
1.1.1 Wireless Data Offloading . . . . .	12
1.1.2 Load Balancing and Traffic Steering . . . . .	12
1.1.3 Mobile Edge Caching . . . . .	13
1.2 Thesis Contributions . . . . .	14
1.2.1 Resource Management and QoS Provision in Delay Tolerant Networks (Chapter 2) . . . . .	14
1.2.2 Performance Modeling for Heterogeneous Mobility and Traffic Patterns (Chapter 3) . . . . .	17
1.2.3 Complex Network Analysis for Opportunistic Device-to-Device Networking (Chapter 4) . . . . .	20
1.2.4 WiFi-based Offloading (Chapter 5) . . . . .	24
1.2.5 User Association in Heterogeneous Wireless Networks (Chapter 6) . . . . .	27
1.2.6 Mobile Edge Caching (Chapter 7) . . . . .	30
<b>I Data Offloading Through Opportunistic Device-to-Device Networking</b>	<b>37</b>
<b>2 Resource Management and QoS Provision in Delay Tolerant Networks</b>	<b>38</b>
2.1 Buffer Management for Best Effort Traffic . . . . .	39
2.1.1 Motivation . . . . .	39
2.1.2 Related Work . . . . .	39
2.2 GBSD . . . . .	40
2.3 HBSD . . . . .	42
2.4 Simulation Results . . . . .	43
2.4.1 Experimental Setup . . . . .	43
2.4.2 Performance evaluation for delivery rate . . . . .	44

2.4.3	Performance evaluation for delivery delay . . . . .	45
2.5	Buffer Management for Traffic Classes with QoS Requirements . . . . .	46
2.5.1	Performance Evaluation . . . . .	48
2.6	Implementation and Extensions . . . . .	50
2.6.1	Statistics Collection . . . . .	50
2.6.2	Generalized Inter-Contacts . . . . .	50
2.6.3	Messages with Heterogeneous Popularities . . . . .	50
2.7	Proof of Theorem 2.1 . . . . .	51
<b>3</b>	<b>Performance Modeling for Heterogeneous Mobility and Traffic Patterns</b>	<b>53</b>
3.1	Performance under Heterogeneous Mobility . . . . .	54
3.1.1	Heterogeneous Contact Rates over Random Connectivity Graphs	54
3.1.2	An Upper Bound for Generic Contact Matrices . . . . .	59
3.2	Performance under Heterogeneous Traffic . . . . .	60
3.2.1	Communication Traffic . . . . .	60
3.2.2	Communication Traffic Heterogeneity: Model and Implications	62
3.2.3	A Case Study . . . . .	64
3.2.4	Extensions . . . . .	66
<b>4</b>	<b>Complex Network Analysis for Opportunistic Device-to-Device Networking</b>	<b>72</b>
4.1	Introduction . . . . .	73
4.2	Contact Aggregation: Preliminaries . . . . .	75
4.2.1	Density-based Aggregation . . . . .	76
4.3	Performance Sensitivity to Aggregation . . . . .	76
4.4	Inferring The Optimal Aggregation Density Online . . . . .	78
4.4.1	Optimal community structure of the social graph . . . . .	79
4.5	Robust Clustering of Similarity Values . . . . .	81
4.5.1	Spectral Analysis . . . . .	82
4.5.2	Modularity Function Q . . . . .	83
4.5.3	Online optimal density tracking algorithm . . . . .	84
4.6	Performance Evaluation . . . . .	84
4.7	Complex Network Analysis of Human Mobility . . . . .	85
4.7.1	Mobility Modeling . . . . .	90
4.8	DTN Meteo: A Generic Framework for Predicting DTN Protocol Performance . . . . .	91
4.8.1	Solution Space . . . . .	92
4.8.2	Exploring the Solution Space . . . . .	94
4.8.3	Modeling a Local Optimization Algorithm . . . . .	95
4.8.4	Modeling Heterogeneous Node Contacts . . . . .	96
4.8.5	A Markov Chain Model for Distributed Optimization . . . . .	101
4.8.6	DTN-Meteo's Realism-Tractability Trade Off . . . . .	102
4.9	Analyzing DTN Optimization Algorithms . . . . .	103
4.9.1	DTN-Meteo for Greedy Algorithms . . . . .	104
4.9.2	DTN-Meteo for Randomized Local Search . . . . .	107
4.10	Applications to Communication Algorithms . . . . .	110

4.10.1	Utilities for Routing and Content Placement . . . . .	111
4.10.2	Traces and Mobility Models . . . . .	113
4.10.3	Measuring the Accuracy of DTN-Meteo . . . . .	115
4.10.4	What Influences the Accuracy of DTN-Meteo? . . . . .	116
4.10.5	The Practical Value of DTN-Meteo . . . . .	118

## **II Network Optimization in Heterogeneous Wireless Networks** **120**

<b>5 WiFi Offloading</b>	<b>121</b>
5.1 Introduction . . . . .	122
5.2 Analysis of Delayed Offloading . . . . .	123
5.2.1 Performance of WiFi queue . . . . .	125
5.2.2 Low utilization approximation . . . . .	131
5.2.3 High utilization approximation . . . . .	133
5.3 Performance evaluation . . . . .	134
5.3.1 Validation of main delay result . . . . .	134
5.3.2 Validation of approximations . . . . .	136
5.3.3 Variable WiFi rates and non-exponential parameters . . . . .	136
5.3.4 Delayed offloading gains . . . . .	137
5.4 Optimizing Delayed Offloading . . . . .	138
5.4.1 Optimization problems . . . . .	139
5.4.2 Optimization evaluation . . . . .	141
5.5 Related Work . . . . .	142
5.6 Conclusion . . . . .	144
<b>6 User Association in HetNets</b>	<b>145</b>
6.1 Introduction . . . . .	146
6.2 System Model and Assumptions . . . . .	148
6.2.1 Traffic Model . . . . .	148
6.2.2 Access Network . . . . .	149
6.2.3 Backhaul Network . . . . .	152
6.3 User-Association for Provisioned Backhaul Networks . . . . .	154
6.3.1 Optimal Split UL/DL User Association . . . . .	156
6.3.2 Optimal Joint UL/DL User Association . . . . .	158
6.3.3 Iterative Algorithm Achieves Optimality . . . . .	159
6.4 User-Association for Under-Provisioned Backhaul Networks . . . . .	160
6.4.1 Optimal Split UL/DL User Association . . . . .	160
6.4.2 Optimal Joint UL/DL User Association . . . . .	164
6.4.3 Iterative Algorithm Achieves Optimality . . . . .	164
6.5 Simulations . . . . .	165
6.5.1 Provisioned Backhaul . . . . .	166
6.5.2 Under-provisioned Backhaul . . . . .	167
6.6 Conclusions and Future Work . . . . .	174

<b>7</b>	<b>Mobile Edge Caching</b>	<b>175</b>
7.1	Delay-tolerant Edge Caching . . . . .	178
7.1.1	Content Access Protocol . . . . .	178
7.1.2	Optimal Cache Allocation . . . . .	179
7.1.3	Performance Evaluation . . . . .	182
7.1.4	Extensions . . . . .	185
7.2	Low Cost Caching for Streamed Video Content . . . . .	187
7.2.1	Problem Setup . . . . .	188
7.2.2	Low Density Regime . . . . .	190
7.2.3	Generic Density Regime . . . . .	193
7.2.4	Performance Evaluation . . . . .	196
7.2.5	Caching Strategy Evaluation . . . . .	196
7.2.6	Mobile vs. Static Helpers . . . . .	198
7.3	Soft Cache Hits: Impact and Optimization of Related Content Recommendation . . . . .	199
7.3.1	Introduction . . . . .	199
7.3.2	Problem Setup . . . . .	203
7.3.3	Soft Cache Hits . . . . .	204
7.3.4	Single Cache with Soft Cache Hits . . . . .	205
7.3.5	Optimal SCH for Equal Content Sizes . . . . .	206
7.3.6	Optimal SCH for Different Content Sizes . . . . .	207
7.3.7	Femtocaching with Related Content Recommendation . . . . .	209
7.3.8	Femtocaching with Related Content Delivery . . . . .	211
7.3.9	Performance Evaluation . . . . .	213
7.3.10	Datasets of Content Relations . . . . .	213
7.3.11	Simulation Setup . . . . .	214
7.3.12	Results . . . . .	215
7.3.13	Related Work . . . . .	220
7.4	Coordinated Caching and Communication in 5G: Vision and Initial Results . . . . .	221
7.5	Appendix . . . . .	225
7.5.1	Proof of Lemma 13 . . . . .	225
7.5.2	Proof of Lemma 14 . . . . .	226
7.5.3	Proof of Lemma 16 . . . . .	227
	bibliography	229

# List of Tables

2.1	Notation . . . . .	41
2.2	Simulation parameters . . . . .	44
2.3	Taxi Trace & Limited buffer and bandwidth . . . . .	44
2.4	ZebraNet Trace & Limited buffer and bandwidth . . . . .	44
3.1	Relative Step Delay Error $RE_k$ : Averaged over All Steps and over 100 Network Instances . . . . .	57
3.2	$R_{min}, P_{min}$ : Monotonicity and Asymptotic Limits . . . . .	65
3.3	Multicast Communication . . . . .	70
4.1	Optimal aggregation densities. The first number is the density, the second number is the relative performance increase (compared to DT). MF: Most Frequent, MR = Most recent. . . . .	77
4.2	Relative performance improvement compared to Direct Transmission. The first value for each entry corresponds to performance obtained with algebraic connectivity, the second value to Q modularity. . . . .	84
4.3	Clustering Coefficients and Average Path Lengths using different graph densities. Missing values in cases where there is no giant component.	88
4.4	Number of communities and modularity (Q). . . . .	88
4.5	Important notation for Section 4.8. . . . .	92
4.6	Modeling replication with Eq. (4.8) . . . . .	93
4.7	Important notation in Section 4.9. . . . .	105
4.8	Mobility trace characteristics. . . . .	111
4.9	Absorption probabilities . . . . .	115
4.10	Minimum $L$ to achieve expected delay in ETH with SimBet . . . . .	119
5.1	Variables and Shorthand Notation. . . . .	125
5.2	Probability of reneging for pedestrian and vehicular scenarios. . . . .	135
5.3	Optimal deterministic deadline times vs theory. . . . .	143
6.1	Notation . . . . .	149
6.2	Simulation Parameters . . . . .	166
6.3	Numerical values for Figure 6.3. . . . .	168
6.4	Numerical values for Figure 6.4. . . . .	169
6.5	Mean throughp. for handed-over users (in Mbps). . . . .	171

6.6	UL/DL Split Vs. Joint-association Improvements . . . . .	172
6.7	Comparison with existing work. . . . .	173
7.1	Notation. . . . .	180
7.2	Parameters of the scenarios considered. . . . .	183
7.3	Estimated offloading gains of rounded allocation vs. continuous relaxation for different cache sizes (in percentage of the catalogue size). . . . .	196
7.4	Mobility statistics . . . . .	197
7.5	Important Notation . . . . .	204
7.6	Information contained in datasets. . . . .	214
7.7	Dataset analysis. . . . .	214
7.8	Parameters used in simulations: default scenario. . . . .	216
7.9	Utility matrix density for the MovieLens dataset for different $u_{min}$ thresholds. . . . .	218

# List of Figures

1.1	Interplay between user association and TDD configuration. . . . .	29
2.1	Delivery probability and delay for Epidemic Routing with different scheduling and drop policies. . . . .	45
2.2	QoS Policy vs ORWAR . . . . .	49
2.3	QoS Policy vs CoSSD . . . . .	49
2.4	Overall policies comparison . . . . .	49
3.1	Markov Chain for epidemic spreading over a <i>homogeneous</i> network with $N$ nodes . . . . .	55
3.2	Markov Chain for epidemic spreading over a <i>heterogeneous</i> network with 4 nodes . . . . .	55
3.3	<i>Relative Step Error</i> for the step (a) $k = 0.2 \cdot N$ (i.e. message spreading at 20% of the network) and (b) $k = 0.7 \cdot N$ . Each boxplot correspond to a different network size $N$ (with $\mu_\lambda = 1$ and $CV_\lambda = 1.5$ ). Box-plots show the distribution of the Relative Step Error $RE_k$ for 100 different network instances of the same size. . . . .	57
3.4	Box-plots of the message delivery delay under (a) epidemic, (b) 2-hop routing, and (c) SnW (with $L = 6$ copies) routing. On each box, the central horizontal line is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually as crosses. The thick (black) horizontal lines represent the theoretical values predicted by our model. . . . .	58
3.5	Prediction accuracy for the CCDF of the epidemic delay on real networks	60
3.6	Mean delivery delay of 4 routing protocols, namely <i>Direct Transmission</i> , <i>Spray and Wait (SnW)</i> , <i>2-hop</i> , and <i>SimBet</i> , on the (a) Gowalla and (b) Strathclyde datasets. . . . .	62
3.7	Message delay under Direct Transmission, Spray and Wait ( $L = 5$ ), and Epidemic routing in scenarios with varying traffic heterogeneity; mobility parameters are $\mu_\lambda = 1$ and (a) $CV_\lambda = 1$ and (b) $CV_\lambda = 2$ . . . . .	67
3.8	$P_{(src.)}$ of mobility-aware routing in (a) synthetic scenarios with varying mobility ( $CV_\lambda$ ) and traffic heterogeneity ( $k$ ), and (b) real networks with homogeneous and heterogeneous traffic. . . . .	68

3.9	Delay ratio $R$ in two scenarios with varying traffic heterogeneity $k$ . Relay-assisted routing is SimBet with (a) $L = 5$ and (b) $L = 10$ message copies. . . . .	70
4.1	Converting a sequence of connectivity graph instances into a contact graph. . . . .	74
4.2	Aggregated contacts for the ETH trace at different time instants. . . . .	76
4.3	Delivered messages and delivery delay for SimBet/Bubble Rap vs. Direct Transmission. . . . .	77
4.4	Aggregated contact graph for a scenario with 3 social communities driving mobility (denoted with different colors), and three different aggregation densities. . . . .	79
4.5	Number of links and similarity values as a function of time. . . . .	80
4.6	Histogram of encountered similarity values at three different aggregation levels, with 2 clusters shown for each. . . . .	81
4.7	SimBet and Bubble Rap performance with different aggregation functions compared to direct transmission. . . . .	84
4.8	Mobility traces characteristics. . . . .	86
4.9	Contact Graph for DART trace. . . . .	87
4.10	Community sizes. . . . .	89
4.11	CCDFs for normalize global and community degree in log-linear scale. Insets show the same values with different x-axis range. . . . .	90
4.12	Example state transitions in two 6-node DTNs. . . . .	95
4.13	Summary of the necessary steps when building a DTN-Meteo model. .	97
4.14	Contact model for a toy network with $N = 4$ nodes and $E = 4$ out of 6 node pairs contacting each other. . . . .	99
4.15	Results of the $\chi^2$ goodness-of-fit test for exponentiality . . . . .	113
4.16	Absorption probability validation for <b>purely greedy</b> content placement and routing. . . . .	114
4.17	Expected delay validation for state-of-the-art routing schemes. . . . .	114
4.18	Predictions and meas. for <b>randomized</b> content placement and routing	116
4.19	Effect of (non-)exponentiality on relative errors of predicted delays . .	118
5.1	The WiFi network availability model modelled as an alternating renewal process. During ON periods there is WiFi connectivity while during OFF periods, only cellular connectivity is available. . . . .	124
5.2	The 2D Markov chain for the WiFi queue in delayed offloading. . . . .	126
5.3	The reduced Markov chain for $\rho \rightarrow 0$ . . . . .	131
5.4	Average delay for pedestrian user scenario. . . . .	134
5.5	Average delay for vehicular user scenario. . . . .	134
5.6	The delay for BP ON-OFF periods vs. theory. . . . .	135
5.7	Low utilization delay approximation for $AR = 0.75$ . . . . .	135
5.8	Low utilization $p_r$ approximation for $AR = 0.75$ . . . . .	136
5.9	High utilization delay approximation for $AR = 0.5$ . . . . .	136
5.10	Variable WiFi rates with the same average as theory. . . . .	137
5.11	Deterministic packets . . . . .	137

5.12	The delay for deterministic deadlines vs. theory. . . . .	137
5.13	The delay for BP packet sizes vs. theory. . . . .	137
5.14	Offloading gains for delayed vs. on-the-spot offloading. . . . .	138
5.15	The delay function for the optimization problem. . . . .	142
5.16	The delay vs. cost curve for high cellular rate. . . . .	142
5.17	The delay vs. cost curve for low cellular rate. . . . .	142
6.1	Access network queuing systems for different flows. . . . .	150
6.2	Backhaul topology in future Hetnet. . . . .	153
6.3	DL Optimal user-associations (Spectral efficiency vs. Load balancing and best-effort vs. ded. traffic performance) . . . . .	167
6.4	Optimal user-associations (DL vs. UL traffic performance) . . . . .	168
6.5	DL optimal associations in different scenarios. . . . .	169
6.6	Mean throughputs overall all users in the network. . . . .	171
6.7	Downlink Network Efficiencies (normalized). . . . .	172
7.1	Communication protocol. . . . .	181
7.2	Optimal allocation in semi-log scale. . . . .	182
7.3	Cost comparison according to TTL. . . . .	184
7.4	Cost comparison according to buffer capacity. . . . .	184
7.5	Cost comparison according to number of vehicles. . . . .	184
7.6	Sequence of contacts with three caches (above), and amount of data in end user buffer over time (below, in green). The red region indicates when data is downloaded from $\mathbb{I}$ . . . . .	189
7.7	Proposed queuing model for the playout buffer. The queue inside the small box corresponds to the low density regime model (Section 7.2.2), while the large box (containing both queues) to the high density regime (Section 7.2.3) . . . . .	191
7.8	Data offloading gain as a function of cache number (a) and buffer size (b). . . . .	197
7.9	Offloading gain vs. file size. . . . .	198
7.10	Offloading gain vs. file size. . . . .	199
7.11	Data offloaded for vehicular cloud vs. femto-caching. . . . .	200
7.12	Mobile app example for <i>Soft Cache Hits</i> : (a) related content <i>recommendation</i> (that the user might not accept) , and (b) related content <i>delivery</i> . . . . .	202
7.13	Cache hit ratio for all datasets and caching schemes, for the default scenario. . . . .	216
7.14	Cache hit ratio vs. cache size $C$ . . . . .	217
7.15	Cache hit ratio vs. number of SCs $M$ . . . . .	218
7.16	Cache hit ratio for the MovieLens dataset for scenarios with different $u_{min}$ thresholds; default scenario. . . . .	219
7.17	Relative increase in the cache hit ratio due to soft cache hits (y-axis). Amazon scenarios with different variance of number of related contents (x-axis). . . . .	220
7.18	Average delay in the network for uniform SNR . . . . .	224

# **Chapter 1**

## **Extended Summary**

## 1.1 Introduction

The last few decades have seen the performance and ubiquity of wireless networking increase by leaps and bounds. This has had a great impact on day-to-day life, with people becoming dependent on mobile devices for all sorts of activities ranging from entertainment, to health and wellness, to work, etc. This dependence in turn is fueling a need for yet better wireless network performance: the circle of increased performance generating increased usage that creates the need for yet more performance has led to an accelerating progression of wireless architecture generations, with 4G being the current state-of-the-art, and 5G just around the corner.

Higher base station density and bandwidth, as well as improved communication technology (e.g., MIMO, coding, etc.) arguably hold the lion's share among the factors responsible for the experienced performance improvement. Nevertheless, this has created rather complex, heterogeneous, hard to manage, and most importantly expensive cellular networks. Developments in the context of Software Defined Networks (SDN) [1] and Network Sharing through Network Function Virtualization (NFV) [2] promise to alleviate the complexity of configuring and managing such networks. Nevertheless, we might be reaching the limits of what traditional network upgrades alone can do to further push the performance envelope. Base station densification and higher bandwidth technologies lead to larger, costlier, and energy-hungrier networks, at the same time that operators see their profit margins decreasing.

Another important concern is that, while operators are advertising higher and higher peak rates, these peak rates are available in fewer and fewer locations [3]. 4G coverage can be very limited even in European countries, with patchy coverage only in major cities, and advertised peak rates experienced in even fewer locations and only at off-peak hours. Turning our attention to the developing world, things are yet worse (sometimes radically so), with network upgrades beyond 2G-3G being too expensive and even power supply not to be taken for granted. Hence, while cellular network progress in the last decade is propelling a leap in user experience and mobile data traffic for some, it is also widening the *digital divide* for many.

Both types of concerns have raised an important question: “*can we improve wireless network performance and user experience, without much or any expensive infrastructure and technology upgrades?*”. While similar questions have fueled research efforts along a number of directions, the following have been among the most popular ones:

- *Offloading* traffic from expensive resources (e.g., macro-cell radio bandwidth) to cheaper ones (e.g., WiFi access points, femto-cells, local storage points, user devices).
- *Load Balancing* or *Traffic Steering* to ensure all available resources are well utilized, and reducing the need for “peak traffic design” and overprovisioning.
- *Mobile Edge Caching* to reduce the distance of content from the user, as well as to avoid duplicate transmissions over the transport and core networks.

### **1.1.1 Wireless Data Offloading**

Data offloading has been explored in a variety of formats. Perhaps the most popular one nowadays is the use of WiFi instead of the cellular connection for data-intensive applications (streaming of long videos, file backup services, etc.), as almost every modern mobile device supports both. A user can configure to allow some applications to only access the Internet through WiFi, and some operators may even enforce WiFi connectivity when it is available. It is reported that 60% percent of total mobile data traffic was offloaded onto the fixed network through Wi-Fi or femtocells in 2016. In total, 10.7 exabytes of mobile data traffic were offloaded onto the fixed network each month [4].

Data traffic can also be offloaded within the same Radio Access Technology (RAT), by redirecting traffic from macro-cells to small(er) cells like micro-, pico-, and femto-cells. The higher power of macro-cells can be translated to higher nominal capacity per user, during off-peak hours. Nevertheless, this also leads to much higher coverage and too many users being connected concurrently during peak-hours or in “hotspot” locations. As user experience is not just affected by SINR but often more so by BS load, it is beneficial to “offload” as much of the macro-cell traffic to smaller BSs, serving fewer users, even if they offer lower SINRs [5]. In addition to relieving macro-cell load, small BSs (e.g., femto-nodes) might also have local break-out points to the Internet, thus also relieving the transport and core networks of an operator.

Taking the above idea a step further, it has been argued that data traffic should be offloaded to user devices themselves, when possible. For example, if a user device requesting some content X is near another device already storing that content X, the two devices can establish a local communication link (either by themselves, or with the help of the cellular network), to satisfy this request without involving any BSs. This is often referred to as *Device-to-Device* communication or *D2D-survey*. The fact that the majority of mobile data traffic is content-related, with many users being interested in the same popular content (e.g., latest episode of a popular series, or trending YouTube clips) increases the potential of such D2D approaches. An domain where D2D is promising is when users are interested in location-related content.

### **1.1.2 Load Balancing and Traffic Steering**

The main goal of load-balancing is to move traffic away from congested resources of the system (e.g., BSs, links, cloud servers) towards underutilized ones. No resources should be oversubscribed and underperforming due to congestion, while other equivalent resources remain underutilized or idle. On the one hand, such load balancing improves performance: delay is usually a convex function of load, so having one link underutilized and another congested leads to much higher delays than those same links splitting the total load. One the other hand, intelligent traffic steering also avoids the need for *peak traffic* provisioning, which is common but rather costly when loads fluctuate. Unfortunately, such load fluctuation in both space and time is to be expected when cell sizes grow smaller. The reason is that the laws of large numbers that is able to smooth out loads of macro-cells covering many user do not hold for smaller cells covering just a few users. To avoid costly peak traffic design to hedge for large fluctuations, traffic steering is performed using intelligent algorithms that attempt to

treat distributed and often heterogeneous resources as a “pooled” set of resources to be shared among all, thus offering statistical multiplexing gains. These resources might be base stations, carriers, RATs, antennas and even storage and processing capacity.

*User association* is one example problem, where careful load-balancing is needed. The traditional coverage with carefully positioned macro-cells, combined with the fact that voice was the dominant application, made user-association in 2G or even 3G networks simply a matter of which BS offers the strongest signal. The situation looks quite more complex now. The same coverage area might have overlapping layers of BSs (e.g., macro-cell with a number of small cells in its range). What is more, small cells themselves might have also partially overlapping coverage areas, especially in the context of envisioned UDNs (Ultra Dense Networks) [6]. Finally, each BS might also have multiple carriers, multiple antennas, different frequency reuse areas (to avoid interference at the edge). All this make user association in future wireless networks a hard case of load-balancing optimization problem.

### 1.1.3 Mobile Edge Caching

*Caching* can be seen both as a way to offload data away from (congested) parts of the network, as well as a way to balance the load among links and nodes along the content request path (i.e., from the user asking for a content, to the content provider central server(s)). For example, Information-Centric Networking (ICN) [7] suggest to cache content *anywhere* along that route, in order to both minimize access latency as well as localize traffic and micro-manage congestion along any part of the network. However, the necessity of complex hierarchical caching arising in ICN has been challenged [8].

*Mobile edge caching* has also been recently proposed, namely to store data directly at small cells. The latter, increase the radio access capacity (due to their shorter distance to the user) but are often equipped with low cost backhaul links (e.g., DSL, T1, or wireless), making this backhaul the new “bottleneck”. Caching popular content at the BS itself though can keep most traffic local to it, not burdening the underprovisioned backhaul link(s).

The goal of this thesis has been to explore all these three often inter-connected directions, using an arsenal of:

- (i) *Modeling tools* including markov processes, fluid approximations, queueing theory, and others, in order to derive analytically an appropriate performance objective for these problems, and to quantify the impact of key variables on this objective.
- (ii) *Optimization tools*, such as convex, discrete, and distributed optimization, markov chain monte carlo (MCMC) theory, and others, to identify optimal or at least efficient algorithms for each of these problems.

While the various contributions in these areas could probably be grouped in different ways, we have chosen to present the various contributions in two main parts, the first more closely related to device-based traffic offloading and the second related to BS-based offloading and mobile edge caching. This serves both as a more coherent story, as well as better (albeit not always) capturing the timeline of work, since the author

of this thesis has started exploring these topics from the device side, in relatively challenged networking environments, and has gradually moved to the infrastructure side in wireless networks.

The remainder of this chapter will briefly introduce each of the remaining chapters, providing some historic and related work context for each problem tackled, and summarizing and positioning the various contributions.

## 1.2 Thesis Contributions

### 1.2.1 Resource Management and QoS Provision in Delay Tolerant Networks (Chapter 2)

Delay Tolerant Networks (DTNs) can be broadly described as networks where message delivery delays well over the range we are familiar with, e.g., from a few minutes to hours, *can* or *must* be tolerated. Research in this area was motivated initially by extreme environments such as deep space communication, sparse networks, tactical and unmanned (airborne or underwater) vehicles, etc. However, it was soon recognized that *introducing some delay tolerance to device-to-device or device-to-infrastructure communication can facilitate data offloading*. Applications such as bulk synchronization traffic, software updates, asynchronous communication such as email, etc., could tolerate delays without the user noticing or at least with appropriate incentives [9]. Furthermore, it was observed that delay-tolerant and opportunistic data exchange between mobile devices could also greatly simplify the heavy-duty network routing and topology maintenance algorithms needed in ad-hoc networking, a popular research area at the time. *Opportunistic* message exchange between mobile devices whenever they are in range, and without explicit knowledge about the global network topology, could still achieve eventual (i.e., delay-tolerant) delivery of end-to-end messages or retrieval of requested content even in denser networks found in urban environments.

In the context of end-to-end message delivery in such networks, a number of routing (or, more correctly, *forwarding*) schemes had been proposed. The main goal of such schemes is to *decide if a node currently carrying a message for destination D, should hand over the message or a replica of it, to another encountered node*. The proposed schemes would range from making this decision randomly [10], epidemically [11], based on some replica budget [12], or using predictive metrics (“utilities”) regarding the probability of delivery of each node (for destination  $D$ ). The implicit objective of most of these heuristics was to ensure each message is delivered with a high probability to its destination, usually with some desired deadline (TTL). At the same time, it is important to efficiently utilize the limited resources of these networks: the node buffer capacity, the communication bandwidth during a “contact” between two nodes, and often the node battery.

Nevertheless, the majority of these proposals didn’t really tackle the resource management problem at all. It was common to either assume infinite buffer capacity and/or contact bandwidth, or to heuristically try to maintain a reasonable overhead as a side objective. For example, some proposals considered “ACK” type of messages to clean up valuable buffer space occupied by flooded messages that had already been deliv-

ered [13]. In contrast, our popular early protocol *Spray and Wait* [12] limited beforehand the number of replicas per message to a relatively small number, in order to cap the number of relay nodes (or “custodians”) that have to carry each message, as well as the number of transmissions (and related bandwidth) each new message consumes. At the name suggests, the first  $L - 1$  nodes encountered would immediately receive a copy of the message, and then each of the would “wait” until it encounters the destination. If  $L$  is the replication limit for a message of size  $S$  bits, then each message would consume at most  $L \cdot S$  total bits of buffer and transmission capacity. This choice was partly motivated by the seminal works of [14, 15], as well as by the (intuitive at the time) observation, that the *added value* of each extra replica was diminishing. As a result, choosing  $L > 1$  but  $L \ll N$  (the total number of nodes) would often result in good delivery ratios, without the congestion collapse associated with earlier epidemic-type protocols [11].

Nevertheless, this parameter  $L$  had to be chosen beforehand empirically, or using some foreknowledge about network characteristics. If this value was too conservative (e.g., network traffic was low at the moment), valuable resources could stay *underutilized*. If the value was too high, congestion would occur as in epidemic schemes. While a number of follow up works attempted to rise to this challenge, introducing some intelligence in the spray phase or in the wait phase [16, 17, 18], these still constituted a heuristic manner of resource management. The seminal work of [19] for the first time introduced the allocation of buffer and bandwidth resources as a formal optimization problem. The key observations were the following:

- (i) It is usually suboptimal to leave any buffer space or contact bandwidth unutilized; any such available resource could potentially improve the delivery probability/delay of some live message, without hurting the performance of others;
- (ii) Intelligent forwarding mechanisms need to kick in *only when necessary* (e.g., if the buffer of one of the involved nodes is full); otherwise, epidemic forwarding is optimal;
- (iii) The key role of these mechanism(s) should be to prioritize each message, according to the added value of an extra copy for that message.

In other words, the key insight of this approach is that you don’t need to decide in advance how many resources a message will consume. *One can let messages occupy all available resources, and need only select between two messages if a node’s resources are exhausted.* For example, if a new message arrives but the buffer is full, then that node will have to intelligently choose whether to drop an existing message (copy) or reject the newly arrived one. Similarly, if a node knows that the contact bandwidth during an encounter might not suffice to copy all messages, it *then* has to choose to prioritize some of them.

The key challenge in this context is then how to evaluate the added value or *marginal utility* of each message (copy). Making some simplifying assumptions about node mobility, the authors of [19] showed that this boils down to knowing the number of *already existing* replicas of that message. This number was assumed to be known through some low-bandwidth side channel (e.g., cellular network), which allowed each node to make independent, distributed decisions.

## Chapter Contributions

It is at this point that our first contributions of this chapter, come into the picture. These comprise the following papers [20, 21].

- A. Krifa, C. Barakat, T. Spyropoulos, “*Optimal Buffer Management Policies for Delay Tolerant Networks*”, in Proc. of IEEE SECON 2008. (best paper award)
- A. Krifa, C. Barakat, T. Spyropoulos, “*Message Drop and Scheduling in DTNs: Theory and Practice*”, IEEE Transactions on Mobile Computing. 11(9): 1470-1483, 2012.

**Contribution 2.1** We improved the proposed utility function of [19] to properly account for the probability that the message has already been delivered, based on the number of nodes encountered so far by the message source (which might differ from the nodes currently holding a replica).

**Contribution 2.2** We generalized the framework of [19] and showed how to derive per message utilities for different performance metrics, with delivery delay and delivery probability as two specific examples.

**Contribution 2.3** A side channel or centralized mechanism to keep track of number of replicas per message might not always be present. In that case, estimating the number of copies per message in a distributed manner is an intrinsically hard task in this context, given the disconnected, delay-ridden nature of these networks. To this end, we proposed an efficient *history-based* mechanism to achieve this with the following desirable features: (i) the utility estimates derived are *unbiased* statistics; (ii) the data collection mechanism to derive these statistics is in-band and low overhead in terms of message storage and propagation; (iii) the estimates do not require any knowledge about the current message (which is hard to obtain), but rather on past knowledge about *similar* messages. Simulations suggest that this distributed, online version of the algorithm performs close to the oracle one in a number of scenarios.

These contributions seemed relatively conclusive about the buffer and bandwidth management problem, at the time, at least in the context of simple epidemic schemes. A number of issues remained though, one of which being how to provide some sort of Quality of Service (QoS) and differentiate between different classes of messages. Somewhat surprisingly, this question was only addressed much later. One might argue that this was perhaps due to DTNs being inherently unreliable and delay-tolerant, which makes QoS a contradicting concept in this context. Nevertheless, a number of DTN setups, such as tactical or space networks still gave rise to a need for traffic differentiation in DTNs as well. For example, some messages might be tolerant to larger than usual delays, but might still have to be delivered with high probability before a deadline, otherwise they are useless. On the other side of the fence, some messages might be more time-critical than others, but losing some of them is OK.

To this end, we have extended our resource management framework in this direction in the following papers [22, 23].

- P. Matzakos, T. Spyropoulos, and C. Bonnet, “*Buffer Management Policies for DTN Applications with Different QoS Requirements*,” in Proc. of IEEE GLOBECOM 2015.

- P. Matzakos, T. Spyropoulos, and C. Bonnet, “*Joint Scheduling and Buffer Management Policies for DTN Applications of Different Traffic Classes*,” in IEEE Transactions on Mobile Computing, 2018.

**Contribution 2.4** In addition to “best effort” type of messages (the only class of messages in previous work), we have introduced *classes* of messages with QoS guarantees. E.g., a class (“VIP”) might require a given minimum delivery probability for each message. The objective in this case is to *first* ensure that all QoS classes achieve their minimum required performance, and *then* to allocate remaining resources (if any) to maximize mean performance for all messages.

**Contribution 2.5** We have (re-)formulated the problem of *optimal* resource allocation (buffer and bandwidth) among classes with different QoS requirements as a *constrained optimization problem*. We showed that this problem is convex, and extended the distributed protocol based on per message utilities of [21] in a manner that can be shown to solve the aforementioned QoS-aware problem.

**Contribution 2.6** We have also generalized the mobility assumptions under which the proposed framework can be applied, including: (i) heterogeneous (but still exponential) inter-contact times, (ii) heavy-tailed inter-contact times (Pareto), which are common in some DTN networks.

**Contribution 2.7** Using extensive sets of simulation, based on synthetic and real traces, we have demonstrated that the proposed framework achieves the desired objectives, as outlined by (d), and outperforms previous related work that attempted to heuristically introduce QoS.

### 1.2.2 Performance Modeling for Heterogeneous Mobility and Traffic Patterns (Chapter 3)

The previous discussion revolves around one of the key directions in the early DTN (and Opportunistic Networking) research, namely that of *forwarding algorithm design*. The second important direction has been that of *analytical performance modeling*. The former attempts to find a good message forwarding algorithm under a given network setup. The latter is given the algorithm and network setup, and attempts to derive an analytical, ideally closed form prediction of this algorithm’s performance. Nevertheless, the existence of multiple nodes (message sources, destinations, relays) interacting in a stochastic manner, and the additional dependence on protocol details, makes this task daunting, even in relatively simple network setups.

To this end, the majority of early DTN works that attempted to characterize different routing algorithms almost exclusively made the following simplifying assumptions:

- (i) The inter-contact time between different nodes is *independent* and follows an *exponential* distribution; this assumption allows the message evolution to be modelled by a Markov chain.
- (ii) Pair-wise inter-contact times, in addition to independent, are also *identically distributed*. In other words, the average meeting time between any pair of nodes is the same; this assumption is key to make the Markov chain analysis tractable.

These two assumptions enabled to derive closed form expressions for popular (albeit often quite primitive) forwarding such as epidemic forwarding, 2-hop routing, random forwarding, and spray and wait schemes

While additional simplifying assumptions had to often be made (we will consider some subsequently), these two have been the main “bottleneck” of DTN performance analysis for a while. As device-to-device opportunistic networking became a popular research topic in the mid-2000s, a number of experimental studies emerged that actually tested these two hypotheses. These studies generally studied the contact characteristics of mobile (or nomadic) nodes in different setups including conferences, campuses, public transportation, etc. Initial findings suggested that both assumptions are quite far off from observations. Inter-contact time distributions were shown to follow a power law distribution instead of being exponential, as assumed in theory, and to also have largely heterogeneous behaviors [24]. These findings seemed to invalidate the various analytical findings [24] as well as the relative performance among well-known schemes [25].

Although the debate about the exponentiality of inter-contact times stayed hot for a couple of years (and is still not fully settled), subsequent measurement and theoretical analyses emerged that provided *some* support for the exponential assumption. This support came from a number of directions:

- (i) When looking at the intercontact times of specific node pairs, rather than aggregating *all* inter-contacts of *all* pairs in one histogram (as was done in [24]), a number of pairs *do* show such an exponential behavior (while some others a heavier-tail one) [26];
- (ii) Aggregating inter-contact times of different pairs which are exponential, but with different means, can give rise to a heavy tail aggregate inter-contact distribution, like the ones observed in measured traces [27].
- (iii) Inter-contact times can be modeled as *stopping times* of specific random walks (a lot of mobility models can be modelled as a generalized random walk), whose probability distribution has a power law body but *an exponential tail* [28].

These findings suggested that, while inter-contact times are clearly neither always nor exactly exponential, this can be a reasonable approximation in a number of cases, resulting in reliable performance prediction. Nevertheless, the second problematic assumption remained. Measurement studies (and common sense) suggest that not all node pairs meet each other equally frequently. In fact, some node pairs *never* meet each other [24]. Other nodes, due to their higher mobility (e.g., vehicle vs. pedestrian, smartphone vs. laptop) might meet more nodes during the same amount of time than others. The question thus remained: *Does the entire analysis break down if node meetings are heterogeneous? Could this analysis be generalized?*

While we already mentioned some cases of such generalizations in the previous subsection, these applied to specific forwarding primitives only, and not the general case of epidemic schemes.

## Chapter Contributions

Our contributions, detailed in Chapter 3, start at this point. The following paper makes three contributions in this direction [29].

- P. Sermpezis, and T. Spyropoulos, “*Delay analysis of epidemic schemes in sparse and dense heterogeneous contact environments*”, IEEE Transactions on Mobile Computing, 16(9): 2464-2477, 2017.

**Contribution 2.1** We have kept the assumption of exponentiality but generalized the inter-contact process between nodes using a *2-layer* model: the first layer is a random graph capturing which nodes communicate (at all) with each other; for each existing link  $\{i, j\}$  on the graph, we draw a random weight  $\lambda_{ij}$  from a common distribution  $F(\lambda)$  which corresponds to the meeting rate for that pair. In other words, the inter-contact times between nodes  $i$  and  $j$  are distributed as  $\exp \lambda_{ij}$ . By changing the underlying random graph model or the function  $F(\cdot)$ , a large number of mobility models can be captured.

**Contribution 2.2** We showed that, for an underlying Poisson contact graph, and  $F(\cdot)$  with finite variance, the delay of each epidemic step asymptotically converges to a closed form expression (asymptotically in the size of the network). The same holds for the delivery delay of epidemic forwarding, which is the sum of such steps (as well as other schemes that can be expressed as a simple combination of epidemic steps).

**Contribution 2.3** We derived approximations for finite networks, as well as for generalized underlying contact graphs, captured by the well known *Configuration model* [30].

The above model manages to capture *both* pairs of nodes that don’t meet (i.e., sparse networks) as well as heterogeneous intercontact times. While quite generic, this model requires some structure: pairwise meeting rates, while different, are drawn *IID* from the same probability distribution  $F(\lambda)$ . The above model does not capture *arbitrary* meeting rate combinations. Most importantly, it doesn’t well capture *community structure*, which is ubiquitous in real mobility traces. In reality, there exist some transitivity between meeting rates: if node  $i$  tends to meet often node  $j$  and node  $k$ , then  $j$  and  $k$  probably meet often. This is also referred to as *clustering* [30], and cannot be captured by the model in [29]. To this end, in the following work [31], we extended our analytical results in this direction.

- A. Picu, T. Spyropoulos and T. Hossmann, “*An Analysis of the Information Spreading Delay in Heterogeneous Mobility DTNs*,” in Proc. of IEEE WoWMoM 2012. (best paper award)

**Contribution 2.4** We formalized the weighted graph model for inter-contact rates between pairs as a generic matrix, and showed how epidemic step delay can be captured by properties of this matrix.

**Contribution 2.5** Based on this model, we derived upper bounds for both the mean delay and the delay distribution of epidemic routing that apply to generic mobility scenarios, including community-based ones, and are validated against real traces.

As a final set of contributions in this chapter, we took a step beyond heterogeneity in terms of who contacts whom, and considered for the first time the equally important

question of *traffic heterogeneity*, namely who wants to send a message to whom. In fact, the majority of related work, both analytical and simulation-based, did not only assume IID mobility, an aspect that has been extensively revisited as explained above, but also assumed uniform traffic demand. Somewhat surprisingly, this latter assumption had received to that day almost no attention. We made the following contributions to amend this, elaborated at the end of Chapter 3. The material is based on three articles [32, 33, 34]:

- P. Sermpezis, and T. Spyropoulos, “*Not all content is created equal: effect of popularity and availability for content-centric opportunistic networking*,” in Proc. of ACM MobiHoc 2014.
- P. Sermpezis, and T. Spyropoulos, “*Modelling and analysis of communication traffic heterogeneity in opportunistic networks*”, in *IEEE Transactions on Mobile Computing*, 14(11): 2316-2331, 2015.
- P. Sermpezis, and T. Spyropoulos, “*Effects of content popularity in the performance of content-centric opportunistic networking: An analytical approach and applications*,” *IEEE/ACM Transactions on Networking*, 24(6): 3354-3368, 2016.

**Contribution 2.6** We showed both analytically and using simulations that traffic heterogeneity impacts performance only if its *correlated* with mobility heterogeneity. In other words, if who sends more messages to whom is independent from who encounters whom more frequently, then traffic heterogeneity does not impact performance beyond the *aggregate* mean traffic demand.

**Contribution 2.7** We proposed a class of traffic-mobility correlation models and extended our analytical framework accordingly. We used this framework to derive closed form performance results for a number of simple but well-known opportunistic network routing schemes, under both mobility and traffic heterogeneity. Using both theory and trace-based simulations we showed how configuring well known protocols (e.g., spray and wait) needs to be modified in light of such heterogeneity.

**Contribution 2.8** We extended our analysis beyond end-to-end communication to *content-centric* applications, making some links with device-side caching and D2D. (the topic of caching is more extensively considered in a later chapter).

### 1.2.3 Complex Network Analysis for Opportunistic Device-to-Device Networking (Chapter 4)

The previous two research threads led to new insights, models, and algorithms, in terms of both resource allocation in DTNs, as well as more accurate analytical models that were inline with recent mobility trace insights. Nevertheless, the majority of these works were still mostly based on simple “random” forwarding protocols like epidemic routing, spray and wait, 2-hop routing, and variants. These protocols, which could be arguably referred to as “1st generation” DTN routing, were based on simple forwarding decisions: when a node with a message copy encounters another node without a copy, it can: (i) create a new copy (epidemic), (ii) create one with some probability

(probabilistic routing), (iii) create one only if the copy budget is not depleted (spray and wait), and other variants. In other words, the decision to give a copy to a newly encountered node did not really depend on properties of the relay or the destination (e.g., whether the two might meet soon). This justifies the term “random”, used earlier.

The natural progress in this direction was to amend these protocols to make “smarter” decisions that can assess the usefulness or *utility* of a given relay node. For example,

- (i) the utility might capture the ability of that node to deliver messages to a given destination (e.g., if the two nodes rely in the same “community” or location area and tend to meet often)
- (ii) a node might also have high utility for *any* destination (e.g., a node that tends to explore larger areas of the network compared to average nodes).

*Utility-based* routing protocols for DTNs had already been explored early on. *Prophet* was a popular early variant of epidemic forwarding. There, a node A encountering another node B would give the latter a copy of a message for destination  $X$ , only if the utility of B (related to  $X$ ) was higher than that of A. This utility in turn was based on a mechanism that considered the recency of encounter between the two nodes. It could also consider some transitivity in utilities, e.g., B could have a high utility for  $X$ , not because it saw  $X$  recently, but because it sees other nodes that see  $X$  recently. Similarly, the early “single-copy” study of [10] considered such simple utility metrics but assuming a message is not copied, but rather forwarded. Single-copy DTN forwarding had the advantage of significantly limiting the overhead per message, but did not benefit from the path diversity that multiple-copy schemes exploit [25].

Finally, utility metrics were used to improve specific aspects of existing protocols. For example, Spray and Focus [12] did not create any new copies after there was a total of  $L$  in the network, in order to limit the amount of overhead per message (just as in Spray and Wait), but did allow each such copy to be handed over to a relay with higher utility for the destination, if one was encountered. Similarly, *smart spraying* methods were proposed to not randomly handover the  $L$  copies, but to do so according to some meeting-related utility metric [16, 17, 18].

Nevertheless, the majority of these utility-based methods were simple heuristics, that sometimes improved protocol performance but not always. More importantly, *utilities were pairwise metrics, based on the meetings characteristics between the relay in question and the destination*. A breakthrough came by the seminar works of [35, 36]. Motivated by the intricate structure between node contacts, revealed in the recently studied mobility traces, the authors suggested that contacts between mobile devices are subject to the same *social* relations that the users carrying these devices are subject to. Hence, it is only reasonable to utilize the new science of *Complex Networks* or *Social Network Analysis* to answer questions related to which node might be a better relay for another (these terms, including the term *Network Science* are often used interchangeably, to mean the same thing).

This gave rise to Social Network based opportunistic networking protocols. The main idea behind these first schemes was simple:

- First, collect knowledge about past contacts into a graph structure called a *social* or *contact graph*; a link in this graph could mean, for example, that the two

endpoints of the link have met recently, or have been observed to meet frequently enough in the past.

- Then, use appropriate social network tools or metrics to make forwarding decisions.

For example, BubbleRap [36] uses *Community Detection* algorithms to split the network into communities. Then, epidemic routing was modified using communities and *degree centrality* [30]. The node with the highest degree centrality would be the node that has met, for example, the largest number of other nodes within some time interval. If the message has not yet reached the destination community, a message copy in BubbleRap is given to an encountered node only if the latter has higher *degree centrality*. When the message has reached a relay in the destination community, then *local degree centrality* would be used instead (i.e., a node would become a new relay only if it was better connected *in that community*).

SimBet [35] uses instead *node betweenness* (a different measure of node centrality) to traverse the network. It then “locks” to the destination using *node similarity*. Node similarity captures the percentage of neighbors in common with the destination. Due to the high clustering coefficient of social networks, if two nodes share many neighbors then they belong to the same community with high probability. Hence, trying to find a node with high similarity with the destination in SimBet is equivalent to trying to find a node in the destination community (just as BubbleRap does, in the first phase).

As expected, these two protocols outperformed traditional random and utility-based ones, by intelligently exploiting not just *pairwise structure* in node contacts, but rather *macroscopic structure* such as communities, betweenness, etc., that depend on the interactions of multiple nodes. Nevertheless, this new line of work also raised some important questions:

1. How should one optimally built the contact graph on which the social metrics will be calculated?
2. Do all opportunistic networks exhibit such “social” characteristics, and if so what are the most prominent ones? Do state-of-the-art mobility models capture these?
3. Can one still hope to do useful performance analysis when forwarding protocols become that complex and interdependent with the underlying mobility process?

The goal of this chapter is to present our contributions towards answering these three questions.

### Chapter Contributions

The begining of the chapter is concerned with the first question, which we attempted to answer in the following two works [37, 38]:

- T. Hoßmann, F. Legendre, and T. Spyropoulos, “*From Contacts to Graphs: Pitfalls in Using Complex Network Analysis for DTN Routing*,” in Proc. of IEEE International Workshop on Network Science For Communication Networks (NetSciCom 09), co-located with INFOCOM 2009.

- T. Hossmann, T. Spyropoulos, and F. Legendre, “*Know Thy Neighbor: Towards Optimal Mapping of Contacts to Social Graphs for DTN Routing*,” in Proc. of IEEE INFOCOM 2010.

**Contribution 4.1** We studied different “aggregation methods”, i.e., methods to convert the history of past contacts into a social graph, and showed that the performance of Social Network based protocols is very sensitive to the contact graph creation method; simple empirical tuning or rules of thumb will likely fail to unleash the potential of such protocols.

**Contribution 4.2** We proposed a distributed and online algorithm, based on spectral graph theory. It enables each node to estimate the correct social graph in practice, assessing the role of new nodes in the graph without any training, in a manner reminiscent of unsupervised learning. This algorithm is generic, and could be applied as the first step of *any* opportunistic networking protocol that uses social network metrics.

Regarding the second question, some initial insights about community structure had been observed in [36]. However, we embarked on a systematic study towards answering that question in [39, 40]:

- T. Hossmann, T. Spyropoulos and F. Legendre, “*A Complex Network Analysis of Human Mobility*,” in Proc. of IEEE NetSciCom 11, co-located with IEEE Infocom 2011.
- T. Hossmann, T. Spyropoulos, and F. Legendre, “*Putting Contacts into Context: Mobility Modeling beyond Inter-Contact Times*,” in Proc. of ACM MobiHoc 2011. (best paper award runner-up)

Our contributions in these papers can be summarized as follows:

**Contribution 4.3** We performed a large study of contact traces coming from state-of-the-art mobility models, existing real traces, and traces collected by us (from geo-social networks). We convert each of them into a social graph, using some of the earlier methods from this chapter.

**Contribution 4.4** We investigated whether these graphs exhibit well-known properties of social networks: (i) high clustering and community structure, (ii) small-world connectivity, and (iii) power-law degree distributions. Indeed, we found that the majority of mobility traces exhibit the first two, but not always the third property .

**Contribution 4.5** We showed that state-of-the-art synthetic mobility models, while able to recreate some of these social properties well, are unable to model *bridging nodes*, which seem to be common in real traces. To this end, we suggested a modification of such models using *multi-graphs*, which can be applied as an “add-on” to different mobility models, without interfering with other desirable properties of these models.

These developments were quite positive, corroborating the initial evidence about the strength of social network analysis for opportunistic D2D networking, as well as demonstrating how to properly tap into this potential. Nevertheless, the complexity of protocols based on social graph metrics came on top of the existing challenges of performance analysis for non-IID mobility models, mentioned earlier. The intricate dependence of forwarding decisions on past contacts between many different nodes

creates state history for the Markovian analysis model that is hard to keep track of. What is more, by that time, a large number of utility-based routing protocols had been proposed, making it difficult to come up with a proprietary analytical model (e.g., a specific Markov chain) for each different protocol. To this end, in the following two works, we proposed a unified analytical performance prediction model, coined *DTN-Meteo* [41, 42].

- A. Picu and T. Spyropoulos, “*Forecasting DTN Performance under Heterogeneous Mobility: The Case of Limited Replication*,” in Proc. of IEEE SECON 2012.
- A. Picu, and T. Spyropoulos, “*DTN-Meteo: Forecasting the Performance of DTN Protocols Under Heterogeneous Mobility*,” in IEEE/ACM Transactions on Networking, 23(2): 587-602, 2015.

Our main contributions, can be summarized as follows:

**Contribution 4.6** We attempted to generalize the performance prediction models for heterogeneous mobility of Chapter 3, in order to allow deriving performance metrics for generic utility-based algorithms. Our framework combined the theory of Markov Chain Monte Carlo (MCMC) based optimization and the theory of Absorbing Markov chains as follows:

- Each state of a Markov chain corresponds to a possible *configuration* or *state* of the network (e.g., which nodes have which message(s)) at a time.
- The *mobility process* governs the possible *transitions* and the respective transition rates between states.
- The *algorithm* (e.g., forwarding) governs the *acceptance probabilities* of a potential transition, a probability that may depend on the utility of the previous and the potential new state.
- *Absorbing* states correspond to desired final states of the protocol (e.g. message delivered to its destination, or to all destination in case of multicast, etc.). Performance metrics can be expressed in terms of absorption times and probabilities.

**Contribution 4.7** Using this framework, we managed to successfully model the state-of-the-art protocols mentioned earlier, namely SimBet and BubbleRap, running on top of generic, trace-based mobility models. We also showed that this framework applies to various delivery semantics, beyond end-to-end message delivery, such as multicast, anycast, etc.

### 1.2.4 WiFi-based Offloading (Chapter 5)

Offloading mobile data via WiFi has been common practice for network operators and users alike. Operators might purchase WiFi infrastructure or lease a third-party one, to relieve their congested cellular infrastructure. User prefer to use WiFi at home, cafes, office, for bulk data transfers, video streaming, etc., as much as possible, to avoid depleting their cellular data plans, improve their rates, or sometimes to save their

battery life. As mentioned earlier, it is reported that 60% percent of total mobile data traffic was offloaded onto the fixed network through Wi-Fi or femtocells in 2016 [4].

The ubiquity of WiFi access points, and the interest of network operators in their use, has motivated researchers to study a number of ways to use WiFi-based Internet access as an inexpensive complement to cellular access. One such proposal was to use WiFi access points (AP) while on the move, e.g., from a vehicle, accessing a sequence of encountered WiFi APs to download Internet content [43]. WiFi was originally designed for “nomadic” users, i.e., users who come into the range of the AP and stay for long periods of time, and not for moving users. A number of measurement studies were thus performed to identify the amount of data one could download from such an AP, travelling at different speeds [44, 45]. While the measured amounts suggested that reasonably sized files could already be downloaded even at high speeds, there were a number of shortcomings related to mobile WiFi access:

- The association process with a WiFi AP takes a long time (due to authentication procedures, scanning, and other suboptimalities in the protocol design). This wastes a large amount of the limited time during which a mobile node might be within communication range with that AP (and thus wastes communication capacity).
- The rate adaptation mechanism of WiFi, which reduces the PHY encoding rate (and thus the transmission rate), as a function of the SINR of the receiving node, has dire side-effects when a number of nodes on the move would try to access the same AP. In that case, there always exist at least some nodes at the edge of the AP coverage range, who receive the lowest possible transmission rate. However, it is well known that the way scheduling works in WiFi, the average performance is highly impacted by the existence of edge users [46]. Connecting to WiFi on the move greatly deteriorates the mean WiFi throughput for every node, even the ones close to the AP, as at least some of the mobile nodes are at the edge of the AP, entering its range (or leaving it).

A number of works emerged to address both these issues. Some of the ideas included streamlining the WiFi association procedure [47], maintaining a history of AP location and quality to improve the speed and efficiency of scanning [48], as well as modifications to the scheduling mechanism of WiFi, ensuring that the capacity is allocated to mobile nodes during the time they are close to the AP, a form of *opportunistic scheduling*. While a lot of research activity was taking place in the context of protocol improvements, system design, and experimentation, there was little ongoing in terms of the theoretical understanding for WiFi offloading methods.

### Chapter Contributions

This line of research made us interested in the following question: *Assuming a stochastic traffic mix (i.e., both random traffic arrival, and random flow/session sizes), what is the expected performance of WiFi offloading, as a function of AP deployment and characteristics, mobile node behavior, and traffic characteristics?*. We have attempted to answer it in the following works [49, 50]:

- F. Mehmeti, and T. Spyropoulos, “*Performance Analysis of On-the-spot Mobile Data Offloading*,” in Proceedings of IEEE GLOBECOM 2013.
- F. Mehmeti, and T. Spyropoulos, “*Performance analysis of mobile data offloading in heterogeneous networks*,” in IEEE Transactions on Mobile Computing, 16(2): 482-497, 2017.

Our contributions can be summarized as follows:

**Contribution 5.1** We proposed a queueing-theoretic model, an M/G/1 with different levels of service rate. Assuming a user generating random download requests and a common queue on top of the WiFi and cellular interfaces: when a WiFi AP is in range, these requests are always served through the WiFi interface (with one servicerate); if there is no WiFi in range, the requests are served by cellular interface (with a different service rate).

**Contribution 5.2** We analyzed the performance of this system for both First Come First Serve (FCFS) and Processor Sharing (PS) queueing disciplines, and derived closed form expressions for the expected amount of offloaded data, as well as the mean download performance. These expressions are a function of user mobility, WiFi coverage, and WiFi/Cellular capacity characteristics.

A second research direction that emerged was that of *Delayed WiFi Offloading*. This was partly motivated by the study and exploitation of delay-tolerance (the application and/or the user might be tolerant to delays). As explained earlier, this delay tolerance might sometimes be natural, and sometimes requires some incentives. To this end, researchers suggested that some traffic does not need to be immediately transmitted over the cellular interface (as required in the previous scenario), if there is no WiFi connectivity. Instead, such delay-tolerant download (or upload) requests could be queued at the WiFi interface, until a WiFi AP is encountered. If such an AP is not encountered until a maximum wait timer expires, *only then* must the request be turned over to the cellular interface [51, 52]. This more “aggressive” offload policy was shown to be able to offload even more data, at the expense of a delay increase for some traffic.

To this end, in the following works, we extended our performance analysis framework to investigate such delayed offloading policies as well [53, 54].

- F. Mehmeti, and T. Spyropoulos, “*Is it worth to be patient? Analysis and optimization of delayed mobile data offloading*,” in Proc. of IEEE INFOCOM 2014.
- F. Mehmeti, and T. Spyropoulos, “*Performance modeling, analysis and optimization of delayed mobile data offloading under different service disciplines*,” in ACM/IEEE Transactions on Networking, 25(1): 550-564, 2017.

**Contribution 5.3** We used a queueing theoretic model with service interruptions and abandonments, to model the intermittent access to WiFi APs (the interruptions) and the possibility that a request waiting in the WiFi queue might expire and move back to the cellular interface (the abandonments).

**Contribution 5.4** We derived closed form expressions for the amount of offloaded data and mean per flow delay in this system, as a function of the mean delay-tolerance of the application mix considered, and other network parameters.

**Contribution 5.5** In a scenario where it is the user that can decide this delay tolerance, we showed our analytical expressions and convex optimization theory to show how the wait threshold can be optimized to achieve different delay-vs-offloaded data or delay-vs-energy efficiency tradeoffs.

### 1.2.5 User Association in Heterogeneous Wireless Networks (Chapter 6)

In addition to using third-party WiFi or own WiFi platforms, operators are increasingly considering denser, heterogeneous network (HetNet) cellular deployments. In a HetNet, a large number of small cells (SC) are deployed along with macrocells to improve spatial reuse [55, 56, 57]. The higher the deployment density, the better the chance that a user equipment (UE) can be associated with a nearby base station (BS) with high signal strength, and the more the options to balance the load. Additionally, the line between out-of-band WiFi “cells”, and in-band cellular small cells (femto, pico), is getting blurred, due to developments like LTE-unlicensed. The trend towards base station (BS) densification will continue in 5G systems, towards *Ultra-Dense Networks (UDN)* where: (i) many different small cells (SCs) are in range of most users; (ii) a small number of users will be active at each SC [58, 59, 6]. The resulting traffic variability makes optimal user association a challenging problem in UDNs [58], as the goal here is not to simply choose between WiFi or cellular as in the previous chapter, but to choose between a large range of heterogeneous and overlapping cells.

As a result, a number of research works emerged that studied the problem of user association in heterogeneous networks, optimizing user rates [60, 61], balancing BS loads [62], or pursuing a weighted tradeoff of them [63]. Range-expansion techniques, where the SINR of lightly loaded BSs is biased to make them more attractive to the users are also popular [56, 57]. The main goal of these works is to offload or “steer” traffic away from overloaded base stations, towards underloaded ones, while maintaining (or improving) user performance. Nevertheless, the majority of these works focused on DL traffic and the radio access link only. Future user association algorithms should be sophisticated enough to consider a number of other factors.

- *Uplink:* While optimization of current networks revolves around the downlink (DL) performance, social networks, machine type communication (MTC), and other upload-intensive applications make uplink (UL) performance just as important. Some SCs might see their UL resources congested, while others their DL resources, depending on the type of user(s) associated with that SC. What is more, the same SC might experience higher UL or DL traffic demand over time.
- *Traffic Classes:* Most existing studies of user association considered homogeneous traffic profiles. For example, [63, 64, 65] assume that all flows generated by a UE are “best-effort” (or “elastic”). Modern and future networks will have to deal with high traffic differentiation, with certain flows being able to require specific, *dedicated* resources [66], [67, 68]. Such dedicated flows do not “share” BS resources like best-effort ones, are sensitive to additional QoS metrics, and affect cell load differently.

- *Backhaul Network*: Ignoring the backhaul during user association is reasonable for legacy cellular networks, given that the macrocell backhaul is often over-provisioned (e.g., fiber). However, the considerably higher number of small cells, and related Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) suggest that backhaul links will mostly be inexpensive wired or wireless (in licensed or unlicensed bands), and underprovisioned [69]. Multiple BS might also have to share the capacity of a single backhaul link due to, e.g, point-to-multipoint (PMP) or multi-hop mesh topologies to the aggregation node(s) [70]. Hence, associating a user to a given BS might lead to backhaul congestion and low end-to-end performance, even if that BS can provide a high radio access rate to the user.

### Chapter Contributions

In the following three works, we have addressed these exact questions, within a common unifying framework [71, 72, 73].

- N. Sapountzis, T. Spyropoulos, N. Nikaein, and U. Salim, “*An Analytical Framework for Optimal Downlink-Uplink User Association in HetNets with Traffic Differentiation*,” in Proc. of IEEE GLOBECOM 2015.
- N. Sapountzis, T. Spyropoulos, N. Nikaein, U. Salim, “*Optimal Downlink and Uplink User Association in Backhaul-limited HetNets*,” in Proc. of IEEE INFOCOM 2016.
- N. Sapountzis, T. Spyropoulos, N. Nikaein, U. Salim, *User Association in HetNets: Impact of Traffic Differentiation and Backhaul Limitations*,” in ACM/IEEE Transactions on Networking, 25(6): 3396-3410, 2017.

The related contributions can be summarized as follows:

**Contribution 6.1** Using the flow-level performance framework of  $\alpha$ -fair user association [63] as our starting point, we extended it considerably to include (i) traffic differentiation: splitting scheduler resources between GBR and non-GBR traffic, and assuming a different scheduler for each class; (ii) UL traffic: considering both legacy scenarios where UL and DL traffic must go through the same BS, as well as the envisioned UL/DL split [74]; (iii) backhaul characteristics: considering underprovisioned backhaul links as well as different network topologies (star or tree).

**Contribution 6.2** We derived optimal association rules for each combination of the above scenarios that can be implemented efficiently in a distributed manner and proved convergence to the global optimum (i.e., without any central element requiring global knowledge of all network state).

**Contribution 6.3** We used extensive simulations to understand the impact of backhaul topology, and interplay of UL/DL traffic, and showed that our algorithm outperformed state of the art algorithms at that time.

In addition to this work, presented in detail in Chapter 6, we have very recently made new contributions related to user association in heterogeneous, dense networks. In the interest of space, we will not elaborate on them further in the respective chapter, but only provide a short summary here.

## Joint Optimization of User Association and Flexible TDD

Conventional networks usually operate with the same amount of resources for UL and DL (FDD or static TDD) [75]. Consider for example the standard FDD which uses some fixed, separate bands for uplink and for downlink. Our results from [73] suggested that this can often be suboptimal. In TDD, it is possible to configure more resources for DL than UL (or vice versa, although not common), but each macro-cell is usually configured with the same ratio to be used permanently. However, recently proposed *Dynamic or Flexible TDD* systems can better accommodate UL/DL traffic asymmetry, by varying the percentage of LTE subframes used for DL and UL transmission [76]. Hence, more UL resources can be allocated to SCs with UL intensive users, and vice versa. Nevertheless, dynamic TDD introduces new challenges.

First, *there is a strong interplay between user association and dynamic TDD policies*. Consider the simple example of Fig. 1.1, where a UL intensive user (1) and a DL intensive user (2) are both in range of a SC A (which is close) and a SC B (which is further away). Assume that each SC has initially the same amount of DL and UL resources (50%). Both the DL and the UL user will connect to A, as it offers the best SINR. Now, notice that any change in the TDD schedule of SC A will hurt one of the two users. However, assume that A increases its UL resources to 80%. There are now 8/5 (i.e. 60%) more resources for the UL user, which could lead to 60% higher rate. Furthermore, assume that SC B increases its DL resources to 80%. Connecting the DL user to B can increase the available resource blocks for her also by a factor of 8/5. If the resulting SINR decrease has a smaller impact than this factor, then *both users can win by revisiting both the TDD schedules and the association decisions of the two base stations, in coordination*.

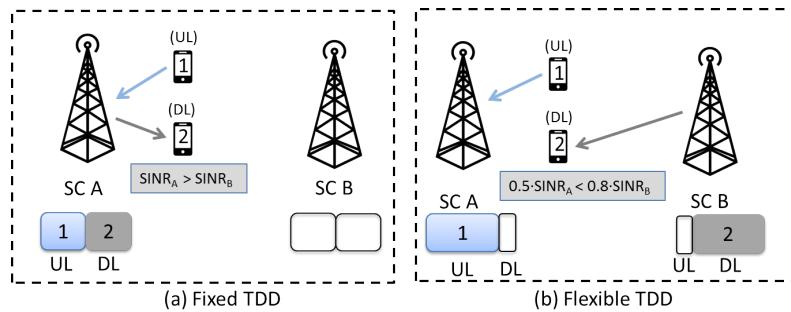


Figure 1.1: Interplay between user association and TDD configuration.

The previous example, while oversimplified, helps illustrate some of the dependencies at hand. One important omission in the above example, is that we ignored the DL-to-UL cross interference that might arise if nearby BSs have different schedules (see Fig. 1.1). E.g. a macro-cell transmitting on the DL, can really hurt a nearby SC transmitting on the UL [77]. Hence, excessive liberty in tuning UL and DL resources might hurt rather than help.

To this end, our main goal in a recent paper was to use the above framework to jointly optimize user association and TDD allocation per BS in order to:

- Associate users with BSs to optimize a chosen user- or network- centric performance metric (e.g. spectral efficiency, load-balancing, etc.).
- Choose the TDD UL/DL configuration for each SC to best match the UL/DL traffic demand for that metric.
- Consider the TDD UL/DL configuration of nearby SCs to avoid cross-interference.

The respective paper is [78]

- N. Sapountzis, T. Spyropoulos, N. Nikaein, and U. Salim, “*Joint Optimization of User Association and Dynamic TDD for Ultra-Dense Networks*,” in Proc. of IEEE INFOCOM 2018,

and the related contributions where the following:

**Contribution 6.4** We modified our analytical framework further, in order to include the possibility to tune the DL/UL schedule per BS, as well as capture cross-interference from nearby schedule discrepancies.

**Contribution 6.5** We showed that the joint problem is non-convex in general, and propose a primal decomposition algorithm that reduces complexity and can be implemented in a distributed manner. We then prove that this algorithm converges to the optimal solution of the joint problem.

**Contribution 6.6** Using simulations, we show our approach can *concurrently* improve UL and DL performance compared to the state of the art, showing more than  $2\times$  aggregate improvement, in the scenarios considered.

### 1.2.6 Mobile Edge Caching (Chapter 7)

As already argued in the previous chapters, a promising and highly-investigated way of dealing with the tremendous increase in traffic demand is via the deployment of many small cell base stations (e.g pico/femto-cells, or out-of-band WiFi APs) together with the standard macro-cellular network. The bulk of current state of the art for small cell solutions tackle mainly aspects of self-organizing radio resource management, inter-cell and interference coordination (ICIC), energy-efficiency and MIMO techniques, as well as traffic steering techniques like the ones considered in the last two chapters. A common problem for a lot of work on small cell networks is the impact of the backhaul network.

While the reduced distances to the user and better spatial reuse, offered by small cells, can greatly increase the (radio) capacity in the last hop, in order to support high cell densification the capacity of the backhaul network needs to scale at unrealistic rates. As explained, small cells are expected to be mostly connected by inexpensive wired or wireless links, and thus will often be under-provisioned, according to the Small Cell Forum [79]. This makes the backhaul network a potential new bottleneck that can significantly reduce the envisioned gains by cell densification.

In the previous chapter, we investigated how to take such backhaul bottlenecks into account, even when optimizing radio access functions like user association. While this approach can redirect some traffic to different BSs, possibly connected with less congested transport links, some important issues remain: (a) this is not always possible or it introduces considerable suboptimalities on the radio access (e.g., in terms of spectral efficiency); (b) inter-cell interference coordination (ICIC) techniques and Network MIMO coordination occurs over the backhaul and further taxes such links; (c) the transport network (backhaul/fronthaul) might simply not be able to support the aggregate radio traffic, for any routing path(s). To further alleviate the backhaul bottleneck, but also to reduce content access latency, the idea of edge caching has been proposed [80, 81, 82, 83].

The increasing demand for Internet content, and the inherent overlap between user data requests, suggests that storing popular content closer to the user could: (a) avoid congesting the capacity-limited backhaul links by avoiding duplicate backhaul transmissions, and (b) reduce the content access latency. Specifically, in [80], local content caching has been identified as one of the five most disruptive enablers for 5G networks, sparking a tremendous interest of academia and industry alike, as a cost-efficient way of tackling the data tsunami. Using real measurement data, [84] has explored the potential benefits of forward caching inside the core of a 3G network (e.g. in the SGSNs or GGSNs), suggesting that hit ratios up to 33% can be achieved for such caches. Similar benefits based on real workloads have been also found for LTE networks [85]. While these works already demonstrate the potential benefits of caching, a number of key questions remain open.

**Edge cache size can be a major bottleneck:** First, the above measurement-based studies have only considered caching in core nodes or backhaul aggregation points. Such nodes see a significant amount of the global network traffic and could also be equipped with large storage capacities (comparable to the size of regular CDN servers). It remains widely open whether similar benefits could be achieved with small, local caches at each BS and small cell. Such caches would have a much smaller storage capacity due to CAPEX/OPEX cost reasons: each core node considered as a potential caching location in the above studies would correspond to 100s or 1000s of small cells in future ultra-dense networks. As a result, the storage space per local edge cache must be significantly smaller to keep costs reasonable. This suggests that only a tiny fraction of the constantly and exponentially increasing content catalog could realistically be stored at each edge cache. In most practical cases, a local edge cache would fit less than 0.1 – 0.01% of the entire Internet catalog. E.g., a typical torrent catalog is about 1.5PB and the entire Netflix catalogue about 3PBs. More than 1TB storage would be needed to just store 0.1% of either one [86]. Considering other VoD platforms, YouTube, etc., this number explodes. Even, with a skewed popularity distribution, *local* cache hit ratios would be rather low [87, 88, 89].

**Convergence of cooperative caching and cooperative communication:** Second, a number of technological advances are envisioned for future 5G networks, such as Coordinated Multi-Point (CoMP) transmission, Coded Caching, and the CloudRAN architecture. These technologies are considered as major drivers to improve the performance on the radio access network (RAN) in 5G, but also introduce novel complications and challenges for caching algorithms. To cooperatively transmit a requested

content from multiple BSs, so as to improve radio access performance (e.g. through diversity gains), that content must be available in the caches of all BSs involved in a coordinated transmission. Otherwise, the effective rate of one or more of the links might be severed by a congested backhaul link. Yet, storing the same content in multiple nearby BSs means not storing other contents there, thus leading to a considerable decrease in cache hit rates and increased backhaul costs, if those contents are requested instead. This creates a novel tradeoff: how could a caching policy facilitate cooperative transmission opportunities (and radio access performance) while maximizing local cache hit rates (and backhaul performance)?

**Application-dependent caching optimization** Third, specific applications, especially content streaming, are beginning to dominate the mobile data traffic (e.g., more than 50% percent of the total traffic is consistently reported to be streamed video), and this is not expected to change in the future. Any means to reduce network traffic through caching should thus pay particular attention to this application. This raises the question: could application-driven optimizations targeting video traffic further improve the performance of edge caching?

### Chapter Contributions

The goal of this last chapter is to highlight a subset of our recent contributions towards answering these questions. We would like to also note that the topic of this chapter, namely mobile edge caching, is one of the main directions of our ongoing and planned research.

A first set of contributions is motivated partly by some shortcomings of HetNets, and partly by the delay-tolerance related work exposed in early chapters. Installing a large enough number of small cells with local storage (SCs), to ensure every user equipment (UE) has access to at least one such low range BS, still requires high CAPEX/OPEX costs (e.g. rental costs, maintenance, backhaul). Combined with the stagnating revenues per user, this makes operators reluctant to considerable densification. The more radical proposal to use UEs for storage and relaying (e.g., through Device-to-Device (D2D) communication) [90, 91] largely avoids these costs, but is stifled by device resource constraints (especially battery) and privacy concerns. As a result, SC storage and/or D2D might offer small performance improvements. E.g., a user requesting a content  $k$  might not have an SC in range, in which case  $k$  must be fetched over an expensive macro-BS. Even if the user has some SCs (or other UEs) in range, these might not in fact store content  $k$  (due to their relatively small size compared to the content catalog). Fetching content  $k$  will again be costly.

To improve this situation, one could exploit the mobility of UEs or even of BSs, to tradeoff access delay for cache hit ratio. This, in essence, is a natural follow-up of the ideas of Chapters 2 to 4, adapted for regular networks. The differences here are: (i) that the opportunistic (D2D) communication is only a single hop long; (ii) unlike most of the work there, the D2D part is supported by regular infrastructure; (iii) the infrastructure (operator) has control over (and knowledge of) the D2D part.

For example, if a user  $i$  requests a content  $k$ , which is not “locally” available, the operator can ask the user to wait for some time (e.g. up to some  $TTL$ ). During that time, user  $i$  may encounter some other cache storing  $k$  (either because  $i$  moved, or

because the cache moved, as in the case of mobile relays [92]). E.g., consider a simple example where there are  $M$  total edge caches, and intermeeting time between a user and such a cache is IID with rate  $\lambda$ . Assume that in the baseline (“immediate access”) scenario, a user sees on average  $c$  of these  $M$  caches, where  $c < 1$  in most cases (since SC coverage is limited). In the “delayed access” case, a user sees an additional  $M \cdot \lambda \cdot TTL$  caches, on average. Hence, the *effective* cache size that the average user sees is approximately  $M \cdot \lambda \cdot TTL$  times larger, which may lead to more cache hits.

The challenge here is to optimally allocate this larger (effective) cache space among all contents. E.g. if a node saw at most one cache (e.g. in the static scenario), it is well known that the optimal policy is to store the most popular contents (that fit) in every cache. However, it is easy to see that this is suboptimal in the delayed access case. If the size of each cache is  $C$  contents, and node  $i$  encounters 2 caches during a  $TTL$ , storing contents 1 to  $C$  in one cache, and  $C + 1$  to  $2C$  in the other, would lead to much higher hit rates *for that node* than storing contents 1 to  $C$  in both. While this example provides some intuition how the cache policy could exploit mobility (and the additional cache space it allows node to “see”), finding the optimal policy depends on the mobility characteristics, content popularities, and  $TTL$ .

In the following works, we have addressed these issues in the context of both delay-tolerant content download but also non delay-tolerant streaming [92, 93, 94]:

- L. Vigneri, T. Spyropoulos, and C. Barakat, “*Storage on wheels: Offloading popular contents through a vehicular cloud*,” in Proc. of IEEE WoWMoM 2016.
- L. Vigneri, T. Spyropoulos, and C. Barakat, “*Low Cost Video Streaming through Mobile Edge Caching: Modelling and Optimization*,” in IEEE Trans. on Mobile Computing, 2018.
- L. Vigneri, T. Spyropoulos, and C. Barakat, “*Quality of Experience-Aware Mobile Edge Caching through a Vehicular Cloud*,” to appear in IEEE Transactions on Mobile Computing, 2018.

The main contributions in these works are summarized below, and are described in detail at the beginning of the chapter:

**Contribution 7.1** We model the problem of maximizing the percentage of traffic offloaded through the vehicular cloud when users are downloading an entire content (before they can consume it). We do so, while controlling the user QoE (captured by a maximum delay or slowdown metric), and having two sets of control variables: (i) the number of replicas cached per content in vehicles; (ii) the deadline assigned to each content. Note that the slowdown metric attempts to control the total download delay *related to the file size* (e.g., an extra delay of 3 minutes might be very annoying when downloading a small file, but almost unnoticeable when the file download would anyway take some 10-20min).

**Contribution 7.2** We solve this problem presenting two variable deadline caching policies: QoE-Aware Caching (QAC) introduces a tight approximation on the generic formulation; QoE-Aware Caching for Small Content (QAC-SC) provides better offloading gains for content of small size but performs poorly as content size increases. Simulations with real traces for vehicle mobility and content popularity show that, in an urban

scenario, our system can achieve considerable offloading gains with modest technology penetration ( $< 1\%$  of vehicles participating in the cloud) with reasonable deadlines.

**Contribution 7.3** Instead of downloading the entire content, we consider a scenario where users stream the content in small chunks. We exploit the fact that *streaming of stored video content offers some delay tolerance “for free”*, in order to get some of the offloading benefits of the previous approach, *without any delay impact on the user*. We model the playout buffer dynamics, as new chunks are retrieved and old are played out, as a queueing system, and analyze the characteristics of its idle periods (during which access to the cellular infrastructure is required).

**Contribution 7.4** Based on this model, we formulate the problem of optimal allocation of content in vehicles, in order to minimize the total load on the cellular infrastructure. We provide closed-form expressions for the optimal allocation for two interesting regimes of vehicular traffic densities, assuming a relatively generic setting. We validate our theoretical results, and show that up to 50% of streamed data can be offloaded even in a realistic scenario, with modest technology penetration, *without any delay impact to the user*.

A second research thread in the area of edge caching was to try to understand and exploit the impact of recommendation systems. In an Internet which is becoming increasingly entertainment-oriented and recommendation-driven, *moving away from satisfying a given user request towards satisfying the user* could prove beneficial for caching systems. For example, when a user requests a content not available in the local cache(s), a recommendation system could propose a set of *highly related contents* that *are* locally available. If the user accepts one of these contents, an expensive remote access could be avoided. We will use the term *soft cache hit* to describe such scenarios.

Although many users in today’s cellular ecosystem might be reluctant to accept alternative contents, we believe there are a number of scenarios where soft cache hits could benefit both the user and the operator. As one example, a *cache-aware* recommendation system could be a plugin to an existing application (e.g., the YouTube app). The operator can give incentives to users to accept the alternative contents when there is congestion (e.g., *zero-rating* services [95, 96]) or letting the user know that accessing content  $X$  from the core infrastructure would be slow and choppy, while contents  $A, B, C, \dots$  might have much better performance. The user can still reject the recommendation and demand the original content. In a second example the operator might “enforce” an alternative (but related) content, e.g.: (i) making very low rate plans (currently offering little or no data) more interesting by allowing regular data access, except under congestion, at which time only locally cached content can be served; (ii) in developing areas [97] or when access to only a few Internet services is provided, e.g., the Facebook’s Internet.org (Free Basics) project [98, 99].

We believe such a system is quite timely, given the increased convergence of content providers with sophisticated recommendation engines (e.g., NetFlix and YouTube) and Mobile Network Operators (MNO), in the context of RAN Sharing [100, 101]. More importantly, the idea of soft cache hits is complementary and can be applied *on top* of existing proposals for edge caching, like the ones described earlier. As an example, in [102] we have considered the idea of soft cache hits on top of a setup with delayed access setup and mobile relays (like the one considered earlier). Although the topic of caching-recommendation interplay is an ongoing research topic for the author,

in this chapter we will focus on the following work:

- T. Giannakas, P. Sermpezis, T. Spyropoulos, and L. Vigneri, Soft Cache Hits: Improving Performance through Recommendation and Delivery of Related Content, to appear in IEEE JSAC special issue on Caching for Communication Systems and Networks, 2018.

The main contributions of this work are:

**Contribution 7.5** We introduce the novel concept of soft cache hits. To our best knowledge, this is the first time that this idea has been applied to edge caching for cellular networks. We propose a generic model for mobile edge caching with soft cache hits that can capture a number of interesting content substitution scenarios and is versatile enough to apply on top of both non-cooperative (i.e. single cache) and cooperative caching frameworks [81], as well as various networking assumptions.

**Contribution 7.6** We prove that the problem of optimal edge caching with SCH is NP-hard even when considering a single cache only. This is in stark contrast to the standard case without SCH. We then prove that, despite the increased complexity, the generic problem of femto-caching with SCH still exhibits properties that can be taken advantage of to derive efficient approximation algorithms with provable performance.

**Contribution 7.7** We corroborate our SCH proposal and analytical findings through an extended evaluation on 5 real datasets containing information about related content, demonstrating that promising additional caching gains could be achieved in practice.

Finally, in the last part of this chapter, we move away from the application itself and consider the radio access network when deciding what to cache. The common denominators between most of the works considered thus far can be summarized as follows:

- (i) The main bottleneck is the backhaul link (in the femto-caching setup) or the macro-cell link (in the vehicular cloud setup), (ii) the transmission phase is ignored (assuming requests are asynchronous and non-interfering) or simplified,
- (iii) caching gains are in terms of cache-hit ratio and amount of traffic offloaded away from expensive links.

Nevertheless, when considering a wireless setup, content delivery over the radio access link becomes just as important as the placement problem. If multiple nearby base stations (BS) have the same content cached, they can coordinate in order to improve performance on the radio access link. For example, several base stations can perform Joint Transmission (JT) to simultaneously transmit the same file to a single user, e.g. for power and diversity gains, which is particularly useful to edge users. Alternatively, multiple user requests could be satisfied in parallel by forming a MU-MIMO channel, between the BSs and users involved. Such techniques are often referred to as Coordinated Multi-point (CoMP) transmission [103]. With the data cached locally on each BS involved, only channel state information (CSI) information needs to be exchanged over the backhaul to coordinate the transmission, which is a much smaller burden, compared to exchanging whole video files. These ideas have led researchers to argue that caching and transmission algorithms at each involved BS must be jointly designed in

order to facilitate such CoMP opportunities, whether this is a distributed BS setup or cloudRAN scenario [104].

As a first step in this direction, we have focused on the JT technique for the radio access part. Every time the requested file is cached at several base stations in the user's range, the base stations can jointly transmit the file to the user. The transmission rate of JT is higher than that of each separate base station. Hence, storing the same (popular) files is optimal with respect to radio access transmission. On the other hand, storing different files in these base stations might lead to fewer cache misses and thus accesses to the backhaul network, which is important if the latter is the bottleneck. Our preliminary contribution in this direction is captured in:

- A. Tuholukova, G. Neglia, T. Spyropoulos, “*Optimal cache allocation for femto helpers with joint transmission capabilities*,” in Proc. of IEEE ICC 2017.

Our main contribution is the following:

**Contribution 7.8** We prove that the femtocaching problem with joint transmission (JT) capabilities is NP-hard, but its objective is monotonic and submodular. Based on this, we propose a polynomial time allocation algorithm that has a provable approximation ratio. Initial simulation results suggest that the optimal policy differs depending on whether the radio access is the bottleneck (caching then tends to reduce diversity in nearby caches, to facilitate JT) or the backhaul is the bottleneck (the cache allocation tends to store different content in overlapping base stations, to maximize the number of cache hits and minimize the number of contents fetched over the backhaul).

**Part I**

**Data Offloading Through  
Opportunistic Device-to-Device  
Networking**

## **Chapter 2**

# **Resource Management and QoS Provision in Delay Tolerant Networks**

## 2.1 Buffer Management for Best Effort Traffic

### 2.1.1 Motivation

Despite a large amount of effort invested in the design of efficient routing algorithms for DTNs, there has not been a similar focus on queue management and message scheduling. Yet, the combination of long-term storage and the, often expensive, message replication performed by many DTN routing protocols [11, 105] imposes a high bandwidth and storage overhead on wireless nodes [12]. As a result, it is expected that nodes' buffers, in this context, will often operate at full capacity. Similarly, the available bandwidth during a contact could be insufficient to communicate all intended messages. Consequently, *regardless of the specific routing algorithm used*, it is important to have:

- (i) efficient *drop policies* to decide which message(s) should be discarded when a node's buffer is full;
- (ii) efficient *scheduling policies* to decide which message(s) should be chosen to exchange with another encountered node when bandwidth is limited.

To this end, *the goal of our work has been to model the problem of buffer management and scheduling as an optimal resource allocation problem, and propose efficient yet practical algorithms.*

### 2.1.2 Related Work

In [13], Zhang et al. present an analysis of buffer constrained *Epidemic* routing, and evaluate some simple drop policies like drop-front and drop-tail. The authors conclude that drop-front, and a variant of it giving priority to source messages, outperform drop-tail in the DTN context. A somewhat more extensive set of combinations of *heuristic* buffer management policies and routing protocols for DTNs is evaluated in [106], confirming the performance of drop-front. In [107], Dohyung et al. present a drop policy which discards a message with the largest expected number of copies first to minimize the impact of message drop. However, all these policies are heuristic, i.e. not explicitly designed for optimality in the DTN context. Also, these works do not address scheduling.

RAPID was the first protocol to explicitly assume both bandwidth and (to a lesser extent) buffer constraints exist, and to handle the DTN routing problem as an optimal resource allocation problem. As such, it is the most related to our proposal, and we have compared our scheme(s) directly against it. Despite the elegance of the approach, and performance benefits demonstrated compared to well-known routing protocols, RAPID suffers from the following drawbacks: (i) its policy is based on suboptimal message utilities (we elaborate on this shortly); (ii) in order to derive these utilities, RAPID requires the flooding of information about all the replicas of a given message in the queues of all nodes in the network; yet, the information propagated across the network might arrive stale to nodes (a problem that the authors also note) due to change in the number of replicas, change in the number of messages and nodes, or if the

message is delivered but acknowledgements have not yet propagated in the network; and (iii) RAPID does not address the issue of signalling overhead. Indeed, in [19], the authors showed that whenever the congested level of the network starts increasing, their meta-data channel consumes more bandwidth. This is rather undesirable, as meta-data exchange can start interfering with data transmissions amplifying the effects of congestion. Our contributions in this chapter can be summarized as follows:

1. We formulate the optimal resource allocation problem and derive an optimal and distributed joint scheduling and drop policy, GBSD (Global knowledge Based Scheduling and Drop). While distributed, i.e. each node locally decides which of its messages to keep or forward, this policy assumes knowledge about the total number of copies per message, in order to derive the utility of each message in its buffer. It is thus not implementable in practice, and used as reference.
2. We propose a second policy, HBSD (History Based Scheduling and Drop) which is also distributed but is based on unbiased estimates of per message utilities based on collected statistics of network history.
3. We discuss how these statistics could be collected in practice, and the related overhead for this collection.
4. We extend the basic policy to consider different classes of messages with QoS requirements.

## 2.2 GBSD

Our network setup consists of  $L$  nodes, each able to store up to  $B$  messages in its buffer (we assume for simplicity equal message sizes). New messages are generated by each node randomly and all messages are unicast. Unless otherwise stated, we assume that inter-contact times between nodes are IID with meeting rate  $\lambda$  and have an exponential tail (we also discuss how this can be generalized). We summarize some useful notation in Table 2.1.

We first assume that our goal is to maximize the delivery rate of this network. Each message has a lifetime  $TTL_i$ , beyond which it is no longer useful to its destination and must be discarded. Buffer management consists of the following two actions:

**(Message Scheduling)** When node  $i$  encounters a node  $j$ , nodes  $i$  and  $j$  must exchange messages not in common (according to the epidemic routing protocol). As the contact between nodes  $i$  and  $j$  are of limited (and unknown) duration,  $i$  has to decide which messages to send first.

**(Message Drop)** If node  $j$  receives from  $i$  a new message, but its buffer is full, it needs to pick a message to drop.

The next theorem shows that both scheduling and message drop can be performed optimally, by deriving a marginal utility per message that a node can use to make local decision.

**Theorem 2.1.** *[GBSD Policy] Let  $K$  denote the total number of non-expired messages in the network, and  $T_i$  the elapsed time for message  $i$ . Let further  $n_i(T_i)$*

Table 2.1: Notation

Variable	Description
$L$	Number of nodes in the network
$K(t)$	Number of distinct messages in the network at time $t$
$TTL_i$	Initial Time To Live for message $i$
$R_i$	Remaining Time To Live for message $i$
$T_i = TTL_i - R_i$	Elapsed Time for message $i$ . It measures the time since this message was generated by its source
$n_i(T_i)$	Number of copies of message $i$ in the network after elapsed time $T_i$
$m_i(T_i)$	Number of nodes (excluding source) that have <i>seen</i> message $i$ since its creation until elapsed time $T_i$
$\lambda$	Meeting rate between two nodes
$B$	Buffer capacity per node

be the number of nodes who have a copy of the message at that time instant, and  $m_i(T_i)$  those that have “seen” the message (excluding the source) since it’s creation<sup>1</sup> ( $n_i(T_i) \leq m_i(T_i) + 1$ ). To maximize the average delivery rate of all messages, each node should use the following utility per message  $i$ ,

$$U_i(DR) = \left(1 - \frac{m_i(T_i)}{L-1}\right) \lambda R_i \exp(-\lambda n_i(T_i) R_i). \quad (2.1)$$

as follows: (i) transmit messages to an encountered node in decreasing order of utility; (ii) drop the lowest utility message when its buffer is full.

*Proof.* The proof can be found at the end of this chapter (Section 2.7).  $\square$

This utility can be viewed as the *marginal utility value* for a copy of a message  $i$  with respect to the total delivery rate.

Optimal (but different) per message utilities can also be similarly derived for the case of minimizing the average per message delivery delay (note that in that case we assume that there is no  $TTL$ ).

**Theorem 2.2.** *To minimize the average delivery delay of all messages, a DTN node should apply the GBSD policy using the following utility for each message  $i$ :*

$$U_i(DD) = \frac{1}{n_i(T_i)^2 \lambda} \left(1 - \frac{m_i(T_i)}{L-1}\right). \quad (2.2)$$

In other words, during each contact the nodes involved make a greedy decision based on marginal message utilities that locally optimize the expected delivery ratio (or expected delivery delay). Given the convexity of both problems, this can be seen as a distributed implementation of a gradient method for the centralized problem subject to buffer constraints, where during a contact only a subset of the control variables ( $n_i$ )

---

<sup>1</sup>We say that a node  $A$  has “seen” a message  $i$ , when  $A$  had received a copy of message  $i$  sometime in the past, regardless of whether it still has the copy or has already removed it from the buffer.

are improved (these are the variables corresponding only to the messages in a node's buffer).

Nevertheless the value of these utilities is a function of the global state of the message ( $n_i$  and  $m_i$ ) in the network. The current buffer state of remote nodes cannot be known in practice, due to the DTN and disconnected nature of such networks. Hence, this method cannot be known in practice and the above quantities must be estimated.

## 2.3 HBSD

For the purpose of our analysis, let us suppose that the variables  $m_i(T)$  and  $n_i(T)$  at elapsed time  $T$  are instances of the random variables  $N(T)$  and  $M(T)$ . We develop our estimators  $\hat{n}(T)$  and  $\hat{m}(T)$  so that when plugged into the GBSD's delivery rate and delay per-message utilities calculated earlier we get new per-message utilities (i.e. estimates of the actual utilities) that can be used by a DTN node without any need for global information about messages. This results in a new scheduling and drop policy, called HBSD (History Based Scheduling and Drop), a deployable variant of GBSD.

When global information is unavailable, one can calculate the average delivery rate of a message over all possible values of  $M(T)$  and  $N(T)$ , and then try to maximize it. In the framework of the GBSD policy, this is equivalent to choosing the estimators  $\hat{n}(T)$  and  $\hat{m}(T)$  so that the calculation of the average delivery rate is unbiased:

$$E\left[\left(1 - \frac{M(T)}{L-1}\right) * (1 - \exp(-\lambda N(T) R_i)) + \frac{M(T)}{L-1}\right] = \\ \left(1 - \frac{\hat{m}(T)}{L-1}\right) * (1 - \exp(-\lambda \hat{n}(T) R_i)) + \frac{\hat{m}(T)}{L-1}$$

Plugging any values for  $\hat{n}(T)$  and  $\hat{m}(T)$  that verify this equality into the expression for the per-message utility of Eq.(2.1), one can make sure that the obtained policy maximizes the average delivery rate. This is exactly our purpose. Suppose now that the best estimator for  $\hat{m}(T)$  is its average, i.e.,  $\hat{m}(T) = \bar{m}(T) = E[M(T)]$ . This approximation is driven by the observation we made that the histogram of the random variable  $M(T)$  can be approximated by a Gaussian distribution with good accuracy. To confirm this, we have applied the Lillie test [108], a robust version of the well known Kolmogorov-Smirnov goodness-of-fit test, to  $M(T)$  for different elapsed times ( $T = 25\%, 50\% \text{ and } 75\%$  of the TTL). This test led to acceptance for a 5% significance level. Consequently, the average of  $M(T)$  is at the same time the unbiased estimator and the most frequent value among the vector  $M(T)$ . Then, solving for  $\hat{n}(T)$  gives:

$$\hat{n}(T) = -\frac{1}{\lambda R_i} \ln\left(\frac{E\left[\left(1 - \frac{M(T)}{L-1}\right) \exp(-\lambda N(T) R_i)\right]}{\left(1 - \frac{\bar{m}(T)}{L-1}\right)}\right) \quad (2.3)$$

Substituting this expression into Eq.(2.1) we obtain the following new per message utility for our approximating HBSD policy:

$$\lambda R_i E\left[\left(1 - \frac{M(T)}{L-1}\right) \exp(-\lambda R_i N(T))\right] \quad (2.4)$$

The expectation in this expression is calculated by summing over all known values of  $N(T)$  and  $M(T)$  for past messages at elapsed time  $T$ . Unlike Eq.(2.1), this new per-message utility is a function of past history of messages and can be calculated locally. It maximizes the average message delivery rate calculated over a large number of messages. When the number of messages is large enough for the law of large numbers to work, our history based policy should give the same result as that of using the real global network information. Similar utility estimators can be derived for the case of delay minimization.

Finally, we note that  $L$ , the number of nodes in the network, could also be calculated from the statistics maintained by each node in the network. In this work, we assume it to be fixed and known, but one could estimate it similar to  $n$  and  $m$ , or using different estimation algorithms like the ones proposed in [109].

## 2.4 Simulation Results

### 2.4.1 Experimental Setup

To evaluate our policies, we have implemented a DTN framework into the Network Simulator NS-2 [110]. This implementation includes (i) the Epidemic routing protocol with *FIFO* for scheduling messages queued during a contact and *drop-tail* for message drop, (ii) the RAPID routing protocol based on flooding (i.e. no side-channel) as described, to our best understanding, in [19], (iii) a new version of Epidemic routing enhanced with our optimal joint scheduling and drop policy (GBSD), (iv) another version using our statistical learning based distributed algorithm (HBSD), and (v) the VACCINE anti-packet mechanism described in [13]<sup>2</sup>.

In our simulations, each node uses the 802.11b protocol to communicate, with rate 11Mbits/sec. The transmission range is 100 meters, to obtain network scenarios that are neither fully connected (e.g. MANET) nor extremely sparse. Our simulations are based on three mobility scenarios, a synthetic one, based on the Random Waypoint model and two real-world mobility traces: the first trace was collected as part of the ZebraNet wildlife tracking experiment in Kenya described in [111]. The second mobility trace tracks San Francisco's Yellow Cab taxis. Many cab companies outfit their cabs with *GPS* to aid in rapidly dispatching cabs to their costumers. The Cabspotting system [112] talks to the Yellow Cab server and stores the data in a database. We have used an API provided by the Cabspotting system in order to extract mobility traces<sup>3</sup>.

To each source node, we have associated a CBR (Constant Bit Rate) application, which chooses randomly from  $[0, TTL]$  the time to start generating messages of 5KB

---

<sup>2</sup>We have also performed simulations without any anti-packet mechanism, from which similar conclusions can be drawn.

<sup>3</sup>Note that this trace describes taxi's positions according to the *GPS* cylindrical coordinates (*Longitude*, *Latitude*). In order to uses these traces as input for the NS-2 simulator, we have implemented a tool [110] based on the Mercator cylindrical map projection which permit us to convert traces to plane coordinates.

Table 2.2: Simulation parameters

<b>Mobility pattern:</b>	RWP	ZebraNet	Taxis
Simulation's Duration(h):	7	14	42
Simulation' Area ( $m^2$ ):	3000*3000	3000*3000	-
Number of Nodes:	70	70	70
Average Speed (m/s):	2	-	-
$TTL(h)$ :	1	2	6
CBR Interval(s):	360	720	2160

Table 2.3: Taxi Trace & Limited buffer and bandwidth

<b>Policy</b>	GBSD	HBSD	RAPID	FIFO\DT
D. Rate(%)	72	66	44	34
D. Delay(s)	14244	15683	20915	36412

for a randomly chosen destination. We have also considered other message sizes (see e.g. [20]), but found no significant differences in the qualitative and quantitative conclusions drawn regarding the relative performance of different schemes<sup>4</sup>. Unless otherwise stated, each node maintains a buffer with a capacity of 20 messages to be able to push the network towards a congested state without exceeding the processing and memory capabilities of our simulation cluster. We compare the performance of the various routing protocols using the following two metrics: the average delivery rate and average delivery delay of messages in the case of infinite  $TTL$ <sup>5</sup>. Finally, the results presented here are averages from 20 simulation runs, which we found enough to ensure convergence.

#### 2.4.2 Performance evaluation for delivery rate

First, we compare the delivery rate of all policies for the three scenarios shown in Table 2.2.

Table 2.4: ZebraNet Trace & Limited buffer and bandwidth

<b>Policy</b>	GBSD	HBSD	RAPID	FIFO\DT
D. Rate(%)	68	59	41	29
D. Delay(s)	4306	4612	6705	8819

---

<sup>4</sup>In future work, we intend to evaluate the effect of variable message size and its implications for our optimization framework. In general, utility-based scheduling problems with variable sized messages can often be mapped to Knapsack problems (see e.g. [113]).

<sup>5</sup>By infinite TTL, we mean any value large enough to ensure almost all messages get delivered to their destination before the TTL expires.

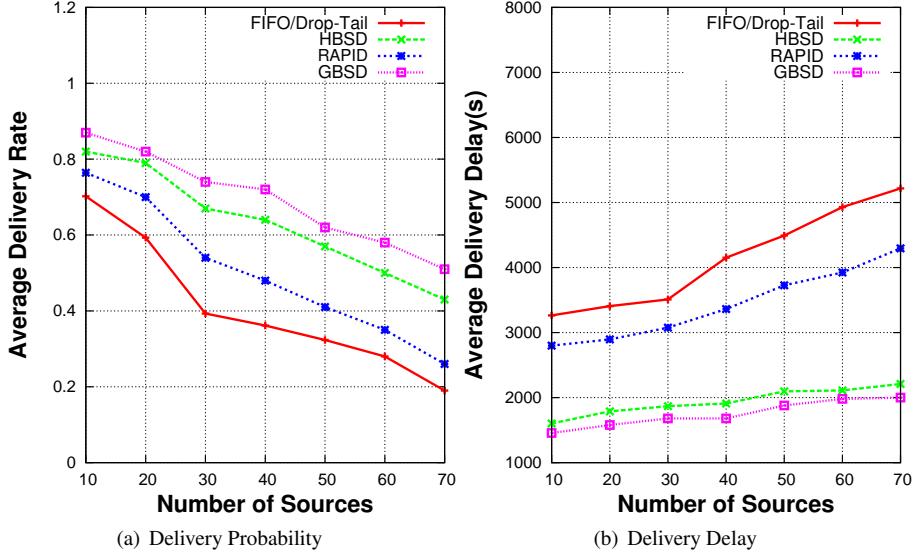


Figure 2.1: Delivery probability and delay for Epidemic Routing with different scheduling and drop policies.

Figure 2.1(a) shows the delivery rate based on the Random Waypoint model. From this plot, it can be seen that: the GBSD policy plugged into Epidemic routing gives the best performance for all numbers of sources. When congestion-level decreases, so does the difference between GBSD and other protocols, as expected. Moreover, the HBSD policy also outperforms existing protocols (RAPID and Epidemic based on FIFO/drop-tail) and performs very close to the optimal GBSD. Specifically, for 70 sources, HBSD offers an almost 60% improvement in delivery rate compared to RAPID and is only 14% worse than GBSD. Similar conclusions can be also drawn for the case of the real Taxi traces or ZebraNet traces and 70 sources. Results for these cases are respectively summarized in Table 2.3 and Table 2.4.

#### 2.4.3 Performance evaluation for delivery delay

To study delays, we increase messages' TTL (and simulation duration), to ensure almost every message gets delivered, as follows. Random Waypoint: (duration 10.5h, TTL = 1.5h). ZebraNet: (simulation duration = 28h, TTL = 4h). Taxi trace: (simulation duration = 84h, TTL = 12h). Traffic rates are as in Section 2.4.2.

For the random waypoint mobility scenario, Figure 2.1(b) depicts the average delivery delay for the case of both limited buffer and bandwidth. As in the case of delivery rate, GBSD gives the best performance for all considered scenarios. Moreover, the HBSD policy outperforms the two routing protocols (Epidemic based on FIFO/drop-tail, and RAPID) and performs close to GBSD. Specifically, for 70 sources and both limited buffer and bandwidth, HBSD average delivery delay is 48% better than RAPID

and only 9% worse than GBSD.

Table 2.3 and Table 2.4 show that similar conclusions can be drawn for the delay under respectively the real Taxi(s) and ZebraNet traces.

## 2.5 Buffer Management for Traffic Classes with QoS Requirements

The proposed algorithms considered thus far assume all end-to-end sessions (and, thus, messages) to be of equal importance. In many envisioned scenarios, network nodes might be running multiple applications in parallel. In this context, ensuring successful data delivery and/or minimizing the delivery delay may be more important for one DTN application than for another. Consider the example of a military operation where we have two applications launched concurrently at the DTN nodes: one reporting position information of friendly forces periodically and another one generating mission debriefings less frequently. We can consider that the delivery delay requirement for the first one is lower than the second one, since, after some time, a reported position may be stale. On the contrary, ensuring that a single mission debriefing message is delivered successfully may be more important than losing some (out of many) position updates. It is thus reasonable to assume that different messages might have different QoS requirements and resource allocation decisions should take these into account.

To model this setup we make some additional assumptions. We assume there are  $C$  different traffic classes different QoS requirements. E.g., focusing on the case of delivery ratio, class  $k$  is assumed to have a minimum acceptable delivery probability  $P_{QoS}^{(k)}$  (a class  $k$  can also be “best effort”, i.e.  $P_{QoS}^{(k)} = 0$ ). Similarly, we use superscript  $(k)$  to refer to a specific quantity for class  $k$  (e.g.  $n_i^{(k)}$ ).

Within this context, our goal is to modify the previous optimization framework with the following objectives:

**Feasible Region:** If there are enough resources in the network,

- Obj. 1 To ensure that the achieved delivery probability for every class  $k$  is at least as high as  $P_{QoS}^{(k)}$ .
- Obj. 2 Provided that minimum delivery ratios are achieved, any remaining resources are allocated towards improving the *total* delivery rate (across classes).

**Infeasible Region:** If there are not enough resources to fulfill *Obj. 1* then

- Obj. 3 Allocate the resources, so as to satisfy the requirements of higher priority classes (without loss of generality we assume priority is decreasing with  $k$ )

Objective (1) above suggest that satisfying QoS requirement of each class is a hard constraint. Objective (2) implies that beyond satisfying the QoS, extra resources should be allocated to messages which can most benefit by these resources. Given the decreasing nature of message marginal utilities (with respect to the number of copies), shown earlier, this results as we shall see to a max-min allocation policy of remaining resources. Finally, Objective (3) says that, if there are not enough resources, then class

priorities are absolute and all resources should be greedily allocated to the highest priority classes. The following formulates the above objectives into an optimization problem.

**Definition 2.5.1** (Buffer Management with QoS Guarantees). *Consider a network snapshot at time  $t$  with  $K$  live messages, and  $R_i$ ,  $T_i$   $m_i^{(k)}$ , being the remaining time, elapsed time, and number of nodes that have seen message  $i$  after elapsed time  $T_i$ , respectively (superscript  $(k)$  denotes that a quantity refers to class  $k$ ). If the following optimization problem has a feasible solution  $n_i^{(k)}$ , then any optimal solution satisfies Obj.1 and Obj.2.*

$$\max_{n_i^{(k)}} \sum_{i=1}^{K(t)} \left(1 - \frac{m_i^{(k)}}{L-1}\right) \cdot P_i^{(k)}(T_i), \quad (2.5)$$

$$g_k(n_i^{(k)}, m_i^{(k)}) = \frac{m_i^{(k)}}{L-1} + \left(1 - \frac{m_i^{(k)}}{L-1}\right) \cdot P_i^{(k)}(T_i) \geq P_{QoS}^{(k)} \quad \forall i \in \text{class } k, \quad (2.6)$$

$$P_i^{(k)}(T_i) = 1 - \exp(-\lambda n_i^{(k)}(T_i) \cdot R_i). \quad (2.7)$$

$$L \cdot B - \sum_{k=1}^C \sum_{i=1}^{L_k(t)} n_i^{(k)} \geq 0, \quad (2.8)$$

$$L - n_i^{(k)} \geq 0 \quad \forall i, k, \quad (2.9)$$

The key additional constraint, compared to the original best effort problem is Eq.(2.6). It says that the delivery probability is 1 if message  $i$  has been delivered already (with probability  $1 - \frac{m_i^{(k)}}{(L-1)}$ ) or  $P_i^{(k)}(t)$  if it hasn't (this probability depends on the number of allocated copies  $n_i^{(k)}$ ) if it hasn't, and the sum of both should be at least as large as desired QoS for this message's class. This constraint is convex on  $n_i^{(k)}$ , so the centralized problem remains convex.

Eq.(2.7) is not a constraint per se (otherwise the problem would not be convex, since this is not affine), but just defines notation  $P_i^{(k)}(t)$  for brevity. If one could centrally and instantaneously choose all values for  $n_i^{(k)}$ , it is easy to see that Eq.(2.6) captures *Obj.1* and the objective captures *Obj.2*. Any interior-point method could solve this problem centrally. Nevertheless, as explained earlier, such a centralized policy cannot be implemented in our DTN context. Our goal instead is to ensure every node takes a drop or scheduling decision independently, in a distributed manner. Additionally, the above problem does not give any guarantees for when the problem is infeasible, and thus does not satisfy *Obj.3*.

Based on the above observations, we propose the following modified utilities, that nodes can use for both scheduling and dropping messages (as in the case of best effort traffic, considered earlier), towards achieving these objectives.

$$U_i^{(k)}(DR) = U_i(DR) \cdot \left[1 + \max \left\{0, c_k \left(P_{QoS}^{(k)} - g_k(n_i^{(k)}, m_i^{(k)})\right)\right\}\right], \quad (2.10)$$

where  $c_k$  is a large constant and  $c_k \gg c_l, \forall k > l$ .

The above utility function achieves the following: (i) It first pushes the solution back into the feasible domain for each QoS constraint (because the term inside the max is much higher than 1, when the constraint is violated). (ii) If there is a feasible solution (i.e. Eq.(2.6) can be satisfied for all messages of all classes), then it performs gradient ascent among feasible solutions (as in the problem without QoS constraints); this ensures *Obj.1-2* above are satisfied eventually. (iii) If there is no feasible solution to the problem, the requirement on constants  $c_k$  ensures that the available resources are allocated to highest priority class, till its constraint is satisfied, then remaining resources to 2nd highest priority class, and so forth.

This can be seen as a distributed implementation of the centralized problem of Def. 2.5.1, where the constraints of Eq.(2.6) are introduced in the objective as penalty functions. The types of penalty functions corresponding to the chosen utility are hard barrier functions, as they are 0 when the constraint is not violated but take a very high value when the constraint is violated even a little. During a contact then the nodes involved update only a subset of the control variables (independently and possibly in parallel with other pairs), corresponding only to the messages inside the two buffers.

Note that it is often more common to assume “soft” penalty functions in distributed implementations (gradually tightening the constraint), in order to ensure the algorithm does not get stuck on the border of the feasible region. However, the convexity of the problem together with the randomized nature of coordinate ascent here (the control variables updated at each step are random, and depend on the nodes that meet each other), a hard constraint like the above does not pose a problem. An implementation with soft penalty functions would be perhaps interesting when quantities  $n_i$  and  $m_i$  are in fact noisy estimates obtained as explained in the next section. We defer this to future work.

### 2.5.1 Performance Evaluation

To evaluate our policy we considered three priority classes, namely Expedited (highest), Normal and Bulk (lowest) (based on the terminology of the bundle protocol specification [114] regarding different QoS classes). The BDR results are presented for various values of total available buffer space in the network. We consider a setup where the nodes create bundles every  $1/r$  seconds, they meet each with exponential inter-contact times with a common rate, and they exchange their non-common bundle copies (and drop copies, if the buffer is full) according to the utilities of Eq.(2.10). We present some sample results, comparing the performance of our policy with two existing QoS-based buffer management policies, namely: ORWAR [115] and CoSSD [116]. More details about the simulation setup and additional results (e.g. with real traces) can be found in [23].

Based on Fig. 2.2, it is clear that our scheme outperforms ORWAR. For low buffer values (i.e.,  $< 500$  buffer spaces), all three classes achieve higher BDR with our scheme. ORWAR fails to capture even the required performance of the Expedited class, even when the resources are adequate to do so. For higher buffer availabilities, ORWAR’s expedited class reaches to higher BDR than the required threshold. However, this is not desired based on the previous discussion, as it comes at the cost of the other

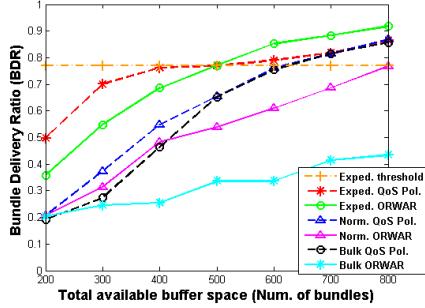


Figure 2.2: QoS Policy vs ORWAR

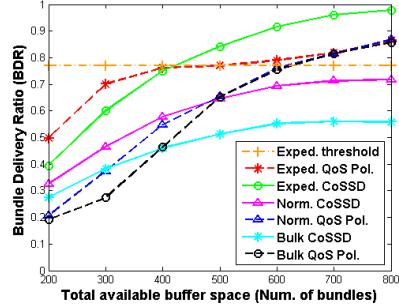


Figure 2.3: QoS Policy vs CoSSD

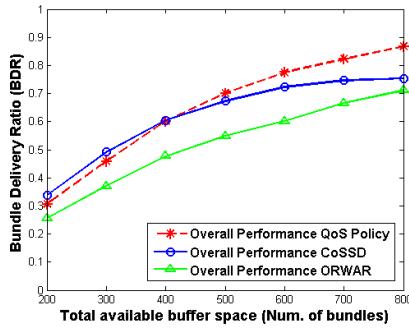


Figure 2.4: Overall policies comparison

two classes, whose performance is much lower than it could be. The superiority of our scheme is also captured by the overall network performance (Fig. 2.4, considering all classes), which is up to 20% higher with our policy, comparing to ORWAR.

In Fig. 2.3, the results of the comparison with CoSSD policy are shown. Inside the infeasible region ( $< 400$ ) the lower classes, as well as the overall performance (Fig. 2.4), are improved comparing to our policy. However, this comes at the cost of significant performance degradation for the Expedited class, which does not manage to reach its required performance threshold for the first values of Buffer sizes ( $< 400$ ). This is obviously contrary to the intended behavior, which dictates that our primal goal is to reach the desired performance for the expedited class. The relative behavior between the two compared policies changes inside the feasible region ( $> 400$ ). The Expedited class's BDR for CoSSD increases beyond its desired QoS threshold, without the lower classes having reached this threshold. As highlighted for the comparison with ORWAR, this is opposite to the optimal behavior. The consequence is that our policy outperforms CoSSD both in terms of lower classes, as well as overall network performance in this buffer availability region.

## 2.6 Implementation and Extensions

In addition to the above basic framework, we have also considered various extensions, generalizations, and implementation details. We summarize them here and refer the interested reader to the respective publications.

### 2.6.1 Statistics Collection

As explained earlier, the HBSD policy requires estimates for quantities  $n_i$  and  $m_i$  in order to derive the (a local estimate of) the utility of each message in the local buffer. However, due to the DTN nature of the network, it is not possible for a node which has a message  $k$  in its buffer, at some time  $t$ , to know how many other nodes currently have message  $k$  in their buffer as well at that time  $t$ . While one could try to use obsolete information obtained from times  $t - T$  for some  $T > 0$ , such information might be quite old and noisy (since the number of copies for a message might be small).

Instead, we choose to *estimate the expected value of variables  $n(t)$  and  $m(t)$ , by collecting data about the history of other, older messages after a similar elapsed time  $t$ .* Nodes keep track of their buffer status (message ID, elapsed time  $T$  tuples) and exchange these with encountered nodes when they meet (updating their known copy estimate). Hence, estimates of these expectations can be built despite the slow nature of DTN propagation, and these estimates are more robust, as they are based on many messages. More details about this statistics collection method and its overhead can be found in [21], which also contains some interesting insights on the behavior of these utilities as a function of elapsed time, for high traffic (“congested”) and low traffic regimes.

### 2.6.2 Generalized Inter-Contacts

The proposed methodology can be applied also for generic, i.e., non exponential intercontact times. E.g., assuming the time until the next contact between two nodes  $i$  and  $j$  is distributed according to  $F(t)$  ( $F(t) = 1 - \exp(\lambda t)$  in the previous analysis). For example, the utility function of Theorem 2.1 could be modified by replacing  $\exp(-\lambda n_i(T_i)R_i)$  with  $(\bar{F}(R_i))^{(n_i)}$ . We can even generalize to heterogeneous contact rates  $\lambda_{ij}$  between nodes, in relatively large networks. We will discuss heterogeneous contact rates in a more general setup in the next chapter. Details about heterogeneous contact rates for the problem of buffer management specifically, can be found in [23].

### 2.6.3 Messages with Heterogeneous Popularities

So far, we have also been assuming that each message has a unique destination and it is equally important with all other messages. In [117], we have extended some of the ideas in this chapter to a setup where each message might be of interest to multiple destinations. Our protocol, CEDO aims at maximizing the total delivery-rate of distributed content in a setting where a range of contents of different popularity may be requested and stored, but nodes have limited resources. It achieves this by extending the delivery-rate utility per content to account for different content popularities as well.

## 2.7 Proof of Theorem 2.1

The probability that a copy of a message  $i$  will not be delivered by a specific node, when meeting times are exponential, is equal to  $\exp(-\lambda R_i)$  (where  $R_i$  is the remaining time until the TTL expires:  $R_i = TTL - T_i$ ). Knowing that message  $i$  has  $n_i(T_i)$  copies in the network, and assuming that the message has not yet been delivered, we can derive the probability that the message itself will not be delivered (i.e. none of the  $n_i$  copies gets delivered):

$$P\{\text{message } i \text{ not delivered} \mid \text{not delivered yet}\} = \prod_{i=1}^{n_i(T_i)} \exp(-\lambda R_i) = \exp(-\lambda n_i(T_i) R_i). \quad (2.11)$$

Note that we have not taken into account that more copies of a given message  $i$  may be created in the future through new node encounters, or a copy of message  $i$  could be dropped within  $R_i$ . Nevertheless, the same assumptions are applied for all messages equally and thus can justify the relative comparison between the delivery probabilities for different messages.

We also need to consider what has happened in the network since the message generation, in the absence of an explicit delivery notification (this part is not considered in RAPID [19], making the utility function derived there suboptimal). Given that all nodes including the destination have the same chance to see the message (due to IID mobility), the probability that a message  $i$  has already been delivered is equal to:

$$P\{\text{msg } i \text{ already delivered}\} = m_i(T_i)/(L-1). \quad (2.12)$$

Combining the above two equations, the probability that a message  $i$  will get delivered before its TTL expires is:

$$P_i = \left(1 - \frac{m_i(T_i)}{L-1}\right) * (1 - \exp(-\lambda n_i(T_i) R_i)) + \frac{m_i(T_i)}{L-1}.$$

So, if we take at instant  $t$  a snapshot of the network, the global delivery rate for the entire whole network will be:

$$DR = \sum_{i=1}^{K(t)} \left[ \left(1 - \frac{m_i(T_i)}{L-1}\right) * (1 - \exp(-\lambda n_i(T_i) R_i)) + \frac{m_i(T_i)}{L-1} \right]$$

The above sum defines the expected *additional* contribution of the messages currently not expired to the total delivery ratio (messages already delivered, will not contribute to this ratio, hence the need for the first term in the parenthesis). In case of a full buffer or limited transfer opportunity, a DTN node should take respectively a drop or replication decision that leads to the best gain in the global delivery rate DR. To define this optimal decision, we differentiate DR with respect to  $n_i(T_i)$ , then we discretize and replace  $dn$  by  $\Delta n$  to obtain:

$$\begin{aligned}\Delta(DR) &= \sum_{i=1}^{K(t)} \frac{\partial P_i}{\partial n_i(T_i)} * \Delta n_i(T_i) \\ &= \sum_{i=1}^{K(t)} \left[ \left(1 - \frac{m_i(T_i)}{L-1}\right) \lambda R_i \exp(-\lambda n_i(T_i) R_i) * \Delta n_i(T_i) \right]\end{aligned}$$

Our aim is to maximize  $\Delta(DR)$ . In the case of message drop, for example, we know that:  $\Delta n_i(T_i) = -1$  if we drop an already existing message  $i$  from the buffer,  $\Delta n_i(T_i) = 0$  if we don't drop an already existing message  $i$  from the buffer, and  $\Delta n_i(T_i) = +1$  if we keep and store the newly-received message  $i$ . Based on this, GBSD ranks messages using the per message utility in Eq.(2.1), then schedules and drops them accordingly.

## **Chapter 3**

# **Performance Modeling for Heterogeneous Mobility and Traffic Patterns**

## 3.1 Performance under Heterogeneous Mobility

Epidemic spreading is probably one of the most popular bio-inspired principles that have made their way into computer engineering. Epidemic algorithms and variants (e.g. gossip) have been used for communication in distributed systems, synchronization of distributed databases, content searching in peer-to-peer systems, etc. As presented in the previous chapter, epidemic-based schemes have also been proposed for routing and data dissemination in *Opportunistic Networks* [118] or *Delay Tolerant Networks* (DTNs). In the previous chapter, we have considered how to modify epidemic algorithms for DTN, in order to efficiently deal with limited buffer capacity and contact bandwidth. In this chapter, we are interested instead in understanding the theoretical performance of such epidemic schemes and variants.

Since the mobility process of nodes involved (e.g. humans or vehicles carrying the devices) is, in most cases, not deterministic, the performance of epidemic algorithms (and variants) heavily depends on the underlying contact patterns between nodes. To this end, epidemic algorithms have been extensively studied through both simulations and analytical models. While simulations with state-of-the-art synthetic models or real mobility traces can provide more reliable predictions for the *specific* scenario tested, analytical models can give quick, qualitative results and intuition, answer “what-if” questions, and help optimize protocols (e.g. choosing the number of copies in [12], or gossip probability [13]).

For the sake of tractability, state-of-the-art analytical models for epidemic spreading mainly rely on simple mobility assumptions (e.g. Random Walk, Random Waypoint), where node mobility is stochastic and independent, identically distributed (IID) (see e.g. [119, 13, 120]). Nevertheless, numerous studies of real mobility traces [121, 122, 26, 36] reveal a different picture. One key finding is that contact rates between different pairs of nodes can vary widely. Furthermore, many pairs of nodes may never meet. This puts in question the accuracy and utility of these models’ predictions. Yet, departures from these assumptions [123, 124, 125, 122] seem to quickly increase complexity and/or limit the applicability of results. This raises the question: can we derive *useful and accurate* closed form expressions for the performance of epidemic schemes, under more generic mobility assumptions? In the next sections, we consider increasingly complex mobility, or more specifically contact models, and derive analytical expressions for the expected delay of epidemic routing in the respective (class of) networks.

### 3.1.1 Heterogeneous Contact Rates over Random Connectivity Graphs

**Definition 3.1.1** (Heterogeneous Full Contact Network). *The sequence of the contact events between each pair of nodes  $\{i, j\}$  is independent from other pairs, and given by a Poisson process with rate  $\lambda_{ij}$ . The contact rates  $\lambda_{ij}$  are independently drawn from an arbitrary distribution  $f_\lambda(\lambda)$ ,  $\lambda \in A \subseteq (0, \infty)$ , with finite mean  $\mu_\lambda$  and variance  $\sigma_\lambda^2$ .*

The above model is a generalization of the standard IID inter-contact model, used also in the previous chapter, which assumes  $\lambda_{ij} = \lambda > 0$  for all pairs. Different choices

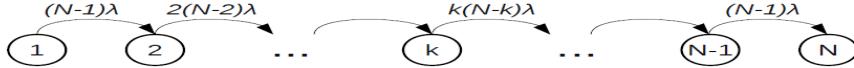


Figure 3.1: Markov Chain for epidemic spreading over a *homogeneous* network with  $N$  nodes

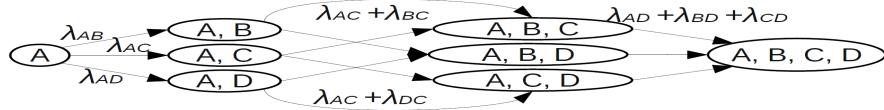


Figure 3.2: Markov Chain for epidemic spreading over a *heterogeneous* network with 4 nodes

of  $f_\lambda(\lambda)$  can describe a significantly broader range of scenarios. For example, large  $\sigma_\lambda^2$  values imply that the contact frequencies between different pairs are very heterogeneous, e.g. some pairs will rarely contact each other while others much more often. An  $f_\lambda(\lambda)$  symmetric around  $\mu_\lambda$  (e.g. uniform distribution) implies a balanced number of high and low contact rates, while a right-skewed  $f_\lambda(\lambda)$  (e.g. Pareto) describes a network with most pairs having large inter-contact times, but few contacting very frequently. Small  $\mu_\lambda$  values could correspond to slow moving nodes, e.g. pedestrians, (or large geographical areas). Finally, multi-modal  $f_\lambda(\lambda)$  functions might approximate scenarios with some hierarchical structure. Note that, conditional on  $f_\lambda$ , we get a class of random matrices  $\Lambda = \{\lambda_{ij}\}$ . Our goal is to analyze the expected performance of epidemic spreading across the possible instances in this class.

To understand the complexity of the problem, let us first assume the simple case of  $\lambda_{ij} = \lambda > 0$ . We can model epidemic spreading with a pure-birth Markov chain, as depicted in Fig. 3.1, where a state  $k$  denotes the number of “infected” nodes (i.e. nodes with the message). In this *homogeneous contact network*, it is easy to show that the *step time*  $T_{k,k+1}$  (i.e. the time to move from state  $k$  to state  $k+1$ ) is exponentially distributed with rate  $k(N - k)\lambda$  (due to independent Poisson pair-wise meetings. Its expected value is then given by  $E[T_{k,k+1}] = \frac{1}{k(N - k)\lambda}$ , and, therefore, one could straightforwardly calculate the expected spreading time (broadcast, anycast or unicast).

While we could still use a Markov chain for the heterogeneous contact network, in order to find  $E[T_{k,k+1}]$ , we now need to know *which nodes exactly* are included in the  $k$  infected nodes. As an example, in Fig. 3.2, we present the Markov Chain of a message epidemic spreading in a heterogeneous network with four nodes,  $\{A, B, C, D\}$ . This Markov Chain is composed of 15 states, whereas the respective Markov Chain of an homogeneous network with 4 nodes would be composed of only 4 states. Hence, it becomes evident that the complexity increases quickly, even for this simple 4-node network. In a network with  $N$  there will be  $\binom{N}{k}$  different states for step  $k$ , each with a potentially different probability.

**Lemma 1.** *The expected delay for the transition from step  $k$  to step  $k + 1$  is given by*

$$E[T_{k,k+1}] = \sum_{m=1}^{\binom{N}{k}} \frac{1}{S_k^m} \cdot P\{\mathbf{C}_k^m\}, \quad (3.1)$$

where  $P\{\mathbf{C}_k^m\}$  is the probability that  $\mathbf{C}_k^m$  is the set of infected nodes at step  $k$ , and

$$S_k^m = \sum_{i \in \mathbf{C}_k^m} \sum_{j \notin \mathbf{C}_k^m} \lambda_{ij} \quad (3.2)$$

While keeping track of the probabilities in the above lemma could be done recursively, the state space grows exponentially fast, so even numerical solutions [126] are infeasible beyond very simple problems. Instead, we prove that, in the limit of large  $N$  (number of nodes), the majority of such starting states become statistically equivalent and the approximation error from using the mean value for  $S_k^m$  goes to 0.

This is captured by the following main result. The proof is technical and can be found in [29]

**Theorem 1.1.** *As the network size increases, the relative error  $RE_k$  between the expected step delay  $E[T_{k,k+1}]$  and the quantity  $\frac{1}{k(N-k)\mu_\lambda}$  converges to zero in probability:*

$$\lim_{N \rightarrow \infty} RE_k = \lim_{N \rightarrow \infty} \frac{E[T_{k,k+1}] - \frac{1}{k(N-k)\mu_\lambda}}{E[T_{k,k+1}]} = 0. \quad (3.3)$$

In Table 3.1, we present the values for the relative error  $RE_k$  (Theorem 1.1) in synthetic simulation scenarios of different network sizes  $N$  and contact rates heterogeneity  $CV_\lambda = \left(\frac{\sigma_\lambda}{\mu_\lambda}\right)$ . The values in Table 3.1 correspond to the relative error  $RE_k$  averaged over all the steps  $k$  of the epidemic process and over 100 different network instances  $\Lambda$  with equivalent characteristics  $(N, f_\lambda)$ . It can be seen that in networks with higher heterogeneity ( $CV_\lambda$ ) the errors are larger, as our theory predicts. However, as the network size increases, the errors for all scenarios become very small.

The decrease of the relative errors can be observed also in Fig. 3.3, where we present the distribution (boxplots) of the values of  $RE_k$  in the different network instances. Here, the relative errors do not correspond to averaged (over different steps) values, but we present the  $RE_k$  at the steps that correspond at the 20% (e.g. in the scenario with  $N = 100$ , we present the relative errors in the step  $k = 20$ ) and 70% of the spreading process, in Fig. 3.3(a) and Fig. 3.3(b), respectively. These plots show that the error not only decreases on average, but it does so for almost all instances from that contact class.

While the above approximation becomes exact for large  $N$ , the error can be non-negligible for finite networks. For such networks we can derive a better, second order approximation using the Delta method known from statistics. Details can be found in [29].

**Lemma 2.** *For a heterogeneous contact network following Definition 3.1.1, the expected step delay can be approximated by*

$$E[T_{k,k+1}] = \frac{1}{k(N-k)\mu_\lambda} \cdot \left(1 + \frac{\sigma_\lambda^2}{k(N-k)}\right). \quad (3.4)$$

Table 3.1: Relative Step Delay Error  $RE_k$ : Averaged over All Steps and over 100 Network Instances

	$N = 50$	$N = 100$	$N = 200$	$N = 500$
$CV_\lambda = 0.5$	2.8%	2.7%	2.6%	2.5%
$CV_\lambda = 1$	4.2%	3.1%	2.7%	2.6%
$CV_\lambda = 1.5$	8.2%	4.6%	3.2%	2.6%
$CV_\lambda = 3$	34.1%	15.3%	8.2%	3.8%

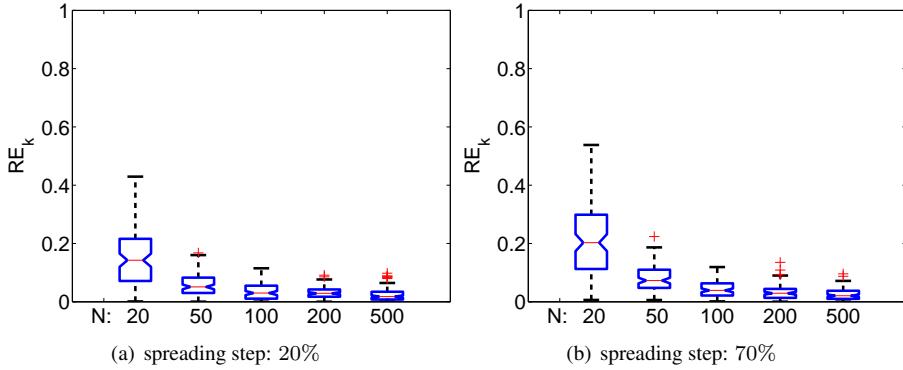


Figure 3.3: *Relative Step Error* for the step (a)  $k = 0.2 \cdot N$  (i.e. message spreading at 20% of the network) and (b)  $k = 0.7 \cdot N$ . Each boxplot correspond to a different network size  $N$  (with  $\mu_\lambda = 1$  and  $CV_\lambda = 1.5$ ). Box-plots show the distribution of the Relative Step Error  $RE_k$  for 100 different network instances of the same size.

The above results can be further generalized for networks where some nodes never meet. Specifically, we can assume an underlying Poisson connectivity graph with parameter  $p_s$ , where a percentage of randomly chosen pairs  $1 - p_s$  do not communicate at all (i.e.  $\lambda_{ij} = 0$ ), and the non-zero contact rates are drawn again according to Def.3.1.1. Then, the following Corollary holds:

**Corollary 1.** *Under a Heterogeneous Poisson Mixing Contact Network, where a contact pair either never meets (with probability  $1 - p_s$ ), or meets regularly (with probability  $p_s$ ) and according to Definition 3.1.1, all previous theoretical results hold, by substituting the moments of the contact rate distribution ( $\mu_\lambda$  and  $\sigma_\lambda^2$ ) with the expressions*

$$\mu_{\lambda(p)} = p_s \cdot \mu_\lambda, \quad (3.5)$$

$$\sigma_{\lambda(p)}^2 = p_s \cdot [\sigma_\lambda^2 + \mu_\lambda^2 \cdot (1 - p_s)] \quad (3.6)$$

With these basic results in hand, we can derive or approximate the performance of epidemic routing as well as other DTN routing variants (e.g. Spray and Wait). Details and additional plots for synthetic scenarios can be found in [29], corroborating our results.

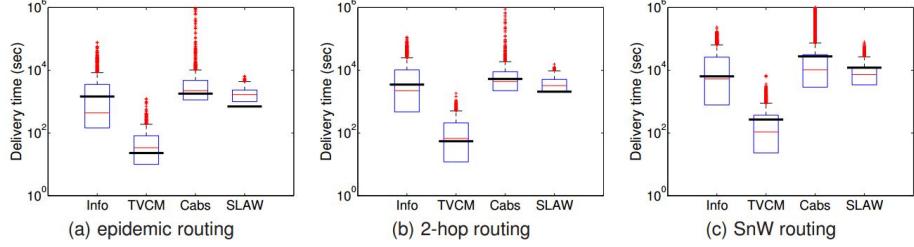


Figure 3.4: Box-plots of the message delivery delay under (a) epidemic, (b) 2-hop routing, and (c) SnW (with  $L = 6$  copies) routing. On each box, the central horizontal line is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually as crosses. The thick (black) horizontal lines represent the theoretical values predicted by our model.

In Table, we just show the analytical expressions for these protocols, for reference.

We also show sample performance results for real mobility traces and realistic mobility models. Analytical results against synthetic contact scenarios conforming to the assumptions in our model show remarkable accuracy, even for modestly sized networks, and are omitted here. The traces we consider are: (i) *Cabspotting* [127], which contains GPS coordinates from 536 taxi cabs collected over 30 days in San Francisco, and (ii) *Infocom* [128], which contains traces of Bluetooth sightings of 78 mobile nodes from the 4 days iMotes experiment during Infocom 2006. In addition to these real traces, we generated mobility traces with two recent mobility models that have been shown to capture well different aspects of real mobility traces, namely, TVCM [129] and SLAW[130]. In order to compare with analysis, we parse each trace and estimate the mean contact rate for all pairs  $\{i, j\}$ . We then produce estimates for the 1<sup>st</sup> and 2<sup>nd</sup> moments of these rates,  $\hat{\mu}_\lambda$  and  $\hat{\sigma}_\lambda^2$ , as well as the percentage of connected pairs  $\hat{p}$  and use them in our analytical expressions.

Fig.3.4 compares the theoretical performance of three protocols, epidemic routing, 2-hop routing, and spray and wait (SnW), according to our analytical expressions, to simulation results. Source-destination pairs are chosen randomly in different runs and messages are generated in random points of the trace. In the next chapter, we will consider scenario where traffic demand is not random among pairs of nodes.

The first thing to observe is that delay values span a wide range of values for different source-destination pairs. This implies a large amount of heterogeneity in the “reachability” of different nodes. Our analytical predictions are shown as thick dark horizontal lines. As it can be seen, our result is in most cases close to the median and in almost all cases between the 25<sup>th</sup> and 75<sup>th</sup> percentile of the delay observed in both the real traces and mobility models.

It is somewhat remarkable that our delay predictors are close to the actual results (qualitatively or even quantitatively in some cases) in a range of real or realistic scenarios; studies of these scenarios reveal considerable differences to the much simpler contact classes for which our results are derived. We should also be careful not to jump

to generalizations about the accuracy of these results in all real scenarios, as we are aware of situations that could force our predictors to err significantly. Nevertheless, we believe these results are quite promising in the direction of finding simple, usable analytical expressions even for complex, heterogeneous contact scenarios.

### 3.1.2 An Upper Bound for Generic Contact Matrices

While the above results are applicable to many different contact matrices  $\Lambda$  the entries of these matrices cannot be arbitrary, but are drawn from the same distribution. This is what allows us to invoke law of large number types of arguments when the network instances become large. However, these do not apply to arbitrary contact matrices, and thus arbitrary underlying mobility processes. In some real networks, the contact matrix  $\Lambda$  tends to exhibit clear community structure, which cannot be well captured by random models, even heterogeneous ones, like the ones considered earlier.

To understand why the above results might not hold, consider a network with 2 communities, which grow as  $N$  grows, but the number of links between community 1 and 2 remain finite, say  $m$ . This suggests that at some step  $k$ , the number of ways we can go from  $k$  to  $k+1$  infected nodes will not grow with  $N$ , and we cannot invoke the law of large numbers, e.g. approximating the infection rate between the communities with  $m \cdot \mu_\lambda$  (if  $m$  is small). It is easy to see that the “bottleneck” delay for this network is for the infection to spread between the two communities, and the delay of that step should depend on the total contact rate across the *cut* between the communities. This can be generalized by converting a given contact matrix  $\Lambda$  into an appropriate Markov chain and using transient analysis to derive the following result.

**Theorem 1.2** (Delay Bound.). *Let  $S$  denote a “cut” of an arbitrary contact graph  $\Lambda$ . Let further  $\phi(S)$  denote the conductance of this cut<sup>1</sup>*

$$\phi(S) = \frac{\partial(S)}{|S| \cdot |\bar{S}|} = \frac{\partial(\bar{S})}{|S| \cdot |\bar{S}|} = \phi(\bar{S}), , \quad (3.7)$$

where  $|S|$  denotes the cardinality of the set  $S$ , and  $\partial(S) = \sum_{i \in S, j \notin S} \lambda_{ij}$ . Finally, let  $\Phi$  be defined as  $\Phi = \min_{S \in \Omega} \phi(S)$ . Then, the expected delay of epidemic spreading over contact network  $\Lambda$  can be upper bounded as follows:

$$\mathbb{E}[D_\Lambda] < \frac{2 \ln(N-1)}{N\Phi}, . \quad (3.8)$$

The above result is generic, and can be further elaborated to derive the distribution of the delay. More importantly, the conductance of a graph, which essentially corresponds to finding a *balanced min cut*, relates to the spectral properties of matrix  $\Lambda$ , namely the 2nd eigenvalue of the Laplacian through Cheeger’s inequality. The proof and more details can be found in [31].

Figs. 3.5(a)–3.5(d) show the empirical CCDFs for the spreading delays in comparison with our theoretical CCDFs, obtained from Eq. (3.8). We also depict a tighter

---

<sup>1</sup>This is one of several definitions of conductance to be found in the literature. We use it as it provides the best bound.

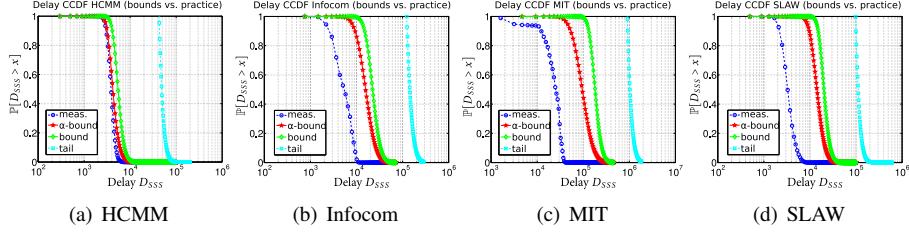


Figure 3.5: Prediction accuracy for the CCDF of the epidemic delay on real networks

(but more complex to calculate) bound based on min cuts for each step  $k$  denoted as “a-bound” as well as the tail bound obtained by Shah in [131], the best known bound up to that time. The empirical results are well within all the predicted bounds. Note that the tail bound is at least one order of magnitude larger than the bounds we obtained here.

A relatively loose bound, as in the case of MIT attests the presence of at least two strong communities connected through a relatively small set of edges (i.e., small conductance), as observed in [40]. On the other hand, for the community-free traces Infocom and SLAW [40], as well as for the HCMM trace (which has built-in very small communities of 10 nodes each) the bounds are, as expected, (much) closer to the empirical results. We will revisit some of these traces, in terms of their contact graph properties, in Ch. 4.

With the results of Section 3.1.1 and this section, we are now able to approximate the performance of epidemic routing and variants in a very large class of contact networks, and thus underlying mobility properties. The only remaining properties observed in real traces that are not explicitly captured, is potential dependence between pair-wise contact properties and non-stationarity (e.g. periodicity). The latter could be captured by identifying periods in the mobility model or trace, and apply different contact matrices  $\Lambda$  for each. Pair-wise dependence is significantly more complex to be captured and lead to closed-form expression. Nevertheless, appropriate bounds could be sought for, if some knowledge about the type of dependence is known. Finally, the fact that  $\Lambda$  is in principle a random matrix, and epidemic performance depends on the eigenvalues of this matrix, *random matrix theory* could be a useful tool to further elaborate on these results [132].

## 3.2 Performance under Heterogeneous Traffic

### 3.2.1 Communication Traffic

In all previous analyses we have assumed that any source-destination pair is equally likely to generate a message. We did this explicitly in simulations, and we also implicitly assumed it in our theory (as we shall explain). Nevertheless, in addition to who contacts whom and how often (i.e. mobility and the resulting contact process), another major question that should be raised in opportunistic networks (but rarely is) is *who wants to communicate with whom and how much traffic do they exchange?*

Intuition suggests that every pair of nodes will not exchange the same amount of traffic. To support intuition, studies from fields related to technological and social networks [133, 134, 135] have demonstrated the existence of heterogeneous traffic patterns. The same studies further suggest that this heterogeneity depends on the *spatial* and *social* characteristics of these networks. Since *location-based services* [136] and *social networking* [137] are considered among the major applications supported by opportunistic networks, such traffic dependencies on social and/or spatial factors are very probable to appear. What is more, mobility characteristics have also been found to depend on spatial and social characteristics [138, 139, 140]. This clearly seems to argue for a non-homogeneous traffic model. Moreover, traffic and mobility in such networks are expected to exhibit some correlations [133, 134].

Before we proceed to choose a traffic model, one should consider the following questions: *Would the mere heterogeneity of traffic suffice to affect performance? Is it necessary to consider traffic and mobility correlations?*

As stated earlier, information dissemination is determined by the sequence of contact events. Hence, if traffic characteristics are independent of node mobility, one might expect a limited impact on performance.

Towards examining the validity of the above argument, we decided to compare the performance of some well-known opportunistic protocols (direct transmission [10], spray and wait [12], 2-hop routing [119], and SimBet [35]) through simulations on two real traces, for three traffic scenarios: (i) homogeneous traffic: every pair of nodes has the same chance of being chosen as the source-destination pair for the next message; (ii) heterogeneous traffic that is *mobility independent*: we assign randomly to each pair a different end-to-end traffic demand (with the normalized message generation rate for a pair drawn uniformly in  $[1, 1000]$ ); (iii) heterogeneous traffic that is *mobility dependent*: end-to-end traffic between two nodes is proportional to their contact rate. We generated an equal (sufficiently large) number of messages for all scenarios.

Results for the mean message delivery delay are shown in Fig. 3.6. As is evident from these results, when traffic heterogeneity is independent of mobility (middle bar), the average delay is practically the same to the homogeneous case (left bar), *for all protocols, and across all scenarios* (including additional ones we have tried). In contrast, when traffic is heterogeneous *and* correlated with the contact rates (rightmost bar), Fig. 3.6 shows a clear difference in average delay for all scenarios and protocols. These results provide an initial answer to the above questions. ***It is not traffic heterogeneity itself that affects performance, but rather the joint effect of mobility and traffic (heterogeneity).***

In other words, unless differences in traffic demand correspond also to differences in contact frequency (e.g. frequently meeting pairs tend to also consistently generate more/less traffic for each other), end-to-end performance will not be affected. This statement is also formally proven in [33].

The above observation, together with the initial insight coming from real datasets, motivates us to propose the following simple, yet quite generic, model for end-to-end traffic.

**Definition 3.2.1** (Heterogeneous Communication Traffic). *The end-to-end traffic demand (per time unit) between a pair of nodes  $\{i, j\}$ , is a random variable  $\tau_{ij}$ , such that*

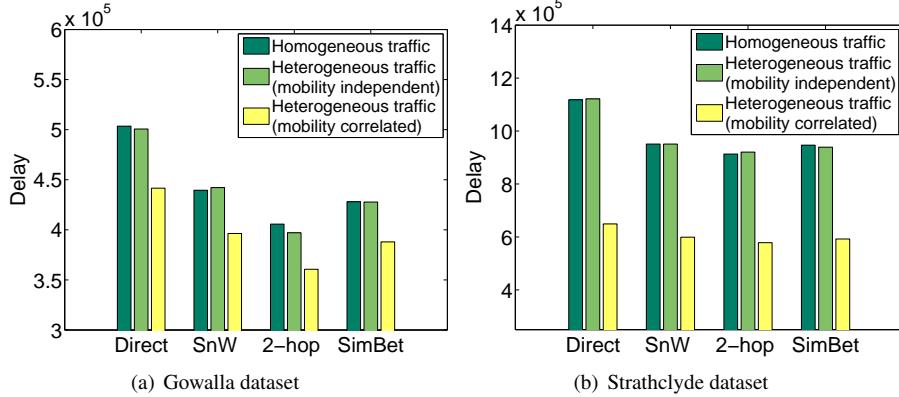


Figure 3.6: Mean delivery delay of 4 routing protocols, namely *Direct Transmission*, *Spray and Wait* (*SnW*), *2-hop*, and *SimBet*, on the (a) Gowalla and (b) Strathclyde datasets.

$$E[\tau_{ij}] = \tau(\lambda_{ij}), \text{ where } \tau(\cdot) \text{ is a continuous function from } \mathbb{R}^+ \text{ to } \mathbb{R}^+.$$

Hence, traffic demand between node pairs can differ and is *on average* correlated with the nodes' contact rate. However,  $\tau_{ij}$  itself is still random, allowing some node pairs to have little traffic demand even if they meet often (e.g. "familiar strangers"). Furthermore, through the function  $\tau(\cdot)$  one can introduce a number of different types and amounts of (positive or negative) correlations between traffic and mobility. While real traffic patterns are clearly expected to have a number of additional nuances and details, not captured by the model of Def. 3.2.1, it turns out that these abstractions are still "rich" enough to allow us to draw useful conclusions.

### 3.2.2 Communication Traffic Heterogeneity: Model and Implications

Let us assume a heterogeneous mobility/contact model as the one introduced earlier in Def. 3.1.1. If a message is equally likely to be generated between any pairs of nodes, then the contact rate between the source and destination of this message in a heterogeneous contact setup should be distributed according to the contact rate  $f_\lambda$ . However, if messages are more likely to come from a frequently meeting pair rather than an "average" pair (e.g. if communication rate is positively correlated with contact rate) then the source-destination contact rate (we refer to it as the *effective* contact rate) would be biased towards higher values. The following proposition captures this formally. Its proof can be found in [33].

**Proposition 1.** *The probability density function  $f_\tau$  of the contact rate between the source and the destination  $\{s, d\}$  of a random message, in a network following Defini-*

tions 3.1.1 and 3.2.1, converges as follows:

$$f_\tau(x) \xrightarrow{P} \frac{1}{C} \cdot \tau(x) \cdot f_\lambda(x) \quad (3.9)$$

where  $f_\tau(x)dx = P\{\lambda_{sd} \in [x, x + dx]\}$ ,  $\xrightarrow{P}$  denotes convergence in probability, and  $C = E[\tau(\lambda)] = \int_0^\infty \tau(x)f_\lambda(x)dx$  is a normalizing constant.

An opportunistic routing protocol tries to deliver the end-to-end traffic demand  $\tau_{ij}$ , and we would like to consider the effects of different contact patterns  $f_\lambda$  and traffic patterns  $\tau(\lambda)$  on its performance. There exists a very large abundance of proposed schemes [141] and it would not be possible, nor would it provide any intuition, to analyze the effect of heterogeneity on each and every one. Instead, we focus on two basic forwarding mechanisms, underlying many DTN protocols, to gain intuition:

**Direct Transmission (“DT”):** This is the simplest imaginable method: nodes wishing to exchange data or information with each other, may do so, only when they are in direct contact, without involving any relays. For instance, *DT* is often assumed in content-centric applications, where a node interested in some content will query directly encountered nodes for content of interest, and retrieve it only if it is available there. *DT* has the minimum amount of overhead and worst amount of delay (in non-limited resource setups), and will serve as the baseline.

**Relay-Assisted:** To improve the performance of direct transmission, replication or *Relay-Assisted* schemes can be used. Extra copies can be handed over to encountered nodes, and the destination can receive the message from either the source or any of the relays, reducing thus the expected delivery delay. In its simplest form, schemes like spray and wait distribute a small number of additional copies to the first few relays encountered, and any of these relays can deliver the message to the destination. In networks with homogeneous mobility and traffic, it is known that using just a few extra copies leads to significant performance gains. For example, in a network of 1000 nodes, simply distributing 10 extra copies to the first 10 nodes encountered provides an almost 10-fold improvement in delay compared to direct transmission [12]. Although this also comes with a 10-fold increase in the amount of (storage and bandwidth) resources needed, it presents a very useful tradeoff to DTN protocol designers, especially since this ratio actually decreases with network size and thus scales well [12].

However, when it comes to heterogeneous mobility and traffic, Proposition 1 suggests that, unlike the above example, the source is no longer equivalent with other random relays, in terms of their probability of contacting an intended destination soon. It is thus of particular interest to examine whether the above trade-off still holds, if one considers the joint effect of realistic mobility and communication traffic patterns.

To compare the performance of Relay-assisted routing and Direct Transmission, in terms of delivery delay and delivery probability (the two main metrics considered in related work), we first define the following metrics:

**(a) Delay Ratio, R:** the ratio of the expected delivery delay of Relay-Assisted routing,  $E[T_R]$ , over the expected delivery delay of Direct Transmission routing,  $E[T_{DT}]$ , i.e.

$$R = \frac{E[T_R]}{E[T_{DT}]}$$

**(b) Source Delivery Probability,  $P_{(src.)}$ :** the probability that a message is delivered to the destination by the source node, rather than by any of the relays.

For instance, when  $R = 0.1$  Relay-assisted routing delivers (on average) a message 10 times faster than Direct Transmission, while a value  $R = 0.5$  implies that Relay-assisted routing is only 2 times faster. Respectively, when  $P_{(src.)} = 0.1$  the probability that the source node  $s$  meets the destination  $d$ , before any other relay node meets  $d$ , is 10%, and  $P_{(src.)} = 0.5$  means that this probability is 50%. And in the limiting cases: when  $R, P_{(src.)} \rightarrow 1$ , it means the message is delivered to the destination by the source node itself; when  $R, P_{(src.)} \rightarrow 0$ , delivery takes place (entirely) due to the relays.

In Result 1, we derive analytical expressions for these two metrics,  $R$  and  $P_{(src.)}$ . The proof can be found in [33].

**Result 1.** *When Relay-assisted routing with  $L$  extra copies is considered, then*

$$R = \frac{1}{E\left[\frac{\tau(\lambda)}{\lambda}\right]} \cdot \int_0^\infty \int_0^\infty \frac{\tau(x)}{x+y} \cdot f_\lambda(x) dx \cdot f_R(y) dy$$

$$P_{(src.)} = \frac{1}{E[\tau(\lambda)]} \cdot \int_0^\infty \int_0^\infty \frac{x \cdot \tau(x)}{x+y} \cdot f_\lambda(x) dx \cdot f_R(y) dy$$

where the expectations are taken over  $f_\lambda$  and  $f_R = f_\lambda^{(*L)}$  is the  $L$ -fold convolution of  $f_\lambda$ .

### 3.2.3 A Case Study

To obtain additional insight, we consider the following setup as an example:

**Mobility:** We will assume the contact rates to be *gamma distributed*, i.e.  $f_\lambda(x) \sim \Gamma(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$ .

Our choice is initially motivated by the findings of Passarella et al. [138], who have shown, through statistical analysis of pervasive social networks' datasets, that the *Gamma distribution* matches well the observed contact rates. Additionally, by selecting appropriately the parameters  $\alpha$  and  $\beta$  of a Gamma distribution, we can assign any desired value to the mean value  $\mu_\lambda$  and the variance  $\sigma_\lambda^2$  of the contact rates<sup>2</sup>. This allows us to describe (or fit up to the first two moments) a large range of scenarios with different mobility heterogeneities captured by  $CV_\lambda = \frac{\sigma_\lambda}{\mu_\lambda}$ .

**Traffic:** We will describe the traffic using a *polynomial function* of the form  $\tau(x) = c \cdot x^k$ ,  $c > 0$ . By choosing different values of  $k$ , one can control both the amount of correlation, as well as the shape (concave or convex).

Under these assumptions, the following closed form expression bounds can be derived for our two metrics of interest. The proof can be found in [34].

**Result 2.** *In a Heterogeneous Contact Network where  $f_\lambda \sim \Gamma(\alpha, \beta)$  with mean value  $\mu_\lambda$  and variance  $\sigma_\lambda^2$  (coefficient of variation  $CV_\lambda = \frac{\sigma_\lambda}{\mu_\lambda}$ ) and  $\tau(x) = c \cdot x^k$ , it holds:*

$$1 \geq R \geq R_{min} = \frac{1 + (k-1) \cdot CV_\lambda^2}{1 + k \cdot CV_\lambda^2 + L} \quad (3.10)$$

---

<sup>2</sup>The mean value and variance of a gamma distribution are given by  $\mu_\lambda = \frac{\alpha}{\beta}$  and  $\sigma_\lambda^2 = \frac{\alpha}{\beta^2}$ , respectively.

Table 3.2:  $R_{min}$ ,  $P_{min}$ : Monotonicity and Asymptotic Limits

Parameter $x$	Monotonicity as parameter $x$ increases ↗	$x \rightarrow \min\{x\}$	Limits for $x \rightarrow \max\{x\}$
mobility heterogeneity: $CV_\lambda \in [0, \infty)$	$R_{min}$ increases ↗, if $k > 1 + \frac{1}{L}$ decreases ↘, otherwise $P_{min}$ increases ↗, if $k > \frac{1}{L}$ decreases ↘, otherwise	$\lim_{CV_\lambda \rightarrow 0} R_{min} = \frac{1}{1+L}$ $\lim_{CV_\lambda \rightarrow 0} P_{min} = \frac{1}{1+L}$	$CV_\lambda \lim_{\lambda \rightarrow \infty} R_{min} = 1 - \frac{1}{k}$ $CV_\lambda \lim_{\lambda \rightarrow \infty} P_{min} = 1 - \frac{1}{k+1}$
traffic heterogeneity: $k \in (k_{min}, \infty)$	$R_{min}, P_{min}$ increase ↗	$\lim_{k \rightarrow k_{min}} R_{min}, P_{min} = 0$	$\lim_{k \rightarrow \infty} R_{min}, P_{min} = 1$
extra copies: $L$ ( $L \ll N$ )	$R_{min}, P_{min}$ decrease ↘	-	-

for  $k > k_{min} = 1 - \frac{1}{CV_\lambda^2}$ , and

$$1 \geq P_{(src.)} \geq P_{min} = \frac{1 + k \cdot CV_\lambda^2}{1 + (k+1) \cdot CV_\lambda^2 + L} \quad (3.11)$$

for  $k > k_{min} = -\frac{1}{CV_\lambda^2}$ .

**Practical Example:** If the measured network characteristics are  $CV_\lambda = 2$  and  $k = 2$ , then from Result 2 we get  $R = \frac{5}{9+L}$ . Therefore, to achieve delivery delay two times faster than Direct Transmission, one extra copy should be used ( $L = 1 \rightarrow R = 0.5$ ), while to achieve 4 times faster delivery,  $L = 11$  relay nodes are needed. In the latter case, if traffic/mobility heterogeneity has not been taken into account [12], the prediction would be  $L = 3$  and this would lead only to 2.5 (instead of 4) times faster delivery (i.e.  $R = \frac{5}{12}$ ).

It is evident from the above example that traffic heterogeneity can have a major impact on performance and thus protocol design. Table 3.2 formalizes this impact, by considering how  $R_{min}$  and  $P_{min}$  (Eq. (3.10) and Eq. (3.11)) behave.

### Gain of Extra Copies

A strong positive correlation (large  $k$ ) between traffic and mobility reduces the added value of extra copies (i.e.  $R_{min}, P_{min} \nearrow$  as  $k \nearrow$ ). This indicates that, as correlation ( $k$ ) increases, one needs to distribute message copies to more relay nodes in order to achieve a certain performance improvement compared to the baseline, Direct Transmission.

In contrast, a negative (or weak positive) correlation renders each extra copy more useful (i.e.  $R_{min}, P_{min} \rightarrow 0$  as  $k \rightarrow k_{min}$ <sup>3</sup>). It becomes evident that it is crucial to identify whether a traffic-mobility correlation exists in a given scenario, and what its nature is, as this could decide whether the overhead of using few or more extra copies is justified or would just waste a lot of valuable resources.

### Routing for Unicast Applications

For high heterogeneity (traffic and mobility), our results imply that a unicast message is likely to arrive to its destination at the time the source and destination come

---

<sup>3</sup>The values of  $k_{min}$  are given in Result 2.

*in contact* (i.e.  $R_{min}, P_{min} \rightarrow 1$  as  $k, CV_\lambda \rightarrow \infty$ ). This raises questions about the usefulness of opportunistic networking for unicast applications in which end-to-end traffic is expected to be highly correlated with contact frequency (e.g. Facebook messaging) [133, 134]. On the other hand, our results suggest that potential unicast applications with an end-to-end traffic demand between nodes with non-frequent meetings, i.e. scenarios with small or negative  $k$ , (e.g. social peers residing in different communities) could benefit a lot (more than normally assumed).

Although these observations might appear somewhat self-evident at first glance, the question of how to tune protocols and choose the right number of replicas stills remains. To our best knowledge, our results are the first to provide closed form, *quantitative* insights into the tradeoffs involved in real scenarios with both mobility and traffic heterogeneity. Furthermore, although our analytical results are derived for the case of random replication protocols, the above qualitative implications hold also for mobility-aware protocols (e.g. EBR [17]), which exploit mobility heterogeneity in order to select better relays.

### Content-Centric Communication

While our results are somewhat pessimistic when it comes to the usefulness of opportunistic networking for unicast applications, the opposite holds when it comes to modern, content-centric applications (e.g. file sharing, D2D-based offloading, service composition). In such applications nodes are looking, for example, for some content of interest [142] or service [143], which they can access directly from *any* encountered node that offers it. If the interests of nodes are heterogeneous (which is known to be the case [144]) and nodes with similar mobility patterns tend to have *some* similarity in their interests too (evidence for this does exist [145]), then our results suggest: (i) that there is a better chance to find a content or service “soon” from a directly encountered node than one would expect in homogeneous scenarios, and (ii) coming up with complex, resource-costly mechanisms, e.g. multi-hop query-response, directories, etc., might not be necessary. We plan to look into such content-centric scenarios in more detail in future work.

In Fig. 3.7 we compare the message delay of (i) Direct Transmission (i.e. the protocol with the *highest* delay), (ii) Relay-assisted routing (Spray and Wait, *SnW*, [12] with  $L = 5$  copies) and (iii) Epidemic routing [11] (i.e. the protocol with the *lowest* delay), in two scenarios, for varying traffic heterogeneity ( $k$ ). Two main observations, with respect to the previous implications, can be made in Fig. 3.7.

We have extensively validated our results against both synthetic and real traces in a number of scenarios. More details can be found in [33]. In the following, we discuss instead some interesting extensions of the above basic framework.

#### 3.2.4 Extensions

We have tried to present our results in the context of simple schemes (e.g. unicast traffic, random relay selection), to keep analysis tractable and illustrate key principles. In this section, we discuss how our framework could be applied in some additional

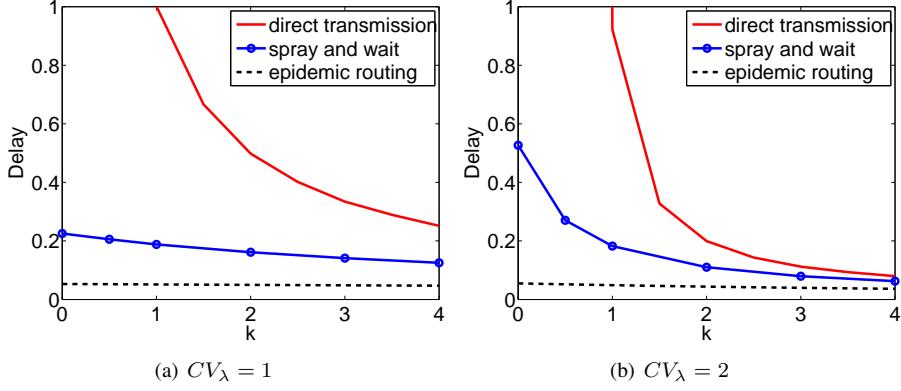


Figure 3.7: Message delay under Direct Transmission, Spray and Wait ( $L = 5$ ), and Epidemic routing in scenarios with varying traffic heterogeneity; mobility parameters are  $\mu_\lambda = 1$  and (a)  $CV_\lambda = 1$  and (b)  $CV_\lambda = 2$ .

cases. Although far from complete, we believe this set of examples, further underlines the utility of our analysis.

### Mobility-Aware Protocols

*Mobility-aware* schemes are often used to select good relays for the intended replicas, rather than picking random ones, e.g. [36, 146, 17, 35]. The selection of the relays is usually based on their social or mobility characteristics. For instance, in *encounter-based routing (EBR)* [17], the more frequently a node  $i$  encounters node  $d$ , the higher the probability to become a relay of a message destined to  $d$ .

The relay-selection mechanism in a number of proposed mobility-aware protocols can be described as following:

**Definition 3.2.2.** *The probability  $p_i$  a node  $i$  to be selected as a relay for a message destined to node  $d$ , is related to their contact rate  $\lambda_{id}$  and this relation is described by a function  $p(\lambda_{id})$ .*

As an example, we present two protocols belonging to the above class and their  $p(\lambda)$  functions: (a) a modified mobility-aware version of *spray and wait* [12] protocol (we refer to it as *U1*), and (b) a variation of the *EBR* [17] protocol (we refer to it as *U2*), where each relay can hold only one message copy.

*U1*: A node  $i$ , which would be selected as a relay by the *spray and wait* mechanism, under *U1* becomes a relay with a probability  $p_i$  that is proportional to its contact rate with the destination  $d$ , i.e.  $p_i = p(\lambda_{id}) = c \cdot \lambda_{id}$ , where  $c$  a normalizing factor such as  $p(\lambda) \in [0, 1]$ .

*U2*: For each message copy, the source node  $s$  selects the relay node  $i$  with a probability  $p_i$  that is computed according to the *EBR* mechanism, i.e.  $p_i = p(\lambda_{id}) = \frac{\lambda_{id}}{\lambda_{id} + \lambda_{sd}}$ .

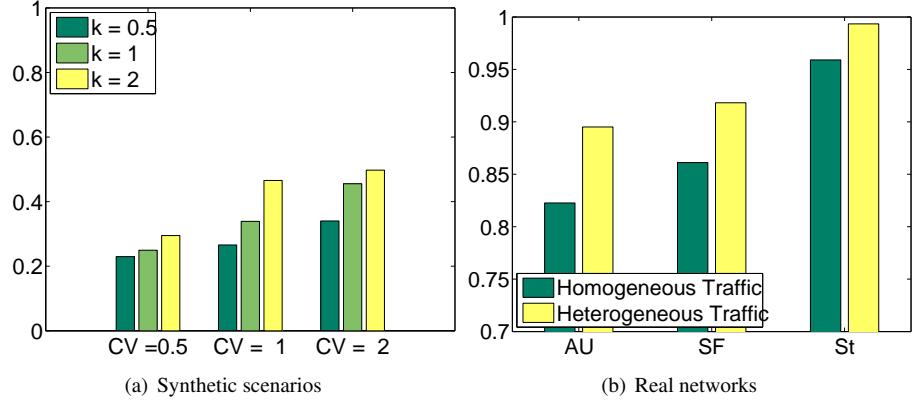


Figure 3.8:  $P_{(src.)}$  of mobility-aware routing in (a) synthetic scenarios with varying mobility ( $CV_{\lambda}$ ) and traffic heterogeneity ( $k$ ), and (b) real networks with homogeneous and heterogeneous traffic.

In the following corollary, we show how Results 1 and 2 can be simply modified to capture such mobility-aware protocols as well. The proof follows similar ideas, and can be found in [33].

**Corollary 2.** *Under a mobility-aware Relay-Assisted protocol conforming to Definition 3.2.2, Results 1 and 2 are modified as:*

*Result 1:*  $f_R$  is given by the  $L$ -fold convolution of  $f_u(\lambda)$ , where

$$f_u(x) = \frac{1}{E[p(\lambda)]} \cdot p(x) \cdot f_{\lambda}(x)$$

*Result 2:* The number of copies  $L$  is multiplied by  $c_u$ , where

$$c_u = \frac{E[\lambda \cdot p(\lambda)]}{E[\lambda] \cdot E[p(\lambda)]}$$

For instance, applying Corollary 2, the expression for the delay ratio  $R$ , becomes

$$1 \geq R \geq R_{min} = \frac{1 + (k - 1) \cdot CV_{\lambda}^2}{1 + k \cdot CV_{\lambda}^2 + c_u \cdot L} \quad (3.12)$$

and for the *U1* protocol presented above,  $c_u$  is given by the expression<sup>4</sup>:

$$c_u^{(U1)} = 1 + CV_{\lambda}^2 \quad (3.13)$$

When mobility is highly heterogeneous (i.e. high  $CV_{\lambda}$ ),  $c_u^{(U1)}$  becomes large, and thus  $R_{min}$  and  $P_{min}$  decrease compared to the random replication mechanism (e.g.

---

<sup>4</sup>An expression for  $c_u$  in the case of the *U2* protocol could also be derived, albeit with more complexity, due to the fact that the function  $p(\lambda)$  involves the source destination contact rate  $\lambda_{sd}$  as well.

random  $SnW$ ). This confirms that the performance gain is larger when mobility-aware protocols are used. However, even in this case, as traffic heterogeneity increases, the performance gain diminishes, i.e.  $R_{min}, P_{min} \rightarrow 1$ .

We further demonstrate some preliminary simulation results suggesting that our conclusions hold also for mobility-aware routing. We use the  $U2$  protocol presented above. In Fig. 3.8(a) we present simulation results for the delivery metric  $P_{(src.)}$  on synthetic scenarios with varying mobility ( $CV_\lambda$ ) and traffic ( $k$ ) heterogeneity. Similarly to the random replication case, for increasing heterogeneity (in mobility and/or traffic) the gain of the extra copies clearly decreases (i.e.  $P_{(src.)}$  increases) under mobility-aware schemes. In Fig. 3.8(b) we compare the probability  $P_{(src.)}$  of scenarios with and without traffic heterogeneity in real networks. As before, the results are consistent with our theory: the gain of extra copies decreases even for protocols using more sophisticated techniques for relay selection.

### Routing based on Contact Graph Structure

A number of mobility-aware routing schemes, e.g. SimBet [35], BubbleRap [36], are based on the structure of the contact graph (centrality, similarity, communities, etc.) rather than the pairwise contact rates. A direct mapping to a function  $p(\lambda)$  for these protocols requires a separate, and rather cumbersome analysis for each such protocol, in most cases not leading to a closed form expression (see, e.g. [42]). However, the contact graphs used to make forwarding decisions by these more sophisticated protocols, still are built based on pair-wise contact rates [38]. We thus expect the utility of such mobility-related information to be similarly affected by the amount of traffic heterogeneity and its relation to mobility patterns.

To test this further, we simulated, as an example, scenarios using the SimBet protocol [35]<sup>5</sup>. In Fig. 3.9 we present the simulation results (continuous lines) for the ratio  $R = \frac{E[T_{SimBet}]}{E[T_{DT}]}$  and the theoretical predictions  $R_{min}$  of Eq. (3.12), for different values of the  $c_u$  parameter (dashed lines). Two main observations that confirm our intuition are: (i) simulated and theoretical curves increase in a similar manner, and (ii) one can find (numerically) the value  $c_u$  that more accurately predicts the performance.

Although this is clearly not conclusive for the applicability of our result to every mobility-aware scheme, we believe it helps to corroborate our findings for the interplay between mobility and traffic heterogeneity on protocol performance.

### Multicast Communication

We have also been assuming unicast messages between a  $\{s, d\}$  pair. However, our results apply also to multicast [122] or anycast (e.g. content sharing or service composition applications) [143] messages from  $s$ , with  $d$  being one of the destinations, since similar mechanism are often used for their dissemination. To demonstrate this, in Table 3.3 we present simulation results for two multicast scenarios, with homogeneous

---

<sup>5</sup>For the contact graph we considered the 10% most frequently meeting pairs following the guidelines of [38], we set the similarity and betweenness weights  $\alpha = \beta = 0.5$  [35], and we generated multiple copies as in [147].

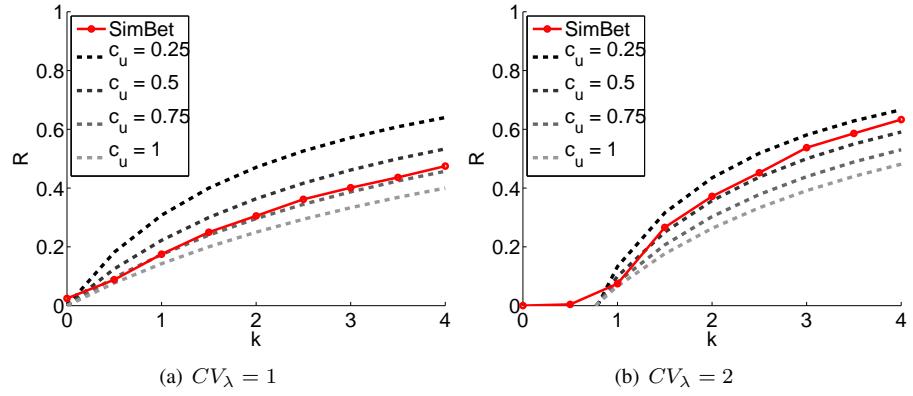


Figure 3.9: Delay ratio  $R$  in two scenarios with varying traffic heterogeneity  $k$ . Relay-assisted routing is SimBet with (a)  $L = 5$  and (b)  $L = 10$  message copies.

Table 3.3: Multicast Communication

	$CV_\lambda$	0.1	0.5	1	1.5	2
HOM	$R$	0.18	0.12	0.01	0	0
	$P_{(src.)}$	0.01	0.01	0.01	0.01	0.01
HET	$R$	0.18	0.26	0.39	0.52	0.61
	$P_{(src.)}$	0.01	0.03	0.12	0.26	0.41

(*HOM*) and heterogeneous (*HET*) traffic ( $\tau(x) = c \cdot x^4$ ), under varying mobility heterogeneity. A source sends messages to 5 destinations (each selected with a probability  $\propto \tau_{ij}$ ) either by Direct Transmission or by Relay-Assisted routing with  $L = 5$  copies. As delivery delay, we consider the delay till all the destinations get the message. It is evident that  $R$  and  $P_{(src.)}$  (i) increase significantly with mobility heterogeneity when traffic is heterogeneous, and (ii) become much larger compared to the homogeneous case (where  $R$  decreases and  $P_{(src.)}$  is constant), which is in agreement with our results.

## Node Selfishness

Another interesting pair-wise property related to the performance of unicast routing protocols is “selfishness”. Most efficient protocols for DTNs require some sort of multi-hop communication, where intermediate nodes that are neither the source nor the destination for a message must act as relays. However, due to limited resources (battery, bandwidth etc.) and privacy concerns, it might happen that nodes are reluctant to cooperate and forward messages of other nodes. For example, a selfish node could only forward and receive its own messages and never act as a relay in order to avoid wasting resources. To avoid such behaviors and isolate selfish nodes, many mechanisms providing incentives for cooperation (e.g. reputation [148] or credit based [149, 150] mechanisms) have been proposed.

In many cases, nodes will be more willing to forward messages for nodes with whom they have a strong social tie (e.g. friends or acquaintances), because they (i) trust them and (ii) have received similar services by them in the past (or it is probable to receive in the future). Additionally, studies from sociology [140], social media [135] and pervasive social networks [151] have shown that the strength of social ties can be predicted by the contact rate and, specifically, the more frequent two nodes contact, the stronger their social tie probably is.

Hence, combining the previous observations, i.e. (a) the level of cooperation is related to social ties and (b) the strength of the ties is related to the contact rate, we can model the nodes' selfishness in a probabilistic way, which allows us to predict its effect on the network performance.

**Definition 3.2.3** (Social Selfishness). *A pair of nodes  $i$  and  $j$  either can exchange messages in every contact event with probability  $p_{ij}$  or can never exchange messages with probability  $1 - p_{ij}$ . The probability  $p_{ij}$  depends on the meeting rate between these nodes, i.e.  $\lambda_{ij}$ , and is described by the relation:*

$$p_{ij} = p(\lambda_{ij}), \quad p_{ij} \in [0, 1] \quad (3.14)$$

The above definition is generic and can be used to describe also scenarios where the cooperation between nodes is not determined only by social factors. For instance, in [152] authors investigate how more efficient communication can be achieved: They propose a content sharing application where the decision for message exchanges depend on the contact rate of nodes and evaluate through simulations different content sharing policies (i.e. different decision rules based on the contact rates).

When node willingness to act as a relay is correlated to the contact rate, similar analytical steps as in the case of traffic heterogeneity can be applied to study the joint impact of the two. We refer the interested reader to [153] for more details on the topic.

## **Chapter 4**

# **Complex Network Analysis for Opportunistic Device-to-Device Networking**

## 4.1 Introduction

As we have seen already in previous chapters, many protocols forward in parallel multiple replicas of the same content, to combat the inherent uncertainty of future contact opportunities, [11, 12]. We have also stressed the fact that contact patterns between nodes are heterogeneous and we studied the impact of simple forwarding schemes under both homogeneous and heterogeneous patterns. However, it is not just that node mobility and resulting contact opportunities are not heterogeneous, but more importantly this mobility is not entirely random. Instead, weak or strong patterns are present that *utility-based* routing schemes attempt to take advantage of to differentiate nodes that are more likely to deliver content or bring it closer to the destination [10]. E.g., rather than handing over an extra message copy to a random encountered node, it makes more sense to wait and hand over that copy to another node if one can predict that the latter will encounter the destination soon, with high probability.

Among the numerous utility-based schemes proposed, even some early ones already implicitly assessed the strength of “social” ties between nodes. For example [105] uses time of last encounter, and [154] uses contact frequency as a hint on the similarity of mobility patterns. [25, 16] use instead a metric much akin to *degree centrality* to identify nodes that are highly mobile/social<sup>1</sup>; the former scheme is reminiscent of search in scale-free networks, while the latter uses centrality to choose which relays to spray a limited budget of message replicas to. However, these simple metrics may only capture one facet of the underlying mobility process, which might not suffice for good contact predictions.

More recently, complex network analysis [30] (CNA) was proposed as a more generic and powerful tool to formulate and solve the problem of future contact prediction in DTNs. The main idea behind this approach is the following: communication devices are most often carried by users, whose mobility pattern is governed by complex social relations with other users (as well as purpose, schedules, etc.); hence, these patterns should also give rise to social properties in the observed contacts between communicating devices, that could be exploited for protocol design, if inferred correctly. To this end, past observed contacts between nodes are aggregated into a social graph, with graph edges representing (one or more) past meetings between the vertices. An edge in this graph conveys the information that two nodes often encounter each other either because they have a strong social tie (friends), or because they are frequently co-located without actually knowing each other (familiar strangers), as shown in Fig. 4.1. Thus, existence of an edge intends to have predictive capacity for future contacts.

Two recently proposed routing protocols, SimBet [35] and BubbleRap [36], make explicit use of CNA metrics and algorithms in order to highlight a node’s position in the aggregated social graph, and assess its utility to act as a relay for messages destined to other nodes. Although the detailed mechanisms of the two protocols differ, they are both based on the same principles: they assume that nodes naturally reside in mobility-related communities (e.g., class, work, home). Increasingly “central” or “well-connected” nodes in the graph are then chosen as carriers to relay content

---

<sup>1</sup>“Degree” is the number of neighbors a node has on a graph, and the node with the highest “degree centrality” is the node with the largest number of neighbors.

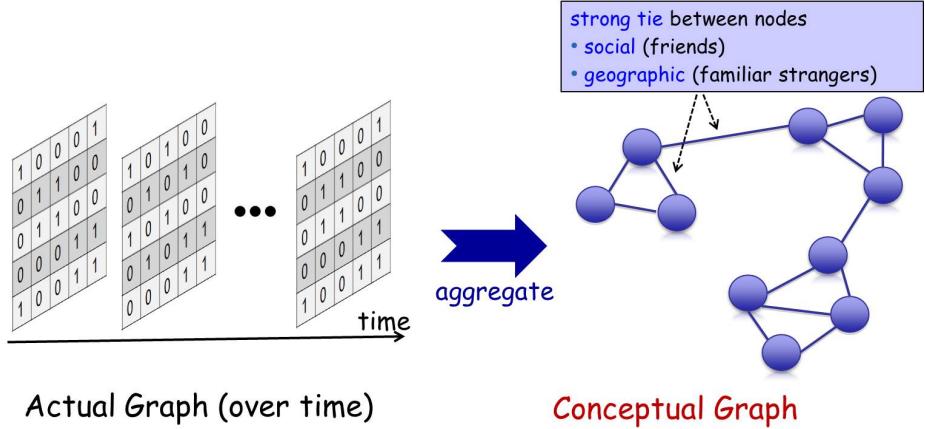


Figure 4.1: Converting a sequence of connectivity graph instances into a contact graph.

over different communities, until a node that shares many neighbors with the destination [35], (i.e., belongs to the destinations community [36]) is reached. These protocols have been reported to often outperform well-known DTN routing schemes that are not explicitly social.

Nevertheless, it is not well understood under what conditions these protocols and their individual components can achieve the suggested performance, nor is it why. What is more, it is actually not (just) the choice or sophistication of social metrics or algorithms that bears the most weight on performance, but rather the mapping from the mobility process generating contacts to the aggregated social graph. This mapping presents a tradeoff, where some information about timing of contacts is lost. As a simple example, one could create a link if at least one contact has occurred in the past between the two nodes [35], but this would result in an overly dense graph after a certain network lifetime. Meaningful differentiation between nodes using complex network analysis will not be possible. The social graph created out of past contacts should best reflect the underlying (mobility or social) structure generating these contacts, so that nodes can be meaningfully differentiated and edges have predictive value.

Given that social-network based schemes are arguably the state-of-the-art in DTN routing, in this chapter we focus on such protocols and make the following contributions:

- We demonstrate that the performance of CNA-based DTN routing is in fact rather sensitive to the “quality” of the inferred social graph: good performance is achieved only for a relatively narrow range of parameters, while otherwise performance can be as bad as that of direct transmission.
- We investigate different methods to identify this optimal operating point “on the fly”. Specifically, we use clustering techniques [155] to identify desirable patterns in observed node similarities, and then use concepts from spectral graph theory [156] to maximize the modularity of such clusters, and compare the be-

havior of various contact models under different aggregation methods.

- We propose a distributed online algorithm that can adjust its contact graph mapping to achieve optimal performance, regardless of the mobility scenario or the specific routing protocol used.

## 4.2 Contact Aggregation: Preliminaries

We will focus our study on the two popular protocols mentioned above, SimBet and BubbleRap.

**SimBet** [35] assesses similarity to detect nodes that are part of the same community, and betweenness centrality to identify bridging nodes, that could carry a message from one community to another. *Similarity* is defined as the percentage of common neighbors (e.g. two nodes A and B, who have one common neighbor C, and each one more neighbor not in common, have a similarity of 0.5). Nodes in the same community will have high similarity. *Betweenness centrality* is related to the importance of a node to route traffic between other pairs. E.g. a node sitting between two communities will tend to have high degree centrality. The decision to forward a message depends on the similarity and centrality values of the newly encountered node, relative to the current one: If the former node has a higher similarity with the destination, the message is forwarded to it; otherwise, the message stays with the most central node. The goal is to first use increasingly central nodes to carry the message between communities, and then use similarity to “home in” to the destination’s community.

**Bubble Rap** [36] uses a similar approach. Again, betweenness centrality is used to find bridging nodes until the content reaches the destination community. Communities here are explicitly identified by a community detection algorithm, instead of implicitly through similarity. Once in the right community, content is only forwarded to other nodes of that community: a local centrality metric is used to find increasingly better relay nodes within the community.

The original versions of these protocols used time-based aggregation methods. A naive such approach is for example to add a link to the contact graph if there has been at least one contact (or a given threshold of contacts) *ever* in the past. However, if nodes visit different locations around the network, even rarely, this method would eventually give rise to an almost *complete graph* (every node is connected to every other node). Such a graph has maintained very little information about the underlying process, and it is clearly not possible to differentiate nodes based on it. To combat this, a sliding time window approach can be applied. However, choosing the “right” time-window is highly dependent on the mobility trace considered (as also mentioned by the authors of [36]) as in some mobility scenarios (and respective traces), many nodes might have encountered each other within some time window  $\Delta T$  while in another almost none. Example contact graphs, resulting for different time windows for a small contact trace of 20 nodes collected at ETH, Zurich are shown in Fig. 4.2.

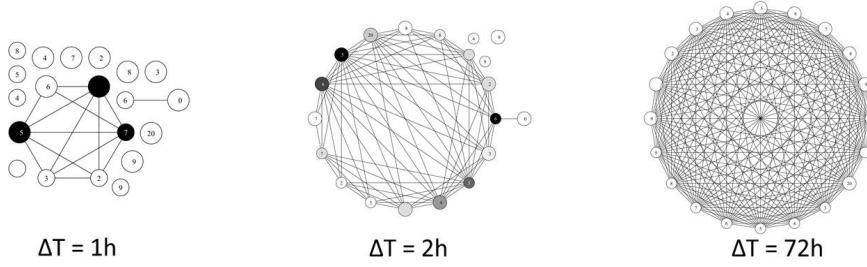


Figure 4.2: Aggregated contacts for the ETH trace at different time instants.

#### 4.2.1 Density-based Aggregation

A more generic approach is instead to fix a (desired) graph density rather than a time window as follows. Since most real social graphs are relatively sparse, one could impose a weight on each edge and then keep the top  $d\%$  of edges with the highest weight. We will refer to  $d(G_n)$  as the *aggregation density* of a given contact graph  $G_n$ : e.g. if  $d(G_n) = 0.2$  it will mean that only the top 20% of all possible node edges (a graph of  $N$  nodes can have at most  $N(N - 1)/2$  edges) will be included in the contact graph. Based on this contact graph the respective protocol will try to derive the needed graph metrics (e.g. similarity, centrality, etc.), and use the to decide which nodes to forward a message to. The ranking of edges is done in terms of weight, where we focus on simplicity on two weight types:

- *Most Recent Contacts (MR)*: the more recent a contact, the higher its weight.
- *Most Frequent Contacts (MF)*: the larger the number of past contacts (possibly within some large time-window to account for non-stationarity), the higher the weight.

A number of more sophisticated mappings are possible, taking additional features into account. E.g., in [40], we have used Principal Component Analysis (PCA) to derive a weight metric that depends on both recency and frequency. Our goal here is not to derive an optimal aggregation function, but rather to demonstrate, that even with simple aggregation functions, one can considerably influence the performance of DTN routing schemes which utilize complex network analysis.

### 4.3 Performance Sensitivity to Aggregation

Having discussed how to create a contact graph, out of a time-sequence of pair-wise contacts, and the need for this graph to be useful for distinguishing good forwarding nodes, two interesting questions arise: (a) *How sensitive is the usefulness of the contact graph to the aggregation density we pick?* and (b) *How can we find the right density?*

Towards answering the first one, we evaluate the performance of SimBet and BubbleRap as a function of aggregation density (as well as type of metric used), we have

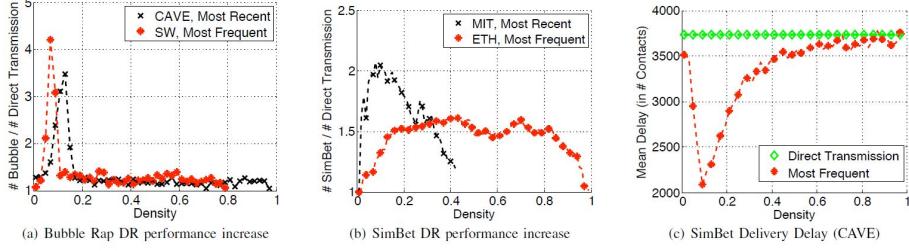


Figure 4.3: Delivered messages and delivery delay for SimBet/Bubble Rap vs. Direct Transmission.

Protocol	SW	CAVE	MIT	ETH	INFO
SimBet MF	.09/4.3	.09/3.3	.03/1.8	.43/1.6	.64/1.3
SimBet MR	.09/3.7	.12/3.5	.09/2.1	.02/1.5	.71/1.4
Bubble MF	.08/4.2	.09/4.5	.08/2.5	.44/1.5	.67/1.4
SimBet MR	.09/3.3	.09/3.9	.14/2.5	.33/1.4	.67/1.4

Table 4.1: Optimal aggregation densities. The first number is the density, the second number is the relative performance increase (compared to DT). MF: Most Frequent, MR = Most recent.

considered two synthetic contact processes with explicit social properties: a *small world* process (**SW**) and a *caveman* process (**CAVE**), popular in social networks literature. We have also considered three real traces, the MIT Reality Mining (**MIT**), the iMotes Infocom 2005 (**INFO**), and the ETH (**ETH**) traces. Details about these scenarios can be found in [38].

In Fig. 4.3 we examine the performance of BubbleRap with SW and CAVE and SimBet with ETH and MIT contacts, respectively, as a function of aggregation density. We compare the respective performance of each protocol with respect to that of Direct Transmission, which has the worst possible performance and serves as a baseline.

The first thing to note is that, for aggregation densities close to 0 or 1 (i.e., the graph is either empty or complete), the performance factors are 1, that is, the performance is the same as for direct transmission<sup>2</sup>.

For SW, CAVE and MIT, we observe clear performance peaks at small densities (around 0.1) and clear performance drops at densities of about 0.2 and higher. For the INFO trace (not shown in the figures) and for ETH, the performance peak is a bit less pointy and at a higher density of around 0.4 for ETH (around 0.65 for INFO). We suspect the reason for this is that both traces are smaller than MIT in terms of nodes and geographical extent.

Table 4.1 summarizes the results of all combinations of the two protocols with the two mapping functions and with the five contact processes, in terms of density at

<sup>2</sup>Note that for the MIT contacts, the graph only reaches a density of about 0.4 during the trace duration. We believe that this is due to the large time granularity, which potentially does not capture many short random co-locations.

which the peak performance occurs and height of the peak performance factor. Let's an example of how to read this table, for SimBet running over a Caveman (CAVE) contact generation model, and with contact graph construction based on the most recent (MR) links (row 2, column 3 on the table): the optimal aggregation density is 0.12 (i.e. the contact graph contains only 12% of all possible pair-wise links, namely the links corresponding to the 12% pairs that most recently met); the performance gains for this scenario and this contact graph density is 3.5, compared to Direct Transmission. Considering that this is a single-copy scheme [10], this is an important improvement.

One important observation from this table is that the peak densities differ from scenario to scenario, but are consistent within a scenario (e.g., peak densities for SW are all along the same value). This indicates that *the optimal point of aggregation does not depend on the forwarding metric, but rather on the contact process*. One exception is the ETH trace, where the peak density for SimBet with MR mapping is at a quite smaller value. One might deduce from this that the few very recent contacts have much more predictive power than slightly older ones. Another possibility is due to the relatively more flat nature of the ETH plot (thus making the optimal point more noisy).

Finally, the performance peaks of SimBet and Bubble Rap are of similar height within a scenario. However, the peak height differs from scenario to scenario. This indicates that different networks show different degrees of structure, which the routing protocols can use to increase their performance compared to direct transmission.

## 4.4 Inferring The Optimal Aggregation Density Online

Assuming a node has collected enough information about past contacts<sup>3</sup>, it could use density-based aggregation to maintain a contact graph around the optimal density point, and apply its preferred forwarding metrics on this graph. However, the above analysis gives rise to an interesting question: *How can each node decide an optimal aggregation density in real time, without any prior knowledge (e.g., mobility context), in order to optimize its performance?*

We conjecture that this optimal density lies at the point where the underlying (social) structure governing mobility best correlates with the social structure that can be observed on the aggregated contact graph. Consider for example a simple CAVE model: each node belongs to exactly one community, and nodes have links to most other nodes in their community (with equal or different weights); we refer to these links/neighbors as *regular*; in terms of mobility each community is assigned to a separate physical location (“cave”), and nodes either move towards a node in their community (with higher probability), or perform a random “roaming” trip in the network (with lower probability). In this mobility model, as nodes move around, they mostly come in contact with *regular* neighbors, but every now and then they might encounter a node from another community (e.g. during a roaming trip). A link to such a node could be included in the contact graph (e.g. if the contact was recent enough, or the aggregation density is high enough). We will call such a node a *random* neighbor (on the

---

<sup>3</sup>given that there are no delay constraints on this collection process, it can be performed with relatively low overhead as nodes meet and exchange info about their past contacts.

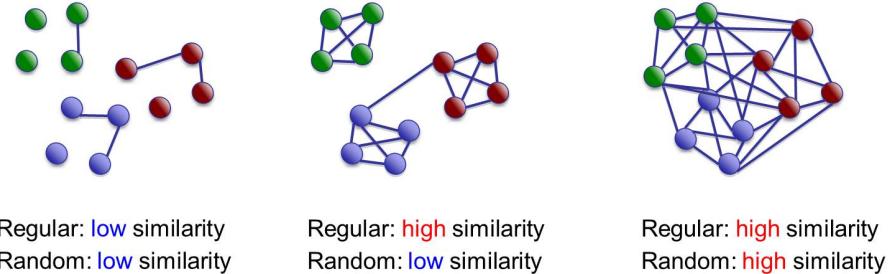


Figure 4.4: Aggregated contact graph for a scenario with 3 social communities driving mobility (denoted with different colors), and three different aggregation densities.

contact graph). Hence, regular links imply future contacts while random links might be incidental, not necessarily implying future contacts.

For CNA-based forwarding to work, we argue that a strong “relationship” in the contact process (e.g., a community/regular link the above model) should also be an observable relationship (edge) in the aggregated graph. At the same time, the number of additional graph edges (e.g., random links) should be as few as possible, as these will most probably be the result of incidental past meetings. There is solid evidence that real life mobility is predominantly small-world, with salient features such as communities [157] and a high clustering coefficient [158]. Our detailed study of real traces in [40, 39] confirms this.

The goal of this section is thus to devise an algorithm that observes the aggregated social graph online (i.e., as new contacts arrive), and tries to assess the density at which this graph has a structure that best reflects the above intuition about connectivity/contact patterns. This is a difficult unsupervised learning problem. We first give hints on how to tackle this problem on a simple example, like the above. Then, we propose two clustering-related methods for identifying distinguishable similarity patterns: one based on spectral graph theory [156] and the other based on established methods of evaluating the quality of graph partitioning [159]. Both can be used at the core of our algorithm.

#### 4.4.1 Optimal community structure of the social graph

One way to distinguish regular neighbors from random neighbors is by their similarity values. Each node will see a set of nodes to which it is highly similar (i.e., many shared regular neighbors in the graph) and another set of nodes to which it is less similar (i.e., random neighbors). Fig. 4.4 shows such an example, for “low”, “good”, and “high” aggregation density. As can be seen there, if density is too low, very few of the regular (intra-community) links appear, and the similarity of most node pairs is 0. On the other hand, if density is too high, the graph is too dense giving the (wrong) impression that random neighbors, belonging in different communities, have high similarity as well (and thus leading to the wrong conclusion that these nodes belong to the same community).

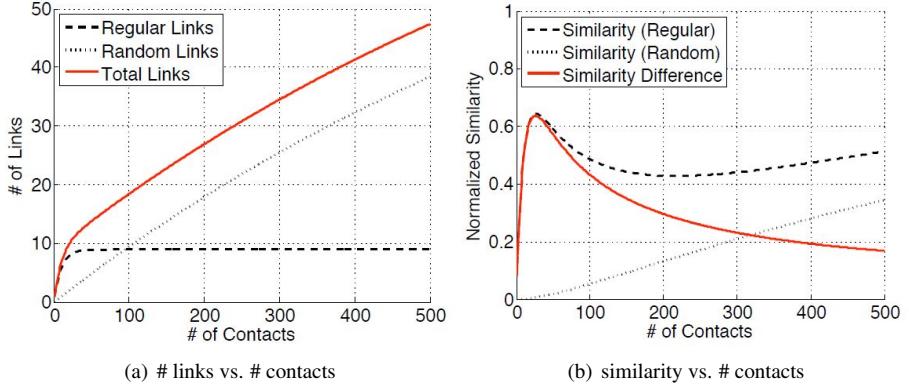


Figure 4.5: Number of links and similarity values as a function of time.

Based on the above observations, our problem could thus be cast as maximizing the similarity to regular neighbors, while minimizing the similarity to random ones. In order to motivate this approach, we use the CAVE model without rewiring, to analyze the expected number of regular neighbors and the expected number of random neighbors for a node in the aggregated graph. Due to the simplicity of this model, these quantities can also be derived in closed form. From these, we get the expected values of similarity that would be observed online, as a function of time.

We omit the detailed analysis here, which can be found in [38], and we just plot the expected number of regular, random and total links in Fig. 4.5(a), as a function of contacts seen by a specific node (this is an implicit measure of time as well). This figure shows that the point when all (or most) regular links are in the graph is reached quickly (after a few 10s of contacts). In this scenario, this is the right point to stop the aggregation and fix the density. After that point, only random links are added to the graph. These have little predictive value and erroneously increase nodes' similarity.<sup>4</sup>

Ideally, assume that a node could measure the average similarity with its regular neighbors (e.g. nodes in the same community) at different aggregation densities  $d$ , denoted as  $\text{sim}_{\text{reg}}(d)$ , and its average similarity with random neighbors, denoted as  $\text{sim}_{\text{rnd}}(d)$ . Then, in order to maximize the predictive capacity of the aggregated graph, it could try to maximize  $\text{sim}_{\text{reg}}(d) - \text{sim}_{\text{rnd}}(d)$ . As nodes do not have any a priori knowledge about community membership and size, or the total number of nodes, it is more sensible to use normalized similarities and divide similarities by the expected node degree  $n_{\text{reg}}(d) + n_{\text{rnd}}(d)$ , where  $n_{\text{reg}}(d)$  and  $n_{\text{rnd}}(d)$  are the number of observed regular and random neighbors, respectively. Consequently, in order for our algorithm to automatically adjust to the optimal aggregation density, it seems reasonable to solve the following maximization problem.

---

<sup>4</sup>We stress here that, in some cases, nodes with little similarity can still share a strong social link with predictive value, as e.g. in the case of two friends who belong each in different communities. This initial discussion serves only as a motivation, based on a simple scenario. The algorithms we propose later, while still partly based on this idea, can be applied to complex setups where such clear distinctions between “random” and “regular” neighbors start getting blurred.

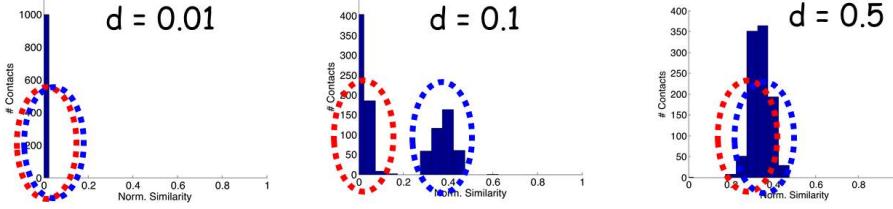


Figure 4.6: Histogram of encountered similarity values at three different aggregation levels, with 2 clusters shown for each.

$$\underset{d}{\text{maximize}} \left( \frac{\text{sim}_{\text{reg}}(d) - \text{sim}_{\text{rnd}}(d)}{n_{\text{reg}}(d) + n_{\text{rnd}}(d)} \right).$$

Figure 4.5(b) depicts the normalized similarities and their difference. It is clear from this plot, that the difference is maximized around the point where enough contacts per node have occurred to fill in most regular links, but only few random links have been instantiated. We will show later that this maximum correlates well with the aggregation density at which the performance of SimBet and Bubble Rap is optimal.

Nevertheless, all the above discussion hinges on the assumption that *a node knows which neighbors are regular and which random, in advance*. In practice however, there are no such pre-assigned labels on nodes. When a node encounters another, it only knows and logs down its similarity value to this node, and whether this similarity value should be counted for a random or regular node is not possible without knowing such labels. Consequently, in order to apply the above maximization, nodes need to first be able to distinguish between these two classes of links.

To solve this classification problem a node can create a histogram of similarity values observed, out of the contacts observed over time. One way to assign “labels” to each past contact (regular vs. random) is to perform clustering on the set of normalized similarity values. If the aggregated density is appropriate, a 2-means algorithm should produce two clusters of similarity values, one for similar regular nodes (with high similarity) and another for similar random nodes (low similarity). Note that at low (close to 0) and high densities (close to 1) only one cluster appears since all nodes have similar similarities close to 0 (no similarity) and 1 (all nodes similar), respectively. Hence, the unsupervised problem of finding the right aggregation density then becomes equivalent to maximizing the distance between the two cluster centroids of low and high similarities. Fig. 4.6 shows an example that captures this intuition, for the CAVE scenario. Two clusters are clearly separable at a density of 0.1 (which is close to the “optimal” densities of Table 4.1).

## 4.5 Robust Clustering of Similarity Values

Although the above discussion provides useful intuition, in practice, the structure guiding mobility (and thus contacts) cannot be captured in a straightforward way using the

2-means clustering approach based on normalized similarity. First, as explained earlier, there might not be 2 clear cluster centers at low and high densities. In this case, the result returned by the simple 2-means approach is unreliable. Second, the real world traces, albeit sharing basic structure with the synthetic models, exhibit more heterogeneity: nodes might belong to more than one community, degree distributions might be skewed, communities might be overlapping, and the line between regular and random contacts becomes blurred. As a result, the two classes of similarities might not be easily distinguishable even around the “correct” aggregation point. Thus, the simple clustering algorithm sketched above, based only on cluster center distance, might draw deceiving conclusions.

To cope with the difficulty of identifying a contact as random or regular, we use two approaches to assess how distinguishable the two clusters returned by the clustering algorithm are. One is *spectral analysis* of a pre-processed similarity matrix and the study of the matrix *algebraic connectivity* [156]. The second is the *Q function* of cluster modularity that has been found to correlate with cluster quality in many examples [159].

### 4.5.1 Spectral Analysis

Let us assume that a node  $u$  has collected a set of  $n$  contacts  $c_i (i = 1 \dots n)$  during a time period. Node  $u$  uses these contacts to build its view of the social graph. Each contact  $c_i$  observed is assigned a real number  $s_i$  measuring the normalized similarity between  $u$  and the node encountered  $v$ :

$$s_i = \frac{|N(u) \cap N(v)|}{\min\{|N(u)|, |N(v)|\}}, \quad (s_i \in [0, 1]), \quad (4.1)$$

where  $N(x)$  is the set of neighbors of node  $x$  in the aggregated social graph, and  $|\cdot|$  denotes cardinality. In other words, each node has now a vector  $\mathbf{s}$  of  $n$  real-valued entries in  $[0, 1]$  representing the various similarity values observed thus far. According to the previous discussion, when the right amount of contacts has been observed, the values in this vector should cluster around small and high values.

In order to formally measure this, spectral clustering [156] converts the vector  $\mathbf{s}$  into an  $n \times n$  affinity matrix  $\mathbf{W}$ .

$$\mathbf{W} = \{w_{ij}\}, \quad (4.2)$$

$$w_{ij} = \exp\left(-\frac{\|s_i - s_j\|^2}{2\sigma^2}\right), \text{ if } i \neq j, \text{ and } w_{ii} = 1, \quad (4.3)$$

where  $\sigma \in [0, 1]$  (threshold value).

Let us further define the symmetric normalized *Laplacian* of  $\mathbf{W}$  as

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}}, \quad (4.4)$$

where  $\mathbf{I}$  is the identity matrix and  $\mathbf{D}$  is a diagonal matrix whose  $(i, i)$ -element  $d_{ii} = \sum_j w_{ij}$  (i.e., is the degree of vertex  $i$  on the matrix  $\mathbf{W}$ ). Spectral Graph Theory studies the structural properties and invariants of the weighted graph defined by  $\mathbf{W}$ , using

eigenvalue decomposition of the Laplacian  $\mathbf{L}$ . Spectral clustering uses this theory to identify  $k$  strongly connected components in  $\mathbf{W}$  with few weak links between them, by projecting the  $n$  points into the eigenspace of  $\mathbf{L}$  consisting of  $\mathbf{L}$ 's first  $k$  eigenvectors. In fact, spectral clustering methods solve a relaxation of the normalized cut problem (a known NP-hard problem), which is a min cut problem under the constraint that the two cut-sets are balanced.

In the ideal case where  $\mathbf{W}$  is block-diagonal, that is, it consists of  $k$  connected components with all weights 0 between blocks, the eigenvalues  $\lambda_i, i = 1 \dots n$  of  $\mathbf{L}$  are

$$\lambda_1 = \dots = \lambda_k < \lambda_{k+1} \dots \lambda_n.$$

This means, it has exactly  $k$  eigenvalues equal to 0, and all other eigenvalues bounded away from and larger than 0. In that case, spectral clustering algorithms are guaranteed to identify clusters correctly even when they form non-convex (nonlinearly separable) regions. In the non-ideal case,  $\mathbf{W}$  is not block diagonal (i.e., is connected with lower weights between clusters), and only the first eigenvalue of  $\mathbf{L}$  is 0. Nevertheless, matrix perturbation theory [160] suggests that if the clusters are compact and modular (in other words, identifiable by a human), the eigenvalues corresponding to these clusters will still be small.

This is the basis of our algorithm that seeks to locally minimize the second eigenvalue  $\lambda_2$  of the Laplacian (known as the *Algebraic Connectivity*) of the similarity vector  $s$  observed *locally* over time.

### 4.5.2 Modularity Function Q

A different approach often used for evaluating community structure in complex networks is the use of an appropriate modularity function  $Q$  [159]:

$$Q(\mathcal{P}_k) = \sum_{c=1}^k \left[ \frac{\mathcal{A}(V_c, V_c)}{\mathcal{A}(V, V)} - \left( \frac{\mathcal{A}(V_c, V)}{\mathcal{A}(V, V)} \right)^2 \right], \quad (4.5)$$

where  $\mathcal{P}_k$  is a partition of the vertices into  $k$  groups and  $\mathcal{A}(V^c, V^c) = \sum_{i \in V^c, j \in V^c} w_{ij}$ , with  $w_{ij}$  defined as earlier. This function attempts to evaluate a particular graph partitioning by measuring the ratios of intra-community links to inter-community links. The more the intra-community link occurrence diverges from what one would expect for a random network, the more clearly separable the communities, and the higher the value of  $Q$ .

Although the modularity function approach and the spectral clustering approach are not completely different (in fact, spectral methods can be used to maximize  $Q$  [161]), these have particular strengths and weaknesses (as we have also observed for our datasets), as well as differences in implementation overhead. For this reason, we are going to present results for both approaches and compare their performance in various scenarios.

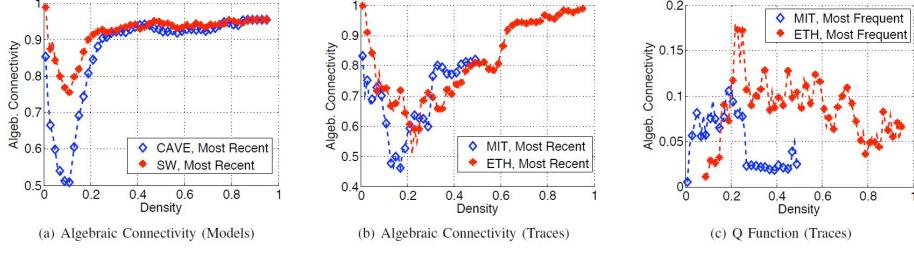


Figure 4.7: SimBet and Bubble Rap performance with different aggregation functions compared to direct transmission.

Protocol	SW	CAVE	MIT	ETH	INFO
SimBet MF	4.1/3.3	3.0/3.0	1.8/1.7	1.5/1.5	1.2/1.2
SimBet MR	3.6/3.2	2.8/2.5	2.1/2.0	1.4/1.5	1.3/1.2
Bubble MF	2.9/2.7	3.6/3.8	2.1/1.7	1.5/1.5	1.3/1.2
SimBet MR	3.2/3.3	3.4/3.2	1.8/1.3	1.4/1.4	1.1/1.2

Table 4.2: Relative performance improvement compared to Direct Transmission. The first value for each entry corresponds to performance obtained with algebraic connectivity, the second value to Q modularity.

### 4.5.3 Online optimal density tracking algorithm

The idea is to find the density at which the Algebraic Connectivity of observed similarity values is minimal, or alternatively, the Q function is maximal. While collecting and logging  $n$  contacts<sup>5</sup>, we compute similarity values, using the aggregated graph at different densities. After  $n$  contacts, we use these similarities to determine the gradient of the Algebraic Connectivity (or Q Function) and adapt the density of operation accordingly, towards the optimum. In order not to get stuck in local extrema, we use occasional random lookaheads.

## 4.6 Performance Evaluation

Using this simple tracking algorithm, we evaluate in Fig. 4.7 how close we get to the optimal performance (that we identified offline in Table 4.1. Fig. 4.7(a) shows the cumulative number of messages BubbleRap delivers, when we create messages between random senders and destinations throughout the simulation. We choose the density with the online algorithm using the Algebraic Connectivity metric, and compare it to the original offline algorithm with the naive growing time window (where density increases over time), and to direct transmission. We observe that after a short initial period, the growing time window graph becomes too dense to be useful, and the performance degrades compared to that of direct transmission (i.e., the two lines are par-

<sup>5</sup>We set the update interval  $n$  empirically to 100 contacts, as a tradeoff between having enough similarity samples, and reacting sufficiently fast to changes in the network.

allel). On the other hand, our online algorithm keeps the good performance throughout the simulation with both the most recent and most frequent mappings. Most frequent performs a bit better in this case, as community links are more frequent than random ones in the CAVE model, by design.

A similar result is shown in Fig. 4.7(b), where we use SimBet and the ETH trace with the same simulation setup. However, in this case the performance increase is a bit smaller, since the trace is less structured, as reported earlier. Fig. 4.7(c) shows the evolution of the chosen density by our online algorithm in the course of trace time (MIT). Our algorithm moves around a density of 0.09, which indeed coincides with the best performance benefit calculated offline. We note here that, in the case of the traces, we have also observed some small variance in the density value that achieves the best performance at different time slices of the trace, indicating some non-stationarity in the process. Nonetheless, our online algorithm seems to be capable, in most cases, to track a close to optimal density value.

Table 4.2 summarizes the performance improvement factor for all combinations of protocols, aggregation functions, and contact processes. Most values are close to the ones shown earlier in Table 4.1. In some cases, for instance SimBet with the MIT trace, we even reach the reported optimal values. Note also, that the performance of the Algebraic Connectivity and the Q Function versions are very similar. In some cases, Algebraic Connectivity is slightly better. However, the computation of the Q values is significantly less complex (i.e., less similarity pre-processing). The choice between the two is thus a tradeoff between complexity and a slightly better performance.

## 4.7 Complex Network Analysis of Human Mobility

Having established an efficient method to infer a contact graph out of a sequence of pairwise contacts online, we then went ahead to analyze a number of mobility traces to understand the extent and type of social properties appearing in such traces. These traces range from actual contact traces (e.g. logs of Bluetooth associations) to WiFi network logs (showing which nodes are associated with which AP and when) and data from geosocial applications. We report here some results from this study. More details can be found in [39].

**Dartmouth (DART)** We use 17 weeks of the WLAN Access Point (AP) association trace [162], between Nov. 2nd 2003 and Feb. 28th 2004. We choose the 1044 nodes which have activities at least 5 days a week on average i.e., the nodes have associations at least  $5 \times 17 = 85$  days. The trace is pre-processed to remove short disconnections ( $< 60s$ ) which we attribute to interference and other non-mobility related effects, as well as the well known ping-pong effect where devices jump back and forth between different APs in reach. We assume that two nodes are in contact when they are associated to the same AP at the same time.

**ETH Campus (ETH)** Using the same methodology as for Dartmouth, we process a trace collected at the ETH campus [163] during almost 15 weeks between Oct. 25th 2006 and Feb. 3rd 2006. Similarly, we choose 285 nodes which connect to the WLAN AP network at least 5 days a week (i.e., 75 days).

**Gowalla (GOW)** Gowalla was a location-based service where users check-in to

	<b>DART</b>	<b>ETH</b>	<b>GOW</b>	<b>MIT</b>
<b># People and context</b>	1044 campus	285 campus	473 Texas	92 campus
<b>Period</b>	17 weeks	15 weeks	6 months	3 months
<b>Type</b>	AP associations	AP associations	Self-reported location	Bluetooth scanning
<b># Contacts total</b>	4'200'000	99'000	19'000	81'961
<b># Contacts per dev.</b>	4'000	350	40	890

Figure 4.8: Mobility traces characteristics.

close-by spots (e.g., restaurants, office buildings), thereby logging their position. We use the publicly available location data of 473 heavy-users who, during the 6 months from Apr. to Sept. 2010, check-in at least 5 days a week somewhere in the State of Texas in the United States. Since users only check-in and do not check-out, we cannot infer the stay duration at a spot. Therefore, we assume users are in contact when they check-in less than 1 hour apart at the same spot. As we do not know the duration of a contact, we assume all contacts have the same duration (1 hour).

**MIT Reality Mining (MIT)** The MIT trace [164] logs contacts between 92 campus students and staff, detected by Bluetooth scans with a scanning interval of 5 minutes. We take a 3 months long piece of the trace. It is the only trace we use where contacts are measured directly (and not inferred from location). However, supposedly, many short contacts are not registered due to the relatively long scanning interval.

Note that these traces differ vastly in their nature, and different traces capture different aspects of mobility. For instance the Gowalla trace, by the nature of the application, mainly captures the mobility of users while they go out and socialize. The DART trace captures students and staff at home and at the university, whereas the ETH trace captures only work behavior, since there are no APs in residential buildings. MIT is the smallest trace but captures work, home and leisure equally. We believe that due to the variety of the traces, our results are general even though individual traces may be biased. Details for these traces are summarized in the table of Fig. 4.8.

Fig. 4.9 shows as an example the DART contact graph (at an appropriately chosen density). We observe that there is strong non-random structure. To quantify this structure, we first measure some standard metrics such as average shortest path lengths and clustering coefficients. With these properties we can examine the graphs for small-world characteristics. Small-world networks, according to [30], manifest short paths between nodes (a typical property of random, Erdos-Renyi or Poisson graphs) and high clustering coefficient (tendency of relations to be transitive).

Table 4.3 shows the average clustering coefficients for different contact graph densities values of 1-4%. Note that for a random (“Poisson”) graph, the clustering coefficient increases linearly with density from 0 to 1. Thus, in a graph where 10% of the node pairs are connected, the expected clustering coefficient is 0.1. The values shown in the table suggest that all scenarios are considerably more clustered than a random graph would be: for example, in the DART trace for a density of 3%, the measured clustering co-efficient is 0.57, which is an order of magnitude higher than that of a Poisson graph with the same density. This finding strongly suggests non random connectivity. We observe that the clustering coefficient of DART, ETH and MIT are very high and strikingly similar, whereas the GOW trace is a bit less clustered. We

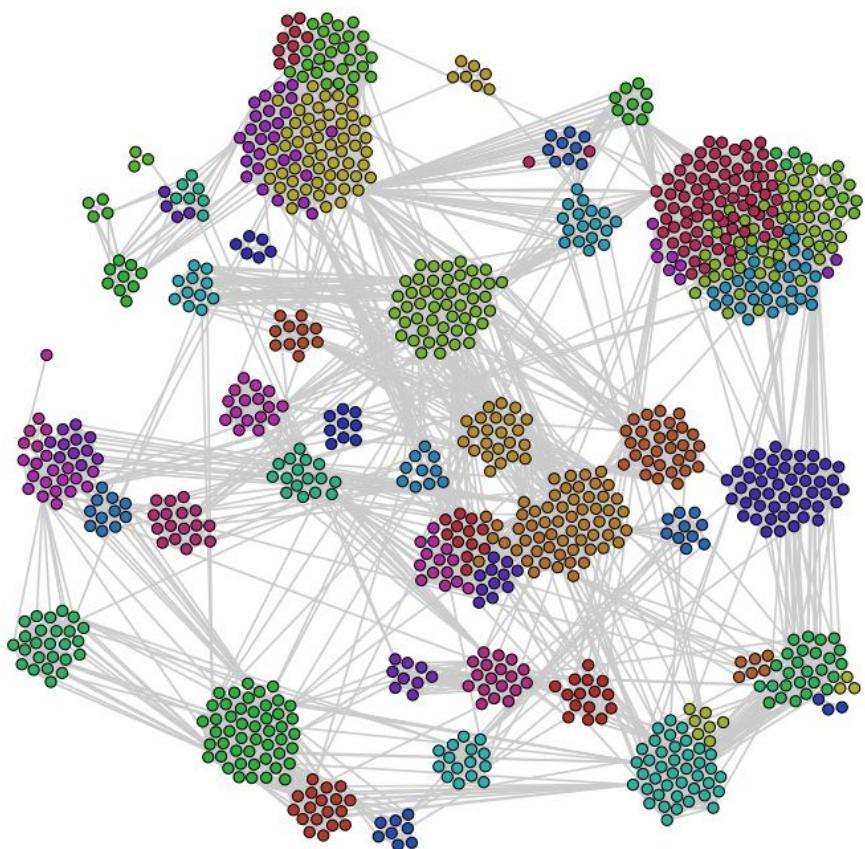


Figure 4.9: Contact Graph for DART trace.

	Clustering Coefficient				Avg. Path Length			
	1%	2%	3%	4%	1%	2%	3%	4%
<b>DART</b>	0.71	0.63	0.57	0.54	7.4	3.7	2.9	2.6
<b>ETH</b>	-	0.66	0.57	0.53	-	6.1	5.6	4.0
<b>GOW</b>	0.28	0.27	0.27	0.26	4.5	3.4	3.0	2.8
<b>MIT</b>	-	-	0.56	0.57	-	-	4.6	3.8

Table 4.3: Clustering Coefficients and Average Path Lengths using different graph densities. Missing values in cases where there is no giant component.

Trace/Model	# Communities	Q
<b>DART</b>	23	0.84
<b>ETH</b>	21	0.81
<b>GOW</b>	29	0.7
<b>MIT</b>	6	0.52

Table 4.4: Number of communities and modularity (Q).

attribute this to the different nature of the traces: Transitivity of ties in work and home environments is stronger than in social activities captured by Gowalla.

Looking at the average shortest path length, we see that paths are only few hops long on average. Thus, *we observe the small-world behavior, typical for social networks, also in the network of physical encounters.*

This latter finding is related to the report of short opportunistic paths by Chaintreau et. al. [165]. However, [165] measures path length in terms of number of relays using epidemic dissemination (i.e., messages are copied at every encounter). Their result means that *short paths exist, when accounting for both, strong and weak ties* (pairs which meet often and ones that only meet randomly). The way we create contact graphs out of mobility traces, we limit the edges to strong ties. Nevertheless, we find that paths are still short. This is an important distinction for designing dissemination protocols, where forwarding decisions must be made based on regular encounters and not random, unpredictable ones, as for example [38].

Additionally to the described small-world characteristics, we were interested in the existence of communities in the contact graph. Communities, informally defined as subsets of nodes with stronger connections between them than towards other nodes, are typical for the structure of social networks. To detect communities in the contact graph, we apply the widely used Louvain community detection algorithm [166], and to measure how strongly modular the resulting partitioning is, we use Newman's Q function, introduced earlier.

The number of identified communities and the respective Q values are reported in Table 4.4. We observe that in all cases, we get high modularity values between 0.52 and 0.84 (Q values of larger than 0.3 suggest strong community structure). Note that high modularity values of the graph of aggregate contact durations have also been reported in [36] (for smaller traces and different community detection algorithms),

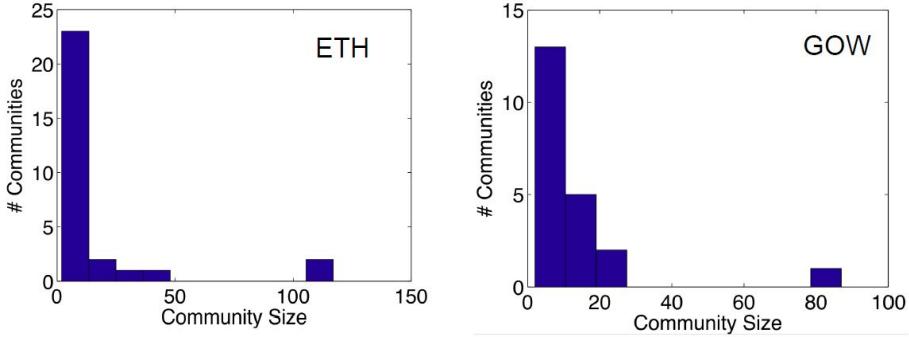


Figure 4.10: Community sizes.

thus our findings are in agreement. Relating the modularities to clustering coefficients (see Table 4.3), we observe on one hand that for DART and ETH, where we measure high clustering coefficients, also modularities are strong. On the other hand, MIT having similarly high clustering coefficient, has the lowest modularity, and GOW has a high modularity despite having the lowest clustering coefficient among all traces. This confirms that clustering coefficient and modularity measure different aspects of clusteredness: While the social nature of the GOW contacts makes them less transitive in terms of forming triangles, they are still grouped into larger communities.

To get an impression of the sizes of such communities, we show two examples of community size histograms in Fig. 4.10. In both, ETH and GOW we observe that the majority of communities are small ( $\leq 10$  nodes), yet, there are few very large communities. The same observation holds for DART, not shown here. MIT is, with only 6 communities, too small to draw conclusions on the community size distribution. These large communities, some of which have 100 and more nodes, raise the question about the community internal structure: *Are communities homogenous entities within which all nodes have similarly strong connections to all other community members, or do they manifest more complex internal structure?*

To study this latter question, we evaluated the degree differences of nodes inside the same community, as well as across communities. We refer the interested reader to [39] for the details of our methodology and findings. We just include here some example plots in Fig. 4.11. The (normalized) degree distributions are plotted both for nodes inside the same community (“Community”) and across all nodes (“Global”). Interestingly, these curves appear as almost straight lines in the log-linear scales shown, for all traces considered (including ones not shown), suggesting an exponential degree distribution with much smaller skewness than usually encountered in complex networks. This finding raises some questions as to the efficiency of intra- and inter-community routing schemes based on increasing degree centrality of the relay (see e.g. [36]), which is known to be an efficient search strategy for networks with scale-free degrees.

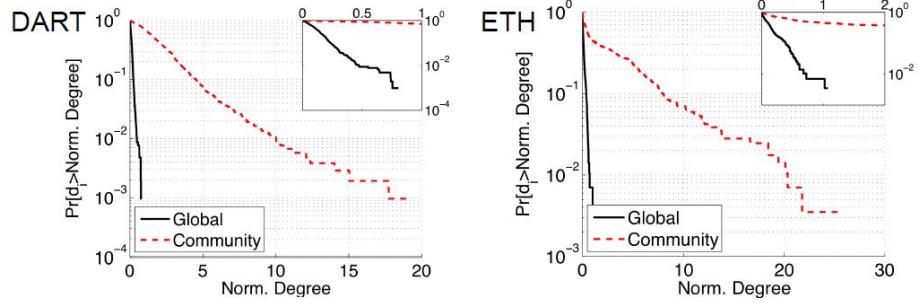


Figure 4.11: CCDFs for normalize global and community degree in log-linear scale. Insets show the same values with different x-axis range.

#### 4.7.1 Mobility Modeling

As a last direction in this complex network analysis of opportunistic networking, we revisited some popular mobility models, namely TVCM [?], HCMM [167], and SLAW [130]. These synthetic models had proven their merit against the well known findings regarding pairwise contact processes [168]. These properties could be characterized as “micro-scopic” in the sense that they focus on specific node pairs and the timing of contacts. Our goal was to investigate whether these model further capture the “macroscopic” properties observed in real mobility traces, that we have seen exhibit many similarities to social or complex networks.

While these models reproduce more or less realistic community structure - reflecting groups of nodes who work or live together - and were in agreement with our real trace findings, they failed to correctly capturing what happens between such communities: they are often connected by few bridging links between nodes who socialize outside of the context and location of their home communities. To this end, we studied this phenomenon in more detailed and proposed simple modifications that can be applied *on top* of existing mobility models. The modified models correctly capture the bridging link behavior without violating any of the previously mentioned micro-scopic or macro-scopic properties. The details of this study can be found in [40].

## 4.8 DTN Meteo: A Generic Framework for Predicting DTN Protocol Performance

The previous approaches are able to provide good approximations or upper bounds for the performance of epidemic routing, and simple variants. The common denominator of all the protocol variants is that they are “random”, in the sense that none of them explicitly differentiates among nodes, based on their characteristics (mobility, resources or other), when making a forwarding/dropping decision. In contrast, almost all newer DTN algorithms and protocols [35, 169, 170] base their decisions on nodes’ utilities with respect to the scheme’s end goal (position in the network, relationship to destination etc). These node utilities are most often based on the observation that human mobility has non-trivial structure, manifesting in mixing/meeting patterns which can be extracted from real world experiments [171, 172].

We propose such a common analytical framework, which we entitle DTN-Meteo, and which is based on the following **three common features** of most DTN protocols. Firstly, the bulk of proposed algorithms, whether for routing, content dissemination, distributed caching etc, essentially solve a combinatorial optimization problem over the state of every node in the network. Each algorithm defines a preference (*utility*) function over possible network states and aims at moving to better states. Secondly, candidate new states, in the DTN context, are presented according to the *stochastic* mobility of the nodes involved. As such, *the traversal of the solution space of a problem is also stochastic*. And thirdly, due to the difficulty (usually impossibility), in this context, of updating nodes’ states globally, protocols resort to *local search* heuristics (with a deterministic or randomized search strategy) to choose between the current and a new possible state, *involving state changes only in the two nodes in contact*.

Using the above insight, DTN-Meteo maps a combinatorial optimization problem into a Markov chain (MC), where each state is a potential solution (e.g. assignment of content replicas to nodes), and transition probabilities are driven by two key factors: (i) *node mobility*, which makes available new potential solutions to the algorithm, over time, based on the (heterogeneous) contact probability of each node pair, and (ii) *the algorithm*, which orders the states using a utility function and accepts better ones either deterministically (*greedy algorithm*) or probabilistically (*randomized algorithm*), to avoid getting stuck in local maxima<sup>6</sup>. This not only decouples the algorithm’s effect and the mobility’s effect from each other, but also enables deriving useful performance metrics (convergence delay, delivery probability) using transient analysis of this MC.

Let  $\mathcal{N}$  be our Opportunistic Network, with  $|\mathcal{N}| = N$  nodes.  $\mathcal{N}$  is a relatively sparse ad hoc network, where node density is insufficient for establishing and maintaining (end-to-end) multi-hop paths, in the sense of [174]. Instead, data is stored and carried by nodes, and forwarded through intermittent contacts established by node mobility. A *contact* occurs between two nodes who are in range to setup a bi-directional wireless link to each other.

We assume an optimization problem over the  $N$ -node network  $\mathcal{N}$  (e.g. multicast under resource constraints) and a distributed algorithm for this problem, run by all

---

<sup>6</sup>In the spirit of solutions like Markov Chain Monte Carlo optimization and simulated annealing, for traditional optimization problems [173].

nodes. Our long term aim is to understand the performance of the algorithm as a function of the nodes' behavior and attributes (mobility, collaboration, resource availability etc). Table 4.5 summarizes this section's notation.

$\mathcal{N}$ and $N$	the network (set of nodes), resp. its size
$i, j, k$	nodes in $\mathcal{N}$
$L$	maximum allowed replication (number of copies)
$\mathcal{S}$ and $\Omega$	node, resp. network state space
$\mathbf{x}, \mathbf{y}, \mathbf{z}$	network states (elements of $\Omega$ )
$U(\mathbf{x}), u(i)$	utility of a network state, resp. of a node
$\delta(\mathbf{x}, \mathbf{y})$	difference between network states $\mathbf{x}$ and $\mathbf{y}$
$A_{\mathbf{xy}}$	acceptance probability for transition $\mathbf{xy}$
$p_{ij}^c$	contact probability between nodes $i$ and $j$
$\mathbf{P} = \{p_{\mathbf{xy}}\}$	Markov transition probability matrix
$T_m^{(ij)}, T_n$	starting times for $(i, j)$ contacts , resp. any contacts
$J_m^{(ij)}, J_n$	intercontact times for $(i, j)$ , resp. any contacts
$K^{(ij)}(t)$	residual intercontact times for $(i, j)$
$\alpha_{ij}$	power law exponent of $(i, j)$ intercontact times
$\beta_{ij}$	rate parameter of $(i, j)$ intercontact times
$F_{ij}, \overline{F}_{ij}$	CDF, resp. CCDF of $(i, j)$ intercontact times

Table 4.5: Important notation for Section 4.8.

#### 4.8.1 Solution Space

We consider a class of DTN problems for which a solution can be expressed in terms of nodes' states. In this paper, we restrict ourselves to *binary node states*, to better illustrate the key concepts; however, in a more realistic variant of DTN-Meteo, nodes' states could be chosen from a set of *B-bit integers* (e.g. to allow the modeling of some storage constraint of  $C \leq B$  messages for each node, out of a maximum  $B$  contemporaneous messages in the network). In all cases, the space of candidate solutions for such problems is a set of  $N$ -element vectors, possibly restricted by a number of constraints. Finally, an algorithm for the problem defines a ranking over these solutions, captured by a utility function  $U(\cdot)$ . The goal is to maximize this utility (or minimize a cost function). We define our class of problems as follows:

- **node state space**  $\mathcal{S} = \{0, 1\}$  or  $\mathcal{S} \subset \mathbb{N}$  (4.6)

- **network state space**  $\Omega \subseteq \mathcal{S}^N$

$$\Omega = \{\mathbf{x} \mid \mathbf{x} = (x_1, x_2, \dots, x_N)\}, \quad x_i \in \mathcal{S}, \quad \forall i \in \mathcal{N} \quad (4.7)$$

- a set of **constraints**  $f_i(x_1, \dots, x_N) \leq \rho_i$  (4.8)

- a **utility function**  $U : \Omega \mapsto \mathbb{R}$  (4.9)

This is, in fact, the combinatorial optimization class, which naturally encompasses several DTN problems, as they are dealing with indivisible entities (nodes, messages, channels etc) and have rules that define a finite number of allowable choices (choice of relays, assignment of channels etc). Below are some examples of DTN problems that can be thus modeled.

**Content placement.** The goal in content placement, is to make popular content (news, software update etc) easily reachable by interested nodes. As flooding the content is unscalable, a small number  $L$  of replicas can be pushed from its source to  $L$  strategic relays, which will store it for as long as it is relevant, and from whom encountered interested nodes retrieve it<sup>7</sup>. In its simplest form, the source of the content distributes the  $L$  replicas to  $L$  initial nodes (e.g. randomly, using binary or source spraying [175]). These initial relays then forward their copies only to nodes that improve the desired utility – which can be based on mobility properties, willingness to help, resources, etc. (see [170] for some examples).

Here, the binary state of a node  $i$  is interpreted as the node being ( $x_i = 1$ ) or not being ( $x_i = 0$ ) a provider for the content of interest<sup>8</sup>. There is a single constraint for any allowed solution, namely  $\sum_{i=1}^N x_i \leq L$ , that is, in Eq. (4.8)  $\rho_i = \rho = L$ .

**Routing.** In routing, be it unicast, multicast, broadcast, or anycast, the binary state of node  $i$  is interpreted as carrying or not carrying a message copy. For example, for unicast routing from source node  $s$  to destination node  $d$ , the initial network state is  $\mathbf{x}_0 = (0, 0, \dots, \{x_s = 1\}, \dots, \{x_d = 0\}, \dots, 0)$ . The desired network state is any  $\mathbf{x}^*$ , with  $x_d = 1$  and  $x_i \in \{0, 1\}$ ,  $\forall i \neq d$ . This can easily be extended to group communication, with multiple sources and/or multiple destinations.

Various replication strategies can be expressed using Eq. (4.8) constraints, as shown in Table 4.6. Different schemes in each category, essentially differ in the utility function used to rank states and the action taken given a utility difference. In this paper, we analyze the first two strategies in Table 4.6.

	Replication		Constraint
i)	single-copy	[15, 12, 35]	$\sum_{i=1}^N x_i = 1$
ii)	fixed budget $L$	[175, 147]	$\sum_{i=1}^N x_i \leq L$
iii)	epidemic	[11]	$\sum_{i=1}^N x_i \leq N$
iv)	utility-based <sup>9</sup>	[105, 176, 169]	$\sum_{i=1}^N x_i \leq N$

Table 4.6: Modeling replication with Eq. (4.8)

As a final note, replication-based schemes have an initial spreading phase, during which the  $L$  copies are distributed to the initial relays. This phase can straightforwardly be included in our model, by considering also all states with 1 to  $L - 1$  copies in the network. This, however, increases the complexity of the model, while offering little added value. Indeed, the delay of this initial phase is negligible in networks with large  $N$  and  $L \ll N$ , which is the case in scenarios of interest (see e.g. [175]). Hence, we will only consider the effect of this phase on the starting configuration (as shown in Section 4.9.1, Eq. (4.25)) of the algorithm and ignore its delay. Thus, our solution

<sup>7</sup> $L$  must be derived to achieve a trade-off between desired performance and incurred costs, for example as in [175].

<sup>8</sup>A  $B$ -bit integer node state extends this to providing  $B$  pieces of content.

<sup>9</sup>Replicating only to higher utility nodes limits copies implicitly, but in the worst case, there will still be  $N$  copies.

space  $\Omega$ , will only be composed of network states with exactly  $L$  copies at  $L$  different nodes.

### 4.8.2 Exploring the Solution Space

In traditional optimization problems, local search methods define a neighborhood around the current solution, evaluate all solutions in the neighborhood and “move” towards the best one therein (purely greedy algorithms). Occasionally, they may also move to lower utility states (using randomization) in order to overcome local maxima, as in randomized local search algorithms (e.g. simulated annealing). This aspect is fundamentally different in DTN optimization. The next candidate solution(s) cannot usually be chosen. Instead, the solution space  $\Omega$  is explored via node mobility (contacts): a new potential solution will only be offered, after some time, when two nodes come in contact. This has two major implications: (i) this new solution can differ (from the current) in the state of at most two nodes: the ones involved in the contact; (ii) which pair of nodes meets next (a random event) will define which potential *new* state will become available, and how much better/worse the new state will be (whether the transition to the new state occurs, is finally decided by the algorithm). As a result, the traversal of the solution space is inherently *stochastic*.

Consider implication (i) first (we treat (ii) when defining our network model, in Section 4.8.4). Every contact between a relay node/content provider  $i$  ( $x_i = 1$ ) and another node  $j$  ( $x_j = 0$ ) offers the chance of moving to a new network state  $\mathbf{y} \in \Omega$ , with  $y_i = 0$  and  $y_j = 1$  (forwarding) or  $y_i = 1$  and  $y_j = 0$  (replication). *If* the replica is transferred from relay  $i$  to  $j$ , then the  $\mathbf{x}\mathbf{y}$  transition happens. Fig. 4.12 provides examples of *potential* state transitions, along with contacts required for the transitions to be possible.

Transition  $\mathbf{x}\mathbf{z}$  in Fig. 4.12 requires two contacts to *start* simultaneously, and is thus not possible<sup>10</sup>. This means that only transitions between *adjacent* states are possible, where we define state adjacency as:

$$\delta(\mathbf{x}, \mathbf{y}) = \sum_{1 \leq i \leq N} \mathcal{I}\{x_i \neq y_i\} \leq 2, \quad (4.10)$$

i.e., the two network states may only differ in one or two nodes ( $\mathcal{I}\{x_i \neq y_i\}$  are binary indicator variables). An encounter of those two nodes is a necessary (but not sufficient) condition for transition  $\mathbf{x}\mathbf{y}$  to happen.

Summarizing, our solution space exploration goes as follows: When at a state  $\mathbf{x}$ , the next contact between two nodes  $i, j$  presents a new potential solution  $\mathbf{y}$  to the algorithm with a (time-homogeneous) probability  $p_{ij}^c$ , the contact probability of node pair  $(i, j)$  (we look deeper into these contact probabilities and the assumptions behind them in Section 4.8.4). The new solution  $\mathbf{y}$  differs from  $\mathbf{x}$  in positions  $i$  and  $j$  only. For example, in Fig. 4.12(a): when at state  $\mathbf{x} = (0, 0, 0, 1, 1, 1)$  the algorithm could move to a new solution  $\mathbf{y} = (0, 1, 0, 0, 1, 1)$  in the next contact, with probability  $p_{24}^c$ .

---

<sup>10</sup>We emphasize that this is not a strict necessity for our networks of interest, but rather a technical assumption of our mobility model, for analytical convenience (see Section 4.8.4 for more details). However, given the relative sparsity of the networks in question, such events occur with low probability.

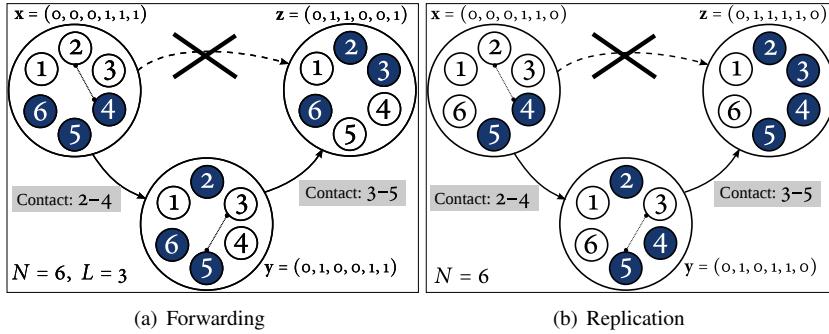


Figure 4.12: Example state transitions in two 6-node DTNs.

### 4.8.3 Modeling a Local Optimization Algorithm

Node contacts merely *propose* new candidate solutions. Whether the relay  $i$  does in fact hand over its message or content replica to  $j$  (or, more generally, whether a new allocation of objects between  $i$  and  $j$  is chosen), is decided by the *algorithm*. In purely greedy (utility-ascent) schemes, a *possible* state transition occurs only if it improves the utility function  $U$ , specific to each problem. Then, for our DTN problems, a possible transition  $xy$  occurs with binary *acceptance probability*:

$$A_{xy} = \mathcal{I}\{U(x) < U(y)\}. \quad (4.11)$$

More generally, the acceptance probability may be any function of the two utilities:  $A_{xy} \in [0, 1]$ . This allows DTN-Meteo to model randomized local search algorithms (e.g. Markov Chain Monte Carlo methods, simulated annealing) as well.

It is important to make some remarks about utilities here. First, the (global) utility values above may sometimes require knowing the state of remote nodes (in the worst case, all nodes). This state may not be readily available locally at each node. Therefore, the algorithm must solve the added problem of collecting information and/or estimating it – usually through an integrated sampling component [105, 176]. Most DTN protocols suppose the existence of a mobility-related node utility  $u : \mathcal{N} \mapsto \mathbb{R}$  (e.g. contact frequency). Thus, if the mobility is stationary and the estimation designed correctly, the online and offline utility rankings will coincide. However, estimating node properties and utilities is beyond the scope of this work.

Second, in any *distributed* algorithm, a node must be able to evaluate *locally* whether a given decision (e.g. to hand over its copy to an encountered node, for routing) will improve the global utility. This is a challenge for the algorithm designer, often solved by using *additive* global utilities (e.g.  $U(x) = \sum_{i=1}^N x_i \cdot u(i)$ , where  $u : \mathcal{N} \mapsto \mathbb{R}$  can be applied to each node separately). Therefore, DTN-Meteo assumes any state utility  $U$  and/or node utility  $u$  are readily available.

Finally, we note that some protocols use packet-based utilities [176], while in DTN-Meteo we assume node-based utilities. Furthermore, the types of packet utilities in [176] are time-dependent. While such time-dependent packet utilities cannot be modelled by our framework (as explained immediately below), some types of static

packet-based utilities could be included in future work.

Summarizing from sections 4.8.1, 4.8.2, and 4.8.3, the transition probability between *adjacent* network states  $\mathbf{x}$  and  $\mathbf{y}$  can be expressed in function of the contact probability (to be discussed in Section 4.8.4) and the acceptance probability as follows:

$$p_{\mathbf{xy}} = p_{ij}^c \cdot A_{\mathbf{xy}}, \quad (4.12)$$

where nodes  $i$  and  $j$  are the two nodes whose encounter *could* provoke the state transition.  $p_{ij}^c$  is the **mobility component** of the transition probability and  $A_{\mathbf{xy}}$  is the **algorithmic component**.

We must stress here that Eq. (4.12) assumes that both variables are time-invariant, such that the resulting Markov chain for the problem will be time-homogeneous. Specifically, this requirement on the algorithm component  $A_{\mathbf{xy}}$  implies time-invariant (or very slowly varying) utilities, such as ones derived from social relationships among humans carrying the devices (nodes) which form our network. Additional types of utilities that are time-invariant (or very slowly varying), that DTN-Meteo can support, include functions based on computational power, operating system, trust relationships tied to social closeness, long-term traffic patterns etc.

The mobility component  $p_{ij}^c$  (contact probability between  $i, j$ ) is also assumed to be *time-homogeneous*. This is not unreasonable, as the above probabilities refer to long-term contact relations of humans, which tend to be stable. Some diurnal patterns might be present, but these could be further handled as discussed in the Section 4.8.6.

Therefore, for any two states  $\mathbf{x}$  and  $\mathbf{y}$ , our algorithm is a discrete-time time-homogeneous Markov chain  $(\mathbf{X}_n)_{n \in \mathbb{N}_0}$  over the solution space  $\Omega$ , described by the transition probability matrix  $\mathbf{P} = \{p_{\mathbf{xy}}\}$ , with:

$$p_{\mathbf{xy}} = \mathbb{P}[\mathbf{X}_{n+1} = \mathbf{y} \mid \mathbf{X}_n = \mathbf{x}] = \begin{cases} 0, & \delta(\mathbf{x}, \mathbf{y}) > 2 \\ p_{ij}^c \cdot A_{\mathbf{xy}}, & 0 < \delta(\mathbf{x}, \mathbf{y}) \leq 2 \\ 1 - \sum_{\mathbf{z} \neq \mathbf{x}} p_{\mathbf{xz}}, & \mathbf{x} = \mathbf{y}. \end{cases} \quad (4.13)$$

with  $i, j$  as before.

This formulation allows us to transform any problem of our defined class into a simple Markov chain, which can be used for performance analysis and prediction. Fig. 4.13 presents the necessary steps to apply DTN-Meteo.

#### 4.8.4 Modeling Heterogeneous Node Contacts

Recall our Opportunistic Network  $\mathcal{N}$ , with  $|\mathcal{N}| = N$  nodes and  $E \leq \binom{N}{2}$  node pairs contacting each other. As data is exchanged only upon contacts in  $\mathcal{N}$ , a mobility model based on contact patterns is sufficient for our analysis.

**Definition 4.8.1 (Node pair processes).** *Every node pair's contact process is an independent, stochastic, stationary, and ergodic renewal process  $(T_m^{(ij)})_{m \in \mathbb{Z}}$ , where  $T_m^{(ij)}$  denotes the starting time of a contact between nodes  $(i, j)$ . The random variables*

## Applying the DTN-Meteo Framework in Practice

1. **Mobility.** Extract all pairwise contact rates  $\beta_{ij} > 0$  from your mobility scenario. To do this, use the assumption that pairwise intercontacts are distributed as a power law with exponential cutoff, as in Eq. (4.14) and a parameter estimation method (e.g., Maximum Likelihood Estimation – MLE).
2. **Problem.** Identify your problem's state space. Many problems can be described with binary node states, which results in a problem state space  $\Omega$  formed of binary vectors of length  $N$  (size of the network). For example, in epidemic routing,  $\Omega$  consists of all possible binary vectors of size  $N$ .
  - Add any constraints your problem may have, to help you reduce the state space size.
3. **Optimization Algorithm.**
  - Based on your state space  $\Omega$  from Step (2) and the contact probabilities from Step (1), create  $\mathbf{M}$ , an  $|\Omega| \times |\Omega|$  probability matrix, describing which contact(s) are needed for which state transition, as in Fig. 4.12.
  - Using the utility function of your protocol, calculate the utility of each state of  $\Omega$ . Based on these utility values, create  $\mathbf{A}$ , another  $|\Omega| \times |\Omega|$  matrix containing acceptance probabilities  $A_{xy}$  (either binary or in  $[0, 1]$ ).
4. **Markov chain.** Take the element-wise (Hadamard) product of  $\mathbf{M}$  and  $\mathbf{A}$ , to obtain the transition matrix of your Markov chain  $\mathbf{P} = \mathbf{M} \circ \mathbf{A}$ .
5. **Analysis.** Using standard Markov chain analysis, obtain performance results for your triple combination of: < mobility – problem – algorithm >.

Figure 4.13: Summary of the necessary steps when building a DTN-Meteo model.

$J_m^{(ij)} = T_m^{(ij)} - T_{m-1}^{(ij)}$  are the times between the initiations of two successive contacts between nodes  $(i, j)$ . The process being renewal implies that the times  $J_m^{(ij)}$  are IID with a generic distribution.

The main implications of the above definition (stationarity and independence) are discussed in detail in Section 4.8.6. We just stress here, that we do not claim all these assumptions to hold in real scenarios (though they are more realistic than in previous models). They just represent our chosen tradeoff between the tractability and prediction accuracy of our model, the latter being validated against real traces.

**Corollary 3 (The network process).** *The entire network  $\mathcal{N}$  is described by the superposition of the  $E$  individual  $(T_m^{(ij)})_{m \in \mathbb{Z}}$  processes (which are mutually independent), forming a new stationary and ergodic marked point process  $(M_n)_{n \in \mathbb{Z}} = \{T_n, \sigma_n\}$ , with  $\sigma_n = (i, j)$ . Therefore, the newly indexed time instants  $(T_n)_{n \in \mathbb{Z}}$  are epochs of a superposition of  $E$  renewal processes. The new holding times are  $J_n = T_n - T_{n-1}$ .*

Making the following two technical provisions, the model is summarized in Fig. 4.14 for a small toy network of  $N = 4$  nodes and  $E = 4$  contacting pairs out of the 6 possible:

- (a)  $T_n < T_{n+1}, \forall n \in \mathbb{Z}$  – the probability of two contacts starting at exactly the same time is negligible<sup>11</sup>, i.e.  $J_n > 0$ .
- (b) the duration of a contact is negligible compared to the time between two contacts, but sufficient for all data transfers to take place.

As made evident in Fig. 4.14, any holding time  $J_n = T_n - T_{n-1}$  of the aggregate process is, in fact, the minimum of (i)  $E - 1$  residual times of the pairwise holding times  $J_m^{(ij)}$ , and (ii) one full pairwise holding time corresponding to the latest renewal. Since we are planning to define a Markov model on the network's contact process, the probability distribution of the superposition's holding times  $J_n = T_n - T_{n-1}$  is crucial.

Based on the above contact model, it is easier to see how our transition probability matrix  $\mathbf{P}$ , defined in Eq. (4.13) fits into the picture. We are essentially embedding a discrete time process at points  $T_n$ , the epochs of the superposition process. Whether this embedded process is indeed Markovian and time-homogeneous will thus depend on the superposition process  $(M_n)_{n \in \mathbb{Z}}$  and its holding times  $J_n$ .

If the  $E$  individual contact processes  $(T_m^{(ij)})_{m \in \mathbb{Z}}$  are Poisson (i.e., pairwise intercontacts  $J_m^{(ij)}$  are exponential), an assumption commonly made in most related literature, then it is easy to see that their residuals are exponential as well. This means that the superposition epochs  $T_n$  define a Poisson process and the (embedded) Markov chain of Eq. (4.13) can be used to analyze performance, despite the heterogeneous contact rates.

However, the assumption of purely exponential intercontact times has been challenged by real collected contact traces [177]. Recently, a solid case has been made [178, 28] for pairwise intercontact times  $J_m^{(ij)}$  being distributed as a power law with an exponential cutoff. Using this as a starting point, we will investigate in the following,

---

<sup>11</sup>Even in reality, this is highly improbable in our sparse networks.

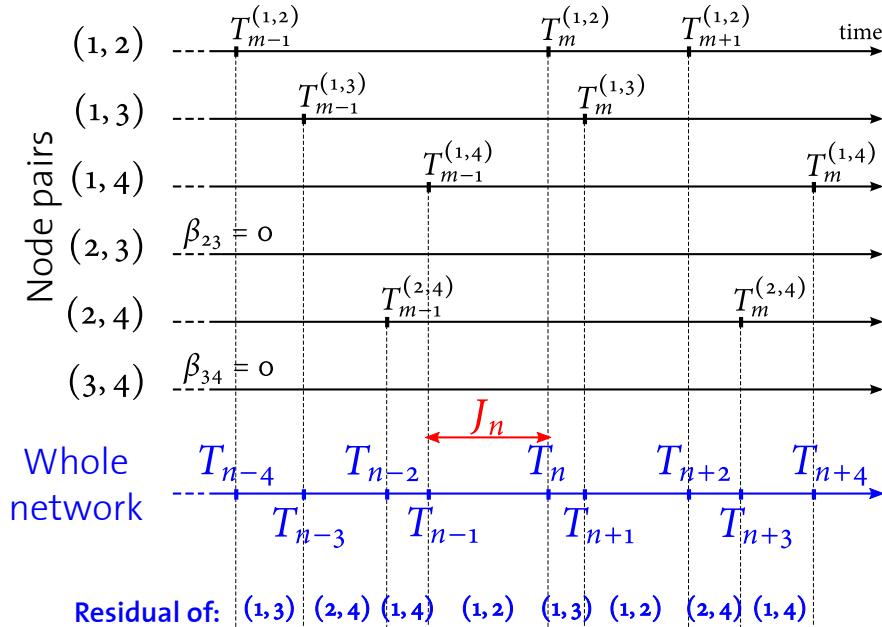


Figure 4.14: Contact model for a toy network with  $N = 4$  nodes and  $E = 4$  out of 6 node pairs contacting each other.

whether the new pairwise residual times (which determine holding times  $J_n$ ) allow us to maintain our Markov model and proceed with the analysis.

#### (Non-)Exponentiality of Intercontact Times.

As shown in Fig. 4.14, the holding times  $J_n$  of our superposition process  $M_n = \{T_n, \sigma_n\}$  are, in fact, minima of  $E - 1$  residual times and one holding time of the original source processes  $\left(T_m^{(ij)}\right)_{m \in \mathbb{Z}}$ , representing *pairwise* intercontacts. Thus, in order to say something about  $J_n$ , we must thoroughly understand the probability distributions of  $J_m^{(ij)}$ .

Cai et al. showed in [178] that, under relatively generic conditions, the *pairwise* intercontact times have a probability distribution, which is a mixture between a power-law and an exponential (i.e., power-law head and exponential tail). This has also been confirmed in real-world traces by Karagiannis et al. [28], who analyzed the empirical complementary cumulative distribution functions (CCDFs) of the intercontacts. Assuming such a distribution for the pairwise intercontacts  $J_m^{(ij)}$ , we will show that their residuals converge to exponential distributions. Based on these findings, we can approximate the random variables  $J_n$  as exponentials, which makes the contact process a suitable substrate for Markov models.

**Definition 4.8.2 (Contact process).** Let the CCDF of the random variables  $J_m^{(ij)}$  be:

$$\overline{F}_{ij}(x) = \begin{cases} C_{ij} \cdot x^{-\alpha_{ij}} e^{-\beta_{ij}x}, & \text{for } x \geq t_0^{(ij)}, \\ 1, & \text{for } 0 < x < t_0^{(ij)}, \end{cases} \quad (4.14)$$

where  $t_0^{(ij)}$  is the minimum intercontact time for the  $(i, j)$  node pair and  $C_{ij} = \left(t_0^{(ij)}\right)^{\alpha_{ij}} e^{\beta_{ij}t_0^{(ij)}}$  is a positive normalization constant. The above function is a combination of a Pareto distribution and an exponential distribution.

For each and every node pair  $(i, j)$ , there uniquely corresponds to any fixed time instant  $t$  exactly one index  $m$ , such that  $T_{m-1}^{(ij)} \leq t < T_m^{(ij)}$ . Then, the **residual time** on  $(i, j)$ 's contact process is the time  $K^{(ij)}(t)$  (random variable), from  $t$  to the next renewal epoch (next  $(i, j)$  contact):

$$K^{(ij)}(t) = T_m^{(ij)} - t. \quad (4.15)$$

Then, applying our earlier observation that aggregate intercontact times are, in fact, minima of  $E - 1$  residuals of the pairwise intercontact times and one intercontact time, the aggregate intercontacts from Fig. 4.14 can be expressed as:

$$J_{n-3} = T_{n-3} - T_{n-4} = \min_{(i,j) \neq (1,2)} \left( K^{(ij)} \left( T_{m-1}^{(1,2)} \right), J_m^{(1,2)} \right), \quad (4.16)$$

$$J_{n-2} = T_{n-2} - T_{n-3} = \min_{(i,j) \neq (1,3)} \left( K^{(ij)} \left( T_{m-1}^{(1,3)} \right), J_m^{(1,3)} \right), \quad (4.17)$$

and so on for the following ones. Note that, within the aggregate contact process  $(M_n)_{n \in \mathbb{Z}}$ , the residual times  $K^{(ij)}(t)$  are correlated, as a result of the superposition. However, it has been shown [179] that approximating a superposition of independent renewal processes as a renewal process yields accurate results. This amounts to approximating residual times  $K^{(ij)}(t)$  as independent, and the aggregate intercontact times above as minima of independent random variables.

With Definition 4.8.2 as a starting point, we will show, in the following, that the residual times  $K^{(ij)}(t)$  converge to the exponential distribution of rate  $\beta_{ij}$ . A validation of this finding on our real and synthetic traces is also presented in Section 4.10.2, when we introduce and describe each trace.

The mean residual lifetime (MRL) of a non-negative random variable  $X$  with CDF denoted  $F$  is defined as:

$$\text{MRL}(t) = \mathbb{E}[X - t \mid X > t] = \frac{\int_t^\infty \overline{F}(x) dx}{\overline{F}(t)}. \quad (4.18)$$

Meilijson showed in [180] that if  $\text{MRL}(t)$  converges to a constant  $A$  as  $t \rightarrow \infty$ , then the conditional distribution of  $(X - t)$  given that  $X > t$  (which is none other than the residual of  $X$ ) converges to the exponential distribution of rate  $A^{-1}$ .

**Theorem 8.1 (Residual Convergence to  $\text{Exp}(\beta_{ij})$ ).** The pairwise residual intercontact times  $K^{(ij)}(t)$  for a superposition of heterogeneous pairwise contact processes

obeying Definition 4.8.2, converge to the exponential distribution of rate  $\beta_{ij}$ , as the pairwise contact processes approach equilibrium<sup>12</sup> ( $t \rightarrow \infty$ ).

*Proof.* In our case, the MRL is the expectation of the pairwise residual intercontact times  $K^{(ij)}(t)$ . Using Eq. (4.14) for  $x \geq t_0^{(ij)}$ , we obtain the following MRL for the pairwise intercontact time variables  $J_m^{(ij)}$ :

$$\text{MRL}(t) = \mathbb{E}[K^{(ij)}(t)] = \frac{1}{\beta_{ij}} \cdot \frac{\Gamma(1 - \alpha_{ij}, \beta_{ij}t)}{(\beta_{ij}t)^{-\alpha_{ij}} e^{-\beta_{ij}t}}. \quad (4.19)$$

The second fraction in the above equation is known to converge to 1 as  $t \rightarrow \infty$ . Therefore,

$$\lim_{t \rightarrow \infty} \text{MRL}(t) = \frac{1}{\beta_{ij}}, \quad (4.20)$$

which means that the residual time  $K^{(ij)}(t)$  converges to the exponential distribution of rate  $\beta_{ij}$ , as the renewal process reaches equilibrium ( $t \rightarrow \infty$ ). This implies that, if our pairwise mobility process  $(M_m^{(ij)})_{m \in \mathbb{Z}}$  has been going on for a very long time ( $t \rightarrow \infty$ ) before we start observing it, the residual time of the  $(i, j)$  intercontacts will be exponential with rate  $\beta_{ij}$ .  $\square$

Considering Eqs. (4.16–4.17) and recalling that the minimum of independent exponentials is itself exponential, we can approximate the distribution of aggregate intercontact times  $(J_n)_{n \in \mathbb{Z}}$  by  $\text{Exp}(\sum \beta_{ij})$ .

#### 4.8.5 A Markov Chain Model for Distributed Optimization

Therefore, the pairwise contact probabilities  $p_c^{ij}$ , that we used in the definition of our Markov chain  $\mathbf{P} = \{p_{xy}\}$  from Eq. (4.13) can now be defined as:

$$p_{ij}^c = \frac{\beta_{ij}}{\sum_{1 \leq i < j \leq N} \beta_{ij}}, \quad (4.21)$$

and  $\mathbf{P} = \{p_{xy}\}$  defines indeed a discrete-time homogeneous Markov chain, as we have initially claimed.

To analyze this chain in the rest of the paper, we use Eq. (4.14) and maximum likelihood estimation (MLE) to estimate the pairwise intercontact time parameters  $\alpha_{ij}$  and  $\beta_{ij}$  for all  $(i, j)$  pairs. The resulting  $N \times N$  **contact probability matrix**  $\mathbf{P}^c = \{p_{ij}^c\}$  entirely describes any given mobility scenario with **heterogeneous node mobility**. Note that, since we are using the embedded Markov chain  $(X_n)_{\mathbb{N}_0}$ , all the time related quantities that we calculate will be expressed in *event time* (measured in number of contact events or “contact ticks”) as opposed to *standard time* (wall-clock time). Because our contact process is stationary and ergodic, it is easy to revert from event time to wall time, using Wald’s equation [182].

---

<sup>12</sup>An equilibrium renewal process, in the sense of [181] (page 28), is an ordinary renewal process started in the remote past at  $t_s \rightarrow -\infty$ , with respect to  $t_o = 0$ , when the observation of the process starts.

#### 4.8.6 DTN-Meteo’s Realism-Tractability Trade Off

DTN-Meteo, our analytical framework described in the above sections and summarized in Fig. 4.13, relies on a set of assumptions about the characteristics of node encounters (tail exponentiality, independence, stationarity), aiming at a good trade off between realism and tractability. Below we discuss whether/how each assumption could be relaxed or even lifted and how that would affect DTN-Meteo’s tractability.

**Tail Exponentiality.** As shown in Section 4.8.4, our framework requires that pairwise intercontact times obey an exponentially-tailed distribution (that is, either fully exponential or power law with exponential cutoff). While these two cases seem to well cover many of the related trace studies and findings, some analyses suggest that a subset of pairwise intercontact times may have a heavier tail (most often Pareto). However, attempts at obtaining even just averages of DTN performance metrics in the presence of heavy-tailed intercontact times have only yielded asymptotic results related to finite vs. infinite delay time (e.g. [183]).

**Space/Time Heterogeneity.** Realistic node mobility may feature spatial preference (nodes visiting some locations more than others) as well as time variations (e.g. nodes meeting predominantly at certain times of day). Spatial preference translates, in the contact process, to higher meeting rates among nodes sharing a preferred location; such scenarios can already be captured by DTN-Meteo through its use of heterogeneous (arbitrary) contact rates.

Time-dependent meeting rates can be more challenging to account for. Such *non-stationarity* effects are often observed as time-of-day (e.g. diurnal) periodicity, and could sometimes be captured, with a different contact matrix for each period or time window (as in [?], Section V.B). Depending on the time-scale of interest for the targeted algorithm, it might suffice to choose the appropriate contact matrix, according to the algorithm’s start time. Algorithm runtimes spanning multiple periods require time-dependent Markov Chains, increasing complexity beyond solution. We keep it for future work to investigate if any useful approximations to such time-dependency problems are possible.

**Correlation.** Def. 4.8.1 implies that every node pair’s contacts are mutually *independent* from the contacts of all other node pairs. This is a step forward from the usual assumption that every node moves independently from all other nodes. When all nodes move mutually independently, all contacts happen purely by chance. With our approximation, if nodes  $i$  and  $j$  come in contact, their two mobilities may be correlated (i.e. they may meet on purpose, as in [40]); however, their contact will happen independently of any other node pairs’ mobility, including node pairs involving one of  $i$  or  $j$ .

Further correlation in node mobility and/or node contacts may refer to a number of aspects of the mobility/contact process: almost every paper on the subject of such correlation has a different definition for it. For example:

- [184] loosely defines correlation as the space/time heterogeneity discussed above. This work does not perform any analytical work, but only proposes a protocol to exploit such correlation.
- [185] uses a model of correlated node mobility, in which groups of nodes confined to a disc area move together as a whole, with each group moving uniformly,

identically and independently of other groups. Based on this model, the authors derive asymptotic scaling laws for delay and capacity in MANETs.

- [186] models epidemic spreading using a colouring process and only requires independence among the consecutive times for the epidemic to spread to one additional node. It is unclear what kind of correlation this allows at the level of node mobility or even at the level of (pairwise) contact processes. Moreover, the analysis does not apply to DTN protocols that actually exploit correlation and heterogeneity, thus limiting its practical relevance.
- [187] expresses correlation in the contact process as the conditional intermeeting time between two nodes, relative to meeting with a third node (note that higher order correlations are likely to be present in reality). Similarly to the first example, this work does not propose any analytical evaluation, but only modifies existing DTN protocols to account for such correlations.

All in all, these examples show that: (i) while everyone agrees that there exist correlations in node mobility/contacts, the nature of these correlations is not yet clearly understood and defined; (ii) almost all types of correlation severely limit the usefulness of analytical derivations, by restricting them to simple protocols such as epidemic spreading and/or by only allowing for asymptotic results.

The current version of DTN-Meteo supports spatial heterogeneity and can be extended to also support time heterogeneity, as discussed above. DTN-Meteo could also be extended to support, for instance, conditional intermeeting times, as in the last example. This would essentially imply introducing additional “memory” in our Markov chain (similarly to turning a normal Markov chain into an order-2 Markov Chain or higher). Besides the relatively limited increase in realism, such a step explodes the state space from  $N$  (number of nodes) for the original chain to  $N^2$  for the respective order-2 chain. Hence, we would have a similar impact on the state space of our model, making it prohibitive to provide a prediction for anything usefully large. We thus choose to not include this aspect in the model, and instead put our model’s prediction up against real traces, where such effects might be present.

In conclusion, we believe DTN-Meteo strikes a good balance between realism and analytical tractability. Improving the realism of analytical models without compromising their usefulness is a central theme of our work. An analytical model that includes all possible human and vehicular features, in all contexts covered by existing traces, does not exist, and even if it did, it would probably be of little use. At the same time, we expect that DTN-Meteo might fail in some scenarios. In Section 4.10.4 we further investigate when and how a given setting could lead to prediction accuracies.

## 4.9 Analyzing DTN Optimization Algorithms

In this section, we show some examples of how to use the DTN-Meteo model to quantify the expected performance of an algorithm – Step (5) of the DTN-Meteo recipe from Fig. 4.13. DTN-Meteo can also be used to analyze worst-case performance and correctness; we performed a detailed investigation of these properties using DTN-Meteo in [188], so we focus here on average performance. Specifically, we will: (i) describe how to calculate the expected performance of greedy algorithms for two DTN prob-

lems (routing and content placement), from standard Markov chain analysis; and (ii) present an example of randomized local search algorithm (based on a Markov Chain Monte Carlo (MCMC) method) applied to content placement and derive its expected performance. Table 4.7 summarizes this section’s notation.

To clarify, a “greedy” algorithm will *always* move to a better solution, if one is available. The majority of existing DTN utility-based forwarding algorithms are in fact greedy, always give a message replica to a “better” (according to the chosen utility) relay. As we will show, such protocols might get stuck into local maxima of the problem in hand, the existence of which depends on the interplay between the nodes’ mobility properties and the selected utility function. Such local maxima might materialize, example, as the message not getting delivered on time (for end-to-end message delivery), or a highly suboptimal choice of content storers (for content placement problems). We will start our analysis with such greedy algorithms, in the next section. However, to circumvent the problem of such local maxima, we will show later how our framework can be used to modify such algorithms using the correct amount of randomization to ensure convergence to the optimal solution. We stress here that both the existence of such local maxima (and related protocol troubles) as well as the methodology to cope with them is a *a key contribution of our proposed DTN-Meteo framework*.

#### 4.9.1 DTN-Meteo for Greedy Algorithms

We will focus here on greedy algorithms for two key problems, routing and content placement. We use the theory of absorbing Markov chains [189], to obtain absorption probabilities and delays for all pairs: (transient state → absorbing state). From these, we derive crucial performance metrics for our algorithms, such as *delivery ratio* and *delivery delay* in routing and *probability and delay* in the context of optimally placing content.

For each of the described DTN problems, there exist(s) some state(s) corresponding to the *best* possible solution globally (e.g. best allocation of content replicas, or, for routing: any state in which the destination holds a message copy). This state will have *maximum utility globally*. By definition, a greedy algorithm only moves to better states. Hence, if and when the best state is reached, there will never be any other transition in the respective Markov Chain for the problem (since there are no better states). Formally, we say that the chain gets *absorbed*.

In addition to states with maximum utility globally, there may also be network states of lower utility, but from which it is impossible to greedily advance to higher utility: as a very simple example, the mobility of nodes may be such, that the nodes currently holding a content replica never directly meet the better nodes. This type of lower utility states might or might not exist, depending on the mobility scenario and utility function. When such a state is reached, then further transitions are, once again, no longer possible. Thus, these states are also *absorbing* states in  $\mathbf{P}$ , and we call them *local maxima*. We use the following notation for the two sets of states:

$$\begin{aligned} \mathcal{GM} &= \{\mathbf{x}^* \in \Omega \mid U(\mathbf{x}^*) \geq U(\mathbf{y}), \forall \mathbf{y} \in \Omega\} && \text{(global maxima),} \\ \mathcal{LM} &= \{\mathbf{x} \in \Omega \setminus \mathcal{GM} \mid p_{\mathbf{xy}} = 0, \forall \mathbf{y} \in \Omega\} && \text{(local maxima).} \end{aligned}$$

$\mathcal{N}, N$	the network (set of nodes), resp. its size
$L$	maximum allowed replication (number of copies)
$\Omega$	network state space (set of binary vectors $\mathbf{x}$ )
$\mathbf{x}^*$	optimum network state(s) (element(s) of $\Omega$ )
$\mathbf{P}$	Markov chain under analysis (transition matrix)
$p_{\mathbf{x}}^{(0)}$	initial probability for state $\mathbf{x}$
$p_d, T_d$	convergence probability, resp. delay
$p_d(\mathbf{x}^*)$	prob. of convergence to one of several opt. states $\mathbf{x}^*$
$T_d(\mathbf{x}^*)$	delay of convergence to one of several opt. states $\mathbf{x}^*$
$\mathbf{Q}$	transient part of $\mathbf{P}$ (greedy algo's)
$\mathbf{R}_1, \mathbf{R}_2$	local/global opt. absorbing parts of $\mathbf{P}$ (greedy algo's)
$\mathbf{N} = \{n_{\mathbf{xy}}\}$	fundamental matrix of $\mathbf{P}$ (greedy algo's)
$b_{\mathbf{xx}^*}, \tau_{\mathbf{xx}^*}$	prob., resp. delay of convergence from $\mathbf{x}$ to $\mathbf{x}^*$
$\mathbf{Z} = \{z_{\mathbf{xy}}\}$	fundamental matrix of $\mathbf{P}$ (randomized algo's)
$\pi$	stationary distribution of $\mathbf{P}$ (randomized algo's)
$m_{\mathbf{xy}}$	mean first passage time from $\mathbf{x}$ to $\mathbf{y}$ (rand. algo's)

Table 4.7: Important notation in Section 4.9.

For example, in Spray and Wait routing, global maxima states  $\mathbf{x}^*$  are the states in which one of the  $L$  copies is at the destination node  $d$ . Local maxima states  $\mathcal{LM}$  arise when the probabilities  $p_{\mathbf{xy}}$  for transitions leaving the  $\mathcal{LM}$  state  $\mathbf{x}$  is zero, either because:

- $\delta(\mathbf{x}, \mathbf{y}) > 2$  – the states are not adjacent, or because
- $p_{ij}^c = 0$  – the required nodes never meet, or because
- $U(\mathbf{y}) < U(\mathbf{x})$  – neighboring states  $\mathbf{y}$  have lower utilities.

Every other solution in  $\Omega \setminus \mathcal{GM}$  is a *transient* state. Denote by  $\text{TR} \subset \Omega$ , the set of transient states. Then,  $\Omega = \mathcal{GM} \cup \mathcal{LM} \cup \text{TR}$ .

In order to derive absorption related quantities, we write the matrix  $\mathbf{P}$  in *canonical form*, where states are re-arranged such that transient states ( $\text{TR}$ ) come first, followed by absorbing states corresponding to local maxima ( $\mathcal{LM}$ ), followed by maximum utility states  $\mathcal{GM}$ :

$$\mathbf{P} = \begin{pmatrix} \text{TR} & \mathcal{LM} & \mathcal{GM} \\ \mathbf{Q} & \mathbf{R}_1 & \mathbf{R}_2 \\ \mathbf{0} & \mathbf{I}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_2 \end{pmatrix} \quad \begin{matrix} \text{TR} \\ \mathcal{LM} \\ \mathcal{GM} \end{matrix} \quad (4.22)$$

Let  $|\mathcal{GM}| = r_2$ ,  $|\mathcal{LM}| = r_1$  and  $|\text{TR}| = t$ . That is, there are  $r_2$  optimum states,  $r_1$  local maxima and  $t$  transient states. Then,  $\mathbf{I}_1$  and  $\mathbf{I}_2$  are, respectively, the  $r_1 \times r_1$  and the  $r_2 \times r_2$  identity matrices,  $\mathbf{Q}$  is a  $t \times t$  matrix, and  $\mathbf{R}_1$  and  $\mathbf{R}_2$  are, respectively, non-zero  $t \times r_1$  and  $t \times r_2$  matrices.

We can now define the *fundamental matrix*  $\mathbf{N}$  for the absorbing Markov chain as

follows:

$$\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1} = \mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \dots \quad (4.23)$$

The last equality is easy to derive (see [189], page 45).  $\mathbf{N}$  is a  $t \times t$  matrix whose entry  $n_{\mathbf{xy}}$  is the expected number of times the chain is in state  $\mathbf{y}$ , starting from state  $\mathbf{x}$ , before getting absorbed. Thus, the sum of all elements of line  $k$  of the fundamental matrix of an absorbing Markov chain is the expected number of steps until absorption, when starting from state  $k$ .

Finally, for the derivation of absorption quantities, we also need the **initial probability distribution**,  $p_{\mathbf{x}}^{(0)}$ , of the DTN-Meteo chain  $(\mathbf{X}_n)_{n \in \mathbb{N}_0}$ . For illustrative purposes, let us assume that all  $N$  nodes are equally likely content/message sources. However, we cannot directly use this as the initial probability distribution of our chain, since the model does not include the algorithms' replication phase (as explained in Section 4.8.1). Therefore, we derive the initial probability distribution for all states  $\mathbf{x} \in \Omega$  (with exactly  $L$  copies at  $L$  different nodes) as:

$$p_{\mathbf{x}}^{(0)} = \frac{1}{N} \sum_{i \mid x_i=1} \mathbb{P}[\mathbf{X}_0 = \mathbf{x} \mid \text{source is } i]. \quad (4.24)$$

The conditional probability above may be hard to calculate depending on the initial replication strategy. For the sake of tractability, we assume that simple source spraying [175] is used and that the spreading completes before the forwarding algorithm starts. We defer the treatment of more sophisticated initial spreading conditions to future work. Then:

$$p_{\mathbf{x}}^{(0)} = \frac{1}{N} \sum_{i \mid x_i=1} \frac{\prod_{j=1}^N x_j p_{ij}^c}{\sum_{\mathbf{y} \mid y_i=1} \left( \prod_{j=1}^N y_j p_{ij}^c \right)}. \quad (4.25)$$

**From Absorption Analysis to Practical Metrics.** Based on the fundamental matrix and the initial probability distribution, we can now easily derive the metrics of interest for any algorithm of our class. In the following theorems, we show how to do this for our example problems: routing and content placement. However, the theorems apply to any other problem unchanged, as long as the state space and utility are defined.

**Theorem 9.1 (Success Probability).** Routing: *The end-to-end delivery probability for a greedy routing algorithm modeled by chain  $\mathbf{P}$  starting from any initial source(s) with equal probability, is*

$$p_d = \sum_{\mathbf{x}^* \in \mathcal{GM}} p_d(\mathbf{x}^*) = \sum_{\mathbf{x}^* \in \mathcal{GM}} \left( \sum_{\mathbf{x} \in TR} p_{\mathbf{x}}^{(0)} b_{\mathbf{xx}^*} \right), \quad (4.26)$$

where  $b_{\mathbf{xx}^*}$  is the probability of being absorbed at state  $\mathbf{x}^*$ , given we start at  $\mathbf{x}$  and  $p_{\mathbf{x}}^{(0)}$  is the probability of starting at  $\mathbf{x}$ .

Content Placement: *The success probability of greedy content placement finding the best set of  $L$  relays, starting from any initial source(s) with equal probability obeys the same relation.*

*Proof.* Using first step analysis, it is relatively straightforward to obtain individual absorption probabilities  $b_{\mathbf{x}\mathbf{x}^*}$  in matrix form, as  $\mathbf{B}^* = \mathbf{N}\mathbf{R}_2$ , where  $\mathbf{B}^* = \{b_{\mathbf{x}\mathbf{x}^*}\}$  is a  $t \times r_2$  matrix. We refer the interested reader to [182], for a comprehensive proof of this theorem.  $\square$

In addition to knowing what chances a greedy algorithm has of finding an optimal solution, we are also interested in how long it will take. In the following theorem, we derive the expected end-to-end delivery delay of routing and the convergence delay of content placement using the fundamental matrix  $\mathbf{N}$  and the individual delivery ratios/success probabilities  $p_d(\mathbf{x}^*) = \sum_{\mathbf{x} \in TR} p_{\mathbf{x}}^{(0)} b_{\mathbf{x}\mathbf{x}^*}$  defined in Eq. (4.26) above.

**Theorem 9.2 (Expected Delay).** Routing: *The expected end-to-end delivery delay for a greedy routing algorithm modeled by chain  $\mathbf{P}$ , starting from any source with equal probability, given that it does not get absorbed in any local maximum is:*

$$\mathbb{E}[T_d] = \sum_{\mathbf{x}^* \in \mathcal{GM}} \frac{p_d(\mathbf{x}^*)}{p_d} \mathbb{E}[T_d(\mathbf{x}^*)] = \sum_{\mathbf{x}^* \in \mathcal{GM}} \frac{p_d(\mathbf{x}^*)}{p_d} \left( \sum_{\mathbf{x} \in TR} p_{\mathbf{x}}^{(0)} \tau_{\mathbf{x}\mathbf{x}^*} \right), \quad (4.27)$$

where  $\tau_{\mathbf{x}\mathbf{x}^*}$  is the (discrete) delay of being absorbed at state  $\mathbf{x}^*$ , given we start at  $\mathbf{x}$ , and  $p_{\mathbf{x}}^{(0)}$  is the probability of starting at  $\mathbf{x}$ .

Content Placement: *The expected convergence delay for greedy content placement to find the best set of  $L$  relays, starting from any initial source with equal probability obeys the same relation.*

*Proof.* Assume we start in a transient state  $\mathbf{x}$  of our chain  $\mathbf{X}_n$  and compute all *conditional transition probabilities*, given that the process ends up in optimal state  $\mathbf{x}^*$ . Then, we obtain a new absorbing chain  $\mathbf{Y}_n$  with a single absorbing state  $\mathbf{x}^*$ . The transient states are unchanged, except we have new transition probabilities. Then, the vector of absorption delays  $\tau_{\mathbf{x}^*} = \{\tau_{\mathbf{x}\mathbf{x}^*}\}$  is obtained from the fundamental matrix of the new chain, as the matrix' row sums.

This process must be repeated for all  $\mathbf{x}^* \in \mathcal{GM}$ . Then, using the initial probabilities  $p_{\mathbf{x}}^{(0)}$  and the law of total expectation, Eq. (4.27) is obtained. We refer the reader to [182], for a comprehensive proof of this theorem.  $\square$

#### 4.9.2 DTN-Meteo for Randomized Local Search

So far, we have only considered greedy algorithms, for which the acceptance probability  $A_{\mathbf{x}\mathbf{y}}$  from Section 4.8.3 is always 0 or 1: a better solution is always chosen and a worse one is never chosen. This feature makes the Markov chain (Eq. (4.13)) for the problem absorbing and may also create local maxima, where greedy algorithms may get blocked.

To this end, randomized local search algorithms have been proposed. While a randomized local search still deterministically accepts a better (higher utility) solution, it may also move to a lower utility solution with a probability  $A_{\mathbf{x}\mathbf{y}} > 0$ . These probabilities (of moving to lower utility states) are calibrated so as to provably converge to the/an optimal solution. One example of a class of such randomized local search algorithms

are Markov Chain Monte Carlo (MCMC) methods [173]. While MCMC methods are often used to simulate and sample complex (and non-invertible) functions, they also provide a powerful tool for stochastic optimization. The most commonly used MCMC optimization algorithm and also our choice for this example is Metropolis-Hastings.

As transitions between two adjacent states are now possible in both directions, the DTN-Meteo Markov chain (Eq. (4.13)) for an MCMC algorithm is *not absorbing*. Moreover, since we have established in Section 4.8.4 that mobility is stationary and ergodic, the DTN-Meteo chain is, as well, stationary and ergodic. It has a unique stationary distribution, which depends on both the contact probabilities (mobility) and on the acceptance probabilities (algorithm). The latter can be chosen by the algorithm designer.

Since we are interested in finding a/the global maximum solution, we naturally should try to maximize the stationary probabilities of good (high utility) solutions. One example of such a distribution is the Gibbs distribution [173]:

$$\pi(\mathbf{x}) = \frac{\exp(U(\mathbf{x}) \cdot T^{-1})}{\sum_{\mathbf{y} \in \Omega} \exp(U(\mathbf{y}) \cdot T^{-1})}, \quad (4.28)$$

where  $T$  is an algorithm parameter, the temperature. When  $T$  is small, the distribution is concentrated around the large values of  $U(\mathbf{x})$  and thus, the algorithm converges to a “good” solution with high probability.

Using the Metropolis-Hastings acceptance probability formula [173] and the above-defined stationary distribution, we obtain the following:

$$A_{\mathbf{x}\mathbf{y}} = \min \left( 1, \frac{\pi(\mathbf{y})}{\pi(\mathbf{x})} \right) = \min \left( 1, \exp \left( \frac{U(\mathbf{y}) - U(\mathbf{x})}{T} \right) \right). \quad (4.29)$$

A Metropolis-Hastings algorithm using this acceptance probability has been proved to eventually converge to an/the optimum solution [173].

To sum up, assuming that node mobility creates a connected contact graph (i.e. for all nodes  $i$ , there exists at least one node  $j$  such that their meeting probability  $p_{ij}^c > 0$ ) and recalling that our mobility process is ergodic, the Markov chain of our randomized local search algorithm will be irreducible and aperiodic, with  $\pi(\mathbf{x})$  as its unique stationary distribution. Since we have chosen  $\pi$  (see Eq. (4.28)) to be highly concentrated around the states  $\mathbf{x}^* \in \mathcal{GM}$  with the largest utility, the Markov chain, and hence, the algorithm, will eventually converge to those states with high probability ( $\mathbb{P}[\text{convergence}] \rightarrow 1$ ).

**Remark:** Sometimes, the temperature parameter  $T$  is varied *during* the algorithm’s operation, in order to speed up convergence. This is known as *simulated annealing*, when  $T$  starts relatively high and gradually “cools down”. However, it is beyond the scope of our paper, as we do not focus on “tweaking” the algorithm itself.

**Convergence Analysis for Randomized Local Search.** Since the Markov chain is now irreducible, hitting an optimal state does not guarantee that the algorithm stays there forever. Convergence to one of the optimum states  $\mathbf{x}^* \in \mathcal{GM}$  now corresponds to the *first hitting/passage time(s)* to (one of) the respective state(s)  $\mathbf{x}^* \in \mathcal{GM}$  in an ergodic Markov chain, starting from any other state  $\mathbf{y} \in \Omega \setminus \mathcal{GM}$ .

The first passage time may still correspond exactly to a performance metric of interest, for example end-to-end delay in routing (the algorithm terminates as soon as the destination node has been reached). However, in other cases, the first passage time will be a lower bound, for example, for the delay of reaching the optimum content storers in content placement. If the algorithm is designed correctly (according to the remark above), this bound can become tight.

Several methods are available for the analysis of stationary and ergodic Markov chains, from first step analysis to the electric network analogy, and including spectral properties. In the following, we will use an approach similar to the one in Section 4.9.1, based on an equivalent of the *fundamental matrix for ergodic chains*, derived from first step analysis:

$$\mathbf{Z} = (\mathbf{I} - \mathbf{P} + \boldsymbol{\Pi})^{-1}, \quad (4.30)$$

where  $\boldsymbol{\Pi}$  is a matrix with each of its rows being the *stationary probability vector*  $\pi$  (Eq. (4.28)) for transition matrix  $\mathbf{P}$ .

We will derive a lower bound for the expected convergence delay of randomized content placement and the exact expected end-to-end delay of randomized routing, in an equivalent way to Theorem 9.2. We use the new fundamental matrix  $\mathbf{Z}$  and the initial probability vector derived in Eq. (4.25).

Recall from Section 4.9.1 that, for each DTN problem's state space, there may be more than one optimal (maximum utility) state  $\mathbf{x}^*$ . For example, in fixed-replication routing, all of the  $\binom{N-1}{L-1}$  states in which one of the  $L$  copies is at the destination node  $d$  are equally good and will form the set  $\mathcal{GM}$ . This is less likely in our content placement example, where all of the  $L$  relays defining a network state  $\mathbf{x}$  count towards that state's final utility  $U(\mathbf{x})$ .

When several optimal states are present, each of their first passage times must be combined, in order to obtain the final performance metric of the randomized local search algorithm (e.g., end-to-end delivery delay in routing or convergence delay in content placement). Below, we will first show how to derive the first passage time for a unique optimal state (Theorem 9.3) and then how to combine the results to obtain the final performance metrics (Corollary 4).

**Theorem 9.3 (First Passage Time).** *Assuming a unique optimal state  $\mathbf{x}^*$ , a lower bound for the expected convergence delay of a randomized algorithm modeled by our Markov chain  $\mathbf{P}$ , starting from any source with equal probability is:*

$$\mathbb{E}[T_d(\mathbf{x}^*)] \geq \sum_{\mathbf{y} \neq \mathbf{x}^*} p_{\mathbf{y}}^{(0)} \frac{z_{\mathbf{x}^*\mathbf{x}^*} - z_{\mathbf{y}\mathbf{x}^*}}{\pi(\mathbf{x}^*)}, \quad (4.31)$$

where  $\mathbf{Z} = \{z_{\mathbf{xy}}\}$  is the fundamental matrix for ergodic chains and  $|\Omega|$  is the size of the solution space.

*Proof.* In [189] (p. 78), the authors prove that the matrix  $\mathbf{M}$ , containing the mean first passage times from any state  $\mathbf{x} \in \Omega$  to any other state  $\mathbf{y} \in \Omega$  is given by:  $\mathbf{M} = (\mathbf{I} - \mathbf{Z} + \mathbf{EZ}_{dg})\mathbf{D}$ , where  $\mathbf{E}$  is a square matrix with all entries 1,  $\mathbf{Z}_{dg}$  agrees with  $\mathbf{Z}$  on the main diagonal and is 0 elsewhere, and  $\mathbf{D}$  is a diagonal matrix with diagonal elements  $d_{yy} = (\pi(\mathbf{y}))^{-1}$ .

From a given state  $y$  to the unique optimal configuration  $x^*$ , the mean first passage is element  $m_{yx^*}$  of  $\mathbf{M}$ , which from above can be written as:

$$m_{yx^*} = \frac{z_{x^*x^*} - z_{yx^*}}{\pi(x^*)}. \quad (4.32)$$

Using the initial probability distribution in Eq. (4.25), we calculate the weighted average of Eq. (4.32) over all non-optimal states  $y$  to obtain the expected first passage time starting from any source node with equal probability to the unique optimal state  $x^*$  as shown in Eq. (4.31).  $\square$

To address the case when the several optimal states of equal utility values are present, we must also obtain the probabilities for each of those optimum states to be reached first (i.e., before any other optimum state), in addition to the first passage times. Then, using the first passage times calculated as above, we can derive, e.g., the expected end-to-end delay of randomized routing.

The probability for an optimum state  $x^*$  to be reached before any other optimum state can be easily calculated by setting all states  $x^* \in \mathcal{GM}$  as absorbing in our Markov chain of the randomized local search algorithm and using Theorem 9.1 from Section 4.9.1 to calculate each  $p_d(x^*)$ . (Note that, in this case, it will not make sense to add up the probabilities, like in Theorem 9.1. They will simply sum to 1, as there will be no other absorbing states in the chain.)

**Corollary 4 (Expected Delay).** *The exact expected end-to-end delivery delay for a randomized routing algorithm modeled by chain  $\mathbf{P}$ , starting from any source with equal probability, is:*

$$\mathbb{E}[T_d] = \sum_{x^* \in \mathcal{GM}} p_d(x^*) \mathbb{E}[T_d(x^*)] \quad (4.33)$$

where  $\mathbb{E}[T_d(x^*)]$  is given in Eq. (4.31). A lower bound for the expected convergence delay for randomized content placement to find the best set of  $L$  relays, starting from any initial source with equal probability obeys the same relation.

Summarizing Section 4.9, we have shown detailed examples of how to use the DTN-Meteo model and Markovian analysis, to quantify the worst-case and expected performance of two types of algorithms (purely greedy and randomized local search), for two important DTN problems: routing and content placement. We have used generic utility functions, only requiring that the functions be time-invariant and that they can be locally calculated at each node.

In Section 4.10, we validate the accuracy of the results provided by DTN-Meteo against simulation results from both real world traces and synthetic mobility models. We use state-of-the-art routing and content placement algorithms, whereof the utility functions obey our requirements.

## 4.10 Applications to Communication Algorithms

In this section, we apply DTN-Meteo to the state-of-the-art routing algorithms – Sim-Bet [147] and BubbleRap [169], and to content placement, all of them using the first

two replication strategies in Table 4.6: single-copy and fixed budget  $L$ . None of these utility-based algorithms can be modeled by existing tools. We also describe in more detail the practical value of DTN-Meteo through various usage examples, as well as through comparison with simulations.

#### 4.10.1 Utilities for Routing and Content Placement

First, we briefly describe the utility used by each algorithm. These utilities were chosen by the designer of the proposed algorithms for the respective problem, and are not necessarily optimal (different utilities would define different algorithms).

In many recent protocols (including our case studies), a node's utility is assessed using the strength of its mobility ties to other nodes, e.g., based on contact frequency and/or duration etc. These tie strengths are sometimes used as such. However, predominantly, they are aggregated in a single static (“social”) graph, on which node utility can be mapped to metrics from social network analysis, such as centrality and community membership or similarity. This may be a weighted ( $\mathbf{W}$ ) or a binary ( $\mathbf{W}_{\text{bin}}$ ) graph. For our case studies, we use normalized pairwise meeting frequencies as weights  $w_{ij}$ . When necessary, we obtain  $\mathbf{W}_{\text{bin}}$  from  $\mathbf{W}$  by keeping only the highest weights up to the optimal link density [38], as explained earlier in Chapter 4.

#### Content Placement

Recall the goal of content placement: to make popular content (news, software update etc) easily reachable by interested nodes, by pushing  $L$  copies of it from its source to  $L$  “strategic” relays. The accessibility that a relay offers to the rest of the network is related to the expected meeting delay between the relay and any other node. This delay is minimized (and accessibility maximized) by relays who meet the highest number of (unique) nodes per unit time [190]. Using the graph  $\mathbf{W}$ , this number amounts to a node's degree:  $d_i = \sum_{j=1}^N w_{ij}$ , with  $w_{ii} = 0$  by convention. Thus, we define the utility of a network state  $\mathbf{x}$  as:

$$U(\mathbf{x}) = \sum_{i=1}^N x_i \cdot d_i. \quad (4.34)$$

	<b>MIT</b>	<b>INFO</b>	<b>ETH</b>	<b>TVCM</b>
<b>Scale and context</b>	92 campus students & staff	41 conference attendees	20 lab students & staff	24/104 nodes, 2/4 disjoint communities
<b>Period Scanning Interval</b>	9 months 300s (Bluetooth)	3 days 120s (Bluetooth)	5 days 0.5s (Ad Hoc WiFi)	11 days N/A
<b># Contacts total</b>	81 961	22 459	22 968	1 000 000

Table 4.8: Mobility trace characteristics.

### SimBet

is a DTN routing algorithm based on social network analysis. It assesses similarity (number of neighbors in common) to detect nodes that are part of the same community, and (ego) betweenness centrality to identify bridging nodes, that could carry a message from one community to another. We calculate these metrics on the binary graph  $\mathbf{W}_{\text{bin}}$ . Thus, in SimBet the utility of node  $i$ , for a destination node  $d$  is<sup>13</sup>:

$$u_d(i) = \alpha \cdot \text{Sim}_d(i) + \beta \cdot \text{Bet}(i) \quad (4.35)$$

and the utility of a network state is, as above, the sum of individual relay utilities:  $U_d(\mathbf{x}) = \sum_{i=1}^N x_i u_d(i)$ .

SimBet was first published as a single-copy utility-based protocol. It was later enhanced [147] with the option of using a fixed number of copies  $L$ .

### BubbleRap

uses an approach to routing similar to SimBet. Again, betweenness centrality is used to find bridging nodes until the content reaches the destination community. Communities are explicitly identified by a community detection algorithm, instead of implicitly by using similarity. Once in the right community, content is only forwarded to other nodes of that community: a local centrality metric is used to find increasingly better relays within the community. We use  $\mathbf{W}_{\text{bin}}$  to obtain betweenness and apply the Louvain method [166] on the same graph to detect communities. Thus, in BubbleRap the utility of node  $i$ , for a destination node  $d$  is:

$$u_d(i) = \mathcal{I}\{i \in C_d\} \cdot LBet(i) + \overline{\mathcal{I}\{i \in C_d\}} \cdot GBet(i), \quad (4.36)$$

where  $\mathcal{I}\{i \in C_d\}$  is an indicator variable for node  $i$  belonging to the destination's community, and  $LBet(i)$  and  $GBet(i)$  are the local and global centralities, respectively. The utility of a network state is:  $U_d(\mathbf{x}) = \sum_{i=1}^N x_i u_d(i)$ .

Bubble Rap does not originally limit the number of copies, but this is easily accomplished with only insignificant modification of the algorithm.

For all three problems, the respective Markov chain  $\mathbf{P}$  from Eq. (4.13) is now entirely defined<sup>14</sup> and we can apply the convergence analysis from Section 4.9 to both the purely greedy and to the randomized local search versions of each of them. While content placement fundamentally differs from routing (in one problem, node characteristics are sought for, in the other the nodes themselves), our three example problems are suddenly similar: same state space, just different utilities. This is, to a great extent, the merit of our *unified* framework DTN-Meteo, whose declared goal is to exploit the similarities of DTN problems and algorithms. On a practical level, applying the analysis step of DTN-Meteo to each problem still exhibits some particularities: (i) content placement, has a single utility function per network scenario – in routing, we must evaluate a collection of utilities (one per destination  $d$ ) for each network; (ii) content

---

<sup>13</sup>For parameters  $\alpha$  and  $\beta$ , we use the original paper values:  $\alpha = \beta = 0.5$ .

<sup>14</sup>The state space  $\Omega$  is formed by all  $L$ -node subsets of  $\mathcal{N}$ , i.e.,  $|\Omega| = \binom{N}{L}$ , as we ignore the delay of the initial replication phase.

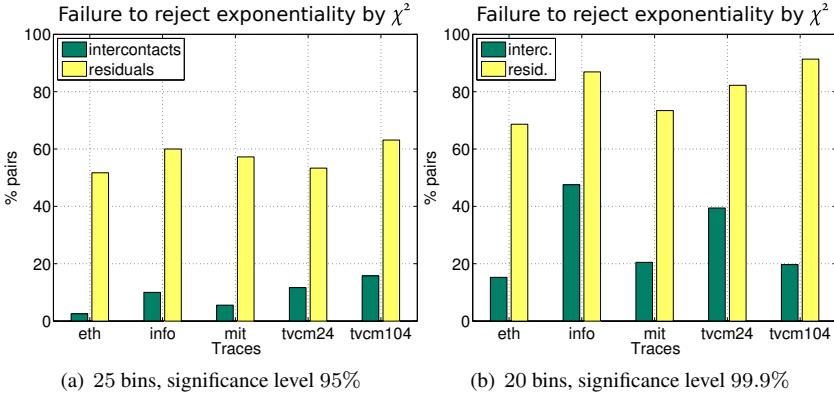


Figure 4.15: Results of the  $\chi^2$  goodness-of-fit test for exponentiality

placement usually has a single optimal state – in routing, for each utility function or destination  $d$ , there are  $\binom{N-1}{L-1}$  optimal states (any subset of  $L$  nodes containing  $d$ ); (iii) local maxima (and thus an algorithm’s behavior) radically change with the utility function, as shown in [182].

#### 4.10.2 Traces and Mobility Models

To cover a broad range of scenarios, we use three real contact traces and two synthetic mobility trace for validation: (i) the *Reality Mining* trace (MIT) [164], (ii) the Infocom 2005 trace (INFO) [191], (iii) the ETH trace [172], and (iv) two synthetic scenarios created with a recent mobility model (TVCM104, TVCM24) [?], which is theoretically tractable and very well-understood.

Their characteristics are summarized in Table 4.8. In addition, we also validate on these traces, the theoretical result we obtained in Section 4.8.4. In that section, we showed that despite the fact that pairwise intercontact times may not always obey the exponential assumption, the pairwise *residual* intercontact times can still be exponential, as long as the pairwise intercontact times have an exponential tail.

To validate this in the above traces, we used MATLAB’s  $\chi^2$  goodness-of-fit test for exponentiality of both pairwise intercontact times and their residuals<sup>15</sup>. Recall that goodness-of-fit tests measure the distance between the empirical CDF (i.e., the one observed in traces) and the CDF of the hypothesized distribution (exponential, in our case). The hypothesis is rejected if the distance is not small enough (this is determined via the significance level). We chose the  $\chi^2$  test as: (i) it makes no strong assumptions about the samples, (ii) it applies even when the parameters of the hypothesized distribution are estimated from the samples, and (iii) it can test for any distribution (discrete or continuous), for which the CDF can be computed. However, the  $\chi^2$  test does require a large enough sample size in order to be valid, therefore we only tested pairs with at

<sup>15</sup>To sample residuals of a given node pair  $(i, j)$ , we determine the times of its first and last contacts in the trace. Between these times, we randomly select geometrically-spaced contacts of any node pair (to ensure Bernoulli arrivals) and measure the time from the selected contact to the next  $(i, j)$  contact. We aim at sampling about the same number of residuals as there are  $(i, j)$  contacts (i.e., about one residual per contact).

least 50 contacts (in all but the INFO trace, this represents more than 70% of all contacts). Finally, the test also requires binning, to be done such that no bin has less than 5 samples. We found that, in most cases, a good number of bins is between 15 and 25 (MATLAB attempts to merge bins with less than 5 values and issues a warning if it fails).

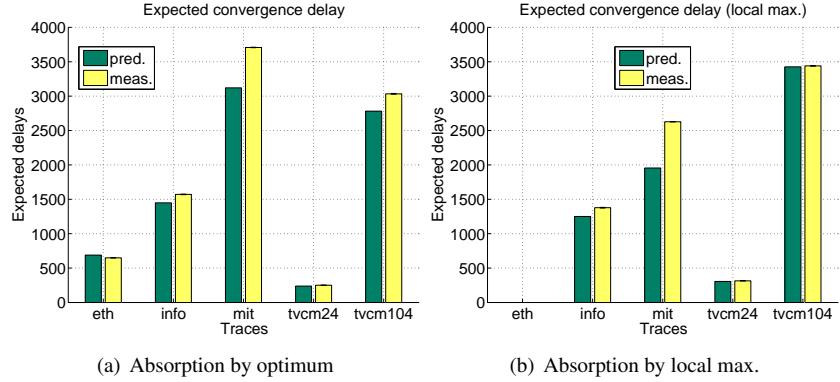


Figure 4.16: Absorption probability validation for **purely greedy** content placement and routing.

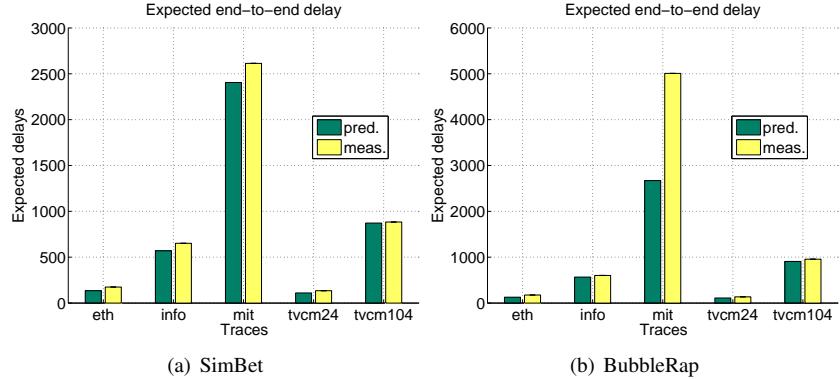


Figure 4.17: Expected delay validation for state-of-the-art routing schemes.

Under these settings, Fig. 4.15 shows that, in all traces, a large percentage of pairwise residual times are exponential, despite the much smaller percentage of corresponding pairwise intercontact times. The results are similar under a variety of significance levels and numbers of bins. This serves to validate our theoretical analysis from Section 4.8.4 in the real world.

### 4.10.3 Measuring the Accuracy of DTN-Meteo

In simulations (trace replays), we measure absorption or convergence quantities – probabilities and average delays – as follows. For every node  $i$  in the network, we set  $i$  as a source of content/messages and do one simulation run. For the two routing algorithms, we do one run per source–destination node pair (using only a subset of randomly chosen destinations, in larger scenarios). In each run, the source node  $i$  generates pieces of content/messages using a Poisson process. This ensures, via the PASTA property [192], that we do not introduce any sampling bias. The content/message generation process produces a sample of at least 1 000 observations per source node/run for shorter traces and up to 15 000 observations per source node/run for longer ones. For all measured delays, we compute the 95<sup>th</sup> percentile using the normal distribution (too small to be visible in most plots).

To obtain the source (and destination) independent metrics, we then average over all source (and destination) nodes.

$L$	Optimum		Best local max.	
	pred.	meas.	pred.	meas.
<b>ETH</b>	5	1.00	1.00	N/A
<b>INFO</b>	3	0.34	0.36	0.18
<b>MIT</b>	2	0.39	0.38	0.42
<b>TVCM24</b>	4	0.10	0.10	0.38
<b>TVCM104</b>	2	0.30	0.28	0.25
				0.26

Table 4.9: Absorption probabilities

Tab. 4.9 shows the measured vs. predicted (Thm. 9.1) success probabilities of greedy content placement. The first two columns give the probability of absorption by the global optimum, the second two – the probability of absorption by a local maximum. In all cases, the prediction is reliably accurate, both with a single absorbing state, the global maximum (in ETH) and when local maxima are present, resulting in multiple absorbing states (in INFO, MIT, TVCM24, TVCM104). The delivery ratios of the greedy routing algorithms show similar accuracy; we omit them due to space limitations.

The first two plots in Fig. 4.17 compare the measured and predicted (Thm. 9.2) values of the convergence delay of greedy content placement, averaged over all initial states ( $L$  values as in Tab. 4.9). The theoretical results coincide once again surprisingly well with the measured delays, both for absorption by the optimum state – Fig. 4.16(a), and for absorption by a local maximum – Fig. 4.16(b). In Fig. 4.16(b), the ETH trace does not have any local maxima with our utility.

Figs. 4.17(a) and 4.17(b) compare the measured and predicted (Thm. 9.2) values of the end-to-end delivery delay of greedy SimBet and BubbleRap routing, averaged over all initial states ( $L$  values as in Tab. 4.9). Again, the theoretical results of both algorithms coincide well with the measured delays, except for BubbleRap routing on the

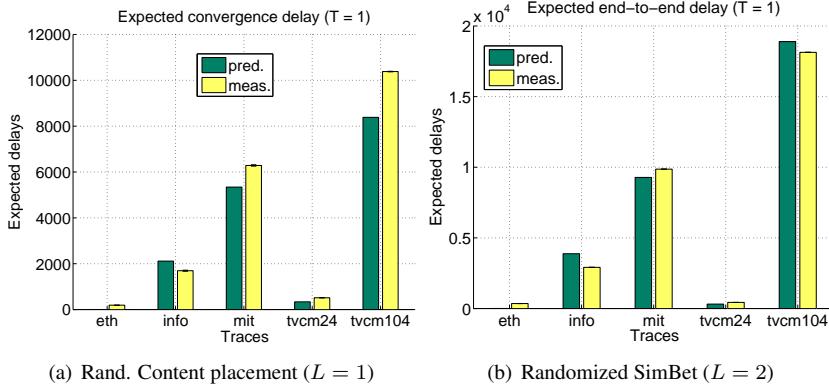


Figure 4.18: Predictions and meas. for **randomized** content placement and routing

MIT trace. We investigate the potential causes of such deviations in the Section 4.10.4.

Finally, in Fig. 4.18, we explore the randomized versions of content placement and SimBet routing. As explained in Section 4.9.2, randomization helps navigate around local maxima, at the cost of convergence time. Figs. 4.18(a) and 4.18(b) show the predicted (Thm. 4) versus measured values for randomized content placement and respectively, randomized SimBet routing. The predictions generally show similar accuracy as their greedy counterparts, with the exception of randomized SimBet routing on the ETH trace and the TVCM104 scenario. This is also examined in the following.

#### 4.10.4 What Influences the Accuracy of DTN-Meteo?

As seen above, in some cases, the prediction of DTN-Meteo may deviate from the results obtained through simulation. In this section, we discuss the factors that lead to such discrepancies. These factors fall generally in two categories: First, the simulation results may be *statistically unreliable*, usually because the underlying trace is not long enough. Second, DTN-Meteo’s operation may be disturbed by breaking one or more of the three basic *assumptions* made in Section 4.8.4 (tail exponentiality, stationarity, independence).

Looking into the first category, simulation results on single trace instances can be deceiving. If the algorithm’s expected time to completion is larger than the trace’s length, simulation results will gravely underestimate that algorithm’s delay: in routing, for example, the delay will be calculated over only the small subset of messages that do get delivered, producing a very biased statistic. This is readily noticeable in the case of randomized algorithms. While these algorithms can provably “escape” local maxima, they might take long to do so, as pointed out in Section 4.9.2. This problem has been explored in detail and illustrated through an example in Fig. 9 from [182]. In contrast, DTN-Meteo can produce a reliable estimate of the expected delay, quickly, even for the scenarios corresponding to such traces, provided there are enough samples to estimate the pairwise contact rates. Furthermore, DTN-Meteo allows one to pick

the right simulation duration when using synthetic simulation models. We consider additional applications of our framework in Section 4.10.5.

Significant departures from the assumptions made by our framework, can create further problems. At the same time, as discussed in Section 4.8.6, dropping one of our three basic assumptions (tail exponentiality, stationarity, independence) can quickly render an already complex framework intractable. Although it is usually hard to isolate each of these factors, synthetic mobility models (e.g. TVCM [?]) have independent node mobility by design and can be made to create stationary scenarios. We can therefore isolate and test the effect of non-exponentiality. In the simplest case of TVCM, nodes exhibit some spatial preference (a common characteristic of human mobility, often responsible for the observed “communities” in a contact graph) as follows: each node has a small, local “home” area and performs random trips inside this area, followed by some roaming trips around the whole network area, then back to doing home trips, and so on, driven by a simple 2-state Markov chain.

Lemma 4.2 of [193] shows that, in order for pairwise intermeeting times in this simple model to have an exponential tail, a *first* requirement is that the nodes’ transmission range must be significantly smaller than the size of their preferred subarea (and, implicitly, than the whole network area, as well). To show the effect of the nodes’ transmission range on DTN-Meteo’s prediction accuracy, we used a simple TVCM scenario with 20 nodes moving within an area of  $200 \times 200$ . Nodes are split in two groups of 12 and, respectively 8 nodes, each with their own preferred subarea. The two subareas are equally sized ( $84 \times 84$ ) and non-overlapping. The strength of the nodes’ spatial preference is controlled by the *home probability*: the probability that the next trip is spent exclusively within the node’s preferred subarea, as opposed to “roaming” over the whole area. Varying the home probabilities and the nodes’ transmission range, Fig. 4.19(a) shows DTN-Meteo’s prediction error relative to simulation results for content placement with  $L = 1$  copy. The error drops significantly at all levels of spatial preference, as the transmission range is reduced and the intercontact times become more exponential.

A second interesting observation is that the error increases as the home probability increases (up to 0.9) and then falls again, when communities become disjoint (i.e. home probability is 1, making nodes move only locally). This is also predicted by Lemma 4.2 in [193], which says that the 2-state chain driving local and roaming trips must have converged between two consecutive contacts of two nodes, for the intermeeting time of these nodes to have an exponential tail. While this convergence is also affected by transmission range (as clearly evident in Fig. 4.19(a)), it results from a more subtle interplay of various spatial factors.

Summarizing, the above example allows us to conclude that: (i) departures from exponentiality in the underlying mobility model are already known to lead to prediction inaccuracies, even for the analytical predictions of expected intercontact times (as can be seen by the prediction errors of [?]); hence, this seems to be the main culprit for DTN-Meteo’s inaccuracies, at least in the case of TVCM; more importantly, DTN-Meteo itself does not lead to additional errors; (ii) in many cases, when it comes to synthetic models, one can tune out such sources of error, or at least predict parameters that would lead to non-conforming scenarios.

The above analysis already provides some hints as to potential problems in real

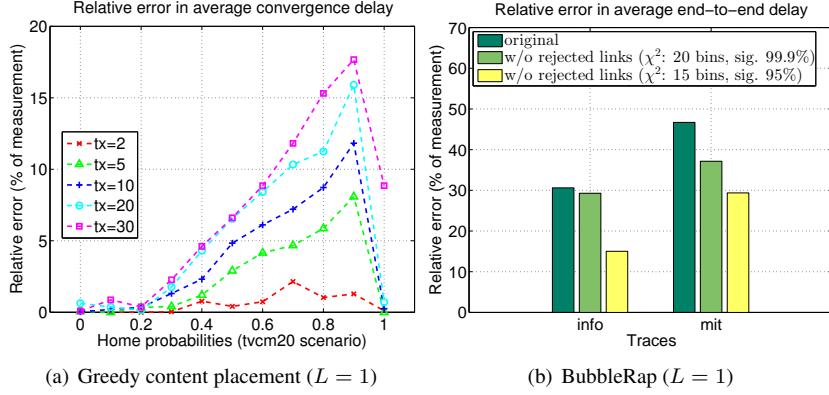


Figure 4.19: Effect of (non-)exponentiality on relative errors of predicted delays

traces: community structure is particularly strong in the MIT trace<sup>16</sup> (compared to ETH and INFO), and it is very likely to originate, at least partially, from spatial preference (even if more complex than described above). Further, there is also evidence of node mobility correlations [40], not present in the TVCM case. Although, traces are not really amenable to a similar “sensitivity” analysis, it may still be possible to get an idea of the effect of (tail) non-exponentiality, by ignoring those links in simulation and analysis, for which the  $\chi^2$  test rejects the exponential hypothesis (see Section 4.10.2). Fig. 4.19(b) shows DTN-Meteo’s prediction error relative to simulation results for BubbleRap with  $L = 1$ , in three variants of the INFO and MIT traces<sup>17</sup>: the original trace, and the trace without links rejected by  $\chi^2$  at significance levels 95%, and respectively 99.9%. Decreasing the significance level makes it easier for the test to reject the exponential hypothesis (i.e., more links to ignore). Thus, Fig 4.19(b) shows that the more non-exponential links we ignore, the better DTN-Meteo’s prediction. In other words, while DTN-Meteo can tolerate a number of non-exponential links, its prediction will slowly worsen, as this number increases.

#### 4.10.5 The Practical Value of DTN-Meteo

Beyond the theoretical aspects of our analysis, we show here how DTN-Meteo may be useful to protocol or system designers. We believe this utility is along three main directions.

**Correctness Analysis:** First, DTN-Meteo offers valuable insight into a protocol’s inner-workings, which simulations alone cannot provide: e.g., a small delivery ratio can be directly linked to the presence of local maxima. This is a crucial issue for *greedy* algorithms in DTNs. Our model can predict whether a given (greedy) algorithm will work for a given mobility scenario; if it does not, some randomization may need

<sup>16</sup>As shown in Table 4.8, the MIT Reality experiment followed people along their daily routine on a larger scale and for a much longer time than the other two traces, making it more likely to observe such structure.

<sup>17</sup>ETH has less nodes, making it harder to perform such operations.

to be introduced (as discussed in Sec. 4.9.2). Moreover, DTN-Meteo allows to identify which properties of mobility or of the utility function are responsible for such convergence or lack thereof. For example, in [182], we treat this issue at large for the case of greedy content placement, and find that this algorithm is only correct under relatively strong requirements on the mobility scenario: for  $L$ -copy greedy content placement, each node should have *at least*  $L$  higher utility neighbor nodes.

**Protocol Tuning:** DTN-Meteo can also be used for tuning key protocol parameters. For example, in Table 4.10 we show how to choose the right number of replicas for SimBet to achieve a desired performance on the ETH scenario. It can further help tune parameters of a utility function (e.g.  $\alpha$  and  $\beta$  in SimBet) or compare two functions, to have as little local maxima as possible and good delays.

Desired Expected Delay <	475	285	190	135
Minimum Required $L$	1	2	3	4

Table 4.10: Minimum  $L$  to achieve expected delay in ETH with SimBet

**Simulation Speedup/Tuning:** Finally, DTN-Meteo has a number of advantages over simulations, when used for the (quantitative) performance evaluation of DTN algorithms. First, as mentioned above, trace-driven simulations often suffer from the rather short time span of the data collection, which makes it hard to achieve statistical significance. This is especially true when evaluating algorithms with (average) completion/convergence times longer than the trace.

Second, even if the trace is long enough, simulations usually require considerably more resources (time, computational power) to achieve the same level of statistical significance as DTN-Meteo. As an example, consider the case of unicast routing. To obtain the average delay from all source nodes to a certain destination  $d$ , DTN-Meteo's largest task consists in inverting a matrix. Depending on the matrix' size (determined by the network size and number of copies allowed) and sparsity, this can require from a few milliseconds to an hour, on a regular laptop computer. In contrast, obtaining the same results through simulations requires the trace to be replayed as many times as there are source nodes, each time generating a large enough number of messages to achieve a good confidence level. In our experience, in function of the trace length, the network size and the number of copies allowed, this may require a computer cluster and may take from half an hour to a day or even more.

## **Part II**

# **Network Optimization in Heterogeneous Wireless Networks**

## **Chapter 5**

# **WiFi Offloading**

So far we have considered offloading mainly through Device-to-Device communication, exploiting the storage resources of mobile nodes and the local bandwidth between nearby devices. Specifically, we have assumed that nodes *opportunistically* (i.e., when they themselves discover a nearby device, and not controlled by an operator as in cellular D2D models) exploit an “out-of-band” channel (e.g., WiFi, 802.11p, Bluetooth) to exchange or relay end-to-end messages or content. Additionally, we have assumed that participating nodes are tolerant to some delays, not requiring the desired files to be delivered or retrieved immediately, but instead can wait (or sometimes *must* wait) until this opportunistic D2D exchange achieves delivery. In this second part of the thesis, we will mostly move away from D2D communication and consider a heterogeneous cellular infrastructure, where the goal is again to offload data traffic but this time from expensive parts/links of the infrastructure (expensive in terms of cost or congestion, etc.) towards cost-efficient ones. In this first chapter of Part II, we will still maintain that aspect of delay tolerance, but consider a HetNet with WiFi and Cellular connectivity.

## 5.1 Introduction

A common cost-effective way to cope with the problem of highly congested mobile networks has been offload some traffic to WiFi networks. In 2016, 60% of the total mobile data traffic was offloaded through WiFi or femto-nodes. Wireless operators have already deployed or bought a large number of WiFi access points (AP) [194].

There are two main types of WiFi offloading. The usual way of offloading is *on-the-spot offloading*: when there is WiFi available, all traffic is sent over the WiFi network; otherwise, *all* traffic is sent over the cellular interface. This is standard practice in today’s industry. More recently, “delayed” offloading has been explored: if there is currently no WiFi availability, (some) traffic can be delayed instead of being sent/received immediately over the cellular interface. In the simplest case, traffic is delayed until WiFi connectivity becomes available. This is already the case with current smartphones, where the user can select to send synchronization or backup traffic (e.g. Dropbox, Google+) only over WiFi. A more interesting case is when the user (or the device on her behalf) can choose a deadline (e.g. per application, per file, etc.). If up to that point no AP is detected, the data are transmitted through the cellular network [195, 51].

As a first contribution on this topic, we analyzed the case of on-the-spot offloading in [49, 50]. We model the transmission of traffic as a queue with multiple transmission states (corresponding to WiFi connectivity, cellular connectivity, or even multiple layers of cellular infrastructure such as macro, pico, femto, etc.), and stochastic transitions between these states that depend on user mobility and network topology. In this chapter we will focus on the case of delay offloading. Delayed offloading offers additional flexibility and promises potential performance gains to both the operator and user. First, more traffic could be offloaded, further decongesting the cellular network. Second, if a user defers the transmission of less delay-sensitive traffic, this could lead to energy savings [196]. Finally, with more operators moving away from flat rate plans towards usage-based plans [197], users have incentives to delay “bulky” traffic to conserve their

plan quotas or to receive better prices [9].

Despite its promise, at the time of our investigation, there was no real consensus as to the added value of delayed offloading. Recent experimental studies largely diverge in their conclusions about the gains of delayed offloading [195, 51]. Additionally, the exact amount of delay a flow can tolerate is expected to depend heavily on (a) the user, and (b) the application type. For example, a study performed in [198] suggests that “more than 50% of the interviewed users would wait up to 10 minutes to stream YouTube videos and 3-5 hours for file downloads”. More importantly, *the amount of patience will also depend on the potential gains for the user*. As a result, two interesting questions arise in the context of delayed offloading:

- *If deadlines are externally defined (e.g. by the user or application), what kind of performance gains for the user/operator should one expect from delayed offloading and what parameters do these depend on?*
- *If an algorithm can choose the deadline(s) to achieve different performance-cost trade offs, how should these deadlines be optimally chosen?*

We will first consider the former question, propose a queueing analytic model for delayed offloading, based on 2D Markov chains, and derive expressions for the average delay, and other performance metrics as a function of the deadlines, and key system parameters. We will also give closed-form approximations for different regimes of interest (*Section 5.2*). Subsequently we will discuss how traffic should be “steered” between the two interfaces, towards achieving different cost-performance tradeoffs.

## 5.2 Analysis of Delayed Offloading

We consider a mobile user that enters and leaves zones with WiFi coverage, with a rate that depends on the user’s mobility (e.g. pedestrian, vehicular) and the environment (e.g. rural, urban). Without loss of generality, we assume that there is always cellular network coverage. We also assume that the user generates flows over time (different sizes, different applications, etc.) that need to be transmitted (uploaded or downloaded) over the network<sup>1</sup>. Whenever there is coverage by some WiFi AP, all traffic will be transmitted through WiFi, assuming for simplicity a First Come First Served (FCFS) queueing discipline. In [54] we show that our results can be applied to Processor Sharing (PS) queueing as well, which often better approximates how a wireless link is shared between concurrent flows. When the WiFi connectivity is lost, we assume that flows waiting in the queue and new flows arriving can be delayed until there is WiFi coverage again. However, each flow has a maximum delay it can wait for (a *deadline*), which might differ between flows and users [198]. If the deadline expires before the flow can be transmitted over some WiFi AP, then it is sent over the cellular network<sup>2</sup>.

---

<sup>1</sup>We will use the terms “flow”, “file”, and “packet” interchangeably throughout the paper, as the most appropriate term often depends on the application and the level at which offloading is implemented.

<sup>2</sup>In practice the switch in connectivity might sometimes occur while some flow is running. Without loss of generality, we will assume that the transmission is resumed from the point it was interrupted when WiFi was lost. It might continue over the cellular network (vertical handover) or paused until WiFi becomes available again or the deadline expires.

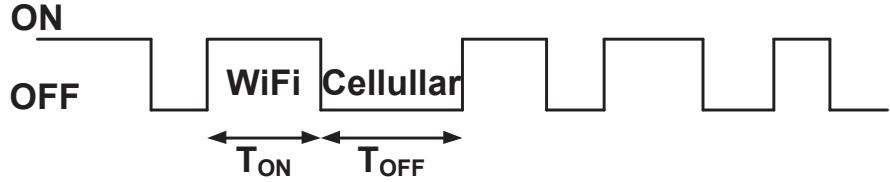


Figure 5.1: The WiFi network availability model modelled as an alternating renewal process. During ON periods there is WiFi connectivity while during OFF periods, only cellular connectivity is available.

We model the WiFi network availability as an ON-OFF alternating renewal process [199]  $(T_{ON}^{(i)}, T_{OFF}^{(i)})$ ,  $i \geq 1$ , as shown in Fig. 5.1. The duration of each ON period (WiFi connectivity),  $T_{ON}^{(i)}$ , is assumed to be an exponentially distributed random variable with rate  $\eta$ , and independent of the duration of other ON or OFF periods. During such ON periods data can be transmitted over the WiFi network with a rate equal to  $\mu$ . Similarly, all OFF periods (Cellular connectivity only) are assumed to be independent and exponentially distributed with rate  $\gamma$ . We further assume that traffic arrives as a Poisson process with rate  $\lambda$ , and file sizes are exponentially distributed. Finally, to capture the fact that each file or flow may have a different deadline assigned to it, we assume that deadlines are also random variables that are exponentially distributed with a mean deadline of  $\frac{1}{\xi}$ .

The above model is flexible enough to describe a large number of interesting settings: high vs. low WiFi availability (by manipulating  $\frac{\gamma}{\gamma+\eta}$ ), low vs. high speed users (low  $\gamma, \eta$  vs. high  $\gamma, \eta$ , respectively), low utilization vs. congested scenarios (via  $\lambda$  and  $\mu$ ), etc. However, the assumptions of exponentiality, while necessary to proceed with any meaningful analysis (as it will be soon made evident), might “hide” the effect of second order statistics (e.g. variability of ON/OFF periods, flow sizes, etc.). To address this, in Section 5.3 we relax most of these assumptions, and validate our results in scenarios with generic ON/OFF periods, generic flow size distributions, and non-exponential deadlines.

Our goal is to analyze this system to answer the following questions: (i) if the deadlines are given (e.g. defined “externally” by the user or application), what is the expected performance as a function of network parameters like WiFi availability statistics, and user traffic load? (ii) if the deadlines are “flexible”, i.e. the user would like to choose these deadlines in order to optimize his overall performance (e.g. trading off some delay to wait for WiFi connectivity, to avoid the often higher energy and monetary cost of cellular transmission), how should they be chosen? Before proceeding, we summarize in Table 5.1 some useful notation. The total time a file spends in the system (queueing+ service time) will be referred to as the *system time* or *transmission delay*.

Table 5.1: Variables and Shorthand Notation.

Variable	Definition/Description
$T_{ON}$	Duration of ON (WiFi) periods
$T_{OFF}$	Duration of periods (OFF) without WiFi connectivity
$\lambda$	Average packet (file) arrival rate at the mobile user
$\pi_{i,c}$	Stationary probability of finding $i$ files in cellular state
$\pi_{i,w}$	Stationary probability of finding $i$ files in WiFi state
$\pi_c$	Probability of finding the system under cellular coverage only
$\pi_w$	Probability of finding the system under WiFi coverage
$p_r$	Probability of reneging
$\eta$	The rate of leaving the WiFi state
$\gamma$	The rate of leaving the cellular state
$\mu$	The service rate while in WiFi state
$\xi$	The reneging rate
$E[S]$	The average service time
$E[T]$	The average system (transmission) time
$T_d$	The deadline time
$\rho = \lambda E[S]$	Average user utilization ratio

### 5.2.1 Performance of WiFi queue

All files arriving to the system are by default sent to the WiFi interface with a deadline assigned (drawn from an exponential distribution). Files are queued if there is another file already in service (i.e. being transmitted) or if there is no WiFi connectivity at the moment, until their deadline expires. If the deadline for a file expires (either while queued or while at the head of the queue, but waiting for WiFi), the file *abandons* the WiFi queue and is transmitted through the cellular network. These kind of systems are known as queueing systems with *impatient* customers [200] or with *reneging* [201]. Throughout our analysis, we'll assume that files will abandon the queue only during periods without WiFi connectivity<sup>3</sup>.

Our main concern is the performance of the WiFi queue, for two reasons: First, this is the place where files accumulate most of the delay. Second, the cellular interface is assumed to be the default option, which has enough capacity to serve all flows at reasonable speed (i.e. a utilization/load of less than 80 – 90%) but is expensive and needs to be avoided (which is why all flows are initially sent to the WiFi interface). At the end of the chapter we briefly discuss situations where either interface can get congested, and how traffic flows should be assigned between interfaces in that case.

Given the previously stated assumptions, the WiFi queue can be modeled with a 2D Markov chain, as shown in Figure 5.2. States with WiFi connectivity are denoted with  $\{i, w\}$ , and states with only cellular connectivity  $\{i, c\}$ .  $i$  corresponds to the number

---

<sup>3</sup>In this manner, abandonments are plausibly associated with the accumulated “opportunity cost”, i.e. the time spent waiting for WiFi connectivity (the “non-standard” option for transmission). Instead, if WiFi is available, but there are some files in front, it might make no sense to abandon, as queueing delays might also occur in the cellular interface.

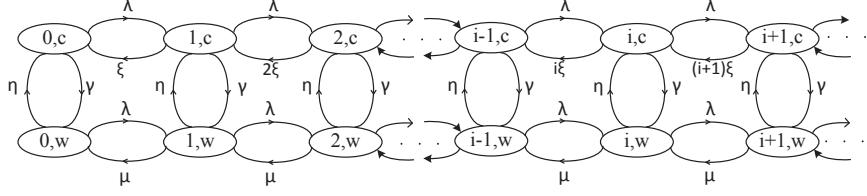


Figure 5.2: The 2D Markov chain for the WiFi queue in delayed offloading.

of customers in the system (service+queue). During WiFi states, the system empties at rate  $\mu$  (since files are transmitted 1-by-1) and during cellular states the system empties at rate  $i \cdot \xi$  since any of the  $i$  queued packets can renege. The following theorem uses probability generating functions (PGF) to derive the mean system time for this queue. The use of PGFs in 2D Markov chains is known for quite a long time [202], [203], [204].

**Theorem 2.1.** *The mean system time for the WiFi queue when delayed mobile data offloading is performed is*

$$E[T] = \frac{1}{\lambda} \left[ \left(1 + \frac{\gamma}{\eta}\right) \frac{\lambda - \mu(\frac{\gamma}{\eta+\gamma} - \pi_{0,w})}{\xi} + \frac{(\lambda - \mu)\frac{\gamma}{\eta+\gamma} + \mu\pi_{0,w}}{\eta} \right], \quad (5.1)$$

where  $\pi_{0,w}$  is the probability of the system being idle during WiFi connectivity, and can be obtained as explained subsequently.

*Proof.* Let  $\pi_{i,c}$  and  $\pi_{i,w}$  denote the stationary probability of finding  $i$  files when there is only cellular network coverage, or WiFi coverage, respectively.

Writing the balance equations for the cellular and WiFi state gives

$$(\lambda + \gamma)\pi_{0,c} = \eta\pi_{0,w} + \xi\pi_{1,c} \quad (5.2)$$

$$(\lambda + \gamma + i\xi)\pi_{i,c} = \eta\pi_{i,w} + (i+1)\xi\pi_{i+1,c} + \lambda\pi_{i-1,c} \quad (5.3)$$

$$(\lambda + \eta)\pi_{0,w} = \gamma\pi_{0,c} + \mu\pi_{1,w} \quad (5.4)$$

$$(\lambda + \eta + \mu)\pi_{i,w} = \gamma\pi_{i,c} + \mu\pi_{i+1,w} + \lambda\pi_{i-1,w} \quad (5.5)$$

The long term probabilities of finding the system in cellular or WiFi state are  $\pi_c = \frac{\eta}{\eta+\gamma}$  and  $\pi_w = \frac{\gamma}{\eta+\gamma}$ , respectively.

We define the PGFs for both the cellular and WiFi states as  $G_c(z) = \sum_{i=0}^{\infty} \pi_{i,c} z^i$ , and  $G_w(z) = \sum_{i=0}^{\infty} \pi_{i,w} z^i$ ,  $|z| \leq 1$ . After multiplying Eq.(5.3) with  $z^i$  and adding to Eq.(5.2), we obtain

$$(\lambda + \gamma)G_c(z) + \xi(1 - \frac{1}{z}) \sum_{i=1}^{\infty} i\pi_{i,c} z^i = \eta G_w(z) + \lambda z G_c(z). \quad (5.6)$$

The summation in the above equation gives  $\sum_{i=1}^{\infty} i\pi_{i,c}z^i = zG'_c(z)$ . Hence, after some rearrangements in Eq.(5.6) we obtain

$$\xi(1-z)G'_c(z) = (\lambda(1-z) + \gamma)G_c(z) - \eta G_w(z). \quad (5.7)$$

Repeating the same procedure for Eq.(5.4)-(5.5) we get

$$(\lambda + \eta)G_w(z) = \gamma G_c(z) + \lambda z G_w(z) + \mu\left(\frac{1}{z} - 1\right)(G_w(z) - \pi_{0,w}),$$

which after some rearrangements yields to

$$((\lambda z - \mu)(1-z) + \eta z)G_w(z) = \gamma z G_c(z) - \mu(1-z)\pi_{0,w}.$$

Next, we make two replacements  $\alpha(z) = \lambda(1-z) + \gamma$ , and  $\beta(z) = (\lambda z - \mu)(1-z) + \eta z$ . Now, we have the system of equations

$$G_w(z) = \frac{\gamma z G_c(z) - \mu(1-z)\pi_{0,w}}{\beta(z)}, \quad (5.8)$$

$$G'_c(z) - \frac{\alpha(z)\beta(z) - \eta\gamma z}{\xi(1-z)\beta(z)}G_c(z) = \frac{\eta\mu\pi_{0,w}}{\xi\beta(z)}. \quad (5.9)$$

The roots of  $\beta(z)$  are

$$z_{1,2} = \frac{\lambda + \mu + \eta \mp \sqrt{(\lambda + \mu + \eta)^2 - 4\lambda\mu}}{2\lambda}. \quad (5.10)$$

It can be shown easily that these roots satisfy the relation  $0 < z_1 < 1 < z_2$ . We introduce the function  $f(z) = -\frac{\alpha(z)\beta(z) - \eta\gamma z}{\xi(1-z)\beta(z)}$ , as the multiplying factor of  $G_c(z)$  in the differential equation of Eq.(5.9). Performing some simple calculus operations, the above function transforms into

$$f(z) = -\frac{\lambda}{\xi} + \frac{\gamma}{\xi(1-z)} \left( \frac{\eta z}{\beta(z)} - 1 \right). \quad (5.11)$$

After some algebra and applying the *partial fraction expansion* the function  $f(z)$  becomes

$$f(z) = -\frac{\lambda}{\xi} + \frac{\gamma}{\xi} \left( \frac{M}{z - z_1} + \frac{N}{z - z_2} \right). \quad (5.12)$$

We determine the coefficients  $M$  and  $N$  in the standard way as  $M = \frac{\frac{\mu}{\lambda} - z}{z_2 - z_1} |_{z=z_1} = \frac{\frac{\mu}{\lambda} - z_1}{z_2 - z_1} = \frac{z_1 z_2 - z_1}{z_2 - z_1} > 0$ , and  $N = \frac{\frac{\mu}{\lambda} - z}{z - z_1} |_{z=z_2} = \frac{\frac{\mu}{\lambda} - z_2}{z_2 - z_1} < 0$ .

In order to solve the differential equation Eq.(5.9) we can multiply it by  $e^{\int f(z)dz}$ . Hence, we get

$$G'_c(z)e^{\int f(z)dz} + f(z)G_c(z)e^{\int f(z)dz} = \frac{\eta\mu\pi_{0,w}}{\xi\beta(z)}e^{\int f(z)dz}. \quad (5.13)$$

We thus need to integrate the function in Eq.(5.15):

$$\int f(z)dz = -\frac{\lambda}{\xi}z + \frac{\gamma M}{\xi} \ln|z - z_1| - \frac{\gamma N}{\xi} \ln(z_2 - z). \quad (5.14)$$

The constant normally needed on the right-hand side of Eq.(5.14) can be ignored in our case. We next raise Eq.(5.14) to the power of  $e$  to get

$$e^{\int f(z)dz} = e^{-\frac{\lambda}{\xi}z} |z - z_1|^{\frac{\gamma M}{\xi}} (z_2 - z)^{-\frac{\gamma N}{\xi}}. \quad (5.15)$$

Now, Eq.(5.13) is equivalent to

$$\frac{d}{dz} \left( e^{-\frac{\lambda}{\xi}z} |z - z_1|^{\frac{\gamma M}{\xi}} (z_2 - z)^{-\frac{\gamma N}{\xi}} G_c(z) \right) = \frac{\eta \mu \pi_{0,w}}{\xi \beta(z)} e^{\int f(z)dz} \quad (5.16)$$

We define  $k_1(z)$  and  $k_2(z)$  as

$$k_1(z) = e^{-\frac{\lambda}{\xi}z} (z_1 - z)^{\frac{\gamma M}{\xi}} (z_2 - z)^{-\frac{\gamma N}{\xi}}, z \leq z_1, \quad (5.17)$$

$$k_2(z) = e^{-\frac{\lambda}{\xi}z} (z - z_1)^{\frac{\gamma M}{\xi}} (z_2 - z)^{-\frac{\gamma N}{\xi}}, z \geq z_1. \quad (5.18)$$

Eq.(5.16) now becomes

$$\frac{d}{dz} (k_1(z)G_c(z)) = \frac{\eta \mu \pi_{0,w}}{\xi \beta(z)} k_1(z), z \leq z_1, \quad (5.19)$$

$$\frac{d}{dz} (k_2(z)G_c(z)) = \frac{\eta \mu \pi_{0,w}}{\xi \beta(z)} k_2(z), z \geq z_1, \quad (5.20)$$

and after integrating we obtain

$$k_1(z)G_c(z) = \frac{\eta \mu \pi_{0,w}}{\xi} \int_0^z \frac{k_1(x)}{\beta(x)} dx + C_1, z \leq z_1 \quad (5.21)$$

$$k_2(z)G_c(z) = \frac{\eta \mu \pi_{0,w}}{\xi} \int_{z_1}^z \frac{k_2(x)}{\beta(x)} dx + C_2, z \geq z_1. \quad (5.22)$$

The bounds of the integrals in Eq.(5.21) and Eq.(5.22) come from the defining region of  $z$  in Eq.(5.19)-(5.20). We need to determine the coefficients  $C_1$  and  $C_2$  in Eq.(5.21) and Eq.(5.22). We take  $z = 0$  in Eq.(5.21). We have  $k_1(0) = z_1^{\frac{\gamma M}{\xi}} z_2^{-\frac{\gamma N}{\xi}}$ , and knowing that  $G_c(0) = \pi_{0,c}$ , we get for  $C_1 = \pi_{0,c} z_1^{\frac{\gamma M}{\xi}} z_2^{-\frac{\gamma N}{\xi}}$ . In a similar fashion we get  $C_2 = 0$ .

Finally, for the PGF in the cellular state we have

$$G_c(z) = \frac{\eta \mu \pi_{0,w} \int_0^z \frac{k_1(x)}{\beta(x)} dx + \xi \pi_{0,c} z_1^{\frac{\gamma M}{\xi}} z_2^{-\frac{\gamma N}{\xi}}}{\xi k_1(z)}, z \leq z_1, \quad (5.23)$$

$$G_c(z) = \frac{\eta \mu \pi_{0,w} \int_{z_1}^z \frac{k_2(x)}{\beta(x)} dx}{\xi k_2(z)}, z \geq z_1. \quad (5.24)$$

In the last equation, the 'zero probabilities'  $\pi_{0,c}$  and  $\pi_{0,w}$  are unknown. We can find them in the following way: We know that  $\pi_c = \frac{\eta}{\eta+\gamma} = G_c(1) = \frac{\eta\mu\pi_{0,w}\int_{z_1}^1 \frac{k_2(x)}{\beta(x)}dx}{\xi k_2(1)}$ . From this we have

$$\frac{\xi k_2(1)}{\eta + \gamma} = \mu\pi_{0,w} \int_{z_1}^1 \frac{k_2(x)}{\beta(x)}dx. \quad (5.25)$$

Similarly, from the boundary conditions in Eq.(5.23) for  $z \leq z_1$ , we get

$$\eta\mu\pi_{0,w} \int_0^{z_1} \frac{k_1(x)}{\beta(x)}dx + \xi\pi_{0,c}z_1^{\frac{\gamma M}{\xi}} z_2^{-\frac{\gamma N}{\xi}} = 0. \quad (5.26)$$

After solving the system of equations Eq.(5.25) and Eq.(5.26), for the 'zero probabilities' we obtain

$$\pi_{0,w} = \frac{\xi k_2(1)}{(\eta + \gamma)\mu} \frac{1}{\int_{z_1}^1 \frac{k_2(x)}{\beta(x)}dx}, \text{ and} \quad (5.27)$$

$$\pi_{0,c} = -\frac{\eta k_2(1) \int_0^{z_1} \frac{k_1(x)}{\beta(x)}dx}{(\eta + \gamma)z_1^{\frac{\gamma M}{\xi}} z_2^{-\frac{\gamma N}{\xi}} \int_{z_1}^1 \frac{k_2(x)}{\beta(x)}dx}. \quad (5.28)$$

The value of the integral  $\int \frac{k_1(x)}{\beta(x)}dx$  is always negative, hence  $\pi_{0,c}$  is always positive.

By using a vertical cut between any two-pairs of neighboring states in Fig. 5.2 and writing balance equations we have

$$\lambda\pi_{i,c} + \lambda\pi_{i,w} = \mu\pi_{i+1,w} + (i+1)\xi\pi_{i+1,c}. \quad (5.29)$$

Summing over all  $i$  yields to

$$\lambda(\pi_c + \pi_w) = \mu(\pi_w - \pi_{0,w}) + \xi \sum_{i=0}^{\infty} (i+1)\pi_{i+1,c}. \quad (5.30)$$

The last equation, obviously reduces to

$$\lambda = \mu(\pi_w - \pi_{0,w}) + \xi E[N_c], \quad (5.31)$$

where  $E[N_c] = G'_c(1)$ , and  $E[N_w] = G'_w(1)$ . Eq.(5.31) yields

$$E[N_c] = \frac{\lambda - \mu(\pi_w - \pi_{0,w})}{\xi}. \quad (5.32)$$

So far, we have derived  $E[N_c]$  as the first derivative at  $z = 1$  of  $G_c(z)$ . In order to find the average number of files in the system, we need  $E[N_w]$  as well. We can get it by differentiating Eq.(5.8):

$$\begin{aligned} G'_w(z) &= \frac{\beta(z) (\gamma G_c(z) + \gamma z G'_c(z) + \mu \pi_{0,w})}{\beta^2(z)} \\ &- \frac{\beta'(z) (\gamma z G_c(z) - \mu(1-z) \pi_{0,w})}{\beta^2(z)}, \end{aligned} \quad (5.33)$$

and setting  $z = 1$ . After some calculus we obtain

$$E[N_w] = \frac{(\gamma E[N_c] + \mu \pi_{0,c}) \eta - \gamma \pi_c (\mu - \lambda)}{\eta^2}. \quad (5.34)$$

Replacing Eq.(5.32) into Eq.(5.34) we get

$$E[N_w] = \frac{\gamma \lambda - \mu (\pi_w - \pi_{0,w})}{\xi} + \frac{\mu \pi_{0,w}}{\eta} - \frac{\gamma \pi_c (\mu - \lambda)}{\eta^2}. \quad (5.35)$$

The average number of files in the system is

$$E[N] = E[N_c] + E[N_w]. \quad (5.36)$$

Finally, using the Little's law  $E[N] = \lambda E[T]$  [199], we obtain the average packet delay in delayed offloading as in Eq.(5.1).  $\square$

The above result gives the total expected delay that incoming flows experience in the WiFi queue. For flows that do get transmitted over WiFi (i.e. whose deadline does not expire) this amounts to their total delay. Flows that end up renegeing (deadline expires before transmission) must be transmitted through the cellular system and thus incur an additional delay  $\Delta$  (related to their transmission time over the cellular link, i.e.  $\frac{\text{packet\_size}}{\text{cellular\_rate}}$ , and possibly some queueing delay as well). The following Corollary gives the probability of renegeing for each.

**Corollary 5.** *The probability that an arbitrary flow arriving to the WiFi queue will renege, i.e. its deadline will expire before it can be transmitted over a WiFi AP is*

$$p_r = \frac{\lambda - \mu(\pi_w - \pi_{0,w})}{\lambda}. \quad (5.37)$$

In other words, the rate of flows sent back to the cellular network is given by  $\lambda \cdot p_r$ . This must be equal to  $\xi \cdot E[N_c]$ , which is the average abandonment rate in Fig. 5.2, i.e.  $\lambda p_r = \xi E[N_c]$ . Replacing  $E[N_c]$  from Eq.(5.32) gives us the above result. This also gives us another important metric, the *offloading efficiency* of our system, namely the percentage of flows that get offloaded over some WiFi network, as  $E_{off} = 1 - p_r$ .

The above expressions can be used to predict the performance of a delayed offloading system, as a function of most parameters of interest, such as WiFi availability and performance, user traffic load, etc. As we shall see later, it does so with remarkable accuracy even in scenarios where many of the assumptions do not hold. However, Eq.(5.1) cannot easily be used to solve optimization problems related to the deadline

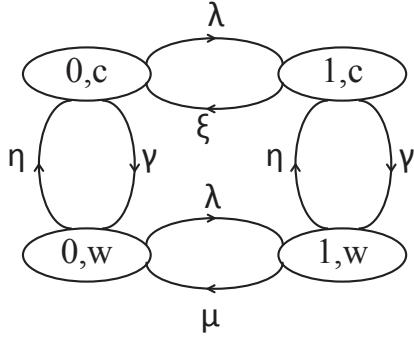


Figure 5.3: The reduced Markov chain for  $\rho \rightarrow 0$ .

( $\xi$ ), analytically, as the parameters  $\pi_{0,c}$  and  $\pi_{0,w}$  involve  $\xi$  in a non-trivial way. To this end, we propose next some closed-form approximations for the low and high utilization regimes.

### 5.2.2 Low utilization approximation

One interesting scenario is when resources are underloaded (e.g. nighttime, rural areas, or mostly low traffic users) and/or traffic is relatively sparse (e.g. background traffic from social and mailing applications, messaging, Machine-to-Machine communication). For very low utilization, the total system time essentially consists of the service time, as there is almost no queueing, so we can use a fraction of the Markov chain from Fig. 5.2 with only 4 states, as shown in Fig. 5.3 to derive  $E[T]$  and  $p_r$ . The system empties at either state  $(0, w)$  if the packet is transmitted while in WiFi connectivity period or state  $(0, c)$ , if the packet spends in queue more than the deadline it was assigned while waiting for WiFi availability.

The goal here is to find the average time until a packet arriving in a WiFi or cellular period finishes its service, i.e. the time until the system, starting from the state  $(1, c)$  or  $(1, w)$  first enters any of the states  $(0, c)$  or  $(0, w)$ . Hence, the average service time is

$$E[S] = \frac{\eta}{\gamma + \eta} E[T_c] + \frac{\gamma}{\gamma + \eta} E[T_w], \quad (5.38)$$

where  $E[T_c]$  ( $E[T_w]$ ) is the average time until a packet that enters service during a cellular (WiFi) network period finishes its transmission. This can occur during a different period. The expression for  $E[T_c]$  is equal to

$$E[T_c] = P[I_c = 1]E[T_c|I_c = 1] + P[I_c = 0]E[T_c|I_c = 0], \quad (5.39)$$

where  $I_c$  is an indicator random variable having value 1 if the first transition from state  $(1, c)$  is to state  $(0, c)$ . This means that the packet is transmitted during the same cellular period. Otherwise, its value is 0. The probabilities of these random variables are

$P[I_c = 1] = \frac{\xi}{\xi+\gamma}$ , and  $P[I_c = 0] = \frac{\gamma}{\xi+\gamma}$ , respectively. For the conditional expectations from Eq.(5.39), we have

$$E[T_c|I_c = 1] = \frac{1}{\xi + \gamma}, \quad (5.40)$$

$$E[T_c|I_c = 0] = \frac{1}{\xi + \gamma} + E[T_w]. \quad (5.41)$$

Eq.(5.40) is actually the expected value of the minimum of two exponentially distributed random variables with rates  $\xi$  and  $\gamma$ . Replacing Eq.(5.40) and (5.41) into Eq.(5.39), we get

$$E[T_c] - \frac{\gamma}{\xi + \gamma} E[T_w] = \frac{1}{\xi + \gamma}. \quad (5.42)$$

Following a similar procedure for  $E[T_w]$  we obtain

$$E[T_w] - \frac{\eta}{\mu + \eta} E[T_c] = \frac{1}{\mu + \eta}. \quad (5.43)$$

After solving the system of equations Eq.(5.42)-(5.43), we have

$$E[T_w] = \frac{\xi + \gamma + \eta}{\xi\mu + \xi\eta + \mu\gamma}, \quad (5.44)$$

$$E[T_c] = \frac{\mu + \gamma + \eta}{\xi\mu + \xi\eta + \mu\gamma}. \quad (5.45)$$

Now, replacing Eq.(5.44)-(5.45) into Eq.(5.38), we have the average service time, and the low utilization approximation is ( $E[T] \approx E[S]$ ):

$$E[T] = \frac{(\eta + \gamma)^2 + \gamma\xi + \eta\mu}{(\xi\mu + \xi\eta + \mu\gamma)(\gamma + \eta)}. \quad (5.46)$$

To find the probability of reneging, we need to know  $\pi_{0,c}$ . We find it solving the local balance equations for Fig. 5.3. After solving the system we get

$$\pi_{0,c} = \frac{\eta}{\eta + \gamma} \frac{\xi(\mu + \lambda + \eta) + \mu\gamma}{\xi(\mu + \lambda + \eta) + \mu\gamma + \lambda^2 + \lambda(\eta + \gamma + \mu)}. \quad (5.47)$$

Replacing Eq.(5.47) and  $\pi_c = \frac{\eta}{\eta + \gamma}$  into Eq.(5.37), we get the probability of reneging for low utilization as

$$p_r = \frac{\theta_1\xi}{\theta_2\xi + \theta_3}, \quad (5.48)$$

where  $\theta_1 = \frac{\eta(\lambda + \eta + \gamma + \mu)}{\eta + \gamma}$ ,  $\theta_2 = \mu + \lambda + \eta$ , and  $\theta_3 = \mu\gamma + \lambda^2 + \lambda(\eta + \gamma + \mu)$ .

Hence, we can write the results for the low utilization regime for both the average delay and probability of reneging as following:

**Low utilization approximation:** The expected system time in the WiFi queue and the probability of reneging for sparse input traffic can be approximated by

$$E[T] = \frac{(\eta + \gamma)^2 + \gamma\xi + \eta\mu}{(\xi\mu + \xi\eta + \mu\gamma)(\gamma + \eta)}, \quad (5.49)$$

$$p_r = \frac{\theta_1\xi}{\theta_2\xi + \theta_3}, \quad (5.50)$$

where  $\theta_1 = \frac{\eta(\lambda+\eta+\gamma+\mu)}{\eta+\gamma}$ ,  $\theta_2 = \mu + \lambda + \eta$ , and  $\theta_3 = \mu\gamma + \lambda^2 + \lambda(\eta + \gamma + \mu)$ .

### 5.2.3 High utilization approximation

Another interesting regime is that of high utilization. As explained earlier, wireless resources are often heavily loaded, especially in urban centers, due to the increasing use of smart phones, tablets, and media-rich applications. Hence, it is of special interest to understand the average user performance in such scenarios. Here, we provide an approximation that corresponds to the region of high utilization ( $\rho \rightarrow 1$ ).

The expected system time in the WiFi queue for a user with heavy traffic can be approximated by

$$E[T] = \frac{1}{\lambda} \left[ \left(1 + \frac{\gamma}{\eta}\right) \frac{\lambda - \mu\pi_w}{\eta} + \frac{(\lambda - \mu)\pi_w}{\eta} \right]. \quad (5.51)$$

The approximation Eq.(5.51) comes from Eq.(5.1) by replacing  $\pi_{0,w} = 0$ .

To find the approximation for the probability of reneging in the high utilization regime we proceed as follows. Since from Eq.(5.37) the only term that depends on  $\xi$  is  $\pi_{0,w}$  (we will need it later to solve optimization problems), we will not take it equal to 0. We will approximate it by a first order Taylor approximation at  $\xi = 1$ . For that purpose, we will denote  $\pi_{0,w}$  as  $\pi_{0,w}(\xi)$ . So, we write

$$\pi_{0,w}(\xi) = \pi_{0,w}(1) + (\xi - 1)\pi'_{0,w}(1), \quad (5.52)$$

where  $\pi_{0,w}(1) = \frac{A(1)}{(\eta+\gamma)\mu \int_{z_1}^1 \frac{A(x)}{\beta(x)} dx}$ , and  $\pi'_{0,w}(1) = \frac{A(1)}{(\eta+\gamma)\mu} \frac{\ln(A(1)e) \int_{z_1}^1 \frac{A(x)}{\beta(x)} dx - \int_{z_1}^1 \frac{A(x) \ln A(x)}{\beta(x)} dx}{\left(\int_{z_1}^1 \frac{A(x)}{\beta(x)} dx\right)^2}$ ,

where  $A(x) = \frac{(x-z_1)^{\gamma M}}{(z_2-x)^{\gamma N}} e^{-\lambda x}$ .

Hence, the probability of reneging in the high utilization regime can be approximated by

$$p_r = \frac{\lambda - \mu\pi_w}{\lambda} + \frac{\mu}{\lambda} \pi_{0,w}, \quad (5.53)$$

where  $\pi_{0,w}$  is given by Eq.(5.52).

Now, we can write jointly the expressions for the average file delay and probability of reneging in a delayed offloading system for high utilization.

**High utilization approximation:** *The expected system time in the WiFi queue and the probability of reneging for a user with heavy traffic can be approximated by*

$$E[T] = \frac{1}{\lambda} \left[ \left(1 + \frac{\gamma}{\eta}\right) \frac{\lambda - \mu\pi_w}{\eta} + \frac{(\lambda - \mu)\pi_w}{\eta} \right], \quad (5.54)$$

$$p_r = \frac{\lambda - \mu\pi_w}{\lambda} + \frac{\mu}{\lambda} \pi_{0,w}, \quad (5.55)$$

where  $\pi_{0,w}$  is the first order Taylor series approximation of Eq.(5.27).

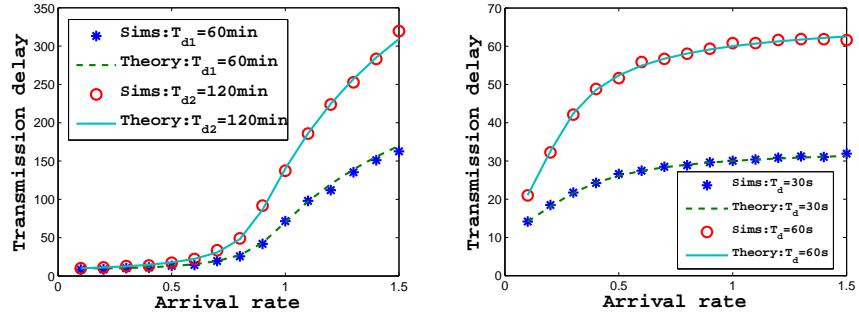


Figure 5.4: Average delay for pedestrian user scenario.

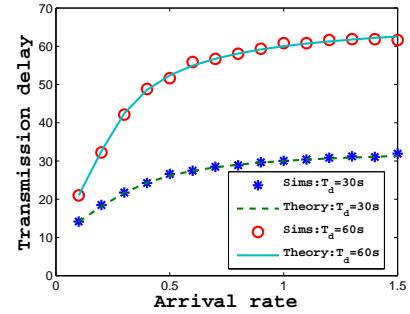


Figure 5.5: Average delay for vehicular user scenario.

### 5.3 Performance evaluation

In this section we will validate our theory against simulations for a wide range of traffic intensities, different values of file sizes, WiFi availability periods with different distributions, and different deadline times. We define the WiFi availability ratio as  $AR = \frac{E[T_{ON}]}{E[T_{ON}] + E[T_{OFF}]} = \frac{\gamma}{\eta + \gamma}$ . Unless otherwise stated the durations of WiFi availability and unavailability periods will be drawn from independent exponential distributions with rates  $\eta$  and  $\gamma$ , respectively. The deadlines are exponentially distributed with rate  $\xi$ , although we will simulate scenarios with deterministic deadlines as well. We mainly focus on two scenarios, related to the user's mobility. The first one considers mostly pedestrian users with data taken from [195]. Measurements in [195] report that the average duration of WiFi availability period is 122 min, while the average duration with only cellular network coverage is 41 min (we use these values to tune  $\eta$  and  $\gamma$ ), with AR=0.75. The second scenario corresponds to vehicular users, related to the measurement study of [51]. An AR of 11 % has been reported in [51], although not all the details are mentioned there. For more details about the measurements we refer the interested reader to [195] and [51]. Finally, unless otherwise stated, file/flow sizes are exponentially distributed, and file arrival at the mobile user is a Poisson process with rate  $\lambda$ .

#### 5.3.1 Validation of main delay result

We first validate here our model and main delay result (Eq.(5.1)) against simulations for the two mobility scenarios mentioned (pedestrian and vehicular). The data rate for WiFi is assumed to be 1 Mbps. The mean packet size is assumed to be 7.5 MB for the pedestrian scenario and 125 kB for the vehicular scenario.

Fig. 5.4 shows the average file transmission delay (i.e. queueing + transmission) for the pedestrian scenario, for two different average deadline times of  $T_{d1} = 1$  hour ( $\xi_1 = 1/3600 s^{-1}$ ) and  $T_{d2} = 2$  hours ( $\xi_2 = 1/7200 s^{-1}$ ), respectively. The range of arrival rates shown corresponds to a server utilization of 0-0.9. We can observe from Fig. 5.4 that there is a good match between theory and simulations. Furthermore, the

Table 5.2: Probability of reneging for pedestrian and vehicular scenarios.

Scenario	Deadline	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 1$	$\lambda = 1.5$
Pedestrian(Theory)	1 hour	0.103	0.109	0.252	0.501
Pedestrian(Simulation)	1 hour	0.1	0.117	0.239	0.508
Vehicular(Theory)	60 s	0.32	0.778	0.889	0.926
Vehicular(Simulation)	60 s	0.32	0.776	0.891	0.925

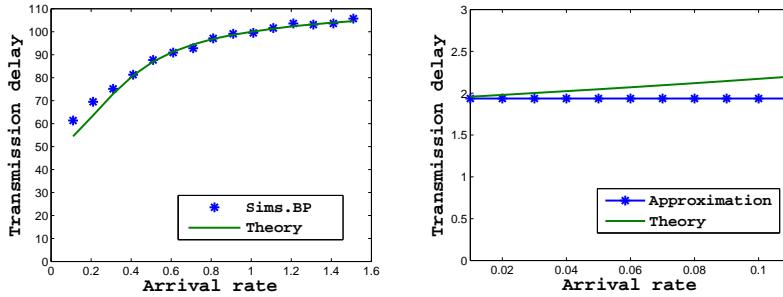


Figure 5.6: The delay for BP ON-OFF periods vs. theory.

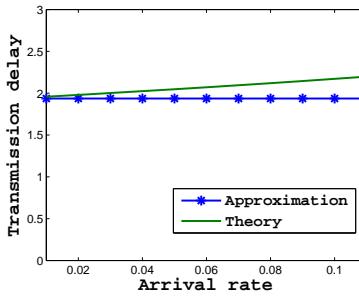


Figure 5.7: Low utilization delay approximation for  $AR = 0.75$ .

average file transmission delay is increased by increasing the arrival rate, as expected, due to queueing effects. On the other hand, the average delay increases for higher deadlines, since flows with lower deadlines leave the WiFi queue earlier, leading to smaller queueing delays. Fig. 5.5 further illustrates the average file transmission delay for the vehicular scenario with average deadline times  $T_{d1} = 30\text{s}$  ( $\xi_1 = 1/30\text{s}^{-1}$ ) and  $T_{d2} = 60\text{s}$  ( $\xi_2 = 1/60\text{s}^{-1}$ ). Despite the differences of the vehicular scenario, similar conclusions can be drawn. Finally, Table 5.2 depicts the respective probabilities of reneging for the two scenarios. The percentage of flows that abandon the WiFi queue is higher in the vehicular scenario, since the availability ratio of the WiFi network is very small (11%), and deadlines are rather small. These observations agree with [51]. Nevertheless, our theory matches simulation values in all scenarios.

So far, we have assumed exponential distributions for ON and OFF periods, according to our model. While the actual distributions are subject to the user mobility pattern, a topic of intense research recently, initial measurement studies ([195, 51]) suggest these distributions to be "heavy-tailed". To this end, we consider a scenario with "heavy-tailed" ON/OFF distributions (Bounded Pareto). Due to space limitations, we focus on the vehicular scenario only. The shape parameters for the Bounded Pareto ON and OFF periods are  $\alpha = 0.59$  and  $\alpha = 0.64$ , respectively. The average deadline is 100s. Fig. 5.6 compares the average file delay against our theoretical prediction. Interestingly, our theory still offers a reasonable prediction accuracy, despite the considerably higher variability of ON/OFF periods in this scenario. While we cannot claim this to be a generic conclusion for any distribution and values, the results underline the utility of our model in practice.

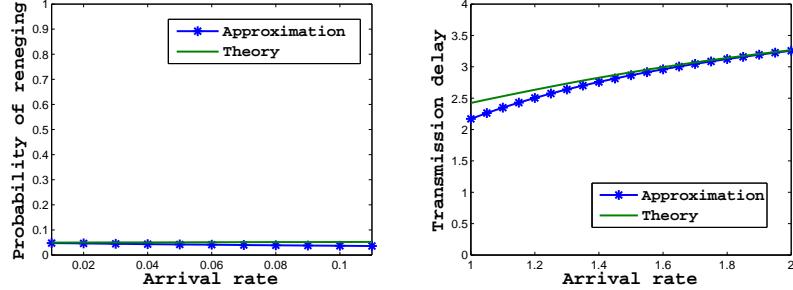


Figure 5.8: Low utilization  $p_r$  approximation for  $AR = 0.75$

Figure 5.9: High utilization delay approximation for  $AR = 0.5$ .

### 5.3.2 Validation of approximations

We next validate the approximations we have proposed in Section 5.2. We start with the low utilization approximation of Section 5.2.2 with  $AR = 0.75$  (similar accuracy levels have been obtained with other values) and with a deadline of 2min. Fig. 5.7 shows the packet delay for low arrival rates in the range  $0.01 - 0.11$ , which correspond to a maximum utilization of up to 0.2. As  $\lambda$  increases, the difference between the approximated result and the actual value increases, since we have considered only the service time for this approximation. The same conclusion holds for the probability of reneging (Fig. 5.8).

Next, we consider the high utilization regime and respective approximation (Eq.(5.54)). We consider  $\rho$  around 0.8. Fig. 5.9 shows the delay for high values of  $\lambda$  ( $AR = 0.5$ ). We can see that our approximation is very close to the actual delay and should become exact as  $\rho$  gets larger.

### 5.3.3 Variable WiFi rates and non-exponential parameters

While in our model we consider a fixed transmission rate for all WiFi hotspots, this is not realistic in practice. For this reason, we have also simulated scenarios where the WiFi rate varies uniformly in the range 0.4-1.6 Mbps. Fig. 5.10 shows the delay for the vehicular scenario with a deadline of 10 minutes. Even in this case, our theory can give solid predictions for the incurred delay.

In all of the above scenarios, we have assumed variable deadlines for each file (drawn from an exponential distribution). In some cases, the user might choose the same deadline for many (or most) flows that can be delayed, which would be a measure of her patience. To this end, we simulate a scenario where the deadline is fixed for an arrival rate of 0.1. The other parameters are the same as for the vehicular scenario. In Fig. 5.12 we compare simulation results for this scenario against our theory (which assumes exponentially distributed deadlines with the same average). It is evident that even in this case, there is a reasonable match with our theory.

To conclude our validation, we finally drop the exponential packet assumption as well, and test our theoretical result vs. generic file size results. Fig. 5.11 compares

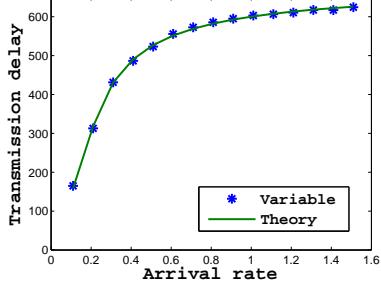


Figure 5.10: Variable WiFi rates with the same average as theory.

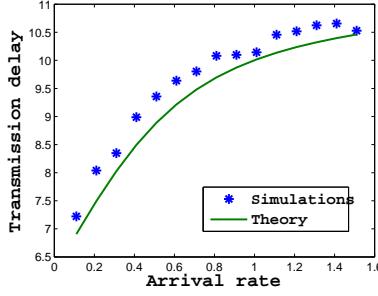


Figure 5.11: Deterministic packets

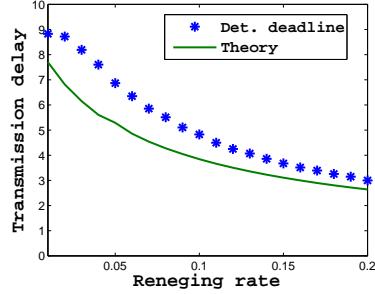


Figure 5.12: The delay for deterministic deadlines vs. theory.

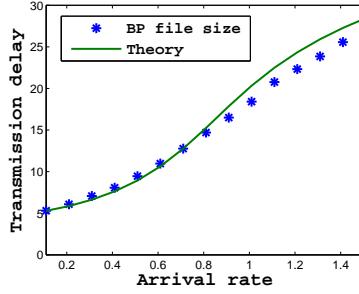


Figure 5.13: The delay for BP packet sizes vs. theory.

analytical and simulation results for deterministic, and Fig. 5.13 does it for Bounded Pareto files sizes (shape parameter  $\alpha = 1.2$  and  $c_v = 3$ ). In both scenarios, the deadline is  $T_d = 20$ s. Mean file size is in both cases 125KB, AR is 0.5, and the rest of the parameters correspond to the vehicular scenario. Our theoretical prediction remains reasonably close, despite higher size variability.

### 5.3.4 Delayed offloading gains

We have so far established that our analytical model offers considerable accuracy for scenarios commonly encountered in practice. In this last part, we will thus use our model to acquire some initial insight as to the actual offloading gains expected in different scenarios. The operator's main gain is some relief from heavy traffic loads leading to congestion. The gains for the users are the lower prices usually offered for traffic migrated to WiFi, as well as the potential higher data rates of WiFi connectivity. There are also reported energy benefits associated [196], but we do not consider them here. In this last part, we will investigate the actual gains from data offloading, in terms of offloading efficiency. Higher offloading efficiency means better performance for both client and operator. We compare the offloading efficiencies for on-the-spot offloading [49] vs.

delayed offloading for different deadline times ( $T_{d1} = 2\text{min}$ ,  $T_{d2} = 1\text{min}$ ). Fig. 5.14 illustrates the offloading efficiency vs. availability ratio for a moderate arrival rate of  $\lambda = 0.2$ . For comparison purposes we also depict the line  $x = y$  (offloading efficiency = availability ratio). First, as expected, we can observe that offloading efficiency increases with AR. However, this increase is not linear. More interestingly, the actual offloading efficiencies are always higher than the respective availability ratio. As expected, the delayed offloading provides higher offloading efficiencies compared to on-the-spot offloading, with higher deadlines leading to higher offloading efficiencies. For the same AR doubling the deadline time, the offloading efficiency is increased by about 10%. Also, although not shown here, the respective offloading efficiency increases even further as traffic loads decreases. Summarizing, these findings are particularly interesting to operators (and users), as they imply that high offloading efficiencies can be achieved for loaded regions, without necessarily providing almost full coverage with WiFi APs.

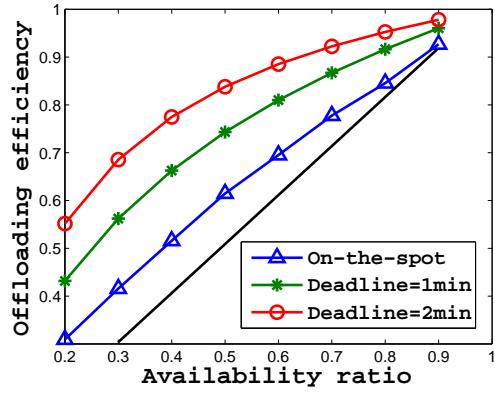


Figure 5.14: Offloading gains for delayed vs. on-the-spot offloading.

## 5.4 Optimizing Delayed Offloading

The results considered so far allow us to predict the expected system delay when the deadlines are defined externally (e.g. by the user or the application). However, the user (or the device on her behalf) could choose the deadline in order to solve an optimization problem among additional (often conflicting) goals, such as the monetary *cost* for accessing the Internet and the *energy consumption* of the device. For example, the user might want to minimize the delay subject to a maximum (energy or monetary) cost, or to minimize the cost subject to a maximum delay the user can tolerate.

To formulate and solve such optimization problems, we need analytical formulas for the average delay and the incurred cost. We already have such formulas for the delay of files sent over WiFi, where we will use the two approximations of Sections 5.2.2 and 5.2.3. Furthermore, we can assume that files transmitted over the cellular network

incur a fixed delay  $\Delta$ , capturing both the service and queueing delays over the cellular interface<sup>4</sup>. To proceed, we need to also assume simple models for energy and cost, in order to get some initial intuition about the tradeoffs involved. We are aware that reality is more complex (for both energy and cost) and may differ based on technology (3G, LTE), provider, etc. We plan to extend our models in future work.

Assume a user has to download or upload a total amount of data equal to  $L$ . On average  $p_r \cdot L$  data units will be transmitted through the cellular interface. Assume further that  $D_c$  and  $D_w$  denote the costs per transmitted data unit for a cellular and WiFi network, where  $D_w < D_c$  (often  $D_w = 0$ ). Finally, let  $c_c$  and  $c_w$  denote the transmission rates, and  $E_c$  and  $E_w$  energy spent per time unit during transmission over the cellular and WiFi network, respectively. It is normally the case that  $c_c < c_w$  as well as  $E_c \approx E_w$  [205]. It follows then that the total monetary and energy costs,  $D$ , and  $E$ , could be approximated by

$$D = (D_c - D_w)p_r + D_w \quad \text{and} \quad E = \left( \frac{E_c}{c_c} - \frac{E_w}{c_w} \right) p_r + \frac{E_w}{c_w}. \quad (5.56)$$

### 5.4.1 Optimization problems

**Optimization problem 1:** Eq.(5.56) suggests that both the average power consumption and cost depend linearly on the probability of renege,  $p_r$ , which we have also derived in Section 5.2, and which is a function of the system deadline  $\frac{1}{\xi}$ . The system delay is also a function of  $\xi$ . We can thus formulate optimization problems of the following form, for both the high and low utilization regimes, where  $\xi$  is the optimization parameter

$$\begin{aligned} \min_{\xi} \quad & E[T] + p_r \Delta \\ \text{s. t.} \quad & p_r \leq P_r^{max}, \end{aligned} \quad (5.57)$$

where  $E[T]$  is given by Eq.(5.49), and  $p_r$  by Eq.(5.50), for low utilization, and Eq.(5.54) and Eq.(5.55), for high utilization, respectively. Due to the linearity of Eq.(5.56), we can express the constraint directly for  $p_r$ , where  $P_r^{max}$  depends on whether we consider monetary cost, energy or a weighted sum of both, and the respective parameters. Finally, we can also exchange the optimization function with the constraint, to minimize the cost, subject to a maximum delay. This provides us with a large range of interesting optimization problems we can solve.

If we express the inequality constraint in Eq.(5.57) through  $\xi$ , we have the equivalent constraint  $\xi \leq \frac{\theta_3 P_r^{max}}{\theta_1 - \theta_2 P_r^{max}}$ . The probability of renege from Eq.(5.50) is an increasing function of  $\xi$ , since  $p_r'(\xi) > 0$ . This implies that maximum  $p_r$  corresponds to maximum  $\xi$ . We denote by  $f(\xi)$  the total average delay of Eq.(5.57) (delay function from now on). Hence we have

---

<sup>4</sup>We could also try to model the cellular queue as an M/M/1 or G/M/1 system, but we are more interested in the dynamics of the WiFi queue, since this is where the renege decisions take place. To keep things simple, we defer this to future work.

$$f(\xi) = \frac{A_1\xi + A_2}{B_1\xi + B_2} + \frac{\theta_1\xi}{\theta_2\xi + \theta_3}\Delta, \quad (5.58)$$

where  $A_1 = \gamma, A_2 = (\eta + \gamma)^2 + \eta\mu, B_1 = (\mu + \eta)(\gamma + \eta), B_2 = \mu\gamma(\gamma + \eta)$ . In order to solve the optimization problem given by Eq.(5.57), we need to know the behavior of the delay function. For that purpose, we analyze the monotonicity and convexity of Eq.(5.58). To do that we need the first and second derivatives, which are

$$\begin{aligned} f'(\xi) &= \frac{A_1B_2 - A_2B_1}{(B_1\xi + B_2)^2} + \frac{\theta_1\theta_3\Delta}{(\theta_2\xi + \theta_3)^2}, \text{ and} \\ f''(\xi) &= \frac{2(A_2B_1 - A_1B_2)}{(B_1\xi + B_2)^3} - \frac{\theta_1\theta_2\theta_3\Delta}{(\theta_2\xi + \theta_3)^3}. \end{aligned}$$

It is worth noting that  $A_1B_2 < A_2B_1$ . This prevents delay function being always concave. The delay function is decreasing in the interval for which  $f'(\xi) \leq 0$ . This happens when

$$\xi \leq \xi_0 = \frac{\theta_3\sqrt{\frac{A_2B_1 - A_1B_2}{\theta_1\theta_3\Delta}} - B_2}{B_1 - \theta_2\sqrt{\frac{A_2B_1 - A_1B_2}{\theta_1\theta_3\Delta}}}.$$

Hence, the delay function is decreasing in the interval  $(0, \xi_0)$ , and increasing in the rest, with  $\xi_0$  being a minimum. Further, the solution of  $f''(\xi) > 0$  gives the interval where the function is convex. This happens when

$$\xi \leq \xi_1 = \frac{\theta_3\sqrt[3]{2\frac{A_2B_1 - A_1B_2}{\theta_1\theta_2\theta_3\Delta}} - B_2}{B_1 - \theta_2\sqrt[3]{2\frac{A_2B_1 - A_1B_2}{\theta_1\theta_2\theta_3\Delta}}}. \quad (5.59)$$

It can be easily proven that  $\xi_0 < \xi_1$ .

Such constrained-optimization problems are often solved with the Lagrangian method and KKT conditions. However, the optimal solution for our problem can be found more easily. The delay function looks like in Fig. 5.15. The optimal deadline depends on the maximum cost, that is proportional to the probability of reneging. So, we can determine the optimal deadline based on the value of  $P_r^{max}$ . If this value of  $P_r^{max}$  is quite high, the corresponding reneging rate  $\xi_{q1}$  (dashed line in Fig. 5.15) will be higher than the global minimum  $\xi_0$ . Consequently, the global minimum of Eq.(5.59) is also the optimal reneging rate. On the other hand, if the maximum cost is quite low (low  $P_r^{max}$ ), the maximum reneging rate  $\xi_{q,2}$  (dotted line in Fig. 5.15) is lower than the global minimum. This implies that the minimum delay will be achieved for the maximum reneging rate of  $\xi_{q,2} = \frac{\theta_3P_r^{max}}{\theta_1 - \theta_2P_r^{max}}$ . In other words, the average deadline time that minimizes the delay for a given maximum cost is

$$T_{d,opt} = \frac{1}{\xi_{opt}} = \frac{1}{\min\left(\xi_0, \frac{\theta_3P_r^{max}}{\theta_1 - \theta_2P_r^{max}}\right)}. \quad (5.60)$$

Similar steps can be followed to solve the same optimization problem for high utilization, as well as other problems. These sometimes involve a quadratic constraint function.

**Optimization problem 2:** After minimizing the transmission delay subject to a maximum renege rate (cost, energy), our next goal now is to minimize the renege probability subject to a maximum transmission delay, which can be for example due to QoS requirements. Hence, the optimization problem in this case would be

$$\begin{aligned} \min_{\xi} \quad & p_r = \frac{\theta_1 \xi}{\theta_2 \xi + \theta_3} \\ \text{s. t.} \quad & E[T] + p_r \Delta \leq T_{max}. \end{aligned} \quad (5.61)$$

Just as in Optimization problem 1, we study the monotonicity and convexity of the delay function, with the only difference that now it is the constrain function. For the probability of renege, we already know that it is an increasing function. Following a similar procedure as in the previous problem, we get for the optimum value of the deadline (from a quadratic constraint)

$$T_{d,opt} = \frac{1}{\max \left( 0, \frac{K_2 - \sqrt{K_2^2 - 4K_1 K_3}}{2K_1} \right)}, \quad (5.62)$$

where  $K_1 = A_1 \theta_2 + \theta_1 \Delta B_1 - T_{max} B_1 \theta_2$ ,  $K_2 = T_{max} B_1 \theta_3 + T_{max} B_2 \theta_2 - A_1 \theta_3 - A_2 \theta_2 - \theta_1 \Delta B_2$ ,  $K_3 = A_2 \theta_3 - T_{max} B_2 \theta_3$ ,  $C_1 = \frac{1}{\lambda} \left( 1 + \frac{\gamma}{\eta} \right) (\lambda - \mu \pi_w)$ .

In [54] we have also considered the respective optimization problems for high utilization regime, where the expressions for  $E[T]$  and  $p_r$  are given by Eq.(5.54) and Eq.(5.55), respectively. The interested reader can find the details there.

### 5.4.2 Optimization evaluation

We will now validate the solutions of the previous optimization problem for two different cases. In both of them the arrival rate is 0.1, and the maximum cost per data unit one can afford is 2.8 monetary units. The transmission of a data unit through WiFi costs 1, and through cellular 5 units. The choice of these values is simply for better visualizing the results; different values yield similar conclusions. Fig. 5.16 shows the delay vs. cost curve for cellular rate being  $2\times$  lower than WiFi rate. First thing that we can observe is that the minimum delay is achieved for the highest possible cost (2.8). The optimal average deadline is  $T_d = 1s$ . This is in agreement with the optimal value predicted from Eq.(5.60), and shown with an asterisk in Fig. 5.16. We replace Eq.(5.50) into Eq.(5.56) to get the relationship between the cost and the renege rate. We have shown in Eq.(5.56) that the cost is directly proportional to  $p_r$ , and the later one is an increasing function of  $\xi$ . This implies that the maximum cost is in fact the maximum  $\xi$  (minimum deadline). This practically means that in Eq.(5.57),  $\Delta$  is small and that the delay in the WiFi queue represents the largest component of the delay. As a consequence of that, it is better to redirect the files through the cellular interface as soon as possible. Hence, in these cases (when cellular rate is comparable to the WiFi),

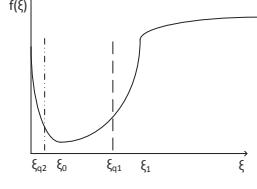


Figure 5.15: The delay function for the optimization problem.

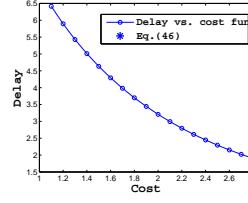


Figure 5.16: The delay vs. cost curve for high cellular rate.

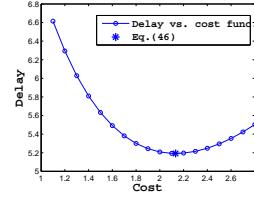


Figure 5.17: The delay vs. cost curve for low cellular rate.

the optimum is to assign the shortest possible deadline constrained by the monetary cost.

Fig. 5.17 corresponds to a scenario with the same parameters as in Fig. 5.16, except that now the cellular rate is much lower ( $10\times$ ). In that case,  $\Delta$  is high, and  $p_r\Delta$  is the largest component of the delay function. As can be seen from Fig. 5.17, it is not the best option to leave as soon as possible from the WiFi queue, i.e. choose the smallest possible deadline. The optimum delay is achieved for  $T_d = 5$ s. This corresponds to an average cost of  $D = 2.1$  which is also very close to the theoretical solution of the problem. This is reasonable since for a large difference between the WiFi and cellular rates it is better to wait and then (possibly) be served with higher rate, than to move to a much slower interface (cellular).

Further, we use the solutions of the four optimization problems for exponentially distributed deadline times to see how accurately our theory can predict the optimal deadline times, but for deterministic deadlines. The optimal policy essentially finds the optimal value for the *average* deadline (assuming these are exponential). In practice, the chosen deadline will be assigned to all files, and will be deterministic. We consider four scenarios, one for each optimization problem. The costs are the same as before. The arrival rate for low utilization scenarios is 0.1, while for the high ones, 1.5. In Table 5.3, we show the optimal deadlines by using our model (e.g. Eq.(5.60)), and the optimal deterministic deadlines by using simulations (delay vs. cost plots) with the same parameters as in theory. As can be seen from Table 5.3, the error in determining the optimal deadline decreases for higher arrival rates. The error is in the range 10%-20%. This is reasonable, since the simulated scenarios are with deterministic deadlines and in our theory we use exponential deadlines. Another reason is that in optimization problems, we are only using the low and high utilization approximations and not the exact result (Eq.(5.1)).

## 5.5 Related Work

Some recent influential work in offloading relates to measurements of WiFi availability [195, 51]. Authors in [195] have tracked the behavior of 100 users (most of which were pedestrians) and their measurements reveal that during 75% of the time there is WiFi connectivity. In [51], measurements were conducted on users riding metropolitan

Table 5.3: Optimal deterministic deadline times vs theory.

Sc.	Constraint	$T_d$ (theory)	$T_d$ (deterministic)	Relative error
1	$D \leq 2.8$	2.71	2.2	18%
2	$T_{max} = 6.6$	1.56	1.22	22%
3	$D \leq 3.8$	1.73	1.55	11%
4	$T_{max} = 15$	7.97	6.82	15%

area buses. In contrast to the previous study, the WiFi availability reported there is only around 10%. The mean duration of WiFi availability and non-availability periods is also different in the two studies, due to the difference in speeds between vehicular and pedestrian users. The most important difference between the two studies relates to the reported offloading efficiency, with [51] reporting values in the range from 20%-33% for different deadlines, and [195] reporting that offloading does not exceed 3%. We believe this is due to the different deadlines assumed together with the different availabilities.

The authors in [206] define a utility function related to delayed offloading to quantitatively describe the trade-offs between the user satisfaction in terms of the price that she has to pay and the experienced delay by waiting for WiFi connectivity. However, their analysis does not consider queueing effects. Such queueing effects may affect the performance significantly, especially in loaded systems (which are of most interest) or with long periods without WiFi. The work in [207] considers the traffic flow characteristics when deciding when to offload some data to the WiFi. However, there is no delay-related performance analysis. Modeling the cost factors is the focus of [208], which also shows where the offloading APs should be installed. A WiFi offloading system that takes into account a user's throughput-delay tradeoffs and cellular budget constraints is proposed in [209]. However, only heuristic algorithms are proposed, and queueing effects are ignored. Finally, in [210], an integrated architecture has been proposed based on opportunistic networking to switch the data traffic from the cellular to WiFi networks. Summarizing, in contrast to our work, these papers either perform no analysis or use simple models that ignore key system effects such as queueing.

To our best knowledge, the closest work in spirit to ours is [52]. The results in [52] are the extension of the results in [195] containing the analysis for delayed offloading. Authors there also use 2D Markov chains to model the state of the system. However, they use matrix-analytic methods to obtain a numerical solution for the offloading efficiency. Such numerical solutions unfortunately do not provide insights on the dependencies between different key parameters, and cannot be used to formulate and analytically solve optimization problems that include multiple metrics.

As a final note, in [49], we have proposed a queueing analytic model for on-the-spot mobile data offloading, and a closed form solution was derived for the average delay. While the model we propose here shares some similarities (ON/OFF availabilities, 2D Markov chain approach) with the basic model in [49], it is in fact considerably more difficult to solve.

## 5.6 Conclusion

Summarizing, in this chapter we have proposed a queueing analytic model for the performance of delayed mobile data offloading, and have validated it against realistic WiFi network availability statistics. We have also considered a number of scenarios where one or more of our model's assumptions do not hold, and have observed acceptable accuracy, in terms of predicting the system delay as a function of the user's patience. Finally, we have also shown how to manipulate the maximum deadlines, in order to solve various optimization problems involving the system delay, monetary costs, and energy costs.

## **Chapter 6**

# **User Association in HetNets**

## 6.1 Introduction

In the previous chapter, we considered offloading through *out-of-band* communication, namely WiFi access points. Here we will generalize to a scenario consisting of various types of base stations, with heterogeneous powers and coverage ranges, such as macro-cells, micro-cells, pico-cells, femto-cells, etc. These networks are often referred to as *HetNets* (or sometimes *Ultra-Dense Networks*, *UDN*) depending on the density and topology of base stations. Note that WiFi can also be seen as a type of small cell. However, we will here assume that all these cells are under the control of a single operator, whose goal is to optimally utilize all its available BSs, balancing their loads, while also ensuring good user QoS.<sup>1</sup> Finally, we also remove in this chapter the assumption of delay-tolerance (Note that a user is still sensitive to delay, but the order of delays experienced here is assumed smaller, and the source of delays is queueing and transmission delay, rather than the explicit and larger delays to encounter a WiFi AP, considered in the previous chapter).

In a HetNet, a large number of small cells (SC) are deployed along with macrocells to improve spatial reuse [55, 56, 57]. The higher the deployment density, the better the chance that a user equipment (UE) can be associated with a nearby base station (BS) with high signal strength, and the more the options to balance the load. At the same time, denser deployments experience high spatio-temporal load variations, and require sophisticated user association algorithms. There are two key, often conflicting concerns when assigning UEs to a BS: (i) maximizing the spectral efficiency, and (ii) ensuring that the load across BSs is balanced to ensure efficient utilization of BSs, and preempt congestion events (i.e., no BS is underutilized while neighboring ones are operating close or above their capacity). The former is usually achieved by associating the UE to the BS with maximum SINR: this association rule was the norm up to LTE (Long-Term Evolution)-release 8. While this rule also maximizes the *instantaneous* rate of a user (i.e., the modulation and coding scheme - MCS - supported), it reflects user QoS only when the BS is lightly loaded. However, user performance, in terms of *per flow delay*, may be severely affected if the BS offering the best SINR is congested [211, 212].

As a result, a number of research works have studied the problem of user association in heterogeneous networks, optimizing user rates [60, 61], balancing BS loads [62], or pursuing a weighted tradeoff of them [63]. For instance, a distributed user-association algorithm is proposed in [213], where the global outage probability and the long term rate maximization are studied. The authors in [214] propose a framework that studies the interplay of user association and resource allocation in future HetNets, by formulating a non-convex optimization problem and deriving performance upper bounds. Range-expansion techniques, where the SINR of lightly loaded BSs is biased to make them more attractive to the users are also popular [56, 57]. Finally, a framework that has received much attention is [63]. This framework jointly considers a family of objective functions, each of which directs the optimal solution towards different goals (e.g. throughput optimal, delay-optimal, load balancing, etc.), using an iterative algorithm. [64, 215, 216] extend this framework to further include

---

<sup>1</sup>Note that energy-efficiency might be another concern of such an operator, sometimes having the opposite intention to “unbalance” loads, so as to turn off underloaded BSs. We have considered such a scenario in [68].

energy management, e.g., by switching off under-loaded BSs.

Nevertheless, the majority of these works are relatively simplified, not taking into account key features of future networks. Firstly, most existing studies only consider homogeneous traffic profiles. For example, [63, 64] assume that all flows generated by a UE are “best-effort” (i.e. elastic). However, modern and future networks will have to deal with high traffic differentiation, with certain flows being able to require specific, *dedicated*<sup>2</sup> (i.e., non-elastic) resources [67]. Such dedicated flows do not share BS resources like best-effort ones, are subject to admission control, and sensitive to different performance metrics [68]. Secondly, the majority of related studies only consider downlink (DL) traffic. Uplink (UL) traffic is becoming important, due to symmetric (e.g. social networking) applications, Machine-Type Communication (MTC), etc. Yet, due to the asymmetric transmit powers of UEs and BSs, leading to different physical data rates, the BS which is optimal for DL traffic might lead to severely degraded performance for UL traffic. Summarizing, a proper user-association scheme should consider all the above dimensions, and attempt to strike an appropriate tradeoff between them.

On top of that, most related works focus on the radio access part (e.g., considering the user rate on the radio interface or BS load), ignoring the backhaul (BH) network. While this might be reasonable for legacy cellular networks, given that the macro-cell backhaul is often over-provisioned (e.g., fiber), this might be quite suboptimal for future cellular networks. The considerably higher number of small cells, and related Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) suggest that backhaul links will mostly be inexpensive wired or wireless links (in licensed or unlicensed bands), and often underprovisioned [69]. Multiple BS might also have to share the capacity of a single backhaul link due to, e.g, point-to-multipoint (PMP) or multi-hop mesh topologies to the aggregation node(s) [70]. Finally, various BS-coordinated schemes have been proposed in the literature as a promising way to better use the available spectrum and further improve system performance, e.g., enhanced Inter-Cell Interference Coordination (eICIC) [217, 218] and Coordinated Multi-Point (CoMP) transmission [219] scenarios. Such schemes are expected to further stress the backhaul network capacities. Hence, as the radio access technologies are constantly improving, it is argued that the backhaul network will emerge as a major performance bottleneck, and user association algorithms that ignore the backhaul load and topology can lead to poor performance [220].

As a result of this increasing focus on the backhaul, some recent works have appeared that attempt to jointly consider radio access and backhaul. These are mostly concerned with joint scheduling issues (for in-band or PMP backhaul links) [220, 221], signaling overhead and performance tradeoffs for cooperative multi-point communication [222], and Software-Defined-Networking (SDN)-based implementation flexibility [223].

The user rate maximization problem is studied in [224] under backhaul capacity constraints, and in [225] jointly with backhaul resource allocation and flow control. Also, a distributed user association scheme was developed for maximizing the network-

---

<sup>2</sup>In terms of LTE systems, dedicated flows are differentiated by their QoS class (QCI) ranging from 1 to 4, whereas best-effort from 5 to 9 [67].

wide spectrum efficiency in [226] using combinatorial optimization. Some backhaul aware association heuristics include: [227] where the sum of user rates is attempted to be optimized in an energy-efficient manner, [228] where the ergodic capacity is maximized under an iterative algorithm, and [229] where resource allocation is investigated in conjunction with carrier aggregation. Other works in this context include investigation of caching capabilities to overcome the backhaul capacity limitations and enhance QoS [230, 81]. Finally, Chen et al. attempt to derive the total expected delay by considering retransmission over the wireless backhaul links [231].

Nevertheless, to our best knowledge, none of these works formally addresses the problem of optimal user association in traffic-differentiated and backhaul-limited HetNets. To this end, in this chapter we revisit the user association problem, jointly considering the radio access and backhaul networks. Specifically, our main contributions can be summarized as follows:

1. We use the popular framework of  $\alpha$ -optimal user association [63] as our starting point, and introduce (i) traffic differentiation, (ii) UL traffic, and (iii) backhaul topology and capacity constraints.
2. We then analytically prove optimal association rules, depending on whether UL and DL traffic of the same UE can be “split” to different BSs or not [74]. We also show how this optimal regime can be attained using an iterative and distributed algorithm that does not require global network knowledge.
3. We use this framework to investigate the various tradeoffs arising in this complex association problem, and provide some initial insights and guidelines about the impact of traffic differentiation and backhaul limitations in optimal user-association policies for future HetNets.

## 6.2 System Model and Assumptions

In the following, we describe our traffic arrival model (Section 6.2.1), then discuss our assumptions related to the access network (Section 6.2.2) and backhaul network (Section 6.2.3).

We use a similar problem setup as the one used in a number of related works [63, 64, 232, 215], and extend it accordingly. To keep notation consistent, for all variables considered, a first superscript “D” and “U” refers to downlink (DL) and uplink (UL) traffic, respectively. A second superscript “b” or “d” refers to best-effort and dedicated traffic, respectively. For brevity, in the following *we present most notation and assumptions in terms of downlink traffic only, assuming that the uplink case and notation is symmetric*. Specific differences will be elaborated, where necessary. In Table 6.1, we summarize some useful notation we use throughout the paper.

### 6.2.1 Traffic Model

**(A.1 - Traffic arrival rates)** Traffic at location  $x \in \mathcal{L}$  consists of file (or more generally flow) requests arriving according to an inhomogeneous Poisson point process with

Table 6.1: Notation

Variable	Best-Effort Flows		Dedicated Flows	
	Downlink	Uplink	Downlink	Uplink
Flow type superscript	D,b	U,b	D,d	U,d
Flow type probability	$z^D \cdot z^b$	$z^U \cdot z^b$	$z^D \cdot z^d$	$z^U \cdot z^d$
Devoted amount of bandwidth for BS $i$	$w_i^D \cdot \zeta^D$	$w_i^U \cdot \zeta^U$	$w_i^D \cdot (1 - \zeta^D)$	$w_i^U \cdot (1 - \zeta^U)$
Traffic arrival rate at location $x$	$\lambda^{D,b}(x)$	$\lambda^{U,b}(x)$	$\lambda^{D,d}(x)$	$\lambda^{U,d}(x)$
Maximum rate   servers at location $x$ for BS $i$	$c_i^{D,b}(x)$	$c_i^{U,b}(x)$	$k_i^D(x)$	$k_i^U(x)$
Load density at location $x$ for BS $i$	$\rho^{D,b}(x)$	$\rho^{U,b}(x)$	$\rho^{D,d}(x)$	$\rho^{U,d}(x)$
Alpha-fair parameter $\in [0, \infty)$	$\alpha^{D,b}$	$\alpha^{U,b}$	$\alpha^{D,d}$	$\alpha^{U,d}$
Utilization (load) of BS $i$	$\rho^{D,b}$	$\rho^{U,b}$	$\rho^{D,d}$	$\rho^{U,d}$
Probability to route a flow from $x$ to BS $i$	$p^{D,b}(x)$	$p^{U,b}(x)$	$p^{D,d}(x)$	$p^{U,d}(x)$
Mean flow size (bits)   flow duration (sec) at $x$	$\frac{1}{\mu^{D,b}(x)}$	$\frac{1}{\mu^{U,b}(x)}$	$\frac{B^D(x), 1}{\mu^{D,d}(x)}$	$\frac{B^U(x), 1}{\mu^{U,d}(x)}$
Capacity of BH link $j$	$C_h^D(j)$	$C_h^U(j)$	-	-
Congestion indicator at BH link $j$	$\mathcal{T}^D(j)$	$\mathcal{T}^U(j)$	-	-

arrival rate per unit area  $\lambda(x)$ <sup>3</sup>. This inhomogeneity facilitates the creation of “hotspot” areas. Each new arriving request is for a *downlink (DL)* flow, with probability  $z^D$ , or *uplink (UL)* flow with probability  $z^U = 1 - z^D$ . Each DL (or UL) flow can further be a *best-effort* flow (e.g., file download) with probability  $z^b$ , or *dedicated* flow (e.g., a VoIP call), with probability  $z^d = 1 - z^b$ .  $z^D$  and  $z^b$  are input parameters that depend on the traffic mix.

Using a Poisson splitting argument [233], it follows that the above gives rise to 4 independent, Poisson flow arrival processes with respective rates

$$\lambda^{D,b}(x) = z^D \cdot z^b \cdot \lambda(x), \quad \lambda^{D,d}(x) = z^D \cdot z^d \cdot \lambda(x) \quad (6.1)$$

$$\lambda^{U,b}(x) = z^U \cdot z^b \cdot \lambda(x), \quad \lambda^{U,d}(x) = z^U \cdot z^d \cdot \lambda(x), \quad (6.2)$$

( $\lambda^{D,b}(x)$  for downlink best-effort flows,  $\lambda^{U,b}(x)$  for uplink best-effort flows, etc.).

**(A.2 - Best effort flow characteristics)** Each *best-effort* flow is associated with a *flow-size* (in bits) drawn from a generic distribution with mean  $1/\mu^{D,b}(x)$ . This can model heterogeneous flow characteristics across locations.

**(A.3 - Dedicated flow characteristics)** Each *dedicated* flow has a *required data-rate* (in bits per second) that is drawn from a generic distribution with mean  $B^D(x)$ . This rate must be guaranteed by the network throughout the flow’s duration. This duration (in seconds) is another, independent random variable with mean  $1/\mu^{D,d}(x)$ .

## 6.2.2 Access Network

**(B.1 - Access network topology)** We assume an area  $\mathcal{L} \subset \mathbb{R}^2$  served by a set of base stations  $\mathcal{B}$ , that are either macro BSs (eNBs) or small cells (SCs). These together constitute the access network.

**(B.2 - DL resources)** Each BS  $i \in \mathcal{B}$  is associated with a transmit power  $P_i$  and a total downlink bandwidth  $w_i^D$ . Out of the total bandwidth,  $\zeta_i^D \cdot w_i^D$  is allocated to

<sup>3</sup>Without loss of generality, we do not distinguish between users at location  $x$ , as we assume that all users/flows related to location  $x$  are treated similarly.

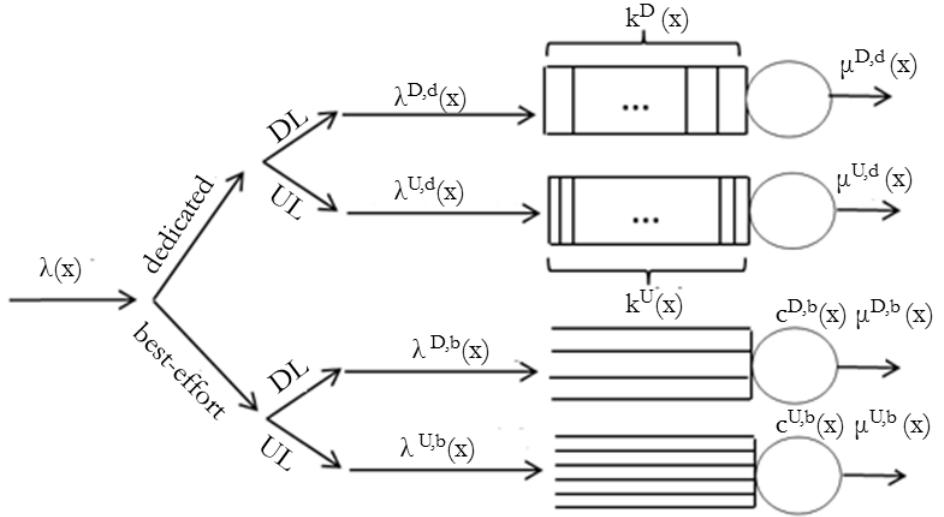


Figure 6.1: Access network queuing systems for different flows.

best-effort traffic and  $(1 - \zeta_i^D) \cdot w_i^D$  for dedicated traffic ( $0 \leq \zeta_i^D \leq 1$ ). Throughout this paper, we will assume that this allocation is static, at least for a given time window of interest (based on long term traffic characteristics and operator policy).

*Remark:* Dynamically updating the  $\zeta_i^D$  parameters could further improve performance. This could, for example, be achieved through dynamic/flexible TDD, where the amount of downlink and uplink resources are jointly optimized with the association parameters. We have recently extended our framework in this direction. We refer the interested reader to [78].

**(B.3 - DL physical data rate)** BS  $i$  can deliver a *maximum* physical data transmission rate of  $c_i^{D,b}(x)$  to a user at location  $x$ , in absence of any other best-effort flows served, which is given by the Shannon capacity<sup>4</sup>

$$c_i^{D,b}(x) = \zeta_i^D \cdot w_i^D \cdot \log_2(1 + \text{SINR}_i(x)), \quad (6.3)$$

where  $\text{SINR}_i(x) = \frac{G_i(x)P_i}{\sum_{j \neq i} G_j(x)P_j + N_0}$ .<sup>5</sup>  $N_0$  is the noise power, and  $G_i(x)$  represents the path loss and shadowing effects between the  $i$ -th BS and the UE located at  $x$  (as well as antenna and coding gains, etc.).<sup>6</sup> We assume that effects of fast fading are filtered out. Our model assumes that the total intercell interference at location  $x$  is static, and

<sup>4</sup>We use Shannon capacity for clarity of presentation. However, our approach could be easily adapted to include modulation and coding schemes (MCS). Furthermore, capacity improving technologies, e.g., the use of MIMO, and modifications to this capacity formula are orthogonal to our framework.

<sup>5</sup>We have assumed that the interference caused to a BS, does not depend on the load of the (neighboring) BSs that interfere with it. This is standard practice to avoid the “coupled queues” problem which is usually not tractable, e.g. see [211],[63]. For a study of random HetNet deployments with such load-based interference, we refer the reader to our recent work of [234].

<sup>6</sup>In the case of UL, we assume that the Tx power of each user is  $P^{UE}$ , and slightly abuse notation

considered as another noise source, as is previously considered in most aforementioned works [63, 64].

The next 4 points (B.4-B.7) describe the scheduling and performance model for best effort traffic only. We return to dedicated traffic in (B.8-B.9).

**(B.4 - Best effort load density)** We introduce the *load density* for best effort flows, at different locations  $x$ ,

$$\rho_i^{D,b}(x) = \frac{\lambda^{D,b}(x)}{\mu^{D,b}(x)c_i^{D,b}(x)}, \quad (6.4)$$

which is the contribution of location  $x$  to the total load of a BS  $i$ , when location  $x$  is associated to BS  $i$ .

**(B.5 - Best effort load)** Each location  $x$  is associated with routing probabilities  $p_i^{D,b}(x) \in [0, 1]$ , which are the probabilities that best effort DL flows generated for users at location  $x$  get associated with (i.e., are served by) BS  $i$ . We can thus define the *total best effort load*  $\rho_i^{D,b}$  for BS  $i$  as

$$\rho_i^{D,b} = \int_{\mathcal{L}} p_i^{D,b}(x)\rho_i^{D,b}(x)dx. \quad (6.5)$$

Similarly to [211, 63], we are interested in the *flow-level dynamics* of this system, and model the service of DL best-effort flows at each BS as a queueing system with load  $\rho_i^{D,b}$  shown in Fig. 7.7. Finally, since we are interested in the aggregation of all flows at BS level (i.e., all flows from all locations  $x$  associated to BS  $i$ ), even if flow arrivals at each location are not Poisson (as in A.1), the Palm-Khintchine theorem [233] suggests that the Poisson assumption could be a good approximation for the input traffic to a BS.

**(B.6 - Best effort scheduling)** Proportionally fair scheduling is often implemented in 3G/4G networks for best-effort flows, due to its good fairness and spectral efficiency properties [67]. This can be modeled as an M/G/1 multi-class processor sharing (PS) system (see, e.g., [211, 63, 64]). It is multi-class, because each flow might get different rates for similarly allocated resources, due to different channel quality and MCS at  $x$ . Channel-based scheduling could also be included in the model and can be accounted for using a multiplicative factor in the average service rate [235].

**(B.7 - Performance for best effort flows)** The stationary number of flows in BS  $i$  is equal to  $E[N_i] = \frac{\rho_i^{D,b}}{1-\rho_i^{D,b}}$  [233]. Hence, minimizing  $\rho_i^{D,b}$  minimizes  $E[N_i]$ , and by Little's law it also minimizes the per-flow delay for that base station [233]. Also, the throughput for a flow at location  $x$  is  $c_i^{D,b}(x)(1 - \rho_i^{D,b})$ . This observation is important to understand how the user's physical data rate  $c_i^{D,b}(x)$  (related to users at location  $x$  only) and the BS load  $\rho_i^{D,b}$  (related to *all* users associated with BS  $i$ ) affect the optimal association rule.

**(B.8 - Dedicated traffic load density)** Unlike best-effort flows which are elastic, dedicated flows are subject to admission control, since they require some resources for

---

for SINR, G, etc., as these don't play a major role in the remaining discussion. Note that since the uplink power of each UE is fixed, the aggregate uplink interference at a given point location  $x$  is independent of the actual uplink association of these UEs. The case of power-controlled uplinks would introduce *all* the control variables in the denominator of the SINR expression, significantly complicating the problem. One can see the chosen setup as a pessimistic bound on the uplink interference.

exclusive usage in order to be accepted in the system. Specifically, let  $c_i^{D,d}(x)$  denote the maximum offered rate to users at location  $x$  corresponding to dedicated flows only (referred to  $(1 - \zeta_i)$  - see B.3 above). If each flow at  $x$  demands, on average, a rate of  $B^D(x)$  (see A.3), then at most  $k_i^D(x) = \frac{c_i^{D,d}(x)}{B(x)}$  dedicated flows from  $x$  could be served in parallel by BS  $i$  (assuming again *no other flows in the system*), and any additional flows would be rejected<sup>7</sup>. Similarly to the best effort case (B.4), we can define a system load density for dedicated traffic at  $x$

$$\rho_i^{D,d}(x) = \frac{\lambda^{D,d}(x)}{\mu^{D,d}(x)k_i^D(x)} = \frac{\lambda^{D,d}(x) \cdot B^D(x)}{\mu^{D,d}(x) \cdot c_i^{D,d}(x)}. \quad (6.6)$$

Hence, a different number of resources  $k_i^D(x)$  can be offered to different locations  $x$ , depending on the rate demand  $B^D(x)$  as well as the channel quality (rate  $c_i^{D,d}(x)$ ) at location  $x$ .

**(B.9 - Dedicated traffic performance)** Given the above heterogeneous blocking model for dedicated flows, we can approximate the allocation of BS  $i$  dedicated resources with an M/G/k/k (or  $k$ -loss) system, where the total load  $\rho_i^{D,d}$  can be calculated as in (B.5) and Eq. (6.5), using the density of Eq. (6.6) and corresponding routing probability  $p_i^{D,d}(x)$  for dedicated flows (see also Fig. 7.7). It is known that for M/G/k/k systems, minimizing  $\rho_i^{D,d}$  is equivalent to minimizing the blocking probability for new flows [233]. This observation is important to understand that a similar tradeoff (as in B.7) exists between choosing a BS at  $x$  that maximizes  $k_i^D(x)$  (related only to flow and channel characteristics at  $x$ ) and choosing a BS whose *total* load  $\rho_i^{D,d}$  (related to *all* users attached to BS  $i$ ).

**(B.10 - UL/DL association split)** We investigate two scenarios, depending on the whether a UE is allowed to be attached to different BSs for its DL and UL traffic [74]:

*Split UL/DL:* Each UE can be associated to different BSs for its DL and UL traffic. This allows one to optimize UL and DL performance independently [236].

*Joint UL/DL:* Each UE must be associated with the same BS for both UL and DL traffic. This is the standard practice in current networks.

### 6.2.3 Backhaul Network

**(C.1 - Backhaul network topology)** Each access network node (either eNB or SC) is connected to the core network through the eNB aggregation gateway via a certain number of backhaul links that constitute the backhaul network. This connection can be either direct (“star” topology) or through one or more SC aggregation gateways (“tree” topology). Fig. 6.2 shows such a backhaul routing topology.

Without loss of generality, we assume that there is a fiber link from the eNB to the core network, and focus on the set of capacity-limited backhaul links (wired or wireless) connecting SCs to the eNB, denoted as  $\mathcal{B}_h$ . We denote as routing path  $\mathcal{B}_h(i)$  the set of all backhaul links  $j \in \mathcal{B}_h$  along which traffic is routed from BS  $i$  to an eNB

---

<sup>7</sup>In fact, since the rate requirement for each flow is a random variable, using its mean  $B^D(x)$  in the denominator yields a lower bound for  $k_i^D(x)$  (by Jensen’s inequality), which can be used as a conservative estimate.

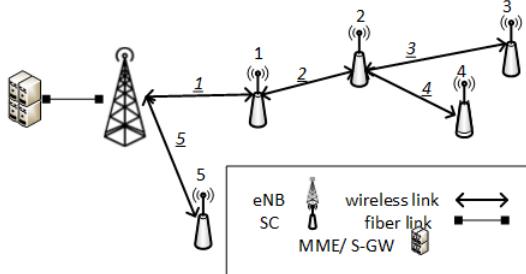


Figure 6.2: Backhaul topology in future Hetnet.

aggregation point. For example, in Fig. 6.2,  $\mathcal{B}_h(1) = \{1\}$ , and  $\mathcal{B}_h(3) = \{1, 2, 3\}$ . We further denote as  $\mathcal{B}(j)$  the set of all BS  $i \in \mathcal{B}$  whose traffic is routed over backhaul link  $j$ . E.g.,  $\mathcal{B}(1) = \{1, 2, 3, 4\}$  and  $\mathcal{B}(2) = \{2, 3, 4\}$  in Fig. 6.2. In the case of a star topology, there is exactly one (unique) backhaul link used for each BS (i.e.,  $\|\mathcal{B}_h(i)\| = \|\mathcal{B}(j)\| = 1, \forall i, j$ ). Finally, we assume that the backhaul route for each BS is given, e.g., calculated in practice as a Layer 2 (L2) spanning tree, and is an input to our problem. In Section 7.3.9, we highlight some limitations of L2 backhaul routing.

**(C.2 - Backhaul load)** Each backhaul link  $j \in \mathcal{B}_h$  is characterized by a DL and UL capacity, denoted as  $C_h^D(j)$  and  $C_h^U(j)$  bps. The capacity on the UL and DL might be the same or different (e.g., Frequency-Division Duplex (FDD)). Additionally, dynamic Time-Division Duplex (TDD) systems [237] could also be assumed for the backhaul links as well, where the UL/DL resource split is yet another control variable (the interested reader can find more details here [238]). Backhaul links usually don't implement any particular scheduling algorithm, and can be seen as a data "pipe".

Without loss of generality, we focus on a scenario with only best-effort traffic, when including backhaul constraints. This will allow us to better understand the impact of backhaul limitations on the wide system performance. Focusing on the DL, the load on a backhaul link  $j \in \mathcal{B}_h$  consists of the sum of downlink loads (corresponding to best-effort traffic) of all BSs using that link:

$$\sum_{i \in \mathcal{B}(j)} \rho_i^{D,b} \tilde{c}_i^D, \quad (6.7)$$

where  $\tilde{c}_i^D$  is a parameter tuned by the operator. It corresponds to the total rate that BS  $i$  can support on the air (and thus has to also carry over the backhaul) when it is busy (i.e. for  $\rho_i$  percent of the time). Note that, while  $\rho_i$  gives the load of the BS, the actual rate for the same load might vary, and it depends on the MCS (Modulation and Coding Scheme) of the users currently being served by it. For example, if the active users (creating the load  $\rho_i$ ) are close to the BS and are being served with the maximum MCS then the actual rate that needs to be carried on the backhaul is very close to the peak nominal rate and  $\tilde{c}_i^D$  shall be relatively high. However, if it is generated by a larger number of users distributed around the entire coverage area of this BS the average MCS (and thus rate) per user will be lower, and thus  $\tilde{c}_i^D$  shall be quite lower than the previous scenario. To that end, depending on how conservative the operator wants to

be (or perhaps depending on past statistics) one could set  $\tilde{c}_i$  close to the peak rate, or close to the average by being more aggressive (it is well known that a heavily loaded BS actually has a lower rate per user than a lightly loaded one [69])<sup>8</sup>.

**(C.3 - Backhaul provisioning)** We have derived the backhaul link load ( $\sum_{i \in \mathcal{B}(j)} \rho_i^{D,b} \tilde{c}_i^D$ ) and defined the backhaul capacity limitation ( $C_h^D(j)$ ) for each backhaul link  $j \in \mathcal{B}_h$  (see C.2). Thus, each of these links is subject to an additional constraint to avoid exceeding its maximum capacity and backhaul congestion ( $\sum_{i \in \mathcal{B}(j)} \rho_i^D \tilde{c}_i^D \prec C_h^D(j) \quad \forall j \in \mathcal{B}_h$ ).

Throughout this paper, we assume that the backhaul network is either *under-provisioned* if the capacity of at least one backhaul link is exceeded, or *provisioned* otherwise. We investigate the user-association problem separately for each scenario in Sections 6.3 and 6.4, by focusing on different tradeoffs.

### 6.3 User-Association for Provisioned Backhaul Networks

We start our discussion for optimal user-association by first assuming that the backhaul network is provisioned. In other words the backhaul constraints are always satisfied and can be safely ignored when deriving the optimal association rules. Our focus will thus be on the radio access network performance: uplink/downlink and traffic-differentiation tradeoffs.

We remind to the reader that based on our system model, the association policy consists in finding appropriate values for the routing probabilities  $p_i^{l,t}(x)$ ,  $l \in \{D, U\}$ ,  $t \in \{b, d\}$ , for DL and UL, best-effort and dedicated traffic, respectively (defined earlier in assumption B.5 and B.9). That is, for each location  $x$ , we would like to optimally choose to which BS  $i$  to route different flow types generated from (UL) or destined at (DL) users in  $x$ <sup>9</sup>. Our goal for this association problem is threefold: (i) ensure that the capacity of no BS is exceeded; (ii) achieve a good tradeoff between spectral efficiency, user QoS and load balancing, (iii) investigate how *UL/DL association split* impacts the optimal rule derivation and the overall performance benefits.

We define the feasible region for the aforementioned routing probabilities, by requiring that no BS capacity being exceeded.

**Definition 6.3.1.** *(Feasible set): Let  $l \in \{U, D\}$ ,  $t \in \{b, d\}$ , and let  $\epsilon$  be an arbitrarily*

---

<sup>8</sup>Note that, since the routing paths on the backhaul network are given (see C.1), if a small cell uses links  $i$  and  $j$  to send its traffic to the macro cell, then all of its traffic will be sent over both links.

<sup>9</sup>The use of a probabilistic association rule convexifies the problem at hand. As it will turn out, the optimal values will turn out to be integer anyway.

small positive constant. The set  $f^{l,t}$  of feasible BS loads  $\rho^{\mathbf{l},\mathbf{t}} = (\rho_1^{l,t}, \rho_2^{l,t}, \dots, \rho_{\|\mathcal{B}\|}^{l,t})$  is

$$\begin{aligned} f^{l,t} = & \left\{ \rho^{\mathbf{l},\mathbf{t}} \mid \rho_i^{l,t} = \int_{\mathcal{L}} p_i^{l,t}(x) \rho_i^{l,t}(x) dx, \right. \\ & 0 \leq \rho_i^{l,t} \leq 1 - \epsilon, \\ & \sum_{i \in \mathcal{B}} p_i^{l,t}(x) = 1, \\ & \left. 0 \leq p_i^{l,t}(x) \leq 1, \forall i \in \mathcal{B}, \forall x \in \mathcal{L} \right\}. \end{aligned} \quad (6.8)$$

**Lemma 3.** The feasible sets  $f^{D,b}, f^{D,d}, f^{U,b}, f^{U,d}$  as well as the  $[f^{D,b}; f^{D,d}], [f^{U,b}; f^{U,d}], [f^{D,b}; f^{U,b}], [f^{D,b}; f^{D,d}; f^{U,b}; f^{U,d}]$ , are convex.

*Proof.* The proof for the feasible set  $f^{D,b}$  is presented in [63]. It can be easily adapted for the other cases, too (e.g., see [239]).  $\square$

Following [63] we extend the proposed objective to also include the DL dedicated traffic (see B.8-B.9). We introduce the parameter  $\theta \in [0, 1]$  that helps the operator weigh the importance of DL best effort versus DL dedicated traffic performance.  $\alpha^{D,b}$  ( $\alpha^{D,d}$ )  $\geq 0$  controls the amount of load balancing desired in the DL best-effort (dedicated) resources. Lets denote  $\alpha^{\mathbf{D}} = [\alpha^{D,b}; \alpha^{D,d}]$  and  $\rho^{\mathbf{D}} = [\rho^{D,b}; \rho^{D,d}]$ .

**Definition 6.3.2.** (Objective function for DL) Our objective is

$$\phi_{\alpha^{\mathbf{D}}}(\rho^{\mathbf{D}}) = \sum_{i \in \mathcal{B}} \theta \frac{(1 - \rho_i^{D,b})^{1-\alpha^{D,b}}}{\alpha^{D,b} - 1} + (1 - \theta) \frac{(1 - \rho_i^{D,d})^{1-\alpha^{D,d}}}{\alpha^{D,d} - 1}, \quad (6.9)$$

when  $\alpha^{D,d}, \alpha^{D,b} \neq 1$ . If  $\alpha^{D,b}$  (or,  $\alpha^{D,d}$ ) is equal to 1, the respective fraction must be replaced with  $\log(1 - \rho_i^{D,b})^{-1}$  (or,  $\log(1 - \rho_i^{D,d})^{-1}$ ).

As a final step, we want to further extend this objective to also capture the UL traffic performance and thus we introduce  $\tau \in [0, 1]$  to trade it off with DL. Lets assume that  $\alpha = [\alpha^{D,b}; \alpha^{D,d}; \alpha^{U,b}; \alpha^{U,d}]$  and  $\rho = [\rho^{D,b}; \rho^{D,d}; \rho^{U,b}; \rho^{U,d}]$ .

**Definition 6.3.3.** (Objective function for DL and UL) The objective that jointly considers DL and UL performance follows

$$\phi_{\alpha}(\rho) = \tau \cdot \phi_{\alpha^{\mathbf{D}}}(\rho^{\mathbf{D}}) + (1 - \tau) \cdot \phi_{\alpha^{\mathbf{U}}}(\rho^{\mathbf{U}}). \quad (6.10)$$

**Lemma 4.** The objective function  $\phi_{\alpha}(\rho)$  is convex.

*Proof.* Since  $\phi_{\alpha}(\rho)$  is a weighted non-negative sum of convex functions [63], convexity is preserved [240].  $\square$

**Definition 6.3.4.** (Optimization Problem) Our problem is

$$\min_{\rho} \{\phi_{\alpha}(\rho) \mid \rho \in \mathcal{F}\}. \quad (6.11)$$

**Lemma 5.** *This is a convex optimization problem.*

*Proof.* This is a convex optimization problem since the objective is convex on the convex feasible set  $\mathcal{F}$ .  $\square$

While  $\theta$  linearly weights the best effort versus dedicated flow performance (see Eq. 6.9), the impact of  $\alpha^b, \alpha^d$  is not obvious. We now discuss their impact on the system performance and refer to [63], [71] for the respective proofs.

- *Optimizing Spectral Efficiency:*  $\alpha^b = 0$  maximizes the average physical rate for best-effort flows (defined in B.3), whereas  $\alpha^d = 0$  maximizes the number of dedicated flows that can be served concurrently (defined in B.8). This is equivalent to maximizing the user  $SINR$  and thus the spectral efficiency.
- *Optimizing flow-level QoS:* if  $\alpha^b = 1$  the corresponding optimal rule tends to maximize the average user throughput. If  $\alpha^b = 2$  the per-flow delay is minimized since the objective for best effort flows corresponds to the delay of an M/G/1/PS system. If  $\alpha^d = 1$  the corresponding optimal rule becomes equivalent to the average *idle* dedicated servers in a k-loss system, and the actual blocking probability is minimized.
- *Optimizing Load-Balancing:* As  $\alpha^b \rightarrow \infty$ , the maximum BS utilization is minimized. Hence, load-balancing has the highest priority. Similarly for  $\alpha^d$  and balancing the loads in terms of dedicated traffic.

As a final note, observe that the association rules optimize the objective with respect to best effort traffic (e.g., maximally balanced loads) will usually be different than the association that optimizes for dedicated traffic. Since both dedicated and best effort traffic for each node must be served by the same BS, this creates a conflict that must be optimally resolved.

### 6.3.1 Optimal Split UL/DL User Association

In the Split UL/DL, since UL and DL traffic can be split at location  $x$ , the problem of optimal DL and UL association decouples into two independent problems. Specifically,  $\min_{\rho} \phi_{\alpha}(\rho) = \min_{\rho^D} \phi_{\alpha^D}(\rho^D) + \min_{\rho^U} \phi_{\alpha^U}(\rho^U)$ .

Note that all DL best-effort and dedicated flows at  $x$  have to be downloaded from the same BS, i.e.,  $p_i^D(x) = p_i^{D,b}(x) = p_i^{D,d}(x)$ . Also, all UL best-effort and dedicated should be offloaded to the same BS, so  $p_i^U(x) = p_i^{U,b}(x) = p_i^{U,d}(x)$ . Nevertheless, as explained in Split UL/DL scenarios  $p_i^D(x)$  and  $p_i^U(x)$  can take different values (see B.10). In that case, the optimal user association rules follow. In the remainder of this section, we focus on the downlink, and we omit the superscripts  $\{D, U\}$  to simplify notation. (We return to the Joint UL/DL association problem in the next subsection.)

**Theorem 3.1.** *If  $\rho^* = (\rho_1^*, \rho_2^*, \dots, \rho_{|\mathcal{B}|}^*)$  denotes the optimal load vector, the optimal association rule at  $x$  is*

$$i(x) = \arg \max_{i \in \mathcal{B}} \frac{(1 - \rho_i^{*b})^{\alpha^b} \cdot (1 - \rho_i^{*d})^{\alpha^d}}{e^b(x) \cdot (1 - \rho_i^{*d})^{\alpha^d} + e^d(x) \cdot (1 - \rho_i^{*b})^{\alpha^b}} \quad (6.12)$$

where  $e^b(x) = \frac{\theta z^D z^b}{\mu^b(x)c_i(x)}$  and  $e^d(x) = \frac{(1-\theta)z^D z^d}{\mu^d(x)k_i(x)}$ , can be seen as weight for the corresponding individual rules, and together forming a (weighted) harmonic mean of these.

*Proof.* We prove that the above association rule (Eq. 6.12) indeed minimizes the objective of Eq. (6.9). This is a convex optimization problem. Let  $\rho^* = [\rho^{*b}; \rho^{*d}]$  be the optimal solution of Problem (6.9). (We will relax this assumption in Section 6.3.3, as the optimal vector  $\rho^*$  is not necessarily known.) Hence, the first order optimality condition is that

$$\langle \nabla \phi_\alpha(\rho^*), \rho - \rho^* \rangle \geq 0 \quad (6.13)$$

for any  $\rho \in f$ . Let  $p(x)$  and  $p^*(x)$  be the associated routing probability vectors for  $\rho$  and  $\rho^*$ , respectively. Using the deterministic cell coverage generated by (6.12), the optimal association rule is given by:

$$p_i^*(x) = \mathbf{1} \left\{ i = \arg \max_{i \in \mathcal{B}} \frac{(1 - \rho_i^{*b})^{\alpha^b} \cdot (1 - \rho_i^{*d})^{\alpha^d}}{e^b(x) \cdot (1 - \rho_i^{*d})^{\alpha^d} + e^d(x) \cdot (1 - \rho_i^{*b})^{\alpha^b}} \right\}. \quad (6.14)$$

Then the inner product in Eq. (6.13) can be written as:

$$\begin{aligned} \langle \nabla \phi_\alpha(\rho^*), \Delta \rho^* \rangle &= \sum_{z=\{b,d\}} \frac{\partial \phi_\alpha}{\partial \rho_z}(\rho^*) (\rho_z - \rho_z^*) \\ &= \frac{\partial \phi_\alpha}{\partial \rho^b}(\rho^*) (\rho^b - \rho^{*b}) + \frac{\partial \phi_\alpha}{\partial \rho^d}(\rho^*) (\rho^d - \rho^{*d}) \\ &= \theta \sum_{i \in \mathcal{B}} \frac{1}{(1 - \rho_i^b)^{\alpha^b}} (\rho_i^b - \rho_i^{*b}) + (1 - \theta) \sum_{i \in \mathcal{B}} \frac{1}{(1 - \rho_i^d)^{\alpha^d}} (\rho_i^d - \rho_i^{*d}) \\ &= \sum_{i \in \mathcal{B}} \frac{\theta \int_L \rho_i^b(x) (p_i(x) - p_i^*(x)) dx}{(1 - \rho_i^b)^{\alpha^b}} + \frac{(1 - \theta) \int_L \rho_i^d(x) (p_i(x) - p_i^*(x)) dx}{(1 - \rho_i^d)^{\alpha^d}} \\ &= \int_L \lambda(x) \sum_{i \in \mathcal{B}} (p_i(x) - p_i^*(x)) \left[ \frac{e^b(x)(1 - \rho_i^{*d})^{\alpha^d} + e^d(x)(1 - \rho_i^{*b})^{\alpha^b}}{(1 - \rho_i^{*b})^{\alpha^b}(1 - \rho_i^{*d})^{\alpha^d}} \right] dx, \end{aligned} \quad (6.15)$$

where  $e^b(x) = \frac{\theta z^D z^b}{\mu^b(x)c_i(x)}$  and  $e^d(x) = \frac{(1-\theta)z^D z^d}{\mu^d(x)k_i(x)}$ . Note that,

$$\begin{aligned} \sum_{i \in \mathcal{B}} p_i(x) \frac{e^b(x)(1 - \rho_i^{*d})^{\alpha^d} + e^d(x)(1 - \rho_i^{*b})^{\alpha^b}}{(1 - \rho_i^{*b})^{\alpha^b}(1 - \rho_i^{*d})^{\alpha^d}} &\geq \\ \sum_{i \in \mathcal{B}} p_i^*(x) \frac{e^b(x)(1 - \rho_i^{*d})^{\alpha^d} + e^d(x)(1 - \rho_i^{*b})^{\alpha^b}}{(1 - \rho_i^{*b})^{\alpha^b}(1 - \rho_i^{*d})^{\alpha^d}} \end{aligned} \quad (6.16)$$

holds because  $p^*(x)$  in (6.14) is an indicator for the minimizer of  $\frac{e^b(x)(1 - \rho_i^{*d})^{\alpha^d} + e^d(x)(1 - \rho_i^{*b})^{\alpha^b}}{(1 - \rho_i^{*b})^{\alpha^b}(1 - \rho_i^{*d})^{\alpha^d}}$ . Hence, (6.13) holds.  $\square$

In the case of split UL/DL association, the above analysis can be applied *separately* on UL and DL traffic, and optimize UL and DL associations independently.

### 6.3.2 Optimal Joint UL/DL User Association

Current cellular networks (e.g. 3G/4G) suggest that a UE should be connected to a single BS for both UL and DL traffic [241], i.e.  $p_i^D(x) = p_i^U(x)$ . In that case, the two problems are coupled and the (single) optimal association rule at  $x$  shall appropriately weigh DL and UL performance as it follows.

**Theorem 3.2.** *If  $\rho^* = (\rho_1^*, \rho_2^*, \dots, \rho_{|\mathcal{B}|}^*)$  denotes the optimal load vector, and given the set of all flow-types  $\Omega = \{(D, b), (D, d), (U, b), (U, d)\}$ , the optimal rule at  $x$  is*

$$i(x) = \arg \max_{i \in \mathcal{B}} \frac{\prod_{c \in \Omega} ((1 - \rho^{*c})^{\alpha^c})}{\sum_{c \in \Omega} e^c(x) \prod_{l \in \Omega \neq c} ((1 - \rho^{*c})^{\alpha^c})}, \quad (6.17)$$

where  $e^{D,b}(x) = \tau \frac{\theta^D z^D z^b}{\mu^{D,b}(x) c_i^D(x)}$ ,  $e^{D,d}(x) = \tau \frac{(1-\theta^D) z^D z^d}{\mu^{D,d}(x) k_i^D(x)}$ ,  $e^{U,b}(x) = (1-\tau) \frac{\theta^U z^U z^b}{\mu^{U,b}(x) c_i^U(x)}$  and  $e^{U,d}(x) = (1-\tau) \frac{(1-\theta^U) z^U z^d}{\mu^{U,d}(x) k_i^U(x)}$  are the corresponding weight factors.

*Proof.* We refer the interested reader to [239].  $\square$

**Remark 1.** The above optimal rule derived in Eq. (6.17) suggests that in the *Joint UL/DL* scenario associated with objectives that potentially conflict with each other (due to the different flow type performances), it is optimal to associate a user with the BS that maximizes a weighted version of the *harmonic mean* of the individual association rules when considering each objective alone. To better understand this, we focus on a simple scenario with only DL and UL best-effort traffic. And assume the following BS options for a user: (BS A) offers 50Mbps DL and only 1Mbps UL; (BS B) 200Mbps DL and 0.5Mbps UL; (BS C) 20Mbps DL and 5Mbps UL. If we care about UL and DL performance equally (i.e.  $\tau = 0.5$ ), one might assume that the BS that maximizes the arithmetic mean (or arithmetic sum) of rates would be a fair choice (i.e. BS B). However, this would lead to rather poor UL performance. Maximizing the harmonic mean would lead to choosing (BS C) instead<sup>10</sup>. Additionally, note that in the case of *split UL/DL*, covered in Section 6.3.1, where each user is free to be associated with two different BSs for the DL and UL traffic offloading, DL traffic would be associated with (BS B), and UL traffic with (BS C) by maximizing the arithmetic mean (or, sum) of their throughputs<sup>11</sup>. These simple examples intuitively explain how split UL/DL impacts the user association policies, by allowing to independently optimize each objective. This also demonstrates why UL/DL split may perform considerably better than the joint association. We will further explore this in the simulation section.

Summarizing, we showed that our framework supports both best-effort and dedicated (in both DL and UL direction) traffic demand and we analytically found the corresponding optimal rules in different scenarios. Nevertheless, we highlight that our derived formulas allow to add more dimensions in our setup and *flexibly* derive

---

<sup>10</sup>While this simple example captures the main principle, the actual rule is more complex, as it weighs each objective with the complex factor  $e^l(x)$ .

<sup>11</sup>The usage of harmonic mean and arithmetic mean/sum appears in a number of physical examples, such as in the calculation of the total resistance in circuits where all resistances are set in series or in parallel.

the optimal rules without any analytical calculations (e.g., by using the arithmetic or harmonic mean maximization). For instance, consider a more modern offloading technique, where different downlink, or uplink, flow types are able to be offloaded to different BSs (e.g., per flow/QCI offloading) with conflicting aims. Using our model we can consider an additional respective  $\alpha$ -function for each flow type, and analytically optimize the complete objective as showed earlier.

### 6.3.3 Iterative Algorithm Achieves Optimality

The above derived association rules (see e.g., Eq. (6.12) or (6.17)) tell a UE at some location  $x$ , where to associate given the optimal BS load vector. However, the loads might not be optimal at that time (i.e. equal to  $\rho^*$ , e.g. see proof of Theorem 3.1). In the following, we show that starting within a feasible point  $\rho^{(0)}$ , iterative application of these rules, push the BS load vector to converge to the global optimal point. In such an iterative framework two parts are involved: the mobile device and the BS. Later, in our summarizing Remark 2, we show that our framework (in both over and under provisioned backhaul cases) (i) is applicable for distributed implementations, as well as (ii) satisfies various important properties (e.g., scalability).

*Mobile Device:* At the beginning of the  $k$  period, each new flow request coming from a user at location  $x$  simply selects the BS  $i$  that maximizes the derived rule (see e.g., Eq. (6.12)), using device-centric information (e.g. data rate) and BS broadcast control messages (e.g. *current* BS load).

*Base Station:* At each iteration  $k$ , each BS measures its utilization  $\tilde{\rho}^{(k)}$  after some required amount of time (e.g., see Eq. (6.5)). Then, based on the previous BS loads  $\rho^{(k)}$  and the latest measurement  $\rho(k)$ , the new BS vector  $\tilde{\rho}^{(k+1)}$  needed for the broadcast message in the next iteration would be

$$\tilde{\rho}^{(k+1)} = \beta^{(k)} \cdot \rho^{(k)} + (1 - \beta^{(k)}) \cdot \tilde{\rho}^{(k)}, \quad (6.18)$$

where  $\beta^{(k)} \in [0, 1]$  is an exponential-averaging parameter. Note that, in the Split UL/DL scenario the UL and DL loads can be independently updated, whereas in the other cases the corresponding loads should be updated using the same  $\beta^{(k)}$ .

This iterative algorithm converges to the global optimum, by requiring a simple modification of the proof found on the original algorithm [63]. The descent direction at  $x$ , improving the objective at the next  $k + 1$  iteration i.e. satisfying

$$\langle \nabla \phi_\alpha(\rho^{(k)}), \rho^{(k+1)} - \rho^{(k)} \rangle < 0, \quad (6.19)$$

is now provided from Eq. (6.12) under joint DL dedicated/DL best-effort association. This formula appropriately projects the direction under the constraint  $p_i^b(x) = p_i^d(x)$ ; as shown in the proof of Theorem 3.1. Similarly for the Joint UL/DL association, where the optimization problem does not decouple.

## 6.4 User-Association for Under-Provisioned Backhaul Networks

While the rules derived above, that try to reflect different performance tradeoffs, always lead to BS loads that are supported from the access network, they perhaps will not be supported from the backhaul link (or the corresponding backhaul link path) for that BS, since they ignore potential backhaul limitations. To that end, in this section we try to extensively consider the backhaul network and related limitations while extracting the optimal association rules, and include to our goals (i) that no backhaul link is congested, (ii) the impact of backhaul topology and capacity on key performance metrics. In order to better elucidate this problem at hand and without loss of generality, we focus on a simple scenario with *only best-effort traffic*. So, in the remainder of the section we drop the corresponding superscripts “b”, “d” to simplify notation.

One of the main challenges when attempting to consider these backhaul constraints is to maintain the user association policy *distributed*, mainly for scalability reasons. To that end, we chose to consider the backhaul constraints in the objective function as appropriate *penalty functions* [72]. First, this facilitates deriving a distributed implementation of the policy. Second, it allows us to treat the backhaul constraint as a “soft” constraint in general (that could gradually converge into a “hard” constraint and a strictly feasible solution, as we shall see).

### 6.4.1 Optimal Split UL/DL User Association

We follow the same presentation as the provisioned case, and start out discussion with the Split UL/DL case. As the association problem can be decoupled, in that case, into two independent problems, we focus on the optimal DL association problem, and we omit the superscripts  $\{D, U\}$ . We return to the Joint UL/DL case in the next section. To better illustrate our approach, we first consider a simple BH star topology, and then generalize for tree topologies.

#### Optimal User Association for Star BH Topology

In the following, since for star topologies there is exactly one backhaul link ( $j$ ) associated with each BS ( $i$ ), it is  $i = j$  (see C.1). Let  $\mathcal{I}(i)$  be an indicator variable showing if the  $i$ -th BH link is congested ( $\mathcal{I}(i)=1$ ) or not ( $\mathcal{I}(i)=0$ ) (see C.2)

$$\mathcal{I}(i) = \begin{cases} 0, & \text{when } \frac{\rho_i \tilde{c}_i}{C_h(i)} < 1 \\ 1, & \text{otherwise.} \end{cases} \quad (6.20)$$

Furthermore, the objective shall be extended to include the penalty functions for the backhaul constraints, as follows:

$$\phi_\alpha(\rho) = \sum_{i \in \mathcal{B}} \frac{(1 - \rho_i)^{1-\alpha}}{\alpha - 1} + \gamma \sum_{i \in \mathcal{B}_h} \mathcal{I}(i) \left( \frac{\rho_i \tilde{c}_i}{C_h(i)} - 1 \right)^2. \quad (6.21)$$

The first term is the standard  $\alpha$ -fair function for each BS  $i$ , already analyzed in the previous section. The second sum introduces a penalty for each backhaul link  $i$  whose

capacity is exceeded ( $\mathcal{I}(i) = 1$ ).  $\gamma$  could be chosen as a small constant, introducing a “soft” constraint for the backhaul links (i.e., backhaul capacity could be slightly exceeded, if this really improves access performance), or, preferably, *be iteratively adapted using increasing values, so as to converge to a “hard” constraint*. This penalty function is quadratic on the amount of excess load (quadratic penalty functions are often considered in convex optimization literature [242]). The corresponding optimal backhaul-aware rules follow.

**Theorem 4.1.** *If  $\rho^* = (\rho_1^*, \rho_2^*, \dots, \rho_{|\mathcal{B}|}^*)$  denotes the optimal load vector, the optimal association rule at location  $x$  is*

$$\arg \max_{i \in \mathcal{B}} c_i(x) \frac{(1 - \rho_i^*)^\alpha}{1 + 2\gamma \cdot (1 - \rho_i^*)^\alpha \cdot \tilde{c}_i \cdot \frac{\mathcal{I}(i)}{C_h(i)} \cdot \left( \frac{\rho_i^* \tilde{c}_i}{C_h(i)} - 1 \right)}. \quad (6.22)$$

*Proof.* We now prove that the above rule indeed minimizes the cost function of Eq. (6.21) within the penalty term  $\gamma$ . This is a convex optimization problem (quadratic penalty functions are convex due to the composition property of convexity [240]). Let  $\rho^*$  be the optimal solution of this problem. Again, it is adequate to check if

$$\langle \nabla \phi_\alpha(\rho^*), \rho - \rho^* \rangle \geq 0 \quad (6.23)$$

for all  $\rho \in f$ . Let  $p(x)$  and  $p^*(x)$  be the associated routing probability vectors for  $\rho$  and  $\rho^*$ , respectively. Using the deterministic cell coverage generated by (6.22), the optimal association rule is given by:

$$p_i^*(x) = \mathbf{1} \left\{ i = \arg \max_{i \in \mathcal{B}} \frac{c_i(x)(1 - \rho_i^*)^\alpha}{1 + 2\gamma \cdot (1 - \rho_i^*)^\alpha \cdot \tilde{c}_i \cdot \frac{\mathcal{I}(i)}{C_h(i)} \cdot \left( \frac{\rho_i^* \tilde{c}_i}{C_h(i)} - 1 \right)} \right\}. \quad (6.24)$$

Before proceeding to the calculation of the inner product, we analytically calculate the derivative of the corresponding cost function  $\phi_\alpha(\rho)$ , described in Eq. (6.21). The derivative is an  $i$ -th dimensional vector; the  $i$ -th element of which has value:

$$\nabla \phi_\alpha(\rho_i) = \begin{cases} (1 - \rho_i)^{-\alpha}, & \text{if } \frac{\rho_i \tilde{c}_i}{C_h(i)} \leq 1 \\ (1 - \rho_i)^{-\alpha} + \gamma \mathcal{I}(i) \frac{2\rho_i \tilde{c}_i^2 - 2\tilde{c}_i C_h(i)}{C_h(i)^2}, & \text{if } \frac{\rho_i \tilde{c}_i}{C_h(i)} \geq 1. \end{cases} \quad (6.25)$$

When  $\rho_i = \frac{C_h(i)}{\tilde{c}_i}$ , we work out explicitly from the definition to calculate the derivative. It is:

$$\lim_{\rho_i \rightarrow \frac{C_h(i)}{\tilde{c}_i}^+} \nabla \phi_\alpha(\rho_i) = \lim_{\rho_i \rightarrow \frac{C_h(i)}{\tilde{c}_i}^-} \nabla \phi_\alpha(\rho_i) = (1 - \rho_i)^{-\alpha}. \quad (6.26)$$

Summarizing, The  $i$ -th element of the derivative of the considered function can be written:

$$\nabla \phi_\alpha(\rho_i) = (1 - \rho_i)^{-\alpha} + \gamma \mathcal{I}(i) \frac{2\rho_i \tilde{c}_i^2 - 2\tilde{c}_i C_h(i)}{C_h(i)^2}. \quad (6.27)$$

Thus, the inner product defined in Eq. (6.23), becomes:

$$\begin{aligned}
\langle \nabla \phi_\alpha(\rho^*), \Delta \rho^* \rangle &= \sum_{i \in \mathcal{B}} \left\{ \frac{1}{(1 - \rho_i^*)^\alpha} + \gamma \mathcal{I}(i) \frac{2\rho_i^* \tilde{c}_i^2 - 2\tilde{c}_i C_h(i)}{C_h(i)^2} \right\} (\rho_i - \rho_i^*) \\
&= \sum_{i \in \mathcal{B}} \frac{1 + 2\gamma \mathcal{I}(i)(1 - \rho_i^*)^\alpha \frac{(\rho_i^* \tilde{c}_i^2 - \tilde{c}_i C_h(i))}{C_h(i)^2}}{(1 - \rho_i^*)^\alpha} \int_{\mathcal{L}} \rho_i(x) (p_i(x) - p_i^*(x)) dx \\
&= \int_L \frac{\lambda(x)}{\mu(x)} \sum_{i \in \mathcal{B}} \left( \frac{1 + 2\gamma(1 - \rho_i^*)^\alpha \tilde{c}_i \frac{\mathcal{I}(i)}{C_h(i)} \left( \frac{\rho_i^* \tilde{c}_i}{C_h(i)} - 1 \right)}{c_i(x)(1 - \rho_i^*)^\alpha} \right) (p_i(x) - p_i^*(x)) dx.
\end{aligned}$$

Note that,

$$\begin{aligned}
\sum_{i \in \mathcal{B}} p_i(x) \left\{ \frac{1 + 2\gamma(1 - \rho_i^*)^\alpha \tilde{c}_i \frac{\mathcal{I}(i)}{C_h(i)} \left( \frac{\rho_i^* \tilde{c}_i}{C_h(i)} - 1 \right)}{c_i(x)(1 - \rho_i^*)^\alpha} \right\} &\geq \\
\sum_{i \in \mathcal{B}} p_i^*(x) \left\{ \frac{1 + 2\gamma(1 - \rho_i^*)^\alpha \tilde{c}_i \frac{\mathcal{I}(i)}{C_h(i)} \left( \frac{\rho_i^* \tilde{c}_i}{C_h(i)} - 1 \right)}{c_i(x)(1 - \rho_i^*)^\alpha} \right\} &
\end{aligned}$$

holds because  $p_i^*(x)$  in (6.24) is an indicator for the minimizer of  $\frac{1+2\gamma\cdot(1-\rho_i^*)^\alpha\cdot\tilde{c}_i\cdot\frac{\mathcal{I}(i)}{C_h(i)}\cdot\left(\frac{\rho_i^*\tilde{c}_i}{C_h(i)}-1\right)}{c_i(x)(1-\rho_i^*)^\alpha}$ . Hence (6.23) holds.  $\square$

Regarding the optimal association rule of Eq. (6.22), we note that when the capacity constraint for the backhaul link  $i$  is not active (i.e.,  $\mathcal{I}(i) = 0$ , in provisioned BH networks), the above theorem states that the optimal association rule is the same as the one found in [63], or the one defined in Eq. (6.12) when  $\theta \rightarrow 1$ . However, when the backhaul link of BS  $i$  gets congested, a second term is added in the denominator that penalizes that BS making it less preferable to UEs at location  $i$ , even if the offered radio access rate  $c_i(x)$  is high, or the radio interface of  $i$  is not itself congested.

### Optimal User Association for Tree BH Topology

We now consider a more complex backhaul scenario, where a single backhaul link might route traffic from multiple BSs, and the traffic of a single BS might be routed over multiple backhaul links (multi-hop path) towards the eNB.  $\mathcal{I}(j)$  is now

$$\mathcal{I}(j) = \begin{cases} 0, & \text{when } \frac{\sum_{i \in \mathcal{B}(j)} \rho_i \tilde{c}_i}{C_h(j)} < 1 \\ 1, & \text{otherwise.} \end{cases} \quad (6.28)$$

Similarly, the backhaul constraint must be modified accordingly, and the cost function becomes

$$\phi_\alpha(\rho) = \sum_{i \in \mathcal{B}} \frac{(1 - \rho_i)^{1-\alpha}}{\alpha - 1} + \gamma \sum_{j \in \mathcal{B}_h} \mathcal{I}(j) \left( \frac{\sum_{i \in \mathcal{B}(j)} \rho_i \tilde{c}_i}{C_h(j)} - 1 \right)^2. \quad (6.29)$$

**Theorem 4.2.** If  $\rho^* = (\rho_1^*, \rho_2^*, \dots, \rho_{||\mathcal{B}||}^*)$  denotes the optimal load vector, the optimal association rule at location  $x$  is

$$\arg \max_{i \in \mathcal{B}} c_i(x) \frac{(1 - \rho_i^*)^\alpha}{1 + 2\gamma \cdot (1 - \rho_i^*)^\alpha \cdot \tilde{c}_i \sum_{j \in \mathcal{B}_h(i)} \frac{\mathcal{I}(j)}{C_h(j)} \cdot \left( \frac{\sum_{k \in \mathcal{B}(j)} \rho_k^* \tilde{c}_k}{C_h(j)} - 1 \right)} \quad (6.30)$$

*Proof.* The steps of this proof are similar to the star case, so we present here directly the corresponding inner product.

$$\begin{aligned} & \langle \nabla \phi_\alpha(\rho^*), \Delta \rho^* \rangle = \\ &= \sum_{i \in \mathcal{B}} \left\{ \frac{1}{(1 - \rho_i^*)^\alpha} + 2\gamma \sum_{j \in \mathcal{B}_h(i)} \mathcal{I}(j) \left[ \frac{\sum_{k \in \mathcal{B}(j)} \rho_k^* \tilde{c}_k}{C_h(j)^2} \tilde{c}_i - \frac{\tilde{c}_i}{C_h(j)} \right] \right\} (\rho_i - \rho_i^*) \\ & \quad \cdot \int_{\mathcal{L}} \rho_i(x) (p_i(x) - p_i^*(x)) dx = \\ &= \int_L \frac{\lambda(x)}{\mu(x)} \sum_{i \in \mathcal{B}} \left( \frac{1 + 2\gamma(1 - \rho_i^*)^\alpha \tilde{c}_i \sum_{j \in \mathcal{B}_h(i)} \frac{\mathcal{I}(j)}{C_h(j)} \cdot \left( \frac{\sum_{k \in \mathcal{B}(j)} \rho_k^* \tilde{c}_k}{C_h(j)} - 1 \right)}{c_i(x)(1 - \rho_i^*)^\alpha} \right) \\ & \quad \cdot (p_i(x) - p_i^*(x)) dx \geq 0, \end{aligned} \quad (6.31)$$

due to the corresponding maximizer  $p_i^*(x)$  derived from (6.30).  $\square$

There are a number of interesting differences between the optimal association rules of star and tree topology. First, the penalty term in the denominator of the rule (Eq. (6.30)) now considers the whole backhaul path  $\mathcal{B}_h(i)$  that traffic from BS  $i$  traverses, and adds a penalty for *every* link along that path that is congested (outer sum in the denominator). This observation provides some support for the number of BH hops heuristic proposed in [243, 227]. However, our analysis also suggests that it can be suboptimal, as a path with few hops might still include one or more congested links, and provides the optimal way to weigh in the amount of congestion on each link.

Second, the actual congestion on each backhaul link  $j$  is now not only dependent on the load of the candidate BS  $i$ , but also on other BSs whose load is routed over  $j$ . Hence, a BS  $i$  which would otherwise be a good candidate for traffic at location  $x$ , might still be penalized and not selected, even if it does not impose itself a large load on a backhaul link  $j$ . This is because *other* BSs sharing the same backhaul link might be heavily loaded or congested.

In the case of split UL/DL traffic, the above analysis can be applied *separately* on UL and DL traffic, and optimize UL and DL associations independently. Finally, although we have provided separate solutions for star and tree topologies, to better illustrate our approach, the optimal rule for the tree topology is generic, and includes star topologies as well.

### 6.4.2 Optimal Joint UL/DL User Association

In the Joint UL/DL case, we remind the reader that each user at  $x$  shall be associated with one BS for both UL and DL traffic, i.e.  $p_i^D(x) = p_i^U(x) \forall i \in \mathcal{B}$ , as discussed in Section 6.3.2. The penalty function should now consider both uplink and downlink capacity being exceeded on each backhaul link, and the objective function  $\phi_\alpha(\rho)$  becomes

$$\tau\phi_{\alpha^D}(\rho^D) + (1 - \tau)\phi_{\alpha^U}(\rho^U) + \gamma \sum_{k \in \{D, U\}} \sum_{j \in \mathcal{B}_h} \mathcal{I}^k(j) \left( \frac{\sum_{i \in \mathcal{B}(j)} \rho_i^k \tilde{c}_i^k}{C_h^k(j)} - 1 \right)^2 \quad (6.32)$$

Here, we present our results directly for the general case of tree backhaul topology, and we remind the reader that this is applicable to star backhaul topologies as well.

**Theorem 4.3.** *If  $\rho^* = (\rho_1^*, \rho_2^*, \dots, \rho_{|\mathcal{B}|}^*)$  denotes the optimal load vector, the optimal user-association rule at location  $x$  is*

$$i(x) = \arg \max_{i \in \mathcal{B}} \frac{(1 - \rho_i^{*D})^{\alpha^D} \cdot (1 - \rho_i^{*U})^{\alpha^U}}{e^D(x) \cdot (1 - \rho_i^{*U})^{\alpha^U} + e^U(x) \cdot (1 - \rho_i^{*D})^{\alpha^D}}, \quad (6.33)$$

where if  $g^D = \tau, g^U = 1 - \tau$ , then for  $l \in \{D, U\}$ :

$$e^l(x) = \frac{z^l \left( g^l + 2\gamma (1 - \rho_i^{*l})^{\alpha^l} \sum_{j \in \mathcal{B}_h(i)} \frac{\mathcal{I}^l(j)}{C_h^l(j)} \left( \frac{\sum_{k \in \mathcal{B}(j)} \rho_k^{*l} \tilde{c}_k^l}{C_h^l(j)} - 1 \right) \right)}{\mu^l(x) c_i^l(x)}.$$

*Proof.* We refer the interested reader to [239].  $\square$

The penalty function for the backhaul network is simply the sum of the respective penalty functions for UL and DL, described in Theorem 4.1. However, despite the similarities of the cost functions, as we can see, the resulting association policy in the Joint UL/DL case is more complex.

### 6.4.3 Iterative Algorithm Achieves Optimality

Our proposed iterative framework stays similar in nature as the one described in Section (6.3.3).

We now focus on the penalty method and the emerging convergence to the global optimal point. Using the proposed (quadratic) penalty functions we now solve a sequence of unconstrained problems (e.g. see Eq. (6.21)) with *monotonically increasing values* of  $\gamma$  at each iteration (chosen so that the solution to the next problem is “close” to the previous one; otherwise we risk getting stuck in steep valleys). Thus, let  $\rho^{(k)} = \{\rho^{(0)}, \rho^{(1)}, \dots, \rho^{(k)}\}$  be a sequence generated within iterations, where  $\rho^{(l)}, 0 \leq l \leq k$ , is the global minimum of  $\phi(\rho)$  with penalty constant  $\gamma^{(l)}$  at the  $l$  iteration. Then any

limit point  $\rho^{(k)}$  of the sequence is a solution to this problem. This is a well known result for such convex objectives proposed from Luenberger in 1984 [244].

**Remark 2.** In this remark we want to underline various important properties of our derived rules (regarding both Sections 6.3 and 6.4). Firstly, no matter the number of traffic types and the backhaul topology design, all derived rules are “device centric”, i.e., the user is able to optimally select where to associate based on own measurements (e.g.  $c_i(x)$  in Eq. (6.22)) and BS broadcast information that jointly capture its access and backhaul performance (e.g., in Eq. (6.22) this broadcast corresponds to the fraction seen). (For dedicated traffic inclusion in backhaul-limited scenarios see e.g. [245]).<sup>12</sup> This allows for distributed implementations that do not require any controller to govern the BSs and the UEs with access to all the necessary information, e.g., as in [249], [250].<sup>13</sup> Our rules also satisfy the following important properties: *scalable*, (constant amount of BS broadcast messages irrespective of the number of users, backhaul topology), *simple* (constant complexity of the rule with respect to the number of BSs), and offers *flexible performance* by properly choosing the  $\alpha$  values in the respective  $\alpha$ -fair cost functions.

## 6.5 Simulations

In this section we briefly present some numerical results and discuss related insights. We consider a  $2 \times 2 \text{ km}^2$  area. Figure 6.3(a) shows a color-coded map of the heterogeneous traffic demand  $\lambda(x)$  (*flows/hour per unit area*) (blue implying low traffic and red high), with 2 hotspots. We assume that this area is covered by two macro BSs and eight SCs. The macro BSs that are shown with asterisks are numbered from 1-2, and the SCs that are shown with triangles are numbered from 3-10, as we can see in Fig. 6.3(b)-(c), Fig. 6.4, and in Fig. 6.5. We also consider standard parameters as adopted in 3GPP [251], listed in Table 6.2<sup>14</sup>. If not explicitly mentioned, we assume  $\theta^D = \theta^U = \tau = 0.5$ , and the split UL/DL scenario as default.

Before proceeding, we need to setup a metric to evaluate load balancing (or, utilization) efficiency. Thus, we introduce the Mean Squared Error ( $MSE^{D,b}$ ), between the DL best-effort utilization of different BSs, normalized to 1. We define the DL load balancing metric for best-effort traffic to be  $1 - MSE^{D,b}$ , that increases on the amount of load balancing<sup>15</sup>. Similarly, we can define them for the other three cases  $1 - MSE^{D,d}$ ,  $1 - MSE^{U,b}$ ,  $1 - MSE^{U,d}$ .

---

<sup>12</sup>Such broadcast quantities can be easily integrated through the newly proposed Access Network Discovery and Selection Function (ANDSF) mechanism [246], or in the absolute/dedicated priority list mechanisms of LTE [247], or in IEEE 802.16m [248].

<sup>13</sup>Such centralized schemes usually imply a high (a) burden of collecting all the necessary information to a central location (usually implemented in a server deep in the core network), processing, and redistributing to every user, (b) computational complexity that increases exponentially in the network size.

<sup>14</sup>As for (i) the sizes and ratios of different flows, (ii) splitting parameters, we can use different values in order to capture different simulation scenarios, and derive similar results.

<sup>15</sup>We should note that different load balancing metrics could have been used, e.g. the *maximum*, *median* and *minimum* BS load; however, we chose to use MSE since it facilitates the visualization of the network performance.

Table 6.2: Simulation Parameters

Parameter	Variable	Value
Transm. Power of eNB/ SC/ UE	$P_{eNB}/P_{SC}/P_{UE}$	43/24/12 dBm
BS Bandwidth for DL, UL	$w/W$	10/10 MHz
Noise Power Density	$N_0$	-174 dBm/Hz
Splitting parameter for DL, UL	$\zeta_i^D, \zeta_i^U$	0.5/0.5
Average DL/UL flow sizes	$\frac{1}{\mu^{D,b}} / \frac{1}{\mu^{U,b}}$	100/20 Kbytes
Average DL/UL flow demands	$B^D(x)/B^U(x)$	512, 128 kbps
Different flow ratios	$z^b, z^D$	0.3,0.6

### 6.5.1 Provisioned Backhaul

We now focus on the case of provisioned backhaul as considered in Section 6.3 and investigate the involved tradeoffs both qualitatively and quantitatively. We will present the impact of our proposed association rules via coverage snapshots to show how users associate in the considered network, while we will also provide values for related performance metrics that complete our study numerically.

**Spectral efficiency vs. Load balancing.** Figure 6.3(b) outlines the optimal DL user-associations if  $\alpha^{D,b} = \alpha^{D,d} = 0$ , i.e., when *spectral efficiency* is maximized. Thus, each UE at  $x$  is attached to the BS that offers the *highest DL SINR* and promises higher DL physical rate for best effort flows  $c_i^{D,b}(x)$ , and more “dedicated” servers  $k_i^D(x)$ ; i.e. most of UEs are attached to macro BSs due to their high power transmission, and fewer to SCs, forming small circles around them. Consequently, macro-cells are overloaded and load imbalance within the cells is sharpened (decreased  $1 - MSE^{D,b}, 1 - MSE^{D,d}$ ; see line 1 of Table 6.3). However, in Fig. 6.3(c) we emphasize the *load-balancing* efficiency and set  $\alpha^{D,b} = \alpha^{D,d} = 10$ . Now, most SCs vastly increase their coverage area in order to offload the overloaded macro BSs (e.g., BSs 6, 8, 10); “heavily” loaded (due to the hotspots) BSs, roughly maintain the same coverage (BS 4 and 7). Thus load balancing is improved, at the cost of  $E[c^{D,b}], E[k^D]$  (see line 2 of Table 6.3). For further implications of  $\alpha$  parameters we refer the reader to [63].

**Best-effort versus dedicated traffic performance.** Although in the previous scenarios the best-effort- and dedicated- related traffic rules (represented from  $\alpha^{D,b}, \alpha^{D,d}$ ) are aligned, one could ask how would two conflicting optimization objectives affect our network? The answer lays in the usage of  $\theta^D$ , that judges which objective carries more importance. E.g., an operator has two main goals: (i) to maximize the average number of servers for “dedicated” traffic captured by  $E[k^D]$  (set  $\alpha^{D,d} = 0$ ), (ii) to better balance the utilization of best-effort resources between BSs (set  $\alpha^{D,b} = 10$ ). As shown in Fig. 6.3(d), if  $\theta \rightarrow 0$   $E[k^D]$  is maximized, whereas as  $\theta \rightarrow 1$ ,  $1 - MSE^{D,b}$  (DL best-effort load balancing) is optimized, and each objective comes at the price of the other.

**DL vs. UL traffic performance** is considered in Figure 6.3(b), 6.4(a)-6.4(b), with respective numerical performance metrics in Table 6.4. The first two figures depict the DL and UL optimal associations, in case of split UL/DL, for each user at  $x$ . However,

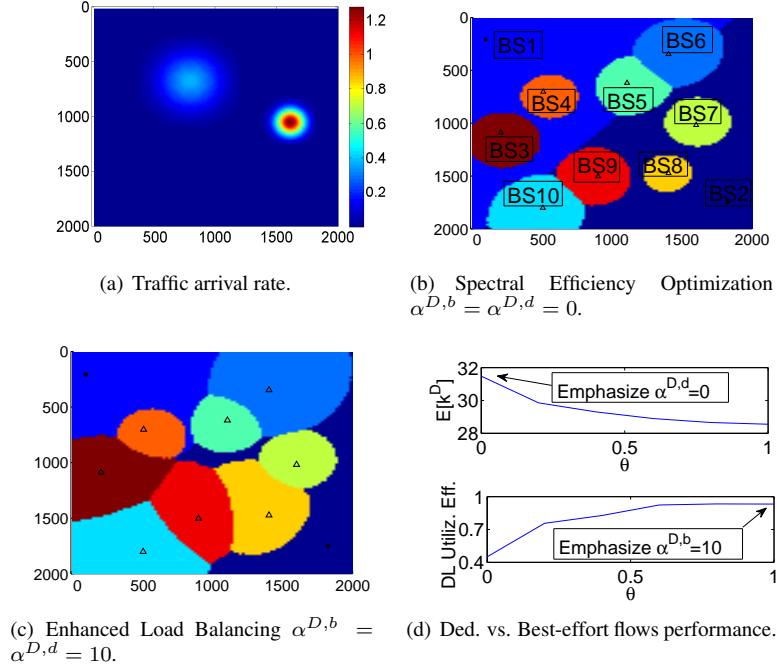


Figure 6.3: DL Optimal user-associations (Spectral efficiency vs. Load balancing and best-effort vs. ded. traffic performance)

if split is not available from the operator point of view, we have to weight whether the DL or UL performance is more important while selecting a *single* BS for Joint UL/DL association, using parameter  $\tau$ . To that end, Figure 6.3(b) (also) outlines the optimal associations in the Joint UL/DL case if the whole emphasis is on the *DL performance* ( $\tau = 1$ ): this hurts the UL performance due to the asymmetric transmission powers of the UEs and BSs (see line 1 of Table 6.4). In Fig. 6.4(a) the emphasis is moved on the *UL performance* ( $\tau = 0$ ), and each UE is attached to the nearest BS, in order to minimize the path loss [236] and enhance the UL performance; this hurts its DL performance though (see line 3 of Table 6.4). Finally, Fig. 6.4(b) shows the optimal coverage areas when one assigns equal importance to the UL and DL performance (i.e.  $\tau = 0.5$ ): this moderates both DL and UL performance (line 2 of Table 6.4). This also corroborates the notion that split is able to simultaneously optimize UL and DL performances, as already discussed in theory.

### 6.5.2 Under-provisioned Backhaul

We now continue with some backhaul-limited network scenarios. We remind to the reader that our focus is on the backhaul links *between the macro cells and SCs* (for

Table 6.3: Numerical values for Figure 6.3.

	<b>Rates and Servers</b>		<b>Load Balancing</b>	
	$E[c^{D,b}]$ (Mbps)	$E[k^D]$	$1-MSE^{D,b}$	$1-MSE^{D,d}$
Fig. 6.3(b)	16.3	32	0.77	0.78
Fig. 6.3(c)	14.3	27	0.96	0.995

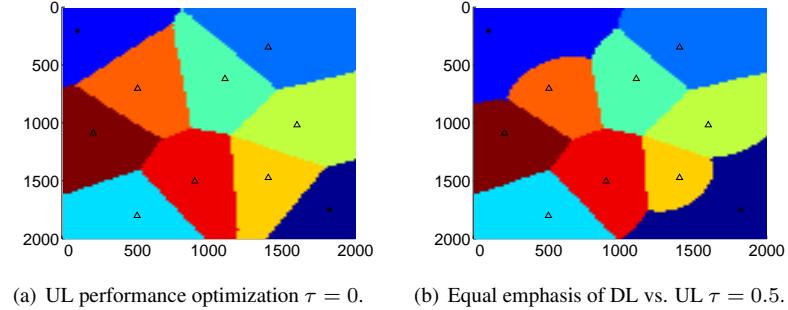


Figure 6.4: Optimal user-associations (DL vs. UL traffic performance)

simplicity we assume provisioned links between the macro cells and core network). As already discussed in assumption C.1, we investigate two different backhaul topology families: (i) “star” topologies (single-hop paths), (ii) “tree” topologies (with multi-hop paths), along with two backhaul links types: *wired and wireless*<sup>16</sup>. Our aim is to evaluate the derived association rules described in Section 6.4 for different *under-provisioned* scenarios, by fixing the aforementioned trade-offs related to the traffic differentiation as it follows:  $\theta^D = \theta^U = 1$  (we only focus on the best-effort flows by dropping the superscripts “b” and “d”), and  $\alpha^D = \alpha^U = 1$  (throughput optimal values). Also, we assume *fixed* backhaul routing paths, pre-established with traditional Layer 2 routing, that the BH capacities on the DL and UL are the same (i.e.  $C_h^D(j) = C_h^U(j) = C_h, \forall j \in \mathcal{B}_h$ ), and if not explicitly mentioned we assume them to be equal to 400Mbps. We maintain this assumption to facilitate our discussion, although our framework works for heterogeneous backhaul links and UL/DL capacities (see C.2). We finally assumed that  $\tilde{c}_i$  is the 80% of the maximum user capacity associated with BS  $i$  (we have also tried higher values for more “conservative” scenarios, and also lower for more “aggressive” with similar conclusions).

Before proceeding, we need to make an assumption about the backhaul link capacities. In case of *wired* backhaul links, we assume that the peak backhaul capacity  $C_h$  is always guaranteed. For *wireless* backhaul links we adopt a simple model associating peak backhaul capacity to distance: if the length of the  $i$ -th link is  $r_i$ , the peak capacity

---

<sup>16</sup>Note that copper and fiber access are the key technologies for wired backhaul links, and microWave and millimeter-wave P2P or P2MP access are the counterpart for the wireless backhaul links [252].

Table 6.4: Numerical values for Figure 6.4.

	<b>DL performance</b>		<b>UL performance</b>	
	$E[c^{D,b}]$ (Mbps)	$E[k^D]$	$E[c^{U,b}]$ (Mbps)	$E[k^U]$
Fig. 6.3(b)	16.3	32	2.3	18
Fig. 6.4(b)	14.7	28	3	24
Fig. 6.4(a)	13.3	26	3.6	28

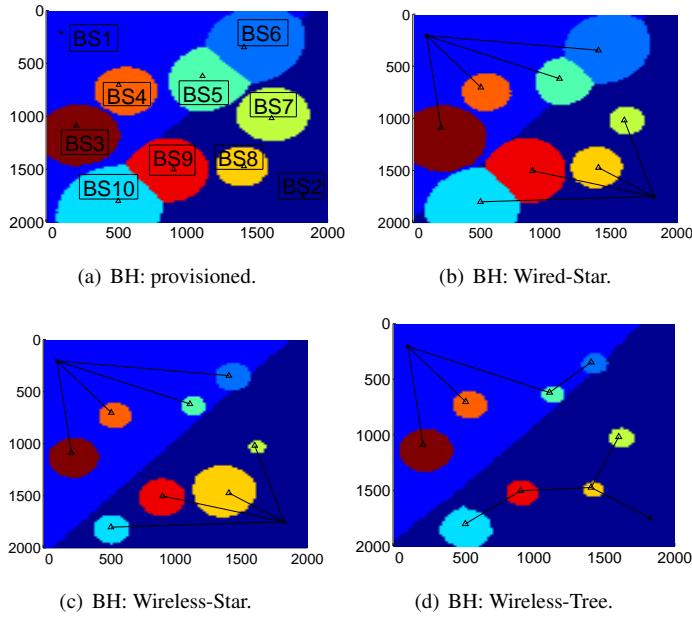


Figure 6.5: DL optimal associations in different scenarios.

drops as:

$$d(r_i) = \begin{cases} 1, & r_i \leq r_0 \\ \left(\frac{r_0}{r_i}\right)^n, & \text{otherwise,} \end{cases} \quad (6.34)$$

where  $r_0$  is some threshold range within which the maximal rate is obtained (e.g. Line-of-Sight), and  $n$  is the attenuation factor. Hence, the available capacity drops to  $d(r_i)C_h(j)$  ( $\leq C_h(j)$ ). For our simulations, we assumed that  $r_0 = 200m$ , and  $n = 3$ . While the above model is perhaps oversimplifying, our main goal is to simply include a generic model for the propagation related impact on wireless backhaul, compared to wired, without getting into the details of specific backhaul implementations. For detailed path loss models for different backhaul technologies, we refer the interested reader to [231].

**Coverage Snapshots.** In Fig. 6.5(a) we depict the optimal DL user-associations for provisioned backhaul network with respect to the traffic arrival rates shown in

Fig. 6.3(a). Compared to the associations showed in Figure 6.3(b) where  $\alpha^D = \alpha^U = 0$ , we note that now some SCs have slightly increased coverage area, in order to improve the mean user throughput [63].

In the following, we focus on different *under-provisioned* backhaul scenarios, and study the DL associations (similar behavior in the UL; we refer the interested reader to [239] for them). In Fig. 6.5(b) we adopt a *wired-star* backhaul topology, where SCs shrink their coverage areas, by handing-over users to other BSs, in order to offload the corresponding (under-provisioned) backhaul links; this phenomenon becomes more intense in the “hot-spot” areas (e.g., BS7 have vastly decreased their coverage areas) due to the higher traffic demand. Similarly, in Fig. 6.5(c), we assume a *wireless-star* backhaul topology, where SCs further decrease their coverage areas, due to the higher backhaul capacity loss caused from the long wireless links (see Eq.(6.34)).

In Fig. 6.5(d) we adopt a *wireless-tree* topology, where some SCs are required to carry also traffic of other SCs, and end up more congested. As a result, most SCs further decrease their coverage area, compared to the star-wireless topology. However, BS7 and BS10 enlarge their coverage areas, compared to the star case. This occurs because these SCs are far from the eNB, and multi-hop topology allows them to route their traffic over shorter wireless links with smaller capacity losses, compared to the star case (Fig. 6.5(c)). Hence, there are two main factors affecting the coverage areas in such wireless backhaul networks: (*topology*) each BS-load might traverse through multi-hop backhaul paths, by “wasting” resources from more than one backhaul links (drawback for tree topologies); (*location*) the higher the  $\eta, r_0$  the worse the capacity loss “wastage” over a dedicated direct backhaul link (drawback for star topologies that require longer links).

As backhaul networks become increasingly complex, e.g. “mesh” topologies, each BS has *multiple* possible routing paths to follow, beyond what is shown in the figures (we remind the reader that the above shown topologies are simply the given spanning routing trees). The above observations thus underline the shortcomings of predetermined, Layer 2 (L2) backhaul routing mechanisms, and call for a *joint* optimization of user-association on the radio access network along with dynamic, Layer 3 (L3) backhaul routing.

**Under-provisioning impact on user performance.** Figure 6.6(a), 6.6(b) depict the *average* DL and UL user throughputs, as a function of the backhaul capacity constraint  $C_h$ , on different scenarios. Generally, as  $C_h$  drops, the mean throughputs are decreased, since users are handed over to (potentially far-away) macro BSs, causing performance degradation. Interestingly, the *slope of the dropping rate* becomes more steep for lower values of  $C_h$ , due to the logarithmic capacity formula chosen in assumption (B.2). Also, as  $C_h$  increases, the average throughputs “converge” to the value corresponding to a provisioned backhaul network. Note that the average UL throughput converges more quickly, compared to the DL. This happens due to the asymmetry between the DL and UL traffic demand on the radio access network: the UL one is much lower, mainly due to the asymmetry between the transmission powers of BSs and UEs, as well as different file sizes assumed in each direction. Beyond this point, the UL backhaul resources will be underutilized. This calls for a *flexible* TDD duplexing scheme, that will dynamically distribute the backhaul resources accordingly, for example by giving more backhaul resources to DL when the UL demand is already

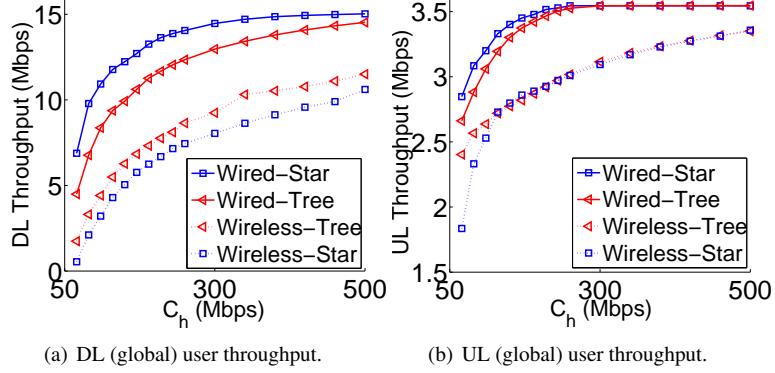


Figure 6.6: Mean throughputs overall all users in the network.

satisfied (e.g. the eIMTA scheme [253]). Finally, in the wired case, star topology is always slightly better than the tree, whereas in the wireless the opposite, as explained earlier.

Table 6.5: Mean throughp. for handed-over users (in Mbps).

Topology	$C_h = 50$	250	500 (Mbps)
DL / UL thr.: Star-Wired	1.1 / 0.2	3.1 / 1.6	4.1 / X
DL / UL thr.: Tree-Wired	0.6 / 0.1	2.4 / 0.7	3.2 / X
DL / UL thr.: Tree-Wirel.	0.2 / 0.03	1.7 / 0.07	2.1 / 0.15
DL / UL thr.: Star-Wirel.	0.1 / 0.001	1.4 / 0.05	1.7 / 0.02

One could notice that user throughputs drop slightly on the  $C_h$  constraint, e.g. in a wired-star topology if  $C_h$  drops 500 → 50 Mbps (10 times), the mean user throughput only drops 15 → 6 Mbps (~ 3 times). This is due to the fact that, under-provisioned backhaul links do not affect the whole network, but specific groups of users associated with the cells that suffer from low backhaul capacity. To better illustrate this, in Table 6.5 we show the average throughput of the *handed-over users*, as a function of  $C_h$ . Indeed, their performance is severely affected: for the same scenario, their DL throughput drops all the way to 1.1 Mbps (~ 15 times). (In scenarios with no handovers, we mark the respective table entry with an X.)

**Under-provisioning impact on Network Performance.** Turning our attention to network-related performance, Fig. 6.7(a) considers spectral efficiency ( $bit/s/Hz$ ), *normalized by the maximum corresponding value when the network is provisioned*. Load-balancing (“utilization”) efficiency is further considered in Fig. 6.7(b) in terms of the MSE metric, described earlier. Both efficiencies converge to 1 as the network gets provisioned. Low  $C_h$  values will push users to handover to far-away BSs, and this will potentially decrease their  $SINR$  (spectral efficiency decrease), and create steep differences between BSs loads, e.g. by congesting macro BSs and under-utilizing the SCs (load balancing decrease). Note that, the joint degradation of these performances

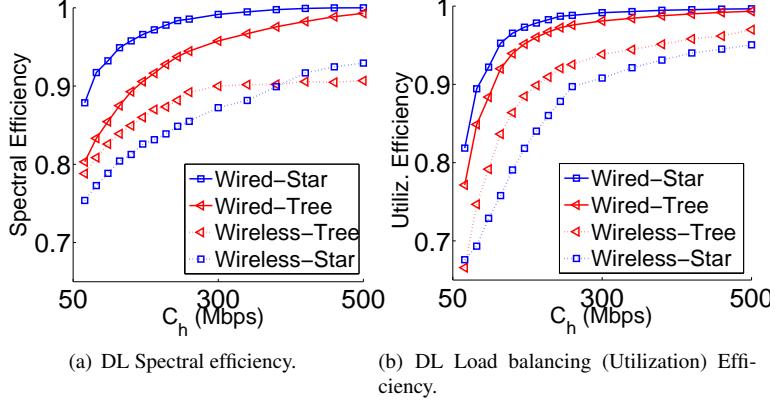


Figure 6.7: Downlink Network Efficiencies (normalized).

also impacts user performance negatively (e.g. user throughput), as explained in Section B.7.

Note that regarding spectral efficiency (for the wireless scenario), there is a crucial point that judges which topology is better than the other in different under-provisioned cases (that is actually the point where the two curves meet). Obviously, this point, that is highly affected by the network topology and the given set of assumptions, can significantly vary between different performance metrics (e.g., in terms of load balancing and user throughput). Note that it is not possible to analytically derive this cross point, as this depends on the value of the objective function of the respective optimization problem. As a result, the actual cross point can only be found via sensitivity analysis. Nevertheless, we provide here the two main factors that affect this tradeoff between star and tree topology, to provide some qualitative insights. These factors are: (i) *path loss characteristics*: the higher the path loss (e.g. higher  $\eta$ ) the worse the capacity loss on a long direct backhaul link required for a star topology, favoring multi-hop connectivity (i.e. a tree topology). (ii) *the number of hops*: the higher the number of backhaul hops the higher the total backhaul resources consumed “per BS” or “per bit of radio access”, which might disfavor a tree topology.

Table 6.6: UL/DL Split Vs. Joint-association Improvements

Performance	$\tau = 0$	$\tau = 0.5$	$\tau = 1$
DL / UL Throughput	6% / 32%	4% / 35%	0% / 37%
DL / UL Spectr. Eff.	4% / 29%	3% / 31%	0% / 33%
DL / UL Uiliz. Eff.	7% / 34%	4% / 38%	0% / 41%

**Split UL/DL impact.** As discussed earlier, while split is able to optimize the DL and UL performance, *simultaneously*, Joint UL/DL association is incapable of this parallel optimization and using  $0 \leq \tau \leq 1$  we can trade-off which dimension carries more importance. Table 6.6 illustrates the *performance improvements* that split promises

over the Joint UL/DL association, in terms of various metrics, for various  $\tau$  when backhaul is underprovisioned. We underline that split enhances the UL performance considerably, e.g. the average UL throughput is increased up to 37%. This is due to the *dependency* that Joint UL/DL generates between the DL and UL associations in the access network, that often makes the DL the bottleneck in the backhaul (due to aforementioned asymmetry between the peak access rates). Thus, DL will often “preempt” the backhaul constraint, and potentially (i) leave some UL resources unused, (ii) cause UL performance degradation.

*Comparison with existing work.* We now compare our proposed algorithm with two others user-association schemes; specifically, we run them in our considered network topology for the wired backhaul case (similar behavior for the wireless). In Table 6.7 we depict the performance decrease compared to our scheme in terms of: Spectral Efficiency (SE), User Performance (UP) (in terms of idle dedicated servers for dedicated traffic and throughput for best-effort) and Load Balancing (LB). We have assumed that all BSs associated with a BH link that is congested decrease their SE and UP proportionally to the amount of congestion by illustrating the effective corresponding performance.

Firstly, we investigate the algorithm proposed from Mesodiakaki et al. that focuses on dedicated traffic demand [227, 243]. This algorithm involves two stages, each of which independently considers access and backhaul performance: in the first, a subset of cells for each users’ association is selected as candidates separately for DL and UL based on the radio access conditions (i.e., the DL data rate or the UL path loss without considering the BS load). Then, the best one among them is selected based on the BH conditions: *the BS with the fewest backhaul hops to the core is selected as optimal*. However, such a simplistic criterion can lead to rather suboptimal performance as explained in Section 6.4.1. Specifically, in our case, in both DL and UL, we see that effective SE and UP are decreased since the path with the fewest hops might include congested backhaul links (obviously, this is more intense in the tree topology where multiple BSs share the resources of a single backhaul link). LB is also slightly hurt (by also hurting UP) since the reference algorithm ignores the BS loads.

Table 6.7: Comparison with existing work.

<b>Algorithm</b>	<b>DL performance</b>			<b>UL performance</b>		
	SE	UP	LB	SE	UP	LB
[227, 243] Star-Wired	1.4	1.3	1.2	1.8	2.1	1.3
[227, 243] Tree-Wired	1.5	1.4	1.2	2.2	3.2	1.4
[228] Star-Wired	5.3	5.6	2.1	3.9	5.1	1.1
[228] Tree-Wired	5.7	5.9	2.4	4.1	5.4	1.2

Secondly, we consider the user-association algorithm proposed by Domenico et al. for DL best-effort flows [228]. There, the ergodic capacity is attempted to be maximized through an iterative algorithm: at each step every user changes its association if the corresponding gain in terms of ergodic capacity is positive. Nevertheless, such an iterative algorithm, strongly dependent on the initial condition as well as the corre-

sponding step directions, does not necessarily converge to the best point by potentially getting stuck in subpar conditions. Also, this algorithm does not take into account heterogeneous traffic demand, and thus BS loads. UL traffic and BH tree topology are not considered, so we (i) assume that UL associations are identical as DL (as in Joint UL/DL, see B.10), and (ii) extend the proposed resource allocation policy to evenly split the available backhaul resources of a link in tree topologies. Simulation results show significant performance degradation since in the (suboptimal) converged point (i) some users end up being attached to far-away BSs, and (ii) some BSs are driven to congestion while attempting to improve ergodic capacity by affecting LB and SE, correspondingly. Joint degradation of them also impacts UP negatively, as explained in (B.7).

## 6.6 Conclusions and Future Work

In this chapter we explored the problem of association in future HetNets and propose a framework which can be derived to find optimized association algorithms that take into account factors like traffic type, traffic direction (UL/DL), and backhaul network connectivity and capacity. A couple of interesting issues arose from the simulation-based evaluation of these optimal policies. These have either been addressed in subsequent work or are part of future work.

First, having the same amount of DL and UL resources in every BS is clearly sub-optimal. Same goes for the resource split in the backhaul network. To this end, in [78] we addressed the first issue, that of jointly optimizing the association variables in our framework, together with the UL/DL resource allocation. This problem appears to be non-convex at first, but with some appropriate transformations it can be convexified and tackled with a distributed hierarchical algorithm, running at different time scales. A preliminary treatment of backhaul resource allocation has also been pursued in [238].

Second, as the backhaul network becomes more complex (e.g. a mesh network) routing should jointly be addressed with the user association problem, to further improve performance. As was evident in this chapter, the user association on the radio access might be limited by the available capacity on the backhaul path. In this chapter we assumed this routing path to be given (e.g., chosen before, or in a slower time scale than the user association variables, as is usually the case). However, re-visiting the routing path together with the BS load-balancing problem is a promising direction. We plan to explore whether the joint problem can be convexified and solved in a distributed/hierarchical manner as well.

Last but not least, we have made throughout this chapter the (strong) assumption that the traffic demand is known at each location  $x$ . In practice, this traffic might not only be unknown (at first) but also non-stationary. In our recent work of [254], we have considered data-driven framework to learn the mean traffic demand while optimizing the associations, in order to achieve robust association maps: to perform well on average, but deteriorate gracefully when the statistics diverge from their mean (e.g., flash crowds, special events, etc.).

## **Chapter 7**

# **Mobile Edge Caching**

As already argued in the previous chapters, a promising and highly-investigated way of dealing with the tremendous increase in traffic demand is via the deployment of many small cell base stations (e.g pico/femto-cells, or out-of-band WiFi APs) together with the standard macro-cellular network. The bulk of current state of the art for small cell solutions tackle mainly aspects of self-organizing radio resource management, inter-cell and interference coordination (ICIC), energy-efficiency and MIMO techniques, as well as traffic steering techniques like the ones considered in the last two chapters. A common problem for a lot of work on small cell networks is the impact of the backhaul network. While the reduced distances to the user and better spatial reuse can greatly increase the (radio) capacity in the last hop, in order to support high cell densification the capacity of the backhaul network needs to scale at unrealistic rates. As a result, small cells are expected to be mostly connected by inexpensive wired or wireless links, and thus will often be under-provisioned, according to the Small Cell Forum [79]. This makes the backhaul network a potential new bottleneck that can significantly reduce the envisioned gains by cell densification.

In the previous chapter, we investigated how to take such backhaul bottlenecks into account, even when optimizing radio access functions like user association. While this approach can redirect some traffic to different BSs, possibly connected with less congested transport links, some important issues remain: (a) this is not always possible or it introduces considerable suboptimalities on the radio access (e.g., in terms of spectral efficiency); (b) the transport network (backhaul/fronthaul) might simply not be able to support the aggregate radio traffic, for any routing path(s). To further alleviate the backhaul bottleneck, but also to reduce content access latency, the idea of edge caching has been proposed [80, 81, 82, 83]

The increasing demand for Internet content, and the inherent overlap between user data requests, suggests that storing popular content closer to the user could: (a) avoid congesting the capacity-limited backhaul links by avoiding duplicate backhaul transmissions, and (b) reduce the content access latency. Specifically, in [80], local content caching has been identified as one of the five most disruptive enablers for 5G networks, sparking a tremendous interest of academia and industry alike, as a cost-efficient way of tackling the data tsunami. Using real measurement data, [84] has explored the potential benefits of forward caching inside the core of a 3G network (e.g. in the SGSNs or GGSNs), suggesting that hit ratios up to 33% can be achieved for such caches. Similar benefits based on real workloads have been also found for LTE networks [85]. While these works already demonstrate the potential benefits of caching, a number of key questions remain open.

**Edge cache size can be a major bottleneck:** First, the above measurement-based studies have only considered caching in core nodes or backhaul aggregation points. Such nodes see a significant amount of the global network traffic and could also be equipped with large storage capacities (comparable to the size of regular CDN servers). It remains widely open whether similar benefits could be achieved with small, local caches at each BS and small cell. Such caches would have a much smaller storage capacity due to CAPEX/OPEX cost reasons: each core node considered as a potential caching location in the above studies would correspond to 100s or 1000s of small cells in future ultra-dense networks. As a result, the storage space per local edge cache must be significantly smaller to keep costs reasonable. This suggests that only a tiny fraction

of the constantly and exponentially increasing content catalog could realistically be stored at each edge cache. In most practical cases, a local edge cache would fit less than 0.1 – 0.01% of the entire Internet catalog. E.g., a typical torrent catalog is about 1.5PB and the entire Netflix catalogue about 3PBs. More than 1TB storage would be needed to just store 0.1% of either one [86]. Considering other VoD platforms, YouTube, etc., this number explodes. Even, with a skewed popularity distribution, *local* cache hit ratios would be rather low [87, 88, 89].

**Convergence of cooperative caching and cooperatice communication:** Second, a number of technological advances are envisioned for future 5G networks, such as Coordinated Multi-Point (CoMP) transmission, Coded Caching, and the CloudRAN architecture. These technologies are considered as major drivers to improve the performance on the radio access network (RAN) in 5G, but also introduce novel complications and challenges for caching algorithms. To cooperatively transmit a requested content from multiple BSs, so as to improve radio access performance (e.g. through diversity gains), that content must be available in the caches of all BSs involved in a coordinated transmission. Otherwise, the effective rate of one or more of the links might be severed by a congested backhaul link. Yet, storing the same content in multiple nearby BSs means not storing other contents there, thus leading to a considerable decrease in cache hit rates and increased backhaul costs, if those contents are requested instead. This creates a novel tradeoff: how could a caching policy facilitate cooperative transmission opportunities (and radio access performance) while maximizing local cache hit rates (and backhaul performance)?

**Application-dependent caching optimization** Third, specific applications, especially content streaming, are beginning to dominate the mobile data traffic (e.g., more than 50% percent of the total traffic is consistently reported to be streamed video), and this is not expected to change in the future. Any means to reduce network traffic through caching should thus pay particular attention to this application. This raises the question: could application-driven optimizations targeting video traffic further improve the performance of edge caching?

In this chapter, we briefly present four main directions of ongoing and future planned research to maximize the amount of traffic offloaded from, as well as to jointly optimize the performance of, radio access and backhaul networks.

1. Using node mobility (UE or Relay) to trade-off content access delay (i.e. user QoE) for cache hit ratio. (Section 7.1)
2. Exploiting the properties of video streaming to offload more data through edge caching *without any QoE impact*. (Section 7.2)
3. Using alternative content recommendation to improve caching performance through “soft cache hits”. (Section 7.3)
4. Coordinated caching and communication. (Section 7.4)

## 7.1 Delay-tolerant Edge Caching

Installing a large enough number of small cells with local storage (SCs), to ensure every user equipment (UE) has access to at least one such low range BS, still requires high CAPEX/OPEX costs (e.g. rental costs, maintenance, backhaul). Combined with the stagnating revenues per user, this makes operators reluctant to perform the necessary densification. The more radical proposal to use UEs for storage and relaying (e.g., through Device-to-Device (D2D) communication) [90, 91] largely avoids these costs, but is stifled by device resource constraints (especially battery) and privacy concerns. As a result, SC storage and/or D2D might offer small performance improvements, in practice. E.g., a user requesting a content  $k$  might not have an SC in range, in which case  $k$  must be fetched over an expensive macro-BS. Even if the user has some SCs (or other UEs) in range, these might not in fact store content  $k$  (due to their relatively small size compared to the content catalog). Fetching content  $k$  will again be costly, e.g. over an SC backhaul link which is usually wireless and capacity-limited.

To improve this situation, one could exploit the mobility of UEs or even of BSs, to tradeoff access delay for cache hit ratio. For example, if a user  $i$  requests a content  $k$ , which is not “locally” available, the operator can ask the user to wait for some time (e.g. up to some  $TTL$ ). During that time, user  $i$  may encounter some other cache storing  $k$  (either because  $i$  moved, or because the cache moved, in the case of mobile relays [92]). E.g., consider a simple example where there are  $M$  total edge caches, and intermeeting time between a user and such a cache is IID with rate  $\lambda$ . Assume that in the baseline (“immediate access”) scenario, a user sees on average  $c$  of these  $M$  caches, where  $c < 1$  in most cases (since SC coverage is limited). In the “delayed access” case, a user sees an additional  $M \cdot \lambda \cdot TTL$  caches, on average. Hence, the *effective* cache size that the average user sees is approximately  $M \cdot \lambda \cdot TTL$  times larger, which may lead to more cache hits.

The challenge here is to optimally allocate this larger (effective) cache space among all contents. E.g. if a node sees at most one cache, the optimal policy (putting aside the further considerations of Coded Caching [255]) is to store the most popular contents in every cache, until it’s full. However, it is easy to see that this is suboptimal in the delayed access case. If the size of each cache is  $C$  contents, and node  $i$  encounters 2 caches during a  $TTL$ , storing contents 1 to  $C$  in one cache, and  $C + 1$  to  $2C$  in the other, would lead to much higher hit rates *for that node* than storing contents 1 to  $C$  in both. While this example provides some intuition how the cache policy could exploit mobility (and the additional cache space it allows node to “see”), finding the optimal policy depends on the mobility characteristics, content popularities, and  $TTL$ .

### 7.1.1 Content Access Protocol

In this section, we state some assumptions about our system.

**A.1 - Network Setup.** There are three types of nodes:

- *Infrastructure Nodes* ( $\mathbb{I}$ ), e.g., macrocells, that provide full coverage, and can serve any content request.

- (*Mobile*) *Helper Nodes* ( $\mathbb{H}$ ), e.g., cars, buses, taxis, where  $|\mathbb{H}| = H$ . Each of these nodes has a cache size of  $C$  bytes, and can serve only requests for content in its cache.
- *End Users* ( $\mathbb{U}$ ), e.g., smartphones, tablets, netbooks, that request different contents.

In the above, we are focusing on the case of mobile relays and storage points. These provide many advantages, which are detailed in [92]. However, the theory is applicable to the case of storage at static helper nodes (e.g., small cells) or even other UEs (see e.g., [256]).

**A.2 - Content Request Model.** We assume the standard IRM request model, where each node independently asks for some content  $i$  out of a catalogue  $\mathbb{K}$  of content ( $|\mathbb{K}| = K$ ), with probability  $\phi_i$  [257, 258, 259]<sup>1</sup>. Without loss of generality, we assume content is sorted by decreasing popularity as  $\phi_1 \geq \phi_2 \geq \dots \geq \phi_K$ . We also assume that content  $i$  has size  $s_i$ .

**A.3 - Delayed Access.** A user requesting content  $i$  will wait up to a maximum time  $TTL_i$  to find a (mobile) helper with that content. If it does, then it retrieves the content locally, imposing 0 cost on the infrastructure. If  $i$  is not found during that time, then the content is fetched from an (expensive)  $\mathbb{I}$  node.

**A.4 - Contact Model** ( $\mathbb{U} - \mathbb{H}$ ). Unless otherwise stated, pairwise inter-contact times between  $\mathbb{H}$  and  $\mathbb{U}$  are independent, drawn from a *generic* distribution  $F_C(t)$  with mean rate  $\lambda$ . Contact durations are drawn from another *generic* distribution  $F_D(t)$  with mean  $\mathbf{E}[D]$ . We assume both distributions have bounded first and second moments.

**A.5 - Cache Model.** Let  $x_{ij} \in \{0, 1\}$ ,  $i \in \mathbb{K}$ ,  $j \in \mathbb{H}$  be an indicator variable denoting if helper node  $j$  stores content  $i$ . We assume  $\mathbb{H}$  nodes *store entire contents*, and fractional storage is not allowed. Let further  $N_i$  denote the number of helper nodes storing content  $i$ , that is

$$N_i = \sum_{j \in \mathbb{H}} x_{ij}. \quad (7.1)$$

$N_i$  will be a key control variable for our optimal cache allocation problem<sup>2</sup>.

The goal of the operator is to optimally choose these number of copies  $N_i$  for each content  $i$  that it should push into the helpers (called “seeding” phase, e.g. the night before). The basic protocol is made up of three phases and is depicted in Fig. 7.1. We summarize some useful notation in Table 7.1.

### 7.1.2 Optimal Cache Allocation

To understand the basic ideas behind the cache allocation problem, let us consider a simple subcase of the above problem setup where  $TTL_i = TTL$  (i.e., all content has the same delay-tolerance). Let us additionally assume that: (i)  $F_C(t) = 1 - e^{\lambda t}$ , i.e. inter-contact times are exponential, and (ii)  $\mathbf{E}[D]$  is much larger than the time to

---

<sup>1</sup>In practice, these quantities can be estimated as an expected request rate during some MNO-defined time window (e.g., a day [260, 83]). Several studies have shown that it is possible to predict the popularity with good accuracy over relatively short periods (e.g., a few weeks for YouTube videos) [261].

<sup>2</sup>Note that given the assumption of IID mobility, it suffices to optimize the total number of copies  $N_i$  without considering the per vehicle variables  $x_{ij}$  any more.

Table 7.1: Notation.

<b>CONTROL VARIABLES</b>	
$N_i$	Number of replicas stored for content $i$
$TTL_i$	Deadline for content $i$
<b>CONTENT</b>	
$K$	Number of content in the catalogue
$\phi_i$	Number of requests for content $i$
$s_i$	Size of content $i$
<b>MOBILITY</b>	
$T_{ij}$	Inter-meeting time between $\mathcal{U}$ and $\mathcal{H}$ nodes
$\lambda$	Mean inter-meeting time with vehicles
$M_i$	Number of contacts within $y_i$
<b>CHUNK DOWNLOAD</b>	
$b_{ij}$	Bytes downloaded per contact
$\mu$	Mean of $b_{ij}$
$\sigma^2$	Variance of $b_{ij}$
$B_i$	Total bytes downloaded per content $i$
$f_{B_i}(x)$	Probability density function of $B_i$
$F_{B_i}(x)$	Cumulative density function of $B_i$
<b>OTHER PARAMETERS</b>	
$H$	Number of vehicles
$C$	Buffer size per vehicle
$\Omega$	Weighted average of maximum slowdown (WAMS)
$TTL_{max}$	Maximum deadline

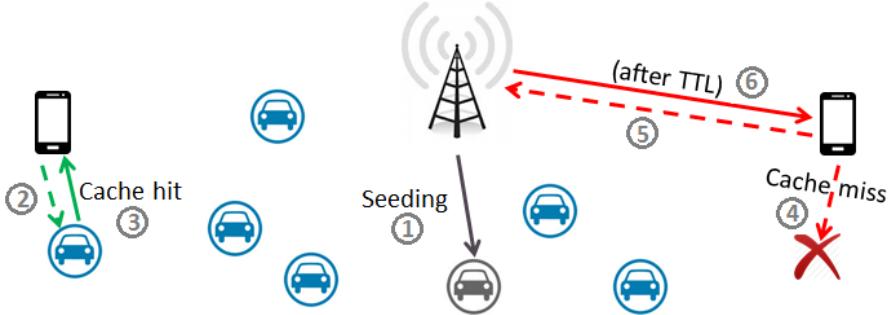


Figure 7.1: Communication protocol.

download any of the contents (i.e., any content  $i$  can be downloaded in a single contact with a helper storing  $i$ ). In this setup, the optimization problem in hand is:

$$\underset{N_i}{\text{minimize}} \quad \sum_{i \in \mathbb{K}} \phi_i \cdot s_i \cdot e^{-\lambda \cdot N_i \cdot TTL}, \quad (7.2)$$

$$\sum_{i \in \mathbb{K}} N_i \cdot s_i \leq H \cdot C, \quad (7.3)$$

$$0 \leq N_i \leq H, \forall i. \quad (7.4)$$

The objective (7.2) corresponds to the total bytes that have to be downloaded after the  $TTL$  expires, and follows from basic properties of the exponential distribution. Constraint (7.3) is a knapsack type constraint related to the *total* storage capacity in  $\mathcal{H}$  nodes, and constraint (7.4) simply says that the number of replicas must be non-negative and at most equal to the total number of helpers (there is no benefit from storing the same content twice in the same cache).

The above problem is a discrete optimization problem (as  $N_i$  must be integer) which could be mapped into a non-linear knapsack variant, and thus is hard in theory. However, if cache size  $C$  is larger than just a few contents (which is typically the case), a very good approximation can be achieved by solving the continuous relaxation of the above problem, i.e. assuming  $N_i \in \mathbb{R}$  (and then rounding appropriately). In this case the above problem is convex, and can in fact be solved analytically using KKT conditions. The following gives the optimal allocation:

**Result 1.** The optimal number for the objective of (7.2) is given by:

$$N^{(i)} = \begin{cases} 0, & \text{if } \phi_i < L \\ \frac{1}{\lambda \cdot TTL} \ln \left( \frac{\lambda \cdot TTL \cdot \phi_i}{\rho} \right), & \text{if } L \leq \phi_i \leq U \\ H, & \text{if } \varphi^{(i)} > U \end{cases}$$

where  $L \triangleq \frac{1+\rho}{\lambda \cdot TTL}$ ,  $U \triangleq \frac{(1+\rho) \cdot e^{H \cdot \lambda \cdot TTL}}{\lambda \cdot TTL}$ , and  $\rho$ ,  $\mu^{(i)}$ ,  $\lambda^{(i)}$  and  $\rho$  are appropriate (non-negative) Lagrangian multipliers.

There are a couple of things to observe here. First, the number of copies is an increasing (as expected) but concave function of the content popularity, suggesting diminishing returns per extra copy (it is useful to compare this to the simple motivating

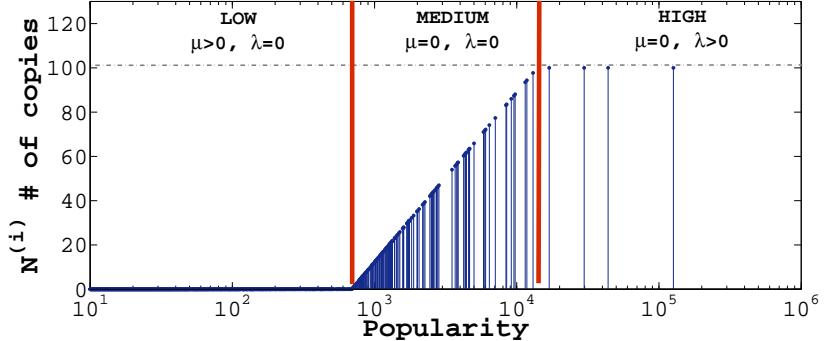


Figure 7.2: Optimal allocation in semi-log scale.

example discussed earlier). The number of copies is also a decreasing function of the TTL, since the larger the delay tolerance, the more caches a node sees during a TTL, and the higher the chance for a cache hit, even if fewer caches store  $i$ . The capacity constraint affects this allocation mainly through multiplier  $\rho$ , which acts as a scaling factor (the higher the capacity the smaller  $\rho$  is). Finally, in some scenarios some low popularity contents get 0 copies while very popular contents are replicated in all helpers. An example of the above allocation (in log-linear scale) can be seen in Fig. 7.2.

### 7.1.3 Performance Evaluation

We validate our model through MATLAB simulations. We simulate the load on the infrastructure and we compare it with other caching systems. We consider the optimal allocation policy described earlier and we analyse the impact of different parameters in the proposed cache system. Furthermore, we consider a square area of 5000m x 5000m in the center of San Francisco. We use the *Cabspotting*<sup>3</sup> trace to compute the average meeting rate  $\lambda$ : we randomly place the  $\mathcal{U}$  nodes and, considering a communication range of 200m, we calculate the meeting rate with each  $\mathcal{H}$  node. We find  $\lambda = 4$  contacts/day. According to the density of the city and to the number of vehicles per capita, we estimate to 100.000 the number of vehicles in the area considered. However, in order to be realistic about initial technology penetration we assume that only 1% of these vehicles is participating in the cloud. We assume that each car can store 100 contents (0, 1% of the catalogue). We set TTL to 3 minutes.

In our analysis, we also consider the content popularity of YouTube videos, since a large percentage of mobile data traffic is represented by video files. We download from [262] a database generated with YouStatAnalyzer that collects statistics for 100.000 YouTube videos. The database includes static (title, description, author, duration, related videos, etc.) and dynamic information (daily and cumulative views, shares, comments, etc.). In our simulations, we only take into account the number of

---

<sup>3</sup>GPS coordinates from 536 taxis in San Francisco over 23 days [112].

views related to the last 360 days. However, these values are equal to the total number of views per day in the world, then we scale them properly<sup>4</sup>.

Note that we have also created synthetic traces based on the work in [263]. Simulations based on these synthetic traces confirm the observations made using the real trace. We therefore focus on the former.

Unless otherwise stated, we will use the parameters summarized in Table 7.2.

DESCRIPTION	PARAM	VALUE
Number of vehicles	$H$	1000 cars
Buffer size	$C$	100 contents/car
Meeting rate	$\lambda$	4 contacts/day
Time-To-Live	TTL	3 minutes
Number of contents	K	100.000 contents
Communication range		200m

Table 7.2: Parameters of the scenarios considered.

Finally, we compare our allocation policies with:

- *No cache*: no contents are stored in the vehicles; the probability of miss is equal to 1, therefore the cost corresponds to the total demand:  $\text{cost} = \Delta t \sum_{i=1}^K \varphi^{(i)}$ ;
- *Random*: contents are allocated randomly in the vehicles;
- *Square root*: this policy also has diminishing returns, similar to our optimal allocation, but it replaces the logarithm with the square root (after an appropriate normalization to satisfy the storage constraint); such policies have been shown to perform well in P2P networks.

Fig. 7.3 shows the cost according to the value of TTL for the different policies. It is very important to note that considerable gains can be achieved with very small TTL values (in the order of a few minutes) and small number of vehicles participating in the cloud (1%). This provides some evidence on the advantages of offloading based on a vehicular cloud, compared to offloading using small cells or WiFi, as for example in [51] or [195]. E.g., [51] reports minor gains for similar small deadlines, while [195] requires a much longer TTL (order of 1-2 hours) to achieve similar gains. In addition, increasing the TTL further has diminishing returns. This implies that even users not willing to wait too long could participate in such a system (benefiting themselves and the operator).

Fig. 7.4 shows the cost according to the buffer size. From the plot we can observe that storing 100 contents/car (only 0,1% of the total number of contents) provides a gain of almost 60%. In a scenario with a larger catalogue (e.g., 100 millions), it seems doable to store 0,1% of the contents (e.g., 100.000 contents/car) needed to achieve good savings. What is more, due to the intrinsic characteristics of the popularity distribution,

---

<sup>4</sup>We scale them linearly taking into account the number of YouTube users and the population of San Francisco.

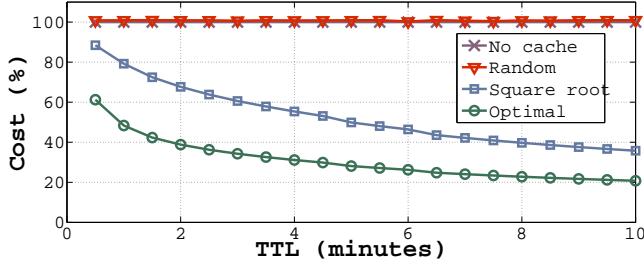


Figure 7.3: Cost comparison according to TTL.

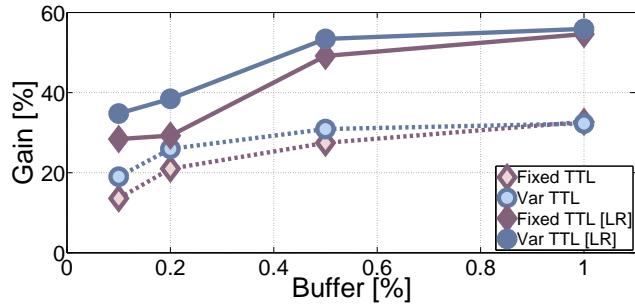


Figure 7.4: Cost comparison according to buffer capacity.

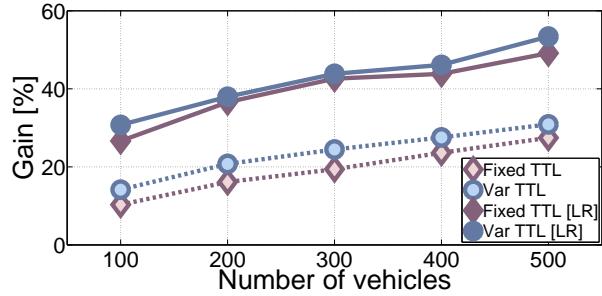


Figure 7.5: Cost comparison according to number of vehicles.

the system might require an even smaller number of storage in order to achieve similar gains.

In an urban environment, the great availability of vehicles leads to large gains for the proposed infrastructure. However, an operator will probably keep using our framework even if the number of vehicles available decreases: in Fig. 7.5 we depict the cost savings according to the number of (vehicular) helper nodes participating in the cloud and the gain observed is not less than 50% when more than 250 vehicles are part of the cloud.

### 7.1.4 Extensions

The above framework and results provide some initial, yet important insights. In addition, we have considered a number of extensions to the above basic framework.

#### Generic Inter-contact Times

A first possible extension is to generalize the inter-contact times. This can be done by replacing  $e^{-\lambda \cdot N_i \cdot TTL}$  in objective (7.2) with  $(1 - F_C(TTL))^N_i$ . It is easy to see that the objective remains convex and optimal values for  $N_i$  can be found with standard methods (albeit not always analytically, depending on  $F_C(TTL)$ ).

#### Limited Contact Duration

Although we have assumed unlimited contact duration (or equivalent, bandwidth), in practice a content might not be downloaded in its entirety during a contact (e.g. the node moves away before the download is finished). In this case, one could consider two possible protocols: (*Repeat*) a node must restart the download of content  $i$  in the next contact (with some node storing  $i$ ); (*Resume*) a node keeps the bytes it has downloaded so far, and continues with the remaining ones during a new contact. For the *Repeat* policy, let us denote with  $p_i = 1 - F_D(\frac{s_i}{r_H})$ , the probability that content  $i$  of size  $s_i$  is fully downloaded during a given contact during which the mean download rate is  $r_H$ . This download is successful if the contact duration is at least equal to  $\frac{s_i}{r_H}$ . Since contact durations are assumed independent, one can use a Poisson thinning argument (in the case of exponential contacts) and replace  $\lambda$  in all the above equations with  $p_i \cdot \lambda$ , to obtain the optimal allocation.

While analytically tractable, clearly the *Repeat* policy is not optimal (large contents, even if popular, will get very few copies, because there is very little chance one can download them in one shot). *Resume* on the other hand is in some sense “work conserving” and can perform better. However, its analysis is more involved. In the case of exponential inter-contact times, the total amount of bytes downloaded during consecutive successful contacts form a Compound Poisson Process whose distribution is a function of both  $F_C(t)$  and  $F_D(t)$ . Nevertheless, one can approximate this with a Gaussian distribution and derive an appropriate modification of (7.2) in closed form. Convexity is maintained (for the continuous relaxation of the problem) and the optimal solution can be obtained. For more details, please see [92, 93].

#### Variable TTL

Although we have assumed an equal  $TTL$  for all contents so far, the above results easily generalize to different  $TTL_i$  per content  $i$ . If different users further have different delay tolerance for the same content, then the value  $TTL_i$  could be seen as the average over all such users. However, objective (7.2) is only a first order approximation in that case. Nevertheless, if one assumes that the TTL per user is a random variable from a known distribution, then one could replace the  $e^{-\lambda \cdot N_i \cdot TTL_i}$  (or even  $(1 - F_C(TTL_i))^N_i$ , for generic inter-contacts) with  $E_{TTL_i}(e^{-\lambda \cdot N_i \cdot TTL_i})$ , where the expectation is taken over the distribution of  $TTL_i$ .

A more interesting case arises if we consider  $TTL_i$  as a second set of control variables. The motivation for this is that the user perceives the enforced delay not in absolute terms, but rather in relation to the total download time of the requested file. E.g., a user requesting a web page (which normally takes a few 10s of seconds to download) would not be happy with an extra delay of 5 minutes. However, these extra 5 minutes could more easily be “absorbed” if a user is downloading a large software or video, which anyway takes some 10s of minutes. Hence, an operator could allocate a higher  $TTL$  to a large content, and lower  $TTL$  to small contents. This can be easily formalized with the concept of *slowdown*, a metric that has recently attracted a lot of attention in queueing theory, and can be loosely defined as the total access delay (wait + download) divided by the download time. In its simplest case, the optimization problem then becomes:

$$\underset{N_i, TTL_i}{\text{minimize}} \quad \sum_{i \in \mathbb{K}} \phi_i \cdot s_i \cdot e^{-\lambda \cdot N_i \cdot TTL_i}, \quad (7.5)$$

$$\sum_{i \in \mathbb{K}} N_i \cdot s_i \leq M \cdot C, \quad (7.6)$$

$$\sum_{i=1}^k \phi_i \cdot \frac{TTL_i \cdot r_H + s_i}{s_i} \leq S, \quad (7.7)$$

$$0 \leq N_i \leq M, \forall i. \quad (7.8)$$

We have just introduced  $TTL_i$  as a control variable, and also added constraint (7.7) which normalizes the total TTL “budget” we have to allocate (for some constant  $S$ ). This is a mixed integer program as some variables are integer ( $N_i$ ) and others continuous ( $TTL_i$ ). We can again let  $N_i$  take real values to simplify the problem. However, note that the resulting problem is not convex in both variables anymore, even in this simple setup. In fact it is “bi-convex”, as it is convex on either  $N_i$  or  $TTL_i$ , but not both. One can approach this problem with various solvers for bi-convex problem (e.g. iteratively optimizing for one set of variables, then the second, alternating the two sets till convergence). However, there are no strong guarantees of global convergence. Things get more complicated if one considers a *Resume* type of policy. Nevertheless, one can replace the respective objective with an approximate one (whose accuracy increases when there are plenty of contacts during a TTL), and which gives rise to a Geometric Program. The latter can be transformed into a convex one with appropriate manipulations and solved using standard methods. We refer the interested reader to [93].

### Offloading through HetNets with SCs and UEs

As a final setup, in [256] we have considered a mixed setup, where a user can get the content both from small cells or from other UEs, storing the content. Additionally, we assume there that every UE that receives a given content  $k$ , can further act as a local cache (with some probability, depending on incentives or other consideration). This setup is different compared to the previous one, as the number of storage points here increases over time, while in all previous scenarios the number of storage points is selected initially (when the optimal cache allocation is selected, e.g. at nighttime) and the number of caches remains unchanged during the delivery phase. While this

evolving number of storage points complicates the analysis, necessary to derive an objective in closed form (that can be then optimized), we have used [256] mean field and fluid approximations for the problem, that were validated both against synthetic and trace-based simulations.

## 7.2 Low Cost Caching for Streamed Video Content

So far, we have been assuming that contents are downloaded in their entirety *before they can be consumed*. We then introduced an additional delay in the system, before such a download starts, in order to allow a user to encounter low cost nodes (e.g. small cells or other UEs storing the content), and optimized the cache allocation in this setup. In the simulation scenarios considered for such a system, we saw that interesting offloading values can be achieved with delays ranging from 2-3 minutes to a few 10s. Such delay values could be acceptable in some scenarios, e.g. large files, delay-tolerant traffic (software updates, synchronization), developing regions and/or low cost plans. Nevertheless, if one considers standard urban environments, such deadlines might require a significant monetary (or other plan-related) incentive by the operator, or might be unacceptable altogether.

In this second part, we focus on streamed (video) content specifically, and try to go one step further by exploiting the fact that *streaming of stored video content offers some delay tolerance “for free”*. In other words, we would like to get some of the offloading benefits of the previous approach, *without any delay impact on the user*. Video content is split into many chunks that are streamed into the user’s playout buffer one by one, and consumed at the playout speed. The user does not have to wait until the whole content is found in a local cache, but can start streaming right away, fetching chunks from the infrastructure or local/mobile caches, depending on availability of the latter and buffer status. For example, consider a user that starts watching a 1h video, and a chunk corresponding roughly to the middle of the video. This chunk does not have to be downloaded until 30 minutes after the user starts streaming. During that time, the user might encounter a mobile cache storing this chunk and fetch it inexpensively. In that case, traffic is offloaded from the main infrastructure (which would otherwise be used to download the chunk), but *without any impact to user experience*.

It is important to note that, while chunks are also “downloaded” in the streaming case (as contents were downloaded before), the difference here is that no extra delay needs to be *imposed* on the user (and thus deteriorate her QoE). Any chunk not available in the playout buffer when its playout time arrives, can be streamed from the main infrastructure. In this context, two interesting questions arise:

1. How many bytes of streamed video get offloaded through such mobile edge caching?
2. How can we optimize the edge cache allocation to maximize the amount of bytes that get offloaded?

Our goal is to provide some initial answers to these questions assuming: (i) vehicular nodes acting as (mobile) SCs and local caches, and (ii) streamed video-on-demand

content as the main application<sup>5</sup>. Our main contributions are the following:

- We model the playout buffer dynamics of a user device in this setup as a queueing system, and analyze the expected amount of offloaded traffic (corresponding to the idle periods of this buffer) as a function of network characteristics (e.g., vehicle density, file characteristics) and a given cache allocation.
- Based on this model, we formulate the problem of optimal allocation of content in vehicles, that minimizes the total load on the cellular infrastructure. We formulate the optimal allocation problem, show it is NP-hard, and propose appropriate approximations for two interesting regimes of vehicular traffic densities.

As a final remark, while we have performed our analysis within the context of vehicular mobile relays, the main framework and a number of our results could again be applied to content streaming from fixed SCs or even UEs.

### 7.2.1 Problem Setup

In addition to the general network setup described in the previous section, we make here the following additional assumptions<sup>6</sup>:

**A.6 - Video Streaming Model.** Each video is characterized by size  $s_i$  and consists of a number of small chunks. New chunks are downloaded into a  $\mathbb{U}$  node's playout buffer in order, and consumed for playout as follows:

- *Helper download.* When a  $\mathbb{U}$  node is in range of (at least) one  $\mathbb{H}$  node that stores the requested content, the next immediate chunks not yet in the playout buffer are downloaded *in order* at mean rate  $r_H$ . E.g., assume that a node is currently viewing chunk  $n$ , and its playout buffer already contains chunks  $n + 1, \dots, n + K$ ; then, chunks starting from chunk  $n + K + 1$  will be downloaded until the connection with that  $\mathbb{H}$  node is lost. This opportunistic connection is represented by the green region in Fig. 7.6 (e.g., between  $t_1$  and  $t_2$  the node will download from cache 1). What is more, we do not allow for simultaneous connections, i.e., a  $\mathbb{U}$  node can download from at most one  $\mathbb{H}$  node at a time (we defer considering multi-connectivity to future work). For this reason, in Fig. 7.6 the user will switch to cache 3 only at  $t_4$ , i.e., after it has finished downloading from cache 2.
- *Infrastructure download.* When a  $\mathbb{U}$  node is not in range of an  $\mathbb{H}$  node that stores the requested content *and* its playout buffer is empty, new chunks are downloaded from the infrastructure at a mean rate  $r_I$ . This corresponds to the red region of Fig. 7.6. However, if the playout buffer is *not* empty, no chunks are downloaded from  $\mathbb{I}$  until the buffer empties.
- *Playout.* Chunks in the playout buffer are consumed at a mean viewing (*playout*) rate  $r_P$ .

---

<sup>5</sup>Note that this scenario does not include live content streaming, which is not usually amenable to caching, and is often optimized using multicast techniques.

<sup>6</sup>Note that we convert all file sizes and cache storage spaces into bits (rather than bytes), as transmission rates are also quoted in bits/sec, to avoid normalization

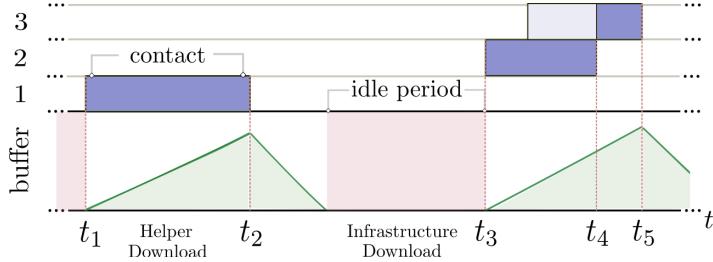


Figure 7.6: Sequence of contacts with three caches (above), and amount of data in end user buffer over time (below, in green). The red region indicates when data is downloaded from  $\mathbb{I}$ .

**A.7 - Download Rates.** We assume  $r_I = r_P + \epsilon$  ( $\epsilon > 0$  small), in order to limit the access to the cellular infrastructure to the minimum required to ensure smooth playout (for simplicity, we assume that  $\epsilon = 0$ ). Furthermore, we assume that  $r_H \geq r_I$ , on average (this is a reasonable assumption due to the reduced communication distance)<sup>7</sup>.

These assumptions are needed to adapt our architecture to facilitate stored (video) content streaming (A.6), and specifically to ensure that the expensive infrastructure nodes are used only when absolutely necessary (A.7). Based on this setup, the following defines the new objective for our system:

**A.8 - Data Offloading.** A request for content  $i$  will download a number of bits from  $\mathbb{I}$  nodes. This number is a random variable  $X_i$  that depends on  $N_i$  as well as the sample path of the contact variable(s). We denote as  $\mathbf{E}[X_i|N_i]$  the expected value of this quantity, where the expectation depends on distributions  $f_C(t)$  and  $f_D(t)$ . Our goal is to minimize it, since  $1 - \mathbf{E}[X_i|N_i]$  is the traffic offloaded on average for requests of content  $i$ . To keep notation simple, we will refer to this quantity as  $\mathbf{E}[X_i]$ .

The goal of the operator is to minimize the average number of bits downloaded per content,  $\mathbf{E}[X_i]$ , among all content  $i \in \mathcal{K}$ , by appropriately choosing the control variables  $N_i$  (number of caches storing  $i$ ). This is captured in the following:

**Lemma 6.** *The optimal allocation  $N_i \in \mathbb{N}$ , that minimizes the expected number of bits downloaded from the infrastructure is given by the solution of the following optimization problem:*

$$\underset{N_i}{\text{minimize}} \quad \sum_{i=1}^K \phi_i \cdot \mathbf{E}[X_i], \quad (7.9)$$

$$\text{subject to} \quad 0 \leq N_i \leq H, \quad i \in \mathbb{K}, \quad (7.10)$$

$$\sum_{i=1}^K s_i \cdot N_i \leq C \cdot H, \quad (7.11)$$

<sup>7</sup>Scenarios where  $r_I$  (and/or  $r_H$ ) are lower than the playout rate require initial buffering, which is known to significantly degrade QoE [264], and are orthogonal issues to the problem addressed here. Nevertheless, our framework could be easily extended to include such initial buffering.

where  $\mathbf{E}[X_i]$  is the expected number of bits downloaded from  $\mathbb{I}$  for content  $i$ , when  $N_i$  helper nodes store that content.

The objective function is subject to two constraints: (i) the number of replicas is limited by the number of vehicles participating in the cloud and cannot be negative (Eq. (7.10)); (ii) the allocation cannot exceed the aggregate buffer capacity  $C \cdot H$  of all vehicles.  $\mathbf{E}[X_i]$  depends on the scenario parameters (content popularity, size, helper node density and mobility, etc.). Also, while in Eq. (7.9) we just consider the *expected* cost, in the limit of many content requests during a time window, this result becomes asymptotically exact for a specific instance of requests (due to the law of large numbers).

In this section, we approximate analytically  $\mathbf{E}[X_i]$  for two regimes of mobile helper density, and we solve the respective optimization problem of Lemma 6 to find the optimal content allocation. Specifically, in Section 7.2.2 we consider first a *low density* scenario, which provides insights on how to solve the *generic density* scenario of Section 7.2.3. In addition to these results, in [94] we have also derived an analytical bound on the approximation error of  $\mathbf{E}[X_i]$ , that we omit here.

### 7.2.2 Low Density Regime

Let us assume first that contacts with vehicles are sparse, i.e., the probability of overlapping contacts in time, with different vehicles *both storing the same content*, is small:

**Definition 7.2.1** (Low Helper Density). *We define a low helper density regime as a scenario where  $\lambda \cdot H \cdot E[D] \ll 1$  and  $\frac{r_H}{r_P} < (\lambda \cdot H \cdot E[D])^{-1}$ .*

This is a reasonable assumption when the number of vehicles utilized in the cloud is small, and/or for medium/low popularity content that do not have replicas in every vehicle. In this scenario, we model the playout buffer as a bulk  $G^Y/D/1$  queue, where new bits arrive in bulks when a helper node with the requested content is encountered, and are consumed at a constant playout rate. This system is depicted inside the small square of Fig. 7.7 (the queue on the left can be ignored for now). The following holds for the expected load on the infrastructure (see A.7):

**Lemma 7.** *For a low density scenario the following expression is asymptotically tight as the content size  $s_i$  becomes large, that is:*

$$\lim_{s_i \rightarrow \infty} \left[ \mathbf{E}[X_i] - s_i \cdot \left( 1 - \lambda N_i \cdot \mathbf{E}[D] \cdot \frac{r_H}{r_P} \right) \right] = 0.$$

*Proof.* Consider a content  $i$  currently stored in  $N_i$  caches. The  $G^Y/D/1$  queue model for the playout buffer has:

- *Service Rate.* Jobs (i.e., bits) in the buffer are served (i.e., viewed by the user) at the constant playout rate  $r_P$ .
- *Bulk Size.* A contact between a device and a helper node storing video  $i$  corresponds to a new (bulk) arrival in the playout buffer of the device. A new arrival brings a random amount of new bits (with mean  $\mathbf{E}[Y] \triangleq \mathbf{E}[D] \cdot r_H$ ) that depends on the contact duration with that  $\mathbb{H}$  node.

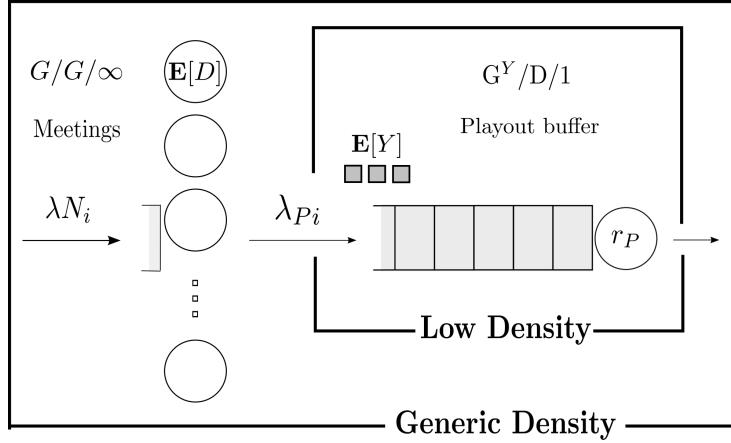


Figure 7.7: Proposed queuing model for the playout buffer. The queue inside the small box corresponds to the low density regime model (Section 7.2.2), while the large box (containing both queues) to the high density regime (Section 7.2.3)

- *Arrival Rate.* The total arrival rate into the playout queue is  $\lambda_{Pi} \triangleq \lambda N_i$  since there are  $N_i$  caches storing content  $i$ .

By Little's law, the long term utilization of the playout queue is:

$$\rho_i = \lambda_{Pi} \cdot \mathbf{E}[Y]/r_P = \lambda N_i \cdot \mathbf{E}[D] \cdot r_H/r_P \quad (7.12)$$

The necessary condition for this queue to be stable and ergodic is that  $\rho_i < 1$ ,  $\forall i \in \mathbb{K}$ . This condition is satisfied since  $\lambda_{Pi} \leq \lambda N_i$  and Definition 7.2.1 applies.

Let  $B_i^{(n)}$  (resp.  $I_i^{(n)}$ ) be the length of the  $n^{th}$  busy (resp. idle) period of the playout buffer for content  $i$ . When the queue is stable,  $(I_i^{(n)}, B_i^{(n)}), n \geq 1$  forms an alternating renewal process (as the queue regenerates at the end of each busy period). Let further  $I_i^{(n)} + B_i^{(n)}$  define a cycle, and  $P_I(t)$  the probability that the playout buffer is empty at time  $t$ . Since  $\mathbf{E}[I_i^{(n)} + B_i^{(n)}] < \infty$  (by stability), and the distribution of  $I_i^{(n)} + B_i^{(n)}$  is non-lattice, we can apply Theorem 3.4.4 of [199] to show that  $\lim_{t \rightarrow \infty} P_I(t) = \frac{\mathbf{E}[I_i]}{\mathbf{E}[I_i] + \mathbf{E}[B_i]}$ . However, by ergodicity it also holds that

$$1 - \rho_i = \lim_{t \rightarrow \infty} P_I(t) = \frac{\mathbf{E}[I_i]}{\mathbf{E}[B_i] + \mathbf{E}[I_i]} \quad (7.13)$$

Let further associate to each cycle a reward equal to the bits downloaded from the cellular infrastructure during that cycle, i.e. the reward in cycle  $n$  is equal to  $I_i^{(n)} \cdot r_P$  (we remind the reader that the download rate from the infrastructure is assumed to be equal to the playout rate  $r_P$ , see A.4). Consider now a video of duration  $T_i$  and remember that variable  $X_i$  is equal to the number of total bits downloaded from the infrastructure (see A.7). However, from the Renewal-Reward theorem (e.g. Theorem

3.6.1 in [199]) we have that

$$\lim_{T_i \rightarrow \infty} \frac{\mathbf{E}[X_i]}{T_i} = \frac{\mathbf{E}[I_i] \cdot r_P}{\mathbf{E}[B_i] + \mathbf{E}[I_i]}.$$

Combining Eq.(7.12) and Eq.(7.13) and the fact that the duration  $T_i = \frac{s_i}{r_P}$  for large  $s_i$ , we get that:

$$\lim_{s_i \rightarrow \infty} \left[ \mathbf{E}[X_i] - s_i \cdot \left( 1 - \lambda N_i \cdot \mathbf{E}[D] \cdot \frac{r_H}{r_P} \right) \right] = 0.$$

□

The above results states that as  $s_i$  becomes large, we can easily express  $\mathbf{E}[X_i]$  in closed form. We will use this result, as an approximation for finite sized content, to introduce the optimal cache allocation problem for the low helper density scenario. As mentioned earlier, in [94] we formally elaborate on the approximation error introduced for small files.

**Lemma 8** (Low Density Optimization). *For a low density scenario, the optimal content allocation is the solution to the following integer linear programming (ILP):*

$$\underset{N_i}{\text{maximize}} \sum_{i=1}^K \phi_i \cdot s_i \cdot N_i, \quad (7.14)$$

subject to the constraints of Eqs. (7.10) and (7.11).

*Proof.* Our objective is to minimize the total number of bits downloaded from  $\mathbb{I}$ . Based on Lemma 7, this is equal to:

$$\underset{N_i}{\text{minimize}} \sum_{i=1}^K \phi_i \cdot s_i \cdot \left( 1 - \lambda N_i \cdot \mathbf{E}[D] \cdot \frac{r_H}{r_P} \right),$$

which is equivalent to maximize the objective function of Eq. (7.14). □

**Theorem 2.1.** *The above optimization problem is NP-hard.*

The above theorem states that the problem is hard, due to the integer nature of variables  $N_i$  (by reduction to a bounded knapsack problem). We therefore propose the “modified greedy” algorithm Algorithm 1 for this problem:

**Lemma 9.** *Algorithm 1 guarantees a  $\frac{1}{2}$ -approximation for the above optimization problem.*

The proofs for Theorem 2.1 and Lemma 9 can be found in [94]. While Algorithm 1 is  $\frac{1}{2}$  optimal, in realistic scenarios it provides much better performance (as cache sizes are large and can fit several content). Also, this algorithm stores the most popular content in every vehicle (except in some very corner scenarios, which is why we refer to it as “modified”). This finding is interesting, because even if contacts do not overlap,

a node still sees multiple caches during the layout of a content. One would expect that these caches should store different content to maximize diversity. E.g., in the femto-caching setup, storing the most popular content in all caches is optimal only when caches are isolated, but it is suboptimal when a node has access to multiple caches [81]. We will see, that this most popular allocation is no longer efficient when vehicle density increases.

---

**Algorithm 1** Caching Algorithm for Low Density.

---

**Input:**  $s, \phi, C, H$

- 1:  $N \leftarrow \emptyset; j \leftarrow 1$
- 2: **while**  $\sum_{i=1}^j s_i \leq C$  **do**
- 3:      $N_i \leftarrow H$
- 4:      $j \leftarrow j + 1$
- 5: **end while**
- 6: **if**  $\sum_{i=1}^{j-1} \phi_i \cdot s_i \cdot N_i \leq \phi_j \cdot s_j \cdot N_j$  **then**
- 7:      $N \leftarrow \emptyset; N_j \leftarrow H$
- 8: **end if**
- 9: **return**  $N$

---

### 7.2.3 Generic Density Regime

We now consider a busy urban environment where contacts with different vehicles (with the same content) might overlap, i.e.,  $\lambda \cdot N_i \cdot E[D]$  is not small. If a user is downloading video  $i$  from node A, and the connection is lost (e.g., user or cache moves away), the user could just keep downloading from another node B storing  $i$ , also in range. Hence, as long as there is *at least* one cache with a copy within range (we denote this time interval with  $B_i$ ), the user will keep downloading content  $i$  at rate  $r_H$ <sup>8</sup>. We can then model these overlapping contacts with an extra G/G/ $\infty$  queue in front of the playout queue (as shown in Fig. 7.7). New vehicles arrive in the G/G/ $\infty$  queue with rate  $\lambda N_i$ , each staying for a random service time (corresponding to a contact duration with mean  $E[D]$ ) and independently of other cars. The number of jobs in the G/G/ $\infty$  queue is the number of cars concurrently within range of the user.

Hence, it is easy to see that: (a) the beginnings of busy periods of the queue on the left correspond to new bulk arrivals in the playout buffer (queue on the right), and (b) the mean duration of such busy periods, multiplied by  $r_H$ , corresponds to the (new) mean bulk size per arrival.

**Lemma 10.** *For a generic density scenario, the bulk arrival statistics into the playout*

---

<sup>8</sup>We ignore for now interruptions from switching between nodes. Such delays can be very small (e.g., in the order of few ms if vehicles are operating as LTE relays [247]). We consider switching and association delays in the simulation scenarios.

buffer are:

$$\lambda_{P_i} = \lambda N_i \cdot e^{-\lambda \mathbf{E}[D] \cdot N_i}, \quad (7.15)$$

$$\mathbf{E}[Y] = \frac{r_H}{\lambda N_i} \cdot \left( e^{\lambda \mathbf{E}[D] \cdot N_i} - 1 \right). \quad (7.16)$$

*Proof.* Let  $\{Z_{C_i}(t), t > 0\}$  be  $N_i$  identical and independent renewal processes corresponding to the inter-contact times with vehicles storing content  $i$  drawn from distribution  $f_C(t)$  with rate  $\lambda$  (see A.5). Let further  $\{Z_C(t), t > 0\}$  be the superposition of these processes. According to the Palm-Kintchine theorem [265],  $\{Z_C(t)\}$  approaches a Poisson process with rate  $\lambda N_i$  if  $N_i$  large and  $\lambda$  small. Thus, the G/G/ $\infty$  queue capturing overlapping meetings (Fig. 7.7 left) can be approximated by an M/G/ $\infty$  queue with arrival rate  $\lambda N_i$  and mean service time  $\mathbf{E}[D]$ .

The probability that there are 0 jobs in the system (idle probability) is  $e^{-\lambda \mathbf{E}[D] \cdot N_i}$  (this result is well known for M/M/ $\infty$  queue, but it also holds for generic contact durations by the service process insensitivity of the M/G/ $\infty$  queue [233]). Furthermore, by ergodicity, it holds that<sup>9</sup>:

$$\frac{\mathbf{E}[I_i]}{\mathbf{E}[B_i] + \mathbf{E}[I_i]} = e^{-\lambda \mathbf{E}[D] \cdot N_i}.$$

Since  $\mathbf{E}[I_i] = 1/\lambda N_i$ , solving for  $\mathbf{E}[B_i]$  gives us the expected busy period of the M/G/ $\infty$  queue and multiplying by  $r_H$  gives as the expected bulk size  $\mathbf{E}[Y]$  of Eq. (7.16).

Additionally, the beginnings of busy periods of the M/G/ $\infty$  queue correspond to (bulk) arrivals into the playout queue. The mean time between such arrivals is simply  $\mathbf{E}[B_i] + \mathbf{E}[I_i]$ . Hence, the arrival rate of bulks into the playout buffer is:

$$\lambda_{P_i} \triangleq \frac{1}{\mathbf{E}[B_i] + \mathbf{E}[I_i]} = \frac{1}{\frac{e^{\lambda \mathbf{E}[D] \cdot N_i} - 1}{\lambda N_i} + \frac{1}{\lambda N_i}} = \lambda N_i \cdot e^{-\lambda \mathbf{E}[D] \cdot N_i}.$$

□

**Lemma 11.** *For a generic density scenario the following expression is asymptotically tight as the content size  $s_i$  becomes large, when  $N_i < \frac{1}{\lambda \mathbf{E}[D]} \cdot \ln \left( \frac{r_H}{r_H - r_P} \right)$ :*

$$\lim_{s_i \rightarrow \infty} \left[ \mathbf{E}[X_i] - s_i \cdot \left[ 1 - \left( 1 - e^{-\lambda \mathbf{E}[D] \cdot N_i} \right) \cdot r_H / r_P \right] \right] = 0. \quad (7.17)$$

*Proof.* The proof follows directly by replacing  $\lambda_{P_i}$  and  $\mathbf{E}[Y]$  (from Lemma 10 above) into the proof of Lemma 7. The extra condition corresponds to the requirement of  $\rho_i < 1$  (for stationarity). Note that when this requirement is not satisfied, this essentially implies that the mean delivery capacity of the helper system is higher than  $r_P$ , and the infrastructure is essentially not needed for large enough content. □

We can now formulate the optimal cache allocation problem for the generic density scenario:

---

<sup>9</sup>We slightly abuse notation for these idle and busy periods of the queue on the left, while in the proof of Lemma 7 we used them for the idle and busy period of the playout buffer (queue on the right).

**Theorem 2.2** (Generic Density Optimization). *For a generic density scenario, the optimal content allocation is the solution to the following integer non-linear programming:*

$$\underset{N_i}{\text{minimize}} \sum_{i=1}^k \phi_i \cdot s_i \cdot e^{-\lambda \mathbf{E}[D] \cdot N_i}, \quad (7.18)$$

*subject to the constraints of Eqs. (7.10) and (7.11).*

*Proof.* Based on Lemma 11, the total number of bits downloaded from  $\mathbb{I}$  is equal to

$$\underset{N_i}{\text{minimize}} \sum_{i=1}^K \phi_i \cdot s_i \cdot \left[ 1 - \left( 1 - e^{-\lambda \mathbf{E}[D] \cdot N_i} \right) \cdot r_H / r_P \right],$$

which is equivalent to minimize the objective function of Eq. (7.18).  $\square$

**Corollary 6.** *The above optimization problem is NP-hard.*

The optimization problem described in Theorem 2.2 corresponds to a nonlinear bounded knapsack problem, which is known to be NP-hard. Branch-and-bound algorithms have been developed for such problems, but they are not efficient when the size of the problem increases. Instead, we will consider here the *continuous relaxation* of the above problem, i.e., assuming that  $N_i \in [0, H]$  are real random variables. It is easy to see that the continuous problem is convex, and we can solve it using standard Lagrangian methods [266]. In addition to reduced complexity, this relaxation allows us to also derive the optimal allocation in closed form thus offering additional insights (e.g., compared to discrete approximation algorithms).

**Corollary 7** (Generic Density Allocation). *For a generic density scenario with continuous allocation variables  $N_i \in [0, H]$ , the optimal cache allocation is given by*

$$N_i^* = \begin{cases} 0, & \text{if } \phi_i < L \\ \frac{1}{\lambda \mathbf{E}[D]} \cdot \ln \left( \frac{\lambda \mathbf{E}[D] \cdot \phi_i}{m_C} \right), & \text{if } L \leq \phi_i \leq U \\ H, & \text{if } \phi_i > U \end{cases}$$

where  $L \triangleq \frac{m_C}{\lambda \mathbf{E}[D]}$  and  $U \triangleq \frac{m_C}{\lambda \mathbf{E}[D]} \cdot e^{\frac{\min\{H, \hat{H}\}}{\lambda \mathbf{E}[D]}}$ , and  $m_C$  is an appropriate Lagrangian multiplier.

The proof is based on standard Lagrangian methodology and is omitted due to space limitations. It can be found in [94]. We use *randomized rounding* [267] on the content allocation of Corollary 7 to go back to an integer  $N_i$  allocation, which is a widely used approach for designing and analyzing such approximation algorithms. As argued earlier, the expected error is small when caches fit several content. To validate this, in Table 7.3 we compare the objective value from our allocation to the one corresponding to the continuous solution of Corollary 7 (we report the percentage of traffic offloaded). As the latter is a lower bound on the optimal solution of the discrete problem of Theorem 2.2, the actual performance gap is upper bounded by the values shown in Table 7.3.

Table 7.3: Estimated offloading gains of rounded allocation vs. continuous relaxation for different cache sizes (in percentage of the catalogue size).

<b>Cache size</b>	0, 02	0, 05	0, 10	0, 20
Rounded	33,148%	45,334%	54,959%	62,751%
Continuous	33,116%	45,323%	54,955%	62,750%

The above results are all based on the underlying assumption of queue stationarity. In simulation results, we can see that, in the scenario considered, this is valid for videos longer than 30min, but is already accurate enough for 15min videos. In any case, in [94] we have derived analytically this stationarity error as a function of file size. One could plug this into the objective as a correction factor and solve the resulting optimization for even better results. Nevertheless, for a realistic catalog of YouTube files, enough offloading gains can already be observed, as shown in the next section.

#### 7.2.4 Performance Evaluation

We perform simulations based on real traces for vehicle mobility and content popularity to confirm the advantages of the vehicular cloud and to validate our theoretical results. To do so we extend our previous simulator to also model *User Mobility*.: We use synthetic traces based on SLAW mobility model [268]. Specifically, according to the model, users move in a limited and defined area around popular places. The mobility is nomadic where users alternate between pauses (heavy-tailed distributed) and travelling periods at constant (but random) speed.

Inline with the proposed protocols (e.g., 802.11p, LTE ProSe), we consider two maximum communication ranges between  $\mathbb{U}$  and  $\mathbb{H}$  nodes: 100m (short range) or 200m (long range). As most wireless protocols implement some *rate adaptation* mechanism, our simulator also varies the communication rate according to the distance between the user and the mobile helper she is downloading from (while we consider the average download rate in the model). Given current wireless rates in the 802.\* family and that near future vehicles will probably carry high speed mobile access points, we use a *mean*  $r_H = 5\text{Mbps}$ . We also set  $r_P = 1\text{Mbps}$ , that approximates the streaming of a 720p video. Additionally, we implement an association setup mechanism according to [247] that introduces a delay of 2s to synchronize a UE with a vehicle (i.e., the download from a cache starts 2s after the beginning of the contact). Finally, we set the cache size per node  $C$  in the range  $0, 02\% - 0, 5\%$  of the total catalogue, which is an assumption that has also been used in [260, 83] (we use  $0, 1\%$  as default value). Unless otherwise stated, the mean video length is 1 hour (i.e., mean content size equal to 450MB).

#### 7.2.5 Caching Strategy Evaluation

We compare the following allocation policies:

- *Optimal*. Caching policy for generic density (see Corollary 7).

Table 7.4: Mobility statistics

Range	$\lambda$	$E[D]$
Short range	$0,964 \text{ day}^{-1}$	31,23 s
Long range	$2,83 \text{ day}^{-1}$	50,25 s

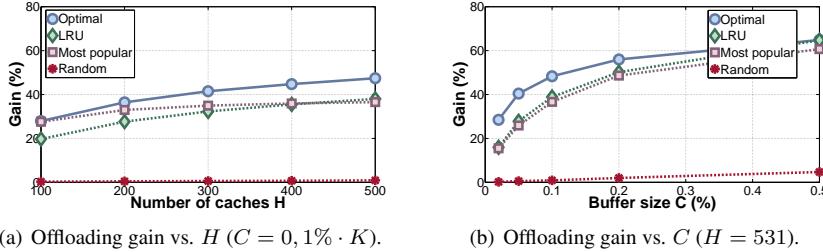


Figure 7.8: Data offloading gain as a function of cache number (a) and buffer size (b).

- *Most popular (MP)*. Caching policy for low density regime (see Algorithm 1).
- *Least Recently Used (LRU)*. Starting from a random initial allocation, it discards the least recently used item when there is a cache miss. Unlike the above two policies, LRU keeps updating the cache content, and thus could incur higher traffic on the backhaul (from  $\mathbb{I}$  to  $\mathbb{H}$  nodes).
- *Random*. Content is randomly allocated in  $\mathbb{H}$  nodes.

**Cache Density.** Fig. 7.8 depicts the fraction of data offloaded by the vehicular cloud as a function of vehicle density, buffer capacity and video length, assuming long range communication. Specifically, in Fig. 7.8(a) we vary the number of vehicles from 100 to 500. While the number of envisioned connected vehicles in the centre of San Francisco is expected to be much larger, it is really interesting to verify that a subset of them can still provide non-negligible offloading gains (more than 30%), which is important to promote the start up phase of the vehicular cloud. As proved in Section 7.2.2, the *MP* policy provides good performance in scenarios with low vehicle density: e.g., for  $H = 100$ , the offloading gain is almost equal to the *optimal*. Conversely, for a larger number of vehicles (that introduce contact overlaps), the gap between the two policies becomes higher than 10%. Finally, we observe that the LRU policy underperforms both policies, in sparse scenarios, while it converges to the *MP* policy in dense ones. This is reasonable, as LRU approximates an LFU policy (least frequently used), i.e., storing the most popular content when the popularity is stationary during the considered window. While in some scenarios LRU is actually not far from the results of our allocation strategy, it should be highlighted that we ignore the extra backhaul cost due to the frequent cache updates for LRU (that *optimal* does not have). In fact, simulations have shown that this ‘‘additional seeding’’ in LRU considerably degrades performance.

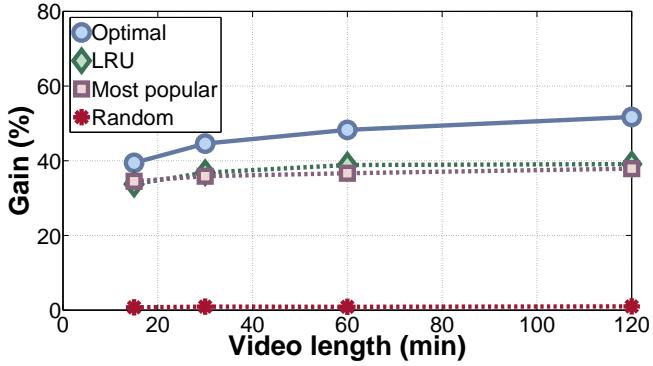


Figure 7.9: Offloading gain vs. file size.

**Buffer Capacity.** In Fig. 7.8(b) we vary the cache storage per vehicle between 0,02% and 0,5% of the catalogue (where  $H = 531$  and mean video length is 1h). Interestingly, the smallest storage capacity still achieves considerable performance gains. E.g., if one considers an entire Torrent or Netflix catalogue ( $\sim 3\text{PB}$ ), a mobile helper capacity of about 500GB (0,02%) already suffices to offload 30% of the total traffic. Moreover, for small  $C$  ( $< 0,05\%$ ), *optimal* almost doubles the gain (from 18% to 30%) compared to the other policies. Finally, the random policy ignores the skewed Internet content popularity, and thus performs very poorly in all scenarios.

**Video Length.** Stationary regime analysis can be considered as a good approximation when there is a reasonable number of busy + idle buffer playout periods, such that the transitory phase becomes negligible. In order to increase the number of these periods, the average content size needs to be large enough, given fixed mobility statistics. Fig. 7.10 shows the fraction of data offloaded by the vehicular cloud for set of content of the same length. One can observe that the gain increases with video length. However, this increase is only marginal in the majority of the scenarios: in fact, even small content (15 minutes) provides a gain which is comparable to the asymptotic gain, validating the stationary regime analysis.

### 7.2.6 Mobile vs. Static Helpers

In this section, we verify the pertinence of the vehicular cloud, that is based on mobile helpers, against the femtocaching framework described in [81], that is based on static SCs equipped with storage. In this second network, SC helpers are distributed in the considered area proportionally to the popularity density, i.e., areas with a higher number of requests have higher SC density (this is a common operator policy since SCs are deployed to alleviate traffic ‘hotspots’). Users move according to the previously described SLAW trace, and they can also download video chunks at low cost from a nearby SC if it stores the requested video. Content is allocated using the algorithm described in [81]. We consider two densities of SCs:

- *Femtocaching (equal number of helpers).* From the analysis of the Cabspotting trace,

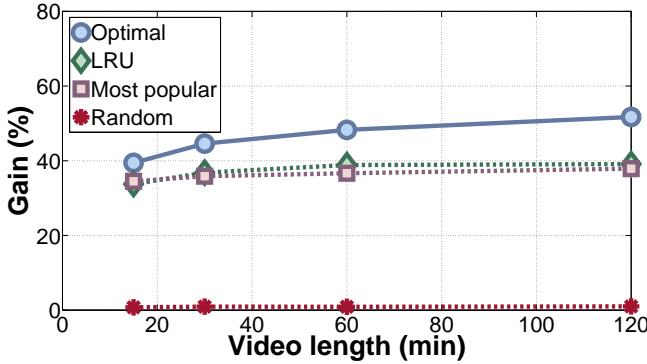


Figure 7.10: Offloading gain vs. file size.

the average number of vehicles *simultaneously* inside the area considered is lower than 200. In order to have a fair comparison with the vehicular cloud, we set the number of SCs to 200.

- *Low cost femto (equal cost).* The CAPEX of a SC consists of base station equipment, professional services (planning, installation and commissioning), and back-haul transmission equipment. This cost may range from 1k euros for a femtocell to 20-30k euros for a microcell [269]. In the proposed vehicular cloud, the equipment might be pre-installed, and a large part of the OPEX could also be avoided as explained earlier. In fact, a first implementation of a similar vehicular cloud, where vehicles act as proxies, has shown a 10-fold cost reduction compared to SCs [270]. We therefore also consider a sparser deployment that equalizes the total cost where we set to 50 the number of SCs.

Fig. 7.11 compares vehicular cloud and femtocaching in terms of data offloaded. We also simulate a femtocaching scenario with the *MP* policy. As expected, gains provided by the vehicular cloud are considerably higher than femtocaching for both short and (mainly) long range communications. This result is even more interesting considering the cost: in fact, storing content in vehicles permits almost 2,5 times higher gains than femtocaching with equal cost.

### 7.3 Soft Cache Hits: Impact and Optimization of Related Content Recommendation

#### 7.3.1 Introduction

##### Background and Motivation

In order to overcome the issue of limited cache size, additional caching gains have been sought by researchers, as already discussed, increasing the “effective” cache size visible to each user. This could be achieved by: (a) *Coverage overlaps*, where each user

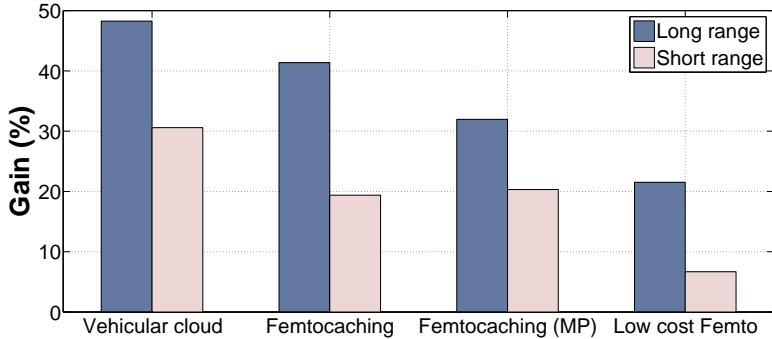


Figure 7.11: Data offloaded for vehicular cloud vs. femto-caching.

is in the range of multiple cells, thus having access to the aggregate storage capacity of these cells, as in the femto-caching framework [81, 271]. (b) *Coded caching*, where collocated users overhearing the same broadcast channel may benefit from cached content in other users’ caches [255]. (c) Last but not least, *delayed content access*, where a user might wait up to a TTL for her request, during which time more than one cache (fixed [272] or mobile [142, 273, 92, 274]) can be encountered, as detailed in the previous sections. Nevertheless, each of these ideas still comes with limitations, e.g., due to high enough cell density required for (a), sub-packetization complexity in (b), and imposed delays in (c).

To get around this seeming impasse, we propose to *move away from trying to satisfy every possible user request, and instead try to satisfy the user*. In an Internet which is becoming increasingly entertainment-oriented, one can make the following observations: (a) a user’s content requests are increasingly influenced by various recommendation systems (YouTube, Netflix, Spotify, or even Social Networks) [275]; (b) some related contents (e.g. two recent NBA games, two funny cat clips) might have similar utility for a user; in micro-economic terms, these are often called *substitute goods*; we will use the terms *alternative*, *related*, and *substitute* content interchangeably.

### Soft Cache Hits: Idea and Implications

Based on these observations, we envision a system where “soft cache hits” can be leveraged to improve caching performance. As one example, consider the following, for the case of YouTube (or, any similar service). A user requests a content, e.g., by typing on the YouTube search bar, and the content is not available in the local cache(s); Then, a local app proxy located near the cache and having knowledge of the cached contents (e.g. a YouTube recommender code running at a Multi-access Edge Computing (MEC) server [276]), could recommend a set of *related contents* that are locally available. If the user prefers or accepts (under some incentives; see below) one of these contents, instead of the one she initially typed/requested, a soft cache hit (SCH) occurs, and an expensive remote access is avoided. We will use the term *soft cache hit* to describe such scenarios.

Of course, appropriate incentives would be needed to nudge a user towards substitute content. While perhaps a somewhat radical concept in today’s ecosystem, we believe there are a number of scenarios where soft cache hits are worth considering, as *they could benefit both the user and the operator*. (i) A *cache-aware recommendation* plugin to an existing application could, for example, let a user know that accessing the original content  $X$  is only possible at low quality and might be choppy, freeze, etc., due to congestion, while related contents  $A, B, C, \dots$  could be streamed at high resolution, as shown in Fig. 7.12(a). (ii) Alternatively, the operator could activate this system only during predicted congestion periods, while giving some incentives to users to accept the alternative contents during that time (e.g., *zero-rating* services [95, 96]). (iii) In some cases, the operator might even “enforce” an alternative (but related) content (see Fig. 7.12(b)), e.g., offering low rate plans with higher data quotas with the agreement that, during congestion, only locally cached content can be served.

While, in the above cases, a potential unwillingness or utility loss needs to be counter-balanced with appropriate incentives, this is not always the case. Sometimes soft cache hits could be leveraged in a relatively seamless manner without the potential psychological impact on user quality of experience (QoE) due to *conscious* content replacement<sup>10</sup>. For example, after a user watches a video  $X$ , the recommendation system could re-order its list of recommendations (among related contents of roughly equal similarity to  $X$ ) to favor a cache hit in the next request, without the user being aware of this change or liking the recommended contents less. Such systems have already been considered, and would be complementary to our proposal [278, 279]. A similar case could be made for online radio type of apps (like last.fm, Pandora, Spotify, etc.). While a lot more can be said about each of the above preliminary incentive ideas, and plenty more thinking might be needed to go from these to concrete business cases, we believe these suffice to motivate an investigation of the potential impact of soft cache hits.

While soft cache hits could provide some benefits on top of an existing caching policy, a first key observation is that the optimal caching policy might differ, sometimes radically, when soft cache hits are allowed. As a simple example, consider a single cache with a tiny content catalog with contents A, B, C of popularities 3, 2, 2, respectively (e.g. number of requests per minute). If the cache could fit only a single content, traditional caching will choose to store the most popular content (A), leading to a cache hit ratio of  $3/(3 + 2 + 2)$ , approx. 43%. However, assume we knew that 1 out of 2 users requesting A, would be willing to watch content C instead (e.g. because C is highly related to A, and available locally at HD). Same for users requesting content B. Then, caching content C would satisfy all requests for C (2), half the requests for B ( $0.5 \cdot 2$ ), and half the requests for A ( $0.5 \cdot 3$ ), leading to a cache hit ratio of  $4.5/7$ , approximately 64% (an almost 50% improvement over the standard policy). This simple example motivates the significant potential of the approach, but also the need for a fundamental reconsideration of caching policies, even in relatively simple networking setups. Finally, while this simple example might tempt the reader to think that the new optimal policy is simply to (re-)rank contents based on *total* hit rate each can achieve (including SCHs), and then apply standard policies (e.g. picking the high-

---

<sup>10</sup>A user that has already chosen a content might over-value her original choice and feel unhappy to swap it to an objectively equally interesting content. This effect is somewhat akin to the well-known *endowment effect* from Behavioral Economics [277].

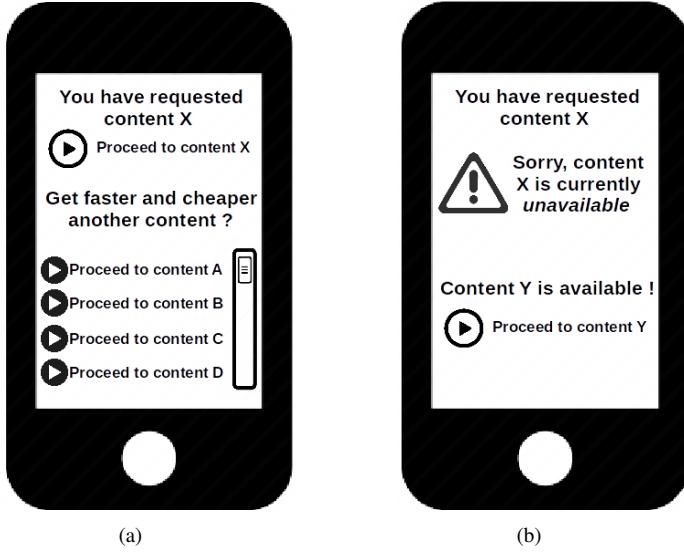


Figure 7.12: Mobile app example for *Soft Cache Hits*: (a) related content *recommendation* (that the user might not accept), and (b) related content *delivery*.

est ranked ones), in fact we will show that the optimal policy is a hard combinatorial (cover) problem.

### Contributions

The main contributions of this work are summarized as follows.

*Soft Cache Hits (SCH) concept:* We introduce the novel concept of soft cache hits. To our best knowledge, this is the first time that this idea has been applied to edge caching for cellular networks (besides our own preliminary work [280]).

*Soft Cache Hits (SCH) model:* We propose a generic model for mobile edge caching with soft cache hits that can capture a number of interesting content substitution scenarios (e.g. both Fig. 7.12(a) and Fig. 7.12(b)) and is versatile enough to apply on top of both non-cooperative (i.e. single cache) and cooperative caching frameworks [81], as well as various networking assumptions.

*Analytical Investigation:* We prove that the problem of optimal edge caching with SCH is NP-hard even when considering a single cache only. This is in stark contrast to the standard case without SCH. We then prove that, despite the increased complexity, the generic problem of femto-caching with SCH still exhibits properties that can be taken advantage of to derive efficient approximation algorithms with provable performance.

*Trace-based Validation:* We corroborate our SCH proposal and analytical findings through an extended evaluation on 5 real datasets containing information about related content, demonstrating that promising additional caching gains could be achieved in practice.

As a final remark, it is important to stress that *we do not propose to modify the recommendation systems themselves* (unlike [278, 279], for example). Instead, *our focus is on the caching policy side*, using the output of a state-of-art recommendation system for the respective content type as input to our problem (this will be further clarified in Section 7.3.2). Of course, a content provider with a recommendation system can benefit from our approach to optimize its caching policies, or even modify its recommendations to incorporate soft cache hits. For example, upon peak hours, a content provider can carefully “steer” recommendations to optimize the network performance and user experience (e.g., from lower latency). Moreover, jointly optimizing both the caching and the recommendation sides of the problem could offer additional benefits. We defer this to future work. Overall, we believe that such a convergence between recommendation and caching systems is quite timely, given that dividing lines between Mobile Network Operators (MNO) and content providers are becoming more blurry, due to architectural developments like Multi-access Edge Computing (MEC) [276] and RAN Sharing [100].

In the following section we introduce the problem setup and our soft cache hits model corresponding to the example application of Fig. 7.12(a). In Section 7.3.4 we formulate and analyze the problem of edge caching with SCH for a single cache, and propose efficient caching algorithms. Then, in Section 7.3.7, we generalize the problem, analysis, and algorithms to the femto-caching case. In Section 7.3.8 we extend our model to capture scenarios as in the example application of Fig. 7.12(b), and show that our analytic findings are applicable to these scenarios as well. The performance evaluation is presented in Section 7.3.9. Finally, we discuss related work and future research directions in Section 7.3.13.

### 7.3.2 Problem Setup

#### Network and Caching Model

**Network Model:** Our network consists of a set of users  $\mathcal{N}$  ( $|\mathcal{N}| = N$ ) and a set of SCs (or, *helpers*)  $\mathcal{M}$  ( $|\mathcal{M}| = M$ ). Users are mobile and the SCs with which they associate might change over time. Since the caching decisions are taken in advance (e.g., the night before, as in [81, 271], or once per few hours or several minutes), it is hard to know the exact SC(s) each user will be associated at the time she requests a content. To capture user mobility, we propose a more generic model than the fixed bipartite graph of [81]:

$$q_{ij} \doteq \text{Prob}\{\text{user } i \text{ in range of SC } j\},$$

or, equivalently,  $q_{ij}$  is the percentage of time a user  $i$  spends in the coverage of SC  $j$ . Hence, deterministic  $q_{ij}$  ( $\in \{0, 1\}$ ) captures the static setup of [81], while uniform  $q_{ij}$  ( $q_{ij} = q, \forall i, j$ ) represents the other extreme (no advance knowledge).

**Content Model:** We assume each user requests a content from a catalogue  $\mathcal{K}$  with  $|\mathcal{K}| = K$  contents. A user  $i \in \mathcal{N}$  requests content  $k \in \mathcal{K}$  with probability  $p_k^i$ .<sup>11</sup> We will initially assume that all contents have the same size, and relax the assumption later.

---

<sup>11</sup>This generalizes the standard femto-caching model [81] which assumes same popularity per user. We can easily derive such a popularity  $p_k$  from  $p_k^i$ .

**Cache Model (Baseline):** We assume that each SC/helper is equipped with storage capacity of  $C$  contents (all our proofs hold also for different cache sizes). We use the integer variable  $x_{kj} \in \{0, 1\}$  to denote if content  $k$  is stored in SC  $j$ . In the traditional caching model (baseline model), if a user  $i$  requests a content  $k$  which is stored in some nearby SC, then the content can be accessed directly from the local cache and a *cache hit* occurs. This type of access is considered “cheap”, while a *cache miss* leads to an “expensive” access (e.g., over the SC backhaul and core network).

For ease of reference, the notation is summarized in Table 7.5.

Table 7.5: Important Notation

$\mathcal{N}$	set of users ( $ \mathcal{N}  = N$ )
$\mathcal{M}$	set of SCs / helpers ( $ \mathcal{M}  = M$ )
$C$	storage capacity of a SC
$q_{ij}$	probability user $i$ in range of SC $j$
$\mathcal{K}$	set of contents ( $ \mathcal{K}  = K$ )
$p_k^i$	probability user $i$ to request content $k$
$x_{kj}$	$k$ is stored in SC $j$ ( $x_{kj} = 1$ ) or not ( $x_{kj} = 0$ )
$u_{kn}^i$	utility of content $n$ for a user $i$ requesting content $k$
$F_{kn}(x)$	distribution of utilities $u_{kn}^i$ , $F_{kn}(x) = P\{u_{kn}^i \leq x\}$
$u_{kn}$	avg. utility for content pair $\{k, n\}$ (over all users)
$s_k$	size of content $k$

### 7.3.3 Soft Cache Hits

Up to this point the above model describes a baseline setup similar to the popular femto-caching framework [81]. The main departure in this paper is the following.

**Related Content Recommendation:** When a user consumes a content (or initially requests a content) that is not found in the local cache, we assume that an app proxy (e.g. YouTube, Netflix, Spotify), collocated or near this cache, looks at the list of contents its recommendation system deems related to the currently consumed content (or the initial request), checks which of them are available in the local cache, and recommends them to the user (see the example in Fig. 7.12(a)). If a user selects to consume next one of them, instead of the original content, a (*soft*) *cache hit* occurs, otherwise there is a cache miss and the network must fetch and deliver the original content.

Below, we first propose a soft cache hit model that captures the scenario of Fig. 7.12(a). We will use this model throughout Sections 7.3.4 and 7.3.7, to develop most of our theory. However, in Section 7.3.8, we will modify our model to also analyze the scenario of Fig. 7.12(b), which we will refer to as *Related Content Delivery*.

**Definition 7.3.1.** A user  $i$  that requests a content  $k$  that is not available, accepts a recommended content  $n$  with probability  $u_{kn}^i$ , where  $0 \leq u_{kn}^i \leq 1$ , and  $u_{kk}^i = 1, \forall i, k$ .

These utilities/probabilities (in the remainder we use these terms interchangeably) define a content relation matrix  $\mathbf{U}^i = \{u_{kn}^i\}$  for each user. They could be estimated from past statistics and/or user profiles, and are closely related to the output of the recommender for that user and that content app. For example, if a collaborative filtering

algorithm suggested that the cosine distance [281] between files  $k$  and  $n$  for user  $i$  is 0.5, we could set  $u_{kn}^i = 0.5^{12}$ .

In some cases, the system might have a coarser view of these utilities (e.g., item-item recommendation [282]). We develop our theory and results for the most generic case of Definition 7.3.1, but we occasionally refer to the following two subcases, which might appear in practice:

**Sub-case 1:** The system does not know the exact utility  $u_{kn}^i$  for each node  $i$ , but only how they are distributed among all nodes, i.e., the distributions  $F_{kn}(x) \equiv P\{u_{kn}^i \leq x\}$ .

**Sub-case 2:** The system knows only the *average utility*  $u_{kn}$  per content pair  $\{k, n\}$ .

### 7.3.4 Single Cache with Soft Cache Hits

In order to better understand the impact of the related content matrices  $\mathbf{U}^i$  on caching performance, we first consider a scenario where a user  $i$  is served by a single small cell, i.e., each user is associated to exactly one SC, but we might still not know in advance which. Such a scenario is in fact relevant in today’s networks, where the cellular network first chooses a single SC to associate a user to (e.g., based on signal strength), and then the user makes its request [247]. In that case, we can optimize each cache independently. We can also drop the second index for both the storage variables  $x_{kj}$  and connectivity variables  $q_{ij}$ , to simplify notation.

In the remainder, we select the cache hit ratio (CHR) as the basic performance metric, similarly to the majority of the related work. However, the analysis for CHR maximization can be generalized to utility maximization [283], where “utility” can be the content access cost or delay, energy consumption, etc.

#### Soft Cache Hit Ratio

A request (from a user to a SC/helper) for a content  $k \in \mathcal{K}$  would result in a (standard) cache hit only if the SC/helper stores the content  $k$  in its cache, i.e., if  $x_k = 1$ . Hence, the (baseline) *cache hit ratio* for this request is simply

$$CHR(k) = x_k$$

If we further allow for soft cache hits, the user might be also satisfied by receiving a different content  $n \in \mathcal{K}$ . The probability of this event is, by Definition 7.3.1, equal to  $u_{kn}^i$ . The following Lemma derives the total cache hit ratio in that case.

---

<sup>12</sup>Going from a content relation value to a value for the willingness of a user to accept that related content arguably entails some degree of subjectivity, given that this also depends on the amount and type of incentives offered to the user. Nevertheless, it is clear that whatever the actual value of  $u_{kn}^i$ , it will be some function of and positively correlated to underlying content relevance, which can be readily available from the respective recommendation system.

**Lemma 12** (Soft Cache Hit Ratio (SCHR)). *Let SCHR denote the expected cache hit ratio for a single cache (including regular and soft cache hits), among all users. Then,*

$$SCHR = \sum_{i=1}^N \sum_{k=1}^K p_k^i \cdot q_i \cdot \left( 1 - \prod_{n=1}^K (1 - u_{kn}^i \cdot x_n) \right). \quad (7.19)$$

*Proof.* The probability of satisfying a request for content  $k$  by user  $i$  with related content  $n$  is  $P\{n|k, i\} = u_{kn}^i \cdot x_n$ , since  $u_{kn}^i$  gives the probability of acceptance (by definition), and  $x_n$  denotes if content  $n$  is stored in the cache (if the content is not stored, then  $P\{n|k, i\} = 0$ ). Hence, it follows easily that the probability of a *cache miss*, when content  $k$  is requested by user  $i$ , is given by<sup>13</sup>  $\prod_{n=1}^K (1 - u_{kn}^i \cdot x_n)$ . The complementary probability, defined as the *soft cache hit ratio* (SCHR), is then

$$SCHR(i, k, \mathbf{U}) = 1 - \prod_{n=1}^K (1 - u_{kn}^i \cdot x_n). \quad (7.20)$$

Summing up over all users that might be associated with that BS (with probability  $q_i$ ) and all contents that might be requested ( $p_k^i$ ) gives us Eq.(7.19).  $\square$

Lemma 12 can be easily modified for the sub-cases 1 and 2 of Definition 7.3.1 presented in Section 7.3.3. We state the needed changes in Corollary 8.

**Corollary 8.** *Lemma 12 holds for the the sub-cases 1 and 2 of Definition 7.3.1, by substituting in the expression of Eq. (7.19) the term  $u_{kn}^i$  with*

$$u_{kn}^i \rightarrow E[u_{kn}^i] \equiv \int (1 - F_{kn}(x)) dx \quad (\text{for sub-case 1}) \quad (7.21)$$

$$u_{kn}^i \rightarrow u_{kn} \quad (\text{for sub-case 2}) \quad (7.22)$$

*Proof.* The proof can be found in [284]  $\square$

### 7.3.5 Optimal SCH for Equal Content Sizes

The (soft) cache hit ratio depends on the contents that are stored in a SC/helper. The network operator can choose the storage variables  $x_k$  to maximize SCHR by solving the following optimization problem.

**Optimization Problem 1.** *The optimal cache placement problem for a single cache with soft cache hits and content relations described by the matrix  $\mathbf{U}^i = \{u_{kn}^i\}$ ,  $\forall i \in \mathcal{N}$ , is*

$$\underset{\substack{x = \{x_1, \dots, x_K\}}}{\text{maximize}} \quad f(X) = \sum_{i=1}^N \sum_{k=1}^K p_k^i \cdot q_i \cdot \left( 1 - \prod_{n=1}^K (1 - u_{kn}^i \cdot x_n) \right) \quad (7.23)$$

$$\text{s.t.} \quad \sum_{k=1}^K x_k \leq C. \quad (7.24)$$

---

<sup>13</sup>To simplify our analysis, throughout our proofs we will assume that the user is informed about *all cached contents  $n$  with non-zero relevance  $u_{kn}^i$  to the original content  $k$* . In practice, only a limited number of them would be recommended (e.g. the  $N$  most related among the cached ones, as in [279])). Our analysis also holds for this case, with limited modifications.

In the following, we prove that the above optimization problem is NP-hard (Lemma 13), and study the properties of the objective function Eq. (7.23) (Lemma 14) that allow us to design an efficient approximate algorithm (Algorithm 2) with provable performance guarantees (Theorem 3.1).

**Lemma 13.** *The Optimization Problem 1 is NP-hard.*

**Lemma 14.** *The objective function of Eq.(7.23) is submodular and monotone (non-decreasing).*

The proofs for the previous two Lemmas can be found in Appendix 7.5.

We propose Algorithm 2 as a greedy algorithm for Optimization Problem 1: to select the contents to be stored in the cache, we start from an empty cache (line 1), and start filling it (one by one) with the content that increases the most the value of the objective function (line 4), till the cache is full. The computation complexity of the algorithm is  $O(C \cdot K)$ , since the loop (lines 2-6) denotes  $C$  repetitions, and in each repetition the objective function is evaluated  $y$  times, where  $K \geq y \geq K - C + 1$ . An efficient implementation of the step in line 4 can be based on the method of *lazy evaluations of the objective function*; due to space limitations, we refer the interested reader to [285].

The following theorem gives the performance bound for Algorithm 2.

**Theorem 3.1.** *Let  $OPT$  be the optimal solution of the Optimization Problem 1, and  $S^*$  the output of Algorithm 2. Then, it holds that*

$$f(S^*) \geq \left(1 - \frac{1}{e}\right) \cdot OPT \quad (7.25)$$

*Proof.* Lemma 14 shows that the Optimization Problem 1 belongs to the generic category of maximization of submodular and monotone functions (Eq. (7.23)) with a cardinality constraint (Eq. (7.24)). For such problems, it is known that the greedy algorithm achieves (in the worst case) a  $(1 - \frac{1}{e})$ -approximation solution [286, 285].  $\square$

While the above is a strict worst case bound, it is known that greedy algorithms perform quite close to the optimal in most scenarios. In Sec. 7.3.9 we show that this simple greedy algorithm can already provide interesting performance gains.

### 7.3.6 Optimal SCH for Different Content Sizes

Till now we have assumed that all contents have equal size. In practice, each content has a different size  $s_k$  and the capacity  $C$  of each cache must be expressed in Bytes. Additionally, if a user requests a video of duration  $X$  and she should be recommended an alternative one of similar duration  $Y$  (note that similar duration does not always mean similar size). While the latter could still be taken care of by the recommendation system (our study of a real dataset in Sec. 7.3.9 suggests that contents of different sizes might still be tagged as related), we need to revisit the optimal allocation problem: the capacity constraint of Eq.(7.24) is no longer valid, and Algorithm 2 can perform arbitrarily bad [285].

---

**Algorithm 2**  $(1 - \frac{1}{e})$ -approximation Greedy Algorithm for Optimization Problem 1.  
computation complexity:  $O(C \cdot K)$

---

**Input:** utility  $\{u_{kn}^i\}$ , content demand  $\{p_k^i\}$ , mobility  $\{q_i\}$ ,  $\forall k, n \in \mathcal{K}, i \in \mathcal{N}$

- 1:  $S_0 \leftarrow \emptyset; t \leftarrow 0$
- 2: **while**  $t < C$  **do**
- 3:    $t \leftarrow t + 1$
- 4:    $n \leftarrow \underset{\ell \in \mathcal{K} \setminus S_{t-1}}{\operatorname{argmax}} f(S_{t-1} \cup \{\ell\})$
- 5:    $S_t \leftarrow S_{t-1} \cup \{n\}$ ,
- 6: **end while**
- 7:  $S^* \leftarrow S_t$
- 8: **return**  $S^*$

---

**Optimization Problem 2.** The optimal cache placement problem for a single cache with soft cache hits and variable content sizes, and content relations described by the matrix  $\mathbf{U}^i = \{u_{kn}^i\}$ ,  $\forall i \in \mathcal{N}$ , is

$$\underset{X=\{x_1, \dots, x_K\}}{\operatorname{maximize}} \quad f(X) = \sum_{i=1}^N \sum_{k=1}^K p_k^i \cdot q_i \cdot \left( 1 - \prod_{j=1}^K \left( 1 - u_{kj}^i \cdot x_n \right) \right) \quad (7.26)$$

$$\text{s.t.} \quad \sum_{k=1}^K s_k x_k \leq C. \quad (7.27)$$

*Remark:* Note that the objective is still in terms of cache hit ratio, and does not depend on content size. This could be relevant, e.g., when the operator is doing edge caching to reduce *access latency* to contents (latency is becoming a core requirement in 5G).

The problem is a set cover problem variant with a knapsack type constraint. We propose the approximation Algorithm 3 for this problem, which is a “fast greedy” algorithm (based on a modified version of the greedy Algorithm 2) and has complexity  $O(K^2)$ .

### Theorem 3.2.

- (1) The Optimization Problem 2 is NP-hard.
- (2) Let  $OPT$  be the optimal solution of the Optimization Problem 2, and  $S^*$  the output of Algorithm 3. Then, it holds that

$$f(S^*) \geq \frac{1}{2} \left( 1 - \frac{1}{e} \right) \cdot OPT \quad (7.28)$$

*Proof.* A sketch of the proof can be found in [284].  $\square$

In fact, a polynomial algorithm with better performance  $(1 - \frac{1}{e})$ -approximation could be described, based on [287]. However, the improved performance guarantees come with a significant increase in the required computations,  $O(K^5)$ , which might not be feasible in a practical scenario when the catalog size  $K$  is large. We therefore just state its existence, and do not consider the algorithm further in this paper (the algorithm can be found in [288]).

---

**Algorithm 3**  $\frac{1}{2} \cdot (1 - \frac{1}{e})$ -approximation Algorithm for Optimization Problem 2.  
computation complexity:  $O(K^2)$

---

```

Input: utility  $\{u_{kn}^i\}$ , content demand  $\{p_k^i\}$ , content size  $\{s_k\}$ , mobility  $\{q_i\}$ ,  $\forall k, n \in \mathcal{K}, i \in \mathcal{N}$ 
1:  $S^{(1)} \leftarrow \text{MODIFIEDGREEDY}(\emptyset, [s_1, s_2, \dots, s_k])$ 
2:  $S^{(2)} \leftarrow \text{MODIFIEDGREEDY}(\emptyset, [1, 1, \dots, 1])$ 
3: if  $f(S^{(1)}) > f(S^{(2)})$  then
4:    $S^* \leftarrow S^{(1)}$ 
5: else
6:    $S^* \leftarrow S^{(2)}$ 
7: end if
8: return  $S^*$ 

9: function MODIFIEDGREEDY( $S_0, [w_1, w_2, \dots, w_k]$ )
10:    $\mathcal{K}^{(1)} \leftarrow \mathcal{K}; c \leftarrow 0; t \leftarrow 0$ 
11:   while  $\mathcal{K}^{(1)} \neq \emptyset$  do
12:      $t \leftarrow t + 1$ 
13:      $n \leftarrow \underset{\ell \in \mathcal{K} \setminus S_{t-1}}{\operatorname{argmax}} \frac{f(S_{t-1} \cup \{\ell\})}{w_\ell}$ 
14:     if  $c + w_n \leq C$  then
15:        $S_t \leftarrow S_{t-1} \cup \{n\}$ 
16:        $c \leftarrow c + w_n$ 
17:     else
18:        $S_t \leftarrow S_{t-1}$ 
19:     end if
20:      $\mathcal{K}^{(1)} \leftarrow \mathcal{K}^{(1)} \setminus \{n\}$ 
21:   end while
22:   return  $\leftarrow S_t$ 
23: end function

```

---

### 7.3.7 Femtocaching with Related Content Recommendation

Building on the results and analytical methodology of the previous section for the optimization of a single cache with soft cache hits, we now extend our setup to consider the complete problem with cache overlaps (referred to as “femtocaching” [81]). Note, however, that we *do* consider user mobility, through variables  $q_{ij}$ , unlike previous works in this framework that often assume static users. Due to space limitations, we focus on the case of fixed content sizes.

In this scenario, a user  $i \in \mathcal{N}$  might be covered by more than one SCs/helpers  $j \in \mathcal{M}$ , i.e.  $\sum_j q_{ij} \geq 1, \forall i$ . A user is satisfied, if she receives the requested content  $k$  or *any* other related content (that she will accept), from *any* of the SCs/helpers within range. Hence, similarly to Eq. (7.20), the total cache hit ratio SCHR (that includes regular and soft cache hits) is written as

$$SCHR(i, k, \mathbf{U}) = 1 - \prod_{j=1}^M \prod_{n=1}^K (1 - u_{kn}^i \cdot x_{nj} \cdot q_{ij}) \quad (7.29)$$

since for a cache hit a user  $i$  needs to be in the range of a SC  $j$  (term  $q_{ij}$ ) that stores the content  $n$  (term  $x_{nj}$ ), and accept the recommended content (term  $u_{kn}^i$ ).

Considering (i) the request probabilities  $p_k^i$ , (ii) every user in the system, and (iii) the capacity constraint, gives us the following optimization problem.

**Optimization Problem 3.** *The optimal cache placement problem for the femtocaching scenario with soft cache hits and content relations described by  $\mathbf{U}^i = \{u_{kn}^i\}, \forall i \in \mathcal{N}$ , is*

$$\begin{aligned} & \underset{X=\{x_{11}, \dots, x_{KM}\}}{\text{maximize}} f(X) = \\ &= \sum_{i=1}^N \sum_{k=1}^K p_k^i \cdot \left( 1 - \prod_{j=1}^M \prod_{n=1}^K \left( 1 - u_{kn}^i \cdot x_{nj} \cdot q_{ij} \right) \right), \end{aligned} \quad (7.30)$$

$$\text{s.t. } \sum_{k=1}^K x_{kj} \leq C, \quad \forall j \in \mathcal{M}. \quad (7.31)$$

The following lemma states the complexity of the above optimization problem, as well as its characteristics that allow us to design an efficient approximation algorithm.

**Lemma 15.**

- (1) *The Optimization Problem 3 is NP-hard,*
- (2) *with submodular and monotone (non-decreasing) objective function (Eq. (7.30)) and a matroid constraint (Eq. (7.31)).*

*Proof.* We prove Lemma 15 by extending the basic ideas of the single-cache case, and following a similar methodology as in the proofs of Lemmas 13 and 14; the detailed proof is given in [288].  $\square$

Lemma 15 states that the Optimization Problem 3 is a maximization problem with a submodular function and a matroid constraint. For this type of problems, a greedy algorithm can guarantee an  $\frac{1}{2}$ -approximation of the optimal solution [285]. The greedy algorithm is similar to Algorithm 2 and is of computational complexity  $O(K^2 M^2)$ ; i.e., instead of considering only contents in the allocation, now tuples {content, helper} need to be greedily allocated until the caches of helpers are full (for a detailed pseudocode of the algorithm, we refer the reader to [288]).

**Theorem 3.3.** *Let  $OPT$  be the optimal solution of the Optimization Problem 3, and  $S^*$  the output of the greedy algorithm. Then, it holds that*

$$f(S^*) \geq \frac{1}{2} \cdot OPT \quad (7.32)$$

Submodular optimization problems have received considerable attention recently, and a number of sophisticated approximation algorithms have been considered (see, e.g., [285] for a survey). For example, a better  $(1 - \frac{1}{e})$ -approximation (with increased computation complexity though) can be found following the “multilinear extension” approach [289], based on a continuous relaxation and pipage rounding. A similar approach has also been followed in the original femto-caching paper [81]. Other methods

also exist that can give an  $(1 - \frac{1}{e})$ -approximation [290]. Nevertheless, minimizing algorithmic complexity or optimal approximation algorithms are beyond the scope of this paper. Our goal instead is to derive fast and efficient algorithms (like greedy) that can handle the large content catalogues and content related graphs  $\mathbf{U}$ , and compare the performance improvement offered by soft cache hits. The worst-case performance guarantees offered by these algorithms are added value.

### 7.3.8 Femtocaching with Related Content Delivery

We have so far considered a system corresponding to the example of Fig. 7.12(a), where a cache-aware system *recommends* alternative contents to users (in case of a cache miss), but users might not accept them. In this section, we consider a system closer to our second example of Fig. 7.12(b), where the system *delivers* some related content that is locally available *instead of the original content*, in case of a cache miss. While a more extreme scenario, we believe this might still have application in a number of scenarios, as explained in Section 7.3.1 (e.g., for low rate plan users under congestion, or in limited access scenarios [98, 99]).

In the following, we model the related content delivery system, formulate the respective optimization problem, and show that it has the same properties with the problems in the previous sections, which means that our results and algorithms apply to this context as well. We present only the more generic femto-cache case of Sec. 7.3.7; the analysis and results for the single cache cases of Sec. 7.3.4 follow similarly.

Since now original requests might not be served, the (soft) cache hit ratio metric does not describe sufficiently the performance of this system. To this end, we modify the definition of content utility:

**Definition 7.3.2.** *When a user  $i$  requests a content  $k$  that is not locally available and the content provider delivers an alternative content  $n$  then the user satisfaction is given by the utility  $u_{kn}^i$ .  $u_{kn}^i \in \mathbb{R}$  is a real number, and does not denote a probability of acceptance, but rather the happiness of user  $i$  when she receives  $n$  instead of  $k$ . Furthermore  $u_{kk}^i = U_{max}, \forall i$ .*

*Note:* we stress that the utilities  $u_{kn}^i$  in Definition 7.3.2 do not represent the probability a user  $i$  to accept a content  $n$  (as in Definition 7.3.1), but the satisfaction of user  $i$  given that she accepted content  $n$ . User satisfaction can be estimated by past statistics, or user feedback, e.g., by asking user to rate the received alternative content.

Let us denote as  $G_i(t) \subseteq \mathcal{M}$  the set of SCs with which the user  $i$  is associated at time  $t$ . Given Definition 7.3.2, when a user  $i$  requests at time  $t$  a content  $k$  that is not locally available, we assume a system (as in Fig. 7.12(b)) that delivers to the user the cached content with the *highest* utility<sup>14</sup>, i.e., the content  $n$  where

$$n \equiv \arg \max_{\ell \in \mathcal{K}, j \in G_i(t)} \{u_{k\ell}^i \cdot x_{\ell j}\} \quad (7.33)$$

Hence, the satisfaction of a user  $i$  upon a request for content  $k$  is

$$\max_{n \in \mathcal{K}, j \in G_i(t)} \{u_{kn}^i \cdot x_{nj}\} \quad (7.34)$$

---

<sup>14</sup>Equivalently, the system can recommend all the stored contents to the user and then allow the user to select the content that satisfies her more.

Using the above expression and proceeding similarly to Section 7.3.7, we formulate the optimization problem that the network needs to solve to optimize the total user satisfaction (among all users and all content requests), which we call *soft cache hit user satisfaction* (SCH-US).

**Optimization Problem 4** (SCH-US). *The optimal cache placement problem for the femtocaching scenario with related content delivery and content relations described by the matrix  $\mathbf{U} = \{u_{kn}^i\}$  is*

$$\begin{aligned} & \underset{X=\{x_1, \dots, x_K\}}{\text{maximize}} \quad f(X) = \\ &= \sum_{i=1}^N \sum_{k=1}^K p_k^i \cdot E_{G_i} \left[ \max_{n \in \mathcal{K}, j \in \mathcal{M}} (u_{kn}^i \cdot x_{nj} \cdot Q_{ij}) \right], \end{aligned} \quad (7.35)$$

$$\text{s.t.} \quad \sum_{k=1}^K x_{kj} \leq C, \quad \forall j \in \mathcal{M}. \quad (7.36)$$

where  $Q_{ij} = \begin{cases} 1 & , \text{if } j \in G_i \\ 0 & , \text{otherwise} \end{cases}$ , and the expectation  $E_{G_i}[\cdot]$  is taken over the probabilities  $P\{G_i\} = \prod_{j \in G_i} q_{ij} \cdot \prod_{j \notin G_i} (1 - q_{ij})$ .

For the sub-cases 1 and 2 of Definition 7.3.1 presented in Sec. 7.3.3, the following corollary holds.

**Corollary 9.** *The expression of Eq. (7.35) needs to be modified as*

$$\begin{aligned} E_{G_i} \left[ \max_{n \in \mathcal{K}, j \in \mathcal{M}} (u_{kn}^i \cdot x_{nj} \cdot Q_{ij}) \right] &\rightarrow E_{G_i} \left[ \max_{n \in S} (u_{kn}^i) \right] = \\ &= E_{G_i} \left[ \int \left( 1 - \prod_{n \in S} F_{kn}(x) \right) dx \right] \end{aligned} \quad (7.37)$$

$$u_{kn}^i \rightarrow u_{kn} \quad (7.38)$$

where  $S = \{\ell : \ell \in \mathcal{K}, m \in \mathcal{M}, x_{\ell m} \cdot Q_{im} = 1\}$ , for the sub-cases 1 and 2 of Definition 7.3.1, respectively.

*Proof.* The proof can be found in [288].  $\square$

We can now derive the following Lemma, which shows that Theorem 3.3 applies also to the Optimization Problem 4, and thus it can be efficiently solved by the same greedy algorithm (where the objective function of Eq. (7.35) is now used).

**Lemma 16.**

- (1) *The Optimization Problem 4 is NP-hard,*
- (2) *with submodular and monotone objective function (Eq. (7.35)).*

*Proof.* The proof is given in Appendix 7.5.3.  $\square$

### 7.3.9 Performance Evaluation

In this section, we investigate the gains of employing soft cache hits and the performance of the proposed algorithms. We first analyze 5 real datasets collected from different content-centric applications/sources, such as YouTube and Amazon-TV, as well as other types of contents (e.g. Android applications) or data sources (like MovieLens) (Sec. 7.3.10). We have also tested our schemes with some data related to personalized radio (lastFM) with similar conclusions. The datasets contain information about content relations, based on which we build the utility matrices  $\mathbf{U}$ . We use these realistic utility matrices  $\mathbf{U}$  in our simulations to study the performance of caching with or without soft cache hits. In Sec. 7.3.11 we describe the simulation setup, and present and discuss the results in Sec. 7.3.12.

### 7.3.10 Datasets of Content Relations

**YouTube dataset.** We consider a dataset of YouTube videos from [291]<sup>15</sup>. The dataset contains several information about the videos, such as their popularity, size, and a list of related videos (as recommended by YouTube). We build the utility matrix  $\mathbf{U} = \{u_{nk}\}$ , where  $u_{nk} = 1$  if video  $n$  is in the list of related videos of  $k$  (or vice-versa), and otherwise  $u_{nk} = 0$ .

**Amazon datasets.** We also analyze 3 datasets of product reviews from Amazon [292] for Android applications (*Amazon-App*), Movies and TV (*Amazon-TV*), and Videogames (*Amazon-VG*). The datasets include for each item a list of contents that are “also bought”.<sup>16</sup> We consider for each dataset 10000 of its items, and build a utility matrix  $\mathbf{U} = \{u_{nk}\}$ , where  $u_{nk} = 1$  if item  $n$  is also bought with item  $k$  (or vice-versa), and otherwise  $u_{nk} = 0$ .

**MovieLens dataset.** We finally consider a movies-rating dataset from the MovieLens website [293], containing 69162 ratings (from 0.5 stars to 5) of 671 users for 9066 movies. As these datasets contain only raw user ratings and not movie relations per se, to obtain content relation matrix  $\mathbf{U}$ , in this case, we do an intermediate step and apply a standard concept from collaborative filtering [281]. Specifically, we calculate the similarity of each pair of contents based on their common ratings as their cosine-distance metric:

$$sim(n, k) = \frac{\sum_{i=1}^{\#users} r_i(n) \cdot r_i(k)}{\sqrt{\sum_{i=1}^{\#users} r_i^2(n)} \cdot \sqrt{\sum_{i=1}^{\#users} r_i^2(k)}}$$

where we normalized the ratings  $r_i$ , by subtracting from each rating the average rating of that item, so that we obtain similarity values  $\in [-1, 1]$ . Due to the sparsity of the dataset (few common ratings), we also apply an item-to-item collaborative filtering (using 10 similar items) in order to predict the missing user ratings per item, and thus the missing similarity values. We build the utility matrix  $\mathbf{U} = \{u_{nk}\}$  with  $u_{nk} =$

---

<sup>15</sup>Data from 27-07-2008, and depth of search up to 3; see details in [291]

<sup>16</sup>Our main motivation to use the game and app datasets was to also validate the robustness of our approach to other types of data. Soft cache hits though might be more relevant for free apps or games, rather than paid products.

Table 7.6: Information contained in datasets.

	content popularity	content size	content relations $u_{kn} \in \{0, 1\}$	$\in [0, 1]$
Amazon-*	✗	✗	✓	✗
MovieLens	✓	✗	✗	✓
YouTube	✓	✓	✓	✗

Table 7.7: Dataset analysis.

	#contents	content relations $E[R] \left( \frac{\text{std}[R]}{E[R]} \right)$	popularity $E[p] \left( \frac{\text{std}[p]}{E[p]} \right)$
Amazon-App	8229	16.0 (2.2)	-
Amazon-TV	2789	7.8 (1.0)	-
Amazon-VG	5614	22.0 (1.1)	-
MovieLens	4622	125.8 (0.5)	15 (1.6)
YouTube	2098	5.3 (0.7)	500 (3.1)

$\max \{0, sim(n, k)\}$ , i.e.,  $u_{nk} \in [0, 1]$ . Finally, we assign to each item a popularity value equal to the number of ratings for this item.

For ease of reference, Table 7.6 presents the information contained in each dataset.

Due to the sparsity of the YouTube dataset, we only consider contents belonging to the largest connected component (defining as adjacencies, the positive entries of the utility matrix). For consistency, we consider only the contents in the largest connected component for the other datasets as well. Moreover, since the Amazon and YouTube datasets do not contain per-user information, and the per-user data in the MovieLens dataset is sparse, we consider the sub-case-2 of Definition 7.3.1, i.e.,  $u_{kn}^i = u_{kn}$  for all users  $i$ .

The number of remaining contents for each dataset are given in Table 7.7. We also calculate for each content the number of its related contents  $R_n = \sum_k u_{nk}$  (or the sum of its utilities for the MovieLens dataset where  $u_{kn} \in [0, 1]$ ), and present the corresponding statistics in Table 7.7 along with the statistics for the content popularity.

### 7.3.11 Simulation Setup

**Cellular network.** We consider an area of  $1 \text{ km}^2$  that contains  $M$  SCs. SCs are randomly placed in the area (following a Poisson point process), which is a common assumption in related work [81, 83]. An SC can serve a request from a user, when the user is inside its communication range, which we set to 200 meters. We also consider  $N$  mobile users.

We select as default parameters:  $N = 50$  users, and  $M = 20$  SCs with caching capacity  $C = 5$  (contents). This creates a relatively dense network, where a random user is connected to 3 SCs *on average*.

**Content demand.** We consider a scenario of content demand for each dataset of Sec. 7.3.10, with the corresponding set of contents, content popularities and relations

(utility matrix). For datasets without information on content popularity (see Table 7.6), we generate a random sample of popularity values drawn from a Zipf distribution in  $[1, 400]$ <sup>17</sup> with exponent  $\alpha = 2$ . For each scenario we generate a set of 20 000 requests according to the content popularity, over which we average our results. When soft cache hits are allowed, we assume the *related content recommendation* model of Definition 7.3.1 (see also Fig. 7.12(a)).

Unless otherwise stated, the simulations use the default parameters summarized in Table 7.8.

**Caching schemes / algorithms.** We consider and compare the following schemes for single-SC (*single*) and multi-SC (*femto*) to user association.

- *Single*: A single cache accessible per user (e.g., the closest one). Only normal cache hits allowed, and the most popular contents are stored in each cache, which is the optimal policy in this simple setup. It will serve as the baseline for single cache scenarios.
- *SingleSCH*: Here soft cache hits are allowed. However, the caching policy is still based on popularity as before (i.e., is not explicitly optimized to exploit SCHs).
- *SingleSCH\**: This is our proposed policy. Here soft cache hits are allowed, and the caching policy is optimized to fully exploit this (according to Algorithm 2 or Algorithm 3).
- *Femto*: Femto-caching without soft cache hits. This is the baseline scheme for this class of scenarios, where the proposed algorithm from [81] is applied.
- *FemtoSCH*: Femto-caching based content placement (same as in *Femto*), but allowing soft cache hits on user requests (a posteriori).
- *FemtoSCH\**: Our proposed policy. Femto-caching is explicitly optimized for soft cache hits, according the greedy algorithm (Sec. 7.3.7).

### 7.3.12 Results

#### 1. Overall performance

We simulate scenarios for all datasets / utility matrices with the default parameters (Table 7.8), both under single and multi user-SC association. Fig. 7.13 shows the achieved cache hit ratio CHR (or soft cache hit ratio, SCHR) under the baseline caching (*Single/SingleSCH/Femto/FemtoSCH*) and the SCHR under a content placement using our algorithms (*SingleSCH\*/FemtoSCH\**).

**Key Message:** *Allowing soft cache hits can lead to a dramatic increase in the cache hit ratio.*

Comparing the cache hit ratio (CHR) under the popularity-based caching (*Single/Femto* - red/pink bars) and the schemes we propose (*SingleSCH\*/FemtoSCH\** - black/grey bars), shows that allowing soft cache hits brings a significant increase in the CHR for all datasets. The relative gain ranges from 60% in the *Amazon-TV* case, up to

---

<sup>17</sup>The max value is selected equal to the max number of requests per user, i.e.,  $\frac{\# \text{requests}}{\# \text{contents}}$ .

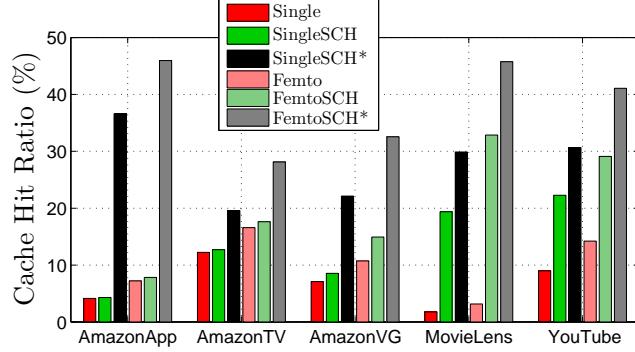


Figure 7.13: Cache hit ratio for all datasets and caching schemes, for the default scenario.

Table 7.8: Parameters used in simulations: default scenario.

Parameter	Value	Parameter	Value
Area	1 x 1 km	Cache size, $C$	5
nb. of SCs, $M$	20	nb. of users $N$	50
SC comm. range	200 m	Zipf distr.	$\in [1, 400]$ , $\alpha = 2$

around 780% in the *Amazon-App* case, for the *Single* scenarios; the relative gains in the *Femto* scenarios are similarly impressive (from 70% up to 530%, respectively). These initial results indicate that soft cache hits can be a promising solution for future mobile networks, by increasing the virtual capacity of mobile edge caching.

**Key Message:** *While gains can sometimes already be achieved just by allowing soft cache hits, to fully take advantage of soft cache hits, the caching policy should be redesigned to explicitly take these into account (through the utility matrix).*

Fig. 7.13 demonstrates that gains could already be achieved by simply introducing soft cache hits on top of existing (state-of-the-art) caching policy (*SingleSCH/FemtoSCH* - dark/light green bars), but these are scenario-dependent. For example, in the *Amazon* scenarios the increase in CHR by allowing soft cache hits is marginal (red vs. green bars), while in the *YouTube* scenario it is 1.5× higher. In contrast, explicitly designing the caching policy to exploit soft cache hits allows for important gains in all scenarios (black/grey bars). Specifically, in the *Amazon* scenarios the performance gains are almost entirely due to the caching algorithm (just allowing soft cache hits, does not improve performance), while in the *YouTube* scenario our utility-aware algorithms outperform by around 40% popularity-based caching. These results show clearly that existing caching policies are not capable to exploit the full potential of soft cache hits.

## 2. Impact of network parameters

We proceed to study the effect of network parameters, on the performance of soft cache hits schemes. We consider the *YouTube* dataset, for which the soft cache hits

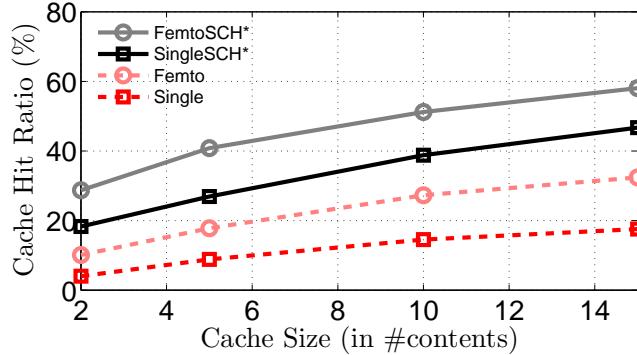


Figure 7.14: Cache hit ratio vs. cache size  $C$ .

schemes (*SingleSCH\*/FemtoSCH\**) have a moderate gain (around  $1.5 - 3 \times$ ) over the baseline schemes. We simulate scenarios where we vary the cache size  $C$  and the number of SCs  $M$ ; the remaining parameters are set as in the default scenario (Table 7.8).

**Key Message:** (a) *Soft cache hits improve performance irrespectively of the underlying network parameters (even for a few SCs with small capacity);* (b) *Combining femto-caching and soft cache hits achieves significantly higher CHR than today's solutions.*

**Cache size impact:** We first investigate the impact of cache size, assuming fixed content sizes. Fig. 7.14 depicts the total cache hit ratio, for different cache sizes  $C$ : we consider a cache size per SC between 2 and 15 contents. The simulations suggest that the *SingleSCH\*/FemtoSCH\** scenarios consistently achieve more than  $2.5 \times$  (single) and  $1.2 \times$  (femto) higher CHR than *Single/Femto*. What is more, these gains are applicable to both single- and femto- caching. The two methods (femto-caching and soft cache hits) together offer a total of  $3.3 \times$  to  $7 \times$  improvement compared to the baseline scenario *Single*. Finally, even with a cache size per SC of about 0.1% of the total catalog ( $C = 2$ ), introducing soft cache hits offers 29% CHR (*SingleSCH\**), whereas today's practices (popularity-based caching without SCH) would achieve only 4% CHR.

**SC density impact:** In Fig. 7.15 we consider the impact of SC density. In sparse scenarios (e.g.,  $M = 5$ ), a user usually is in the range of at most one SC. For this reason, Femto and Single perform similarly. As the SC density increases, the basic *Femto* is able to improve performance, as expected, by exploiting cache cooperation. However, every  $2 \times$  increase in density, which requires the operator doubling the infrastructure cost, offers roughly a relative improvement of 30–50%. Simply introducing soft cache hits instead, suffices to provide a  $2 \times$  improvement.

**Key Message:** *The extra cost to incentivize soft cache hits might be quite smaller than the CAPEX/OPEX costs in infrastructure investment to achieve comparable performance gains.*

### 3. Impact of utility matrix

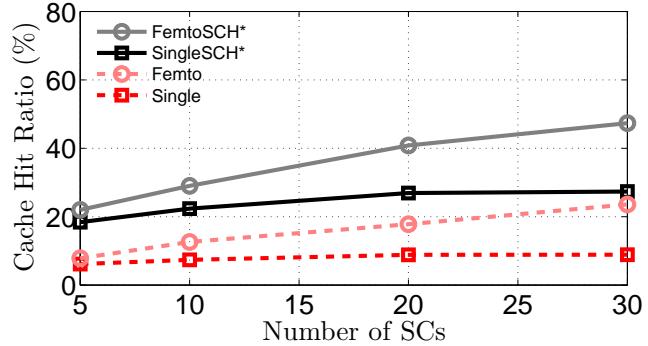


Figure 7.15: Cache hit ratio vs. number of SCs  $M$ .

We further investigate the impact of the content relations as captured by the matrix  $\mathbf{U}$  (and its structure). To quantify the content relations, we use as a metric the sum of the utilities per content  $R_n = \sum_k u_{nk}$  (see also Sec. 7.3.10 and Table 7.7).

**Key Message:** *The CHR increases with the density ( $E[R]$ ) of the utility matrix. Even low utility values  $u_{kn}$  can significantly contribute towards a higher CHR.*

The first important parameter to consider is the average value of  $R_n$ , i.e., the density of the utility matrix  $\mathbf{U}$ . We consider the *MovieLens* dataset, where the utilities  $u_{kn}$  are real numbers in the range  $[0, 1]$ . To investigate the impact of the density of  $\mathbf{U}$ , we consider scenarios where we vary the matrix  $\mathbf{U}$ : for each scenario we set a threshold  $u_{min}$  and take into account only the relations between contents with utility higher than  $u_{min}$ , i.e.,

$$\mathbf{U}' = \{u'_{kn}\} = \begin{cases} u_{kn} & \text{if } u_{kn} \geq u_{min} \\ 0 & \text{if } u_{kn} < u_{min} \end{cases}$$

Table 7.9 gives the density of the utility matrix for the different values of the threshold  $u_{min}$ , and Fig. 7.16 shows the cache hit ratio for these scenarios. When  $u_{min}$  is set to a large value (i.e., only few relations are taken into account and the matrix  $\mathbf{U}$  is sparse), there is no ( $u_{min} = 0.75$ ) or negligible ( $u_{min} = 0.5$ ) improvement from soft cache hits. However, as the density of  $\mathbf{U}$  increases (lower thresholds  $u_{min}$ ), the gains in CHR significantly increase; note that these gains are from content relations with low utility values (i.e., less than 0.5 or 0.25 for the two rightmost scenarios, respectively). Moreover, it is interesting that in the scenario with  $u_{min} = 0.25$ , the gains are almost entirely due to the more efficient caching from our algorithms, i.e., popularity-based caching would not be efficient even if soft cache hits were allowed (green bars).

Table 7.9: Utility matrix density for the MovieLens dataset for different  $u_{min}$  thresholds.

threshold $u_{min}$	0.75	0.5	0.25	0
$E[R]$	0.9	10.8	49.0	125.8

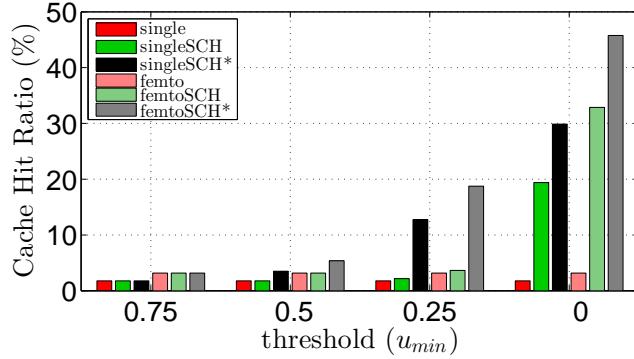


Figure 7.16: Cache hit ratio for the MovieLens dataset for scenarios with different  $u_{min}$  thresholds; default scenario.

**Key Message:** *Highly skewed distributions of #relations,  $R_n$ , can lead to more efficient caching (in analogy to heavy tailed popularity distributions).*

Our simulation study demonstrates the effect of the variance of the number of related contents, i.e.,  $\frac{std[R]}{E[R]}$ . We consider the three *Amazon* scenarios of Fig. 7.13 that have the same content popularity distribution and similar  $E[R]$  values (see Sec. 7.3.11). Fig. 7.17 shows the relative gain (of soft cache schemes over baseline schemes) in these scenarios where the distribution of  $R_n$  has different variance (see Table 7.7). When the variance is very high (*Amazon-App*,  $\frac{std[R]}{E[R]} = 2.2$ ) the CHR under soft cache hit schemes is almost an order of magnitude larger than the baseline scenarios. Large variance means that a few contents have very high  $R_n$ ; thus storing these contents allows to serve requests for a large number of other (non-cached) contents as well. Finally, an interesting observation is that the variance of the distribution plays a more important role than the density of the utility matrix: although the utility matrix in the *Amazon-VG* scenario ( $E[R] = 22$ ,  $\frac{std[R]}{E[R]} = 1.1$ ) is denser than in the *Amazon-App* scenario ( $E[R] = 16$ ,  $\frac{std[R]}{E[R]} = 2.2$ ), the gain of the latter is higher due to the higher variance.

#### 4. Performance of scheme extensions

**Key Message:** *The gain of our algorithms is consistent for all the considered variations of soft cache hits scenarios.*

Finally, we evaluated scenarios with (a) contents of different size and (b) *related content delivery* model (Def. 7.3.2), and observed the following. In the former scenarios (from the *YouTube* dataset), the performance improves considerably for all cache size values (e.g., similarly to the equal content sizes case). In the latter scenarios (from the *MovieLens* dataset; similar to scenarios of Fig. 7.16), the *user satisfaction* significantly increases with related content delivery (i.e., soft cache hits), and denser matrices (i.e., higher willingness of users to accept related contents) lead to better performance.

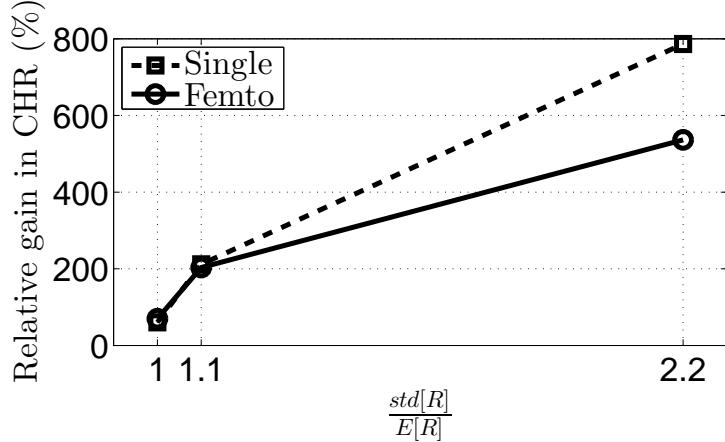


Figure 7.17: Relative increase in the cache hit ratio due to soft cache hits (y-axis). *Amazon* scenarios with different variance of number of related contents (x-axis).

### 7.3.13 Related Work

**Mobile Edge Caching.** Densification of cellular networks, overlaying the standard macro-cell network with a large number of SCs (e.g., pico- or femto-cells), has been extensively studied and is considered a promising solution to cope with data demand [294, 295, 296]. As this densification puts a tremendous pressure on the backhaul network, researchers have suggested storing popular content at the “edge”, e.g., at SCs [81], user devices [272, 142, 274], or vehicles [273, 92].

Our work is complementary to these approaches: it can utilize such mobile edge caching systems and further optimize the cache allocation when there is a cache-aware recommender systems in place. We have applied this approach in the context of mobile (ad-hoc) networks with delayed content delivery [102] as well, and applied it here for the first time in the context of femto-caching [81]. Additional research directions have also recently emerged, more closely considering the interplay between caching and the physical layer such as Coded Caching [255] and caching for coordinated (CoMP) transmission [104, 297]. We believe the idea of soft cache hits could be applied in these settings as well, and we plan to explore this as future work.

**Caching and Recommendation Interplay.** There exist some recent works that have jointly considered caching and recommendation for wireless systems [279], peer-to-peer networks [298], and CDNs [299, 278]. Specifically, [298] studies the interplay between a recommendation system and the performance of content distribution on a peer-to-peer network like BitTorrent (e.g., recommending contents based on the number of “seeders”) towards improving performance.

[299] shows that users tend to follow YouTube’s suggestions, and despite the large catalog of YouTube, the top-10 recommendations are usually common for different users in the same geographical region. Hence, CDNs can use the knowledge from the recommendation system to improve their content delivery. Finally, [278, 279] propose

approaches of recommended list reordering, which can achieve higher cache hit ratios (e.g., for YouTube servers/caches).

These works, except for [279] which is closer to our study, (i) focus on the recommendation side of the problem, ignoring or simplifying the optimal caching algorithm, and (ii) do not consider the wireless cooperative caching aspect of the problem. Nevertheless, the increasing dependence of user requests on the output of recommender systems clearly suggests that there is an opportunity to further improve the performance of (mobile) edge caching by jointly optimizing both, with minimum impact on user Quality of Experience.

A preliminary version of our work appears in [280]. However, there are a number of key additions in this work: (i) we consider in detail the single-cache scenario (Sec. 7.3.4) and propose a more efficient algorithm (Alg. 2), as well as an algorithm for unequal content sizes (Alg. 3); (ii) we introduce the entirely new model of Sec. 7.3.8 and the corresponding analysis; (iii) we collected and analyzed 4 extra real datasets of content relations (*Amazon* and *MovieLens* datasets), in order to extend the evaluation and provide further insights on the gains of SCH (Sec. 7.3.9).

## 7.4 Coordinated Caching and Communication in 5G: Vision and Initial Results

The common denominators between most of the works considered thus far can be summarized as follows: (i) The main bottleneck is the backhaul link (in the femto-caching setup) or the macro-cell link (in the vehicular cloud setup), (ii) the transmission phase is ignored (assuming requests are asynchronous and non-interfering) or simplified, (iii) caching gains are in terms of cache-hit ratio and amount of traffic offloaded away from expensive links.

Nevertheless, when considering a wireless setup, content delivery over the radio access link becomes just as important as the placement problem. If multiple nearby base stations (BS) have the same content cached, they can coordinate in order to improve performance on the radio access link. For example, several base stations can perform Joint Transmission (JT) to simultaneously transmit the same file to a single user, e.g. for power and diversity gains, which is particularly useful to edge users. Alternatively, multiple user requests could be satisfied in parallel by forming a MU-MIMO channel, between the BSs and users involved. Such techniques are often referred to as Coordinated Multi-point (CoMP) transmission [103]. With the data cached locally on each BS involved, only channel state information (CSI) information needs to be exchanged over the backhaul to coordinate the transmission, which is a much smaller burden, compared to exchanging whole video files. These ideas have led researchers to argue that caching and transmission algorithms at each involved BS must be jointly designed in order to facilitate such CoMP opportunities, whether this is a distributed BS setup or cloudRAN scenario [300].

Recent work by Maddah-Ali and Niesen [255] revealed quite interesting findings about the fundamental gains achievable by jointly considering caching and coded transmissions in a broadcast channel. Finally, in the very recent work of [301], ideas from

coded caching are also used to derive fundamental performance bounds on the impact of caching for a simple K-user interference channel.

A dichotomy appears then in the existing literature: the femto-caching line of work aims to reduce backhaul traffic and then focuses on hit rates as main performance metric, the cache-aided communication line of work instead maximizes the transmission rates achievable on the radio access channel, ignoring the effect of cache misses. As a result, a clear understanding of the impact on end-to-end (or network-wide) performance is lacking. Our goal is to address the problem of cache placement that jointly optimizes both radio access and backhaul performance, in a setup where small cells (“femto-nodes”) can coordinate both in terms of what they cache and in how they transmit.

As a first step in this direction, we have focused on the JT technique for the radio access part. Every time the requested file is cached at several base stations in the user’s range, the base stations can jointly transmit the file to the user. The transmission rate of JT is higher than that of each separate base station. Hence, storing the same (popular) files is optimal with respect to radio access transmission. On the other hand, storing different files in these base stations might lead to fewer cache misses and thus accesses to the backhaul network, which is important if the latter is the bottleneck. A recent work looking at the radio access/backhaul tradeoff is [104], where two different CoMP techniques are studied, namely Maximum Ratio Transmission (MRT) and Zero-Forcing BeamForming (ZFBF). The authors consider two caching heuristics: a randomized caching policy for MRT and a threshold policy for ZFBF. While they derive the optimal parameter setting of such heuristics, they are in general suboptimal and there is no theoretical performance guarantee in comparison to the optimal content allocation. On the contrary, we prove and exploit submodularity properties of the objective, and propose an allocation algorithm that has a provable approximation ratio. In the following, we briefly summarize our main theoretical results and provide some initial performance-related insights.

### Cooperative transmission

Let variable  $q_{iu}$  denote again if user  $u$  can download from helper  $i$  ( $q_{iu} = 1$ ) or not ( $q_{iu} = 0$ )<sup>18</sup>. When the download is possible  $g_{iu}$  denotes the corresponding downlink SNR. For convenience, we also define  $g_{iu} = 0$  when  $u$  cannot download from  $i$ .

Variables  $x_{if} = 1$  are still “placement” variables, implying that the file  $f$  is cached on the helper  $i$ , and  $x_{if} = 0$  otherwise. The placement or caching matrix  $\mathbf{X}$  is still the main control variable in our system but now our goal is to choose  $\mathbf{X}$  optimally to minimize the average end-to-end delay per content download, taking into account both backhaul and radio access performance. We use variable  $k(u, f)$  to denote the number of copies of the file  $f$  in the helpers of the user  $u$  under placement  $\mathbf{X}$ :

$$k(u, f) = \sum_{i=1}^H x_{if} e_{iu}.$$

---

<sup>18</sup>These are the same variables as in the SCH case, except there we allowed  $q_{ij}$  to take values smaller than 1 allowing for location uncertainty (e.g. due to mobility). Without loss of generality, we focus here on the simple case.

We assume base stations can cooperate using JT on the downlink, between the base stations that already have the requested file cached and are in the user's neighborhood <sup>19</sup>.

*Cache miss:* Upon a miss, the content has to be fetched over a backhaul link to one of the neighboring base stations (when these have different SNRs, we assume the best one is used), incurring a fixed backhaul delay  $D_{bh}$  and a transmission delay equal to

$$D_{r,miss}(u) = d_0 \cdot \left[ \log_2 \left( 1 + \max_{i=1,\dots,H} g_{iu} \right) \right]^{-1}, \quad (7.39)$$

where  $d_0$  is just a normalizing constant that depends on file size, total bandwidth allocated to that connection, etc.

*Cache hit:* If user  $u$  requests file  $f$ , which is cached at least on one neighboring helper, the *backhaul delay* is 0 and all the helpers will coordinate their transmissions so that the SNRs sum at the mobile. The *radio access delay* is then:

$$D^{r,hit}(u, f) = d_0 \left[ \log \left( 1 + \sum_{h=1..H} x_{if} g_{hu} \right) \right]^{-1}.$$

Putting everything together, the goal is to select variables  $x_{if}$ , in order to minimize the expected delay over all users and all contents, captured in the following objective:

$$\sum_{u,f} p_f (1 - \mathcal{I}_{k(u,f)>0}) \cdot (D_{r,miss}(u) + D_{bh}) + \mathcal{I}_{k(u,f)>0} \cdot D^{r,hit}(u, f). \quad (7.40)$$

In a symmetric SNR scenario, i.e.,  $g_{iu} = g, \forall g_{iu} > 0$ , the following main result has been proven

**Lemma 17.** *When  $D_{bh} \geq d_0 [\log(1 + g \cdot x_{if})]^{-1}$ , the objective of Eq.(7.40) is monotone and submodular on the set of feasible allocation vectors  $\mathbf{X}$ .*

A similar result can also be proven for the more general case of asymmetric SNR.

### Preliminary Performance Results

Based on the above lemma, and the fact that the capacity constraints remain the same as in the original femto-caching problem, and thus are of matroid type, a greedy algorithm can be used to solve these problems guaranteeing a worst case approximation ratio of  $\frac{1}{2}$ . This is a promising results as it means that the problem of optimal cache allocation in a dense femto-caching setup that jointly optimizes radio access and backhaul delay (the latter being equivalent to the original cache hit rate) can be solved with the same complexity and same accuracy as the simpler femto-caching problem with no cooperation on the radio access.

---

<sup>19</sup>Another possibility is to use also other base stations from the user's neighborhood, that do not necessary have the requested file cached. However in this case the base stations, that do not have the requested file cached, have to download it from the backhaul. As this would multiply the backhaul load by a factor equal to the number of BSs cooperating, we do not consider it as an option here.

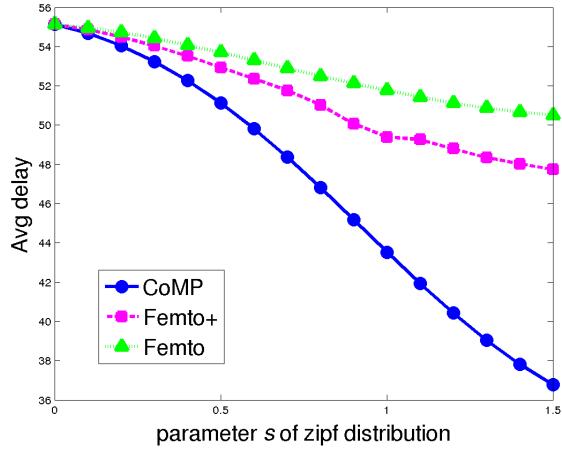


Figure 7.18: Average delay in the network for uniform SNR

We present here some initial results on the performance advantages of coordinated caching and communication. As a baseline, we consider the optimal caching allocation found in a non-cooperative scenario [81]. In order to understand which part of the improvement is simply due to JT, we also consider a third scenario, where contents are allocated according to [81], but then the base stations take advantage of any opportunity to jointly transmit the content. We refer to this third scenario as *Femto+*.

Unless otherwise said, the setting considered in our simulations is the following. We have a squared area with side equal to 300m, where 25 helpers located at the centers of an hexagonal grid, so that the minimum distance between two helpers is 75m. The wireless channel has bandwidth  $W = 5$  MHz, while the backhaul transmission rate is 100 Mbps. The cache of each helper can store 3 files out of a catalogue of 20 files, with size  $M = 1$  Gbit. Content popularities follow a Zipf distribution with parameter  $s$ , i.e.  $p_f \propto 1/f^s$ . 100 users are placed uniformly at random in the area. Each user can connect to any helper less than  $R = 100$  m away.

Figure 7.18 compares the average time to download a file in Femto, Femto+ and CoMP cases, for different values of the parameter  $s$  of the Zipf distribution ( $s = 0$  corresponds to the uniform distribution, the larger  $s$  the more skewed the distribution). First, we notice that Femto+ always outperforms Femto. This is expected because content allocation is exactly the same in the two cases, but in Femto+ case the base stations are allowed to cooperate, if they store the same file. While in the Femto case the user will always download from a single helper, in the Femto+ case he/she will download from the reachable helpers having a copy of the content.

Secondly, CoMP caching always outperforms femto-caching (both in the non-cooperative and cooperative cases). This is also expected because this caching policy is designed to minimize delay, by taking into account all possibility of cooperative transmissions in advance. What is more interesting is the relative performance of the three approaches as  $s$  varies. When contents have the same popularity ( $s \approx 0$ ), the improvement from

storing an additional copy of the same file on one of the helpers is always smaller than the improvement from storing the first copy of a new content. In this case, both femto-caching policies (with or without taking into account JT opportunities) in fact try to make available the largest number possible of files to maximize the hit rate. As  $s$  increases, popularities start becoming different and CoMP caching starts showing significant improvement comparing to femto-caching. For example, for moderately skewed distribution with  $s = 0.6$  femto-caching achieves a 4.84% improvement in comparison to the Femto+ case and 6.52% in comparison to the Femto case. For a more skewed distribution with  $s = 1.5$  the difference between the cachings is more significant: 22.97% comparing to Femto+ case, 27.2% comparing to Femto case. Note that the improvement cannot be simply explained through the possibility to exploit occasional opportunity for joint transmissions, otherwise the delay of CoMP and Femto+ would be much closer. The figure suggests that the two problems produce a very different content allocation.

We refer the reader to [302] for further details on our analysis as well as more simulation scenarios.

## 7.5 Appendix

### 7.5.1 Proof of Lemma 13

We prove here the NP-hardness of the optimal cache allocation for a single cache with soft cache hits. Let us consider an instance of Optimization Problem 1, where the utilities are equal among all users and can be either 1 or 0, i.e.,  $u_{kn}^i = u_{kn}$ ,  $\forall i \in \mathcal{N}$  and  $u_{kn} \in \{0, 1\}$ ,  $\forall k, n \in \mathcal{K}$ . We denote as  $\mathcal{R}_k$  the set of contents related to content  $k$ , i.e.

$$\mathcal{R}_k = \{n \in \mathcal{K} : n \neq k, u_{kn} > 0\} \quad (\text{related content set}) \quad (7.41)$$

Consider the content subsets  $\mathcal{S}_k = \{k\} \cup \mathcal{R}_k$ . Assume that only content  $k$  is stored in the cache ( $x_k = 1$  and  $x_n = 0, \forall n \neq k$ ). All requests for contents in  $S_k$  will be satisfied (i.e. “covered” by content  $k$ ), and thus SCHR will be equal to  $\sum_{i \in \mathcal{N}} \sum_{n \in S_k} p_n^i \cdot q_i$ . When more than one contents are stored in the cache, let  $\mathcal{S}'$  denote the union of all contents covered by the stored ones, i.e.,  $\mathcal{S}' = \bigcup_{\{k: x_k=1\}} S_k$ . Then, the SCHR will be equal to  $\sum_{i \in \mathcal{N}} \sum_{n \in \mathcal{S}'} p_n^i \cdot q_i$ . Hence, the Optimization Problem 1 becomes equivalent to

$$\max_{\mathcal{S}'} \sum_{n \in \mathcal{S}'} p_n^i \cdot q_i \quad s.t. \quad |\{k : x_k = 1\}| \leq C.$$

This corresponds to the the *maximum coverage problem with weighted elements*, where “elements” (to be “covered”) correspond to the contents  $i \in \mathcal{K}$ , weights correspond to the probability values  $p_n^i \cdot q_i$ , the number of selected subsets  $\{k : x_k = 1\}$  must be less than  $C$ , and their union of covered elements is  $\mathcal{S}'$ . This problem is known to be a NP-hard problem [303], and thus the more generic problem (with different  $u_{kn}^i$  and  $0 \leq u_{kn} \leq 1$ ) is also NP-hard.

### 7.5.2 Proof of Lemma 14

The objective function of Eq. (7.23)  $f(X) : \{0, 1\}^K \rightarrow \mathbb{R}$  is equivalent to a set function  $f(S) : 2^K \rightarrow \mathbb{R}$ , where  $\mathcal{K}$  is the finite *ground set* of contents, and  $S = \{k \in \mathcal{K} : x_k = 1\}$ . In other words,

$$f(S) \equiv \sum_{i=1}^N \sum_{k=1}^K p_k^i \cdot q_i \cdot \left( 1 - \prod_{n \in S} (1 - u_{kn}^i) \right). \quad (7.42)$$

A set function is characterised as *submodular* if and only if for every  $A \subseteq B \subset V$  and  $\ell \in V \setminus B$  it holds that

$$[f(A \cup \{\ell\}) - f(A)] - [f(B \cup \{\ell\}) - f(B)] \geq 0 \quad (7.43)$$

From Eq. (7.23), we first calculate

$$\begin{aligned} f(A \cup \{\ell\}) - f(A) &= \sum_{i=1}^N \sum_{k=1}^K p_k^i q_i \left( 1 - \prod_{n \in A \cup \{\ell\}} (1 - u_{kn}^i) \right) \\ &\quad - \sum_{i=1}^N \sum_{k=1}^K p_k^i q_i \left( 1 - \prod_{n \in A} (1 - u_{kn}^i) \right) \\ &= \sum_{i=1}^N \sum_{k=1}^K p_k^i q_i \cdot \left( \prod_{n \in A} (1 - u_{kn}^i) - \prod_{n \in A \cup \{\ell\}} (1 - u_{kn}^i) \right) \\ &= \sum_{i=1}^N \sum_{k=1}^K p_k^i q_i \cdot \left( u_{k\ell}^i \cdot \prod_{n \in A} (1 - u_{kn}^i) \right). \end{aligned}$$

Then,

$$\begin{aligned} [f(A \cup \{\ell\}) - f(A)] - [f(B \cup \{\ell\}) - f(B)] &= \\ &= \sum_{i=1}^N \sum_{k=1}^K p_k^i q_i \left( u_{k\ell}^i \prod_{n \in A} (1 - u_{kn}^i) \right) \\ &\quad - \sum_{i=1}^N \sum_{k=1}^K p_k^i q_i \left( u_{k\ell}^i \prod_{n \in B} (1 - u_{kn}^i) \right) \\ &= \sum_{i=1}^N \sum_{k=1}^K p_k^i q_i \cdot u_{k\ell}^i \cdot \left( \prod_{n \in A} (1 - u_{kn}^i) - \prod_{n \in B} (1 - u_{kn}^i) \right) \\ &= \sum_{i=1}^N \sum_{k=1}^K p_k^i q_i \cdot u_{k\ell}^i \cdot \prod_{n \in A} (1 - u_{kn}^i) \cdot \left( 1 - \prod_{n \in B \setminus A} (1 - u_{kn}^i) \right) \end{aligned}$$

The above expression is always  $\geq 0$ , which proves the submodularity for function  $f$ .

Furthermore, the function  $f$  is characterised as *monotone* if and only if  $f(B) \geq f(A)$  for every  $A \subseteq B \subset V$ . In our case, this property is shown as

$$\begin{aligned}
f(B) - f(A) &= \sum_{i=1}^N \sum_{k=1}^K p_k^i q_i \cdot \left( 1 - \prod_{n \in B} (1 - u_{kn}^i) \right) \\
&\quad - \sum_{i=1}^N \sum_{k=1}^K p_k^i q_i \cdot \left( 1 - \prod_{n \in A} (1 - u_{kn}^i) \right) \\
&= \sum_{i=1}^N \sum_{k=1}^K p_k^i q_i \cdot \left( \prod_{n \in A} (1 - u_{kn}^i) - \prod_{n \in B} (1 - u_{kn}^i) \right) \\
&= \sum_{i=1}^N \sum_{k=1}^K p_k^i q_i \cdot \prod_{n \in A} (1 - u_{kn}^i) \cdot \left( 1 - \prod_{n \in B \setminus A} (1 - u_{kn}^i) \right) \geq 0
\end{aligned}$$

### 7.5.3 Proof of Lemma 16

Item (1): Optimization Problem 4 is of the exact same nature as Optimization Problem 3, so it follows that it is NP-hard.

Item (2): We proceed similarly to the proof of Lemma 14. The objective function of Eq. (7.35)  $f(X) : \{0, 1\}^{K \times M} \rightarrow \mathbb{R}$  is equivalent to a set function  $f(S) : 2^{\mathcal{K} \times \mathcal{M}} \rightarrow \mathbb{R}$ , where  $\mathcal{K}$  and  $\mathcal{M}$  are the finite *ground sets* of contents and SCs, respectively, and  $S = \{k \in \mathcal{K}, j \in \mathcal{M} : x_{kj} = 1\}$ :

$$f(S) \equiv \sum_{i=1}^N \sum_{k=1}^K p_k^i \cdot E_{G_i} \left[ \max_{(n,j) \in S} (u_{kn}^i \cdot Q_{ij}) \right] \quad (7.44)$$

For all sets  $A \subseteq B \subset V$  and  $\{\text{content, SC}\}$  tuples  $(\ell, m) \in V \setminus B$ , we get

$$\begin{aligned}
f(A \cup \{(\ell, m)\}) - f(A) &= \\
&= \sum_{i=1}^N \sum_{k=1}^K p_k^i \cdot E_{G_i} \left[ \max_{(n,j) \in A \cup \{(\ell, m)\}} (u_{kn}^i Q_{ij}) \right] \\
&\quad - \sum_{i=1}^N \sum_{k=1}^K p_k^i \cdot E_{G_i} \left[ \max_{(n,j) \in A} (u_{kn}^i Q_{ij}) \right] \\
&= \sum_{i=1}^N \sum_{k=1}^K p_k^i \cdot E_{G_i} \left[ R \left( u_{k\ell}^i \cdot Q_{im} - \max_{(n,j) \in A} (u_{kn}^i Q_{ij}) \right) \right]
\end{aligned}$$

where in the last equation we use the *ramp function* defined as  $R(x) = x$  for  $x \geq 0$  and  $R(x) = 0$  for  $x < 0$ . Subsequently,

$$\begin{aligned}
[f(A \cup \{(\ell, m)\}) - f(A)] - [f(B \cup \{(\ell, m)\}) - f(B)] &= \\
&= \sum_{i=1}^N \sum_{k=1}^K p_k^i \cdot E_{G_i} \left[ R \left( u_{k\ell}^i Q_{im} - \max_{(n,j) \in A} (u_{kn}^i Q_{ij}) \right) \right. \\
&\quad \left. - R \left( u_{k\ell}^i Q_{im} - \max_{(n,j) \in B} (u_{kn}^i Q_{ij}) \right) \right]
\end{aligned}$$

The above equation is always  $\geq 0$  (which proves that the objective function Eq. (7.35) is *submodular*), since the ramp function is monotonically increasing and comparing the two arguments of the function  $R(x)$  in the above equation, gives

$$\begin{aligned} u_{k\ell}^i Q_{im} - \max_{(n,j) \in A} (u_{kn}^i Q_{ij}) - \left( u_{k\ell}^i Q_{im} - \max_{(n,j) \in B} (u_{kn}^i Q_{ij}) \right) \\ = \max_{(n,j) \in B} (u_{kn}^i Q_{ij}) - \max_{(n,j) \in A} (u_{kn}^i Q_{ij}) \geq 0 \end{aligned}$$

since  $B$  is a superset of  $A$  and therefore its maximum will be at least equal or greater than the maximum value in set  $A$ .

Similarly, since  $A \subseteq B$  it holds

$$\begin{aligned} f(B) - f(A) &= \\ &= \sum_{i=1}^N \sum_{k=1}^K p_k^i \cdot E_{G_i} \left[ \left( \max_{(n,j) \in B} (u_{kn}^i Q_{ij}) - \max_{(n,j) \in A} (u_{kn}^i Q_{ij}) \right) \right] \geq 0 \end{aligned}$$

which proves that the Eq. (7.35) is *monotone*.

# Bibliography

- [1] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.
- [2] S. Vassilaras, L. Gkatzikis, N. Liakopoulos, I. N. Stiakogiannakis, M. Qi, L. Shi, L. Liu, M. Debbah, and G. S. Paschos, “The algorithmic aspects of network slicing,” *IEEE Communications Magazine*, vol. 55, no. 8, pp. 112–119, 2017.
- [3] “White paper for research beyond 5g,” 2015, <http://networld2020.eu/wp-content/uploads/2015/11/B5G-Vision-for-Researchv-1.0-for-public-consultation.pdf>.
- [4] “Cisco visual networking index: Global mobile data traffic forecast update, 2016-2021,” 2016, <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>.
- [5] A. Ghosh, N. Mangalvedhe, R. Ratasuk, B. Mondal, M. Cudak, E. Visotsky, T. A. Thomas, J. G. Andrews, P. Xia, H. S. Jo, H. S. Dhillon, and T. D. Novlan, “Heterogeneous cellular networks: From theory to practice,” *IEEE Communications Magazine*, vol. 50, no. 6, pp. 54–64, June 2012.
- [6] M. Kamel, W. Hamouda, and A. Youssef, “Ultra-dense networks: A survey,” *IEEE Communication Surveys & Tutorials*, 2016.
- [7] I. Psaras, W. K. Chai, and G. Pavlou, “In-network cache management and resource allocation for information-centric networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 11, pp. 2920–2931, Nov 2014.
- [8] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, and S. Shenker, “Less pain, most of the gain: Incrementally deployable icn,” in *Proceedings of the ACM SIGCOMM 2013 Conference*. New York, NY, USA: ACM, 2013, pp. 147–158. [Online]. Available: <http://doi.acm.org/10.1145/2486001.2486023>
- [9] S. Ha, S. Sen, C. Joe-Wong, Y. Im, and M. Chiang, “Tube: time-dependent pricing for mobile data,” in *Proc. of ACM SIGCOMM*, 2012.

- [10] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Efficient routing in intermittently connected mobile networks: The single-copy case,” *to appear in Transactions on Networking*, 2007.
- [11] A. Vahdat and D. Becker, “Epidemic routing for partially connected ad hoc networks,” Duke University, Tech. Rep. CS-200006, 2000.
- [12] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Efficient routing in intermittently connected mobile networks: The multiple-copy case,” *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 77–90, 2008.
- [13] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, “Performance modeling of epidemic routing,” *Comput. Netw.*, vol. 51, no. 10, 2007.
- [14] P. Gupta and P. Kumar, “Capacity of wireless networks,” *Transactions on Information Theory*, vol. 46, no. 2, 2000.
- [15] M. Grossglauser and D. Tse, “Mobility increases the capacity of ad hoc wireless networks,” *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, 2002.
- [16] T. Spyropoulos, T. Turletti, and K. Obrazcka, “Routing in delay tolerant networks comprising heterogeneous populations of nodes,” *IEEE Transactions on Mobile Computing*, 2009.
- [17] S. Nelson, M. Bakht, and R. Kravets, “Encounter-based routing in dtns,” in *Proc. IEEE INFOCOM*, 2009.
- [18] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, “Delegation forwarding,” in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc ’08. New York, NY, USA: ACM, 2008, pp. 251–260. [Online]. Available: <http://doi.acm.org/10.1145/1374618.1374653>
- [19] A. Balasubramanian, B. Levine, and A. Venkataramani, “Dtn routing as a resource allocation problem,” in *Proceedings of ACM SIGCOMM*, 2007.
- [20] A. Krifa, C. Barakat, and T. Spyropoulos, “Optimal buffer management policies for delay tolerant networks,” in *IEEE SECON*, 2008.
- [21] ———, “Message drop and scheduling in dtns: Theory and practice,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 9, pp. 1470–1483, Sept 2012.
- [22] P. Matzakos, T. Spyropoulos, and C. Bonnet, “Buffer management policies for dtn applications with different qos requirements,” in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–7.
- [23] ———, “Joint scheduling and buffer management policies for dtn applications of different traffic classes,” *IEEE Transactions on Mobile Computing*, pp. 1–1, 2018.

- [24] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, “Impact of human mobility on the design of opportunistic forwarding algorithms,” in *Proceedings of IEEE INFOCOM*, 2006.
- [25] V. Erramilli, A. Chaintreau, M. Crovella, and C. Diot, “Diversity of forwarding paths in pocket switched networks,” in *IMC*, 2007.
- [26] V. Conan, J. Leguay, and T. Friedman, “Characterizing pairwise inter-contact patterns in delay tolerant networks,” in *ACM Autonomics*, October 2007.
- [27] A. Passarella and M. Conti, “Analysis of individual pair and aggregate inter-contact times in heterogeneous opportunistic networks,” *IEEE Trans. on Mobile Computing*, vol. 12, no. 12, pp. 2483–2495, 2013.
- [28] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnovic, “Power Law and Exponential Decay of Intercontact Times between Mobile Devices,” *IEEE Trans. Mob. Comp.*, vol. 9, no. 10, 2010.
- [29] P. Sermpezis and T. Spyropoulos, “Delay analysis of epidemic schemes in sparse and dense heterogeneous contact networks,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 9, pp. 2464–2477, Sept 2017.
- [30] M. Newman, *Networks: An Introduction*. New York, NY, USA: Oxford University Press, Inc., 2010.
- [31] A. Picu, T. Spyropoulos, and T. Hossmann, “An analysis of the information spreading delay in heterogeneous mobility dtns,” in *Proc. IEEE WoWMoM*, 2012.
- [32] P. Sermpezis and T. Spyropoulos, “Not all content is created equal: Effect of popularity and availability for content-centric opportunistic networking,” in *Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc ’14. New York, NY, USA: ACM, 2014, pp. 103–112. [Online]. Available: <http://doi.acm.org/10.1145/2632951.2632976>
- [33] ——, “Modeling and analysis of communication traffic heterogeneity in opportunistic networks,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 11, 2015.
- [34] ——, “Effects of content popularity on the performance of content-centric opportunistic networking: An analytical approach and applications,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3354–3368, December 2016.
- [35] E. M. Daly and M. Haahr, “Social network analysis for routing in disconnected delay-tolerant manets,” in *Proc. ACM MobiHoc*, 2007.
- [36] P. Hui, J. Crowcroft, and E. Yoneki, “Bubble rap: Social-based forwarding in delay-tolerant networks,” *IEEE Transactions on Mobile Computing*, vol. 10, pp. 1576–1589, 2011.

- [37] T. Hossmann, F. Legendre, and T. Spyropoulos, “From contacts to graphs: Pitfalls in using complex network analysis for DTN routing,” in *IEEE NetSciCom*, 2009.
- [38] T. Hossmann, T. Spyropoulos, and F. Legendre, “Know thy neighbor: Towards optimal mapping of contacts to social graphs for DTN routing,” in *IEEE Infocom 2010*, March 2010.
- [39] ——, “A complex network analysis of human mobility,” in *IEEE NetSciCom*, 2011.
- [40] ——, “Putting contacts into context: Mobility modeling beyond inter-contact times,” in *Proc. ACM MobiHoc*, 2011.
- [41] A. Picu and T. Spyropoulos, “Forecasting dtn performance under heterogeneous mobility: The case of limited replication,” in *Proc. IEEE SECON*, 2012.
- [42] ——, “Dtn-meteo: Forecasting the performance of dtn protocols under heterogeneous mobility,” *IEEE/ACM Trans. on Networking*, vol. PP, no. 99, 2014.
- [43] J. Ott and D. Kutscher, “A disconnection-tolerant transport for drive-thru internet environments,” in *Proceedings of INFOCOM*, 2005.
- [44] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden, “A measurement study of vehicular internet access using in situ wi-fi networks,” in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom ’06. New York, NY, USA: ACM, 2006, pp. 50–61. [Online]. Available: <http://doi.acm.org/10.1145/1161089.1161097>
- [45] R. Mahajan, J. Zahorjan, and B. Zill, “Understanding wifi-based connectivity from moving vehicles,” in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC ’07. New York, NY, USA: ACM, 2007, pp. 321–326. [Online]. Available: <http://doi.acm.org/10.1145/1298306.1298351>
- [46] G. Arvanitakis, T. Spyropoulos, and F. Kaltenberger, “An analytical model for flow-level performance of large, randomly placed small cell networks,” in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–7.
- [47] J. Eriksson, H. Balakrishnan, and S. Madden, “Cabernet: Vehicular content delivery using wifi,” in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, ser. MobiCom ’08. New York, NY, USA: ACM, 2008, pp. 199–210. [Online]. Available: <http://doi.acm.org/10.1145/1409944.1409968>
- [48] A. J. Nicholson and B. D. Noble, “Breadcrumbs: Forecasting mobile connectivity,” in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, ser. MobiCom ’08. New York, NY, USA: ACM, 2008, pp. 46–57. [Online]. Available: <http://doi.acm.org/10.1145/1409944.1409952>

- [49] F. Mehmeti and T. Spyropoulos, “Performance analysis of on-the-spot mobile data offloading,” in *2013 IEEE Global Communications Conference (GLOBECOM)*, Dec 2013, pp. 1577–1583.
- [50] ———, “Performance analysis of mobile data offloading in heterogeneous networks,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 482–497, Feb 2017.
- [51] A. Balasubramanian, R. Mahajan, and A. Venkataramani, “Augmenting mobile 3G using WiFi,” in *Proc. of ACM MobiSys*, 2010.
- [52] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, “Mobile data offloading: How much can WiFi deliver,” *IEEE/ACM Trans. Netw.*, 2013.
- [53] F. Mehmeti and T. Spyropoulos, “Is it worth to be patient? analysis and optimization of delayed mobile data offloading,” in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, April 2014, pp. 2364–2372.
- [54] ———, “Performance modeling, analysis, and optimization of delayed mobile data offloading for mobile users,” *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 550–564, Feb. 2017. [Online]. Available: <https://doi.org/10.1109/TNET.2016.2590320>
- [55] R. Madan, J. Borran, A. Sampath, N. Bhushan, A. Khandekar, and T. Ji, “Cell association and interference coordination in heterogeneous LTE-A cellular networks,” *IEEE Journal on Selected Areas in Communications (JSAC)*, 2010.
- [56] A. Khandekar, N. Bhushan, J. Tingfang, and V. Vanghi, “LTE-Advanced: Heterogeneous networks,” in *Proc. European Wireless Conference*, 2010.
- [57] A. Damnjanovic, J. Montojo, Y. Wei, T. Ji, T. Luo, M. Vajapeyam, T. Yoo, O. Song, and D. Malladi, “A survey on 3GPP heterogeneous networks,” *IEEE Wireless Communications*, 2011.
- [58] D. López-Pérez, M. Ding, H. Claussen, and A. H. Jafari, “Towards 1 Gbps/UE in cellular systems: Understanding ultra-dense small cell deployments,” *IEEE Communications Surveys & Tutorials*, 2015.
- [59] X. Ge, S. Tu, G. Mao, C. Wang, and T. Han, “5g ultra-dense cellular networks,” *IEEE Wireless Communications*, vol. 23, 2016.
- [60] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, “Providing Quality of Service over a shared wireless link,” *IEEE Communications Magazine*, 2001.
- [61] A. Jalali, R. Padovani, and R. Pankaj, “Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system,” in *Proc. Vehicular Technology Conference*, 2000.

- [62] P. Hande, S. Patil, and H. Myung, “Distributed load-balancing in a multi-carrier wireless system,” in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, 2009.
- [63] H. Kim, G. de Veciana, X. Yang, and M. Venkatachalam, “Distributed alpha-optimal user association and cell load balancing in wireless networks,” *IEEE/ACM Transactions on Networking*, 2012.
- [64] K. Son, H. Kim, Y. Yi, and B. Krishnamachari, “Base station operation and user association mechanisms for energy-delay tradeoffs in green cellular networks,” *IEEE Journal on Selected Areas in Communications (JSAC)*, 2011.
- [65] S. Liu and J. Virtamo, “Inter-cell coordination with inhomogeneous traffic distribution,” in *Proc. IEEE Next G. Int. Design and Eng.*, 2006.
- [66] 3GPP, “Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access (E-UTRAN),” 2010.
- [67] F. Capozzi, G. Piro, L. Grieco, G. Boggia, and P. Camarda, “Downlink packet scheduling in LTE cellular networks: Key design issues and a survey,” *IEEE Communication Surveys and Tutorials*, 2013.
- [68] N. Sapountzis, S. Sarantidis, T. Spyropoulos, N. Nikaein, and U. Salim, “Reducing the energy consumption of small cell networks subject to QoE constraints,” in *Proc. IEEE Globecom*, 2014.
- [69] *Backhaul technologies for small cells*, Small Cell Forum, 2014.
- [70] O. Tipmongkolsilp, S. Zaghloul, and A. Jukan, “The evolution of cellular backhaul technologies: Current issues and future trends,” in *IEEE Communications Surveys and Tutorials*, 2011.
- [71] N. Sapountzis, T. Spyropoulos, N. Nikaein, and U. Salim, “An analytical framework for optimal downlink-uplink user association in hetnets with traffic differentiation,” in *Proc. IEEE Globecom*, 2015.
- [72] ——, “Optimal downlink and uplink user association in backhaul-limited hetnets,” in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, April 2016, pp. 1–9.
- [73] ——, “User association in hetnets: Impact of traffic differentiation and backhaul limitations,” *IEEE/ACM Transactions on Networking*, vol. 25, no. 6, pp. 3396–3410, Dec 2017.
- [74] G. T. . v.12.0.0 Rel.12, *Study on Small Cell enhancements for E-UTRA and E-UTRAN; Higher layer aspects*. Academic press, 2013.
- [75] V. Pauli and E. Seide, *Dynamic TDD for LTE-A and 5G*, 2015.
- [76] 3GPP, “TS 36.300, Release 13 (version 13.2.0),” 2016.

- [77] J. Kerttula, A. Marttinen, K. Ruttik, R. Jäntti, and M. N. Alam, “Dynamic tdd in lte small cells,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 194, Aug 2016.
- [78] N. Sapountzis, T. Spyropoulos, N. Nikaein, and U. Salim, “Joint optimization of user association and dynamic tdd for ultra-dense networks,” in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, May 2018.
- [79] Small Cell Forum, “Backhaul technologies for small cells: Use cases, requirements and solutions,” 2013.
- [80] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, “Five disruptive technology directions for 5g,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, February 2014.
- [81] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, “Femtocaching: Wireless video content delivery through distributed caching helpers,” in *Proc. IEEE INFOCOM*, 2012.
- [82] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, “Cache in the air: exploiting content caching and delivery techniques for 5g systems,” *IEEE Comm. Magazine*, 2014.
- [83] K. Poularakis, G. Iosifidis, A. Argyriou, and L. Tassiulas, “Video delivery over heterogeneous cellular networks: Optimizing cost and performance,” in *IEEE INFOCOM*, 2014.
- [84] J. Erman, A. Gerber, M. Hajiaghayi, D. Pei, S. Sen, and O. Spatscheck, “To cache or not to cache: The 3g case,” *IEEE Internet Computing*, vol. 15, no. 2, pp. 27–34, March 2011.
- [85] S. Woo, E. Jeong, S. Park, J. Lee, S. Ihm, and K. Park, “Comparison of caching strategies in modern cellular backhaul networks,” in *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys ’13. New York, NY, USA: ACM, 2013, pp. 319–332. [Online]. Available: <http://doi.acm.org/10.1145/2462456.2464442>
- [86] G. Paschos and P. Elia, “Caching for wireless networks,” in *IEEE Sigmetrics Tutorials*, 2016.
- [87] G. S. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah, “Wireless caching: Technical misconceptions and business barriers,” <http://arxiv.org/abs/1602.00173>, 2016.
- [88] M. Leconte, G. Paschos, L. Gkatzikis, M. Draief, S. Vassilaras, and S. Chouvardas, “Placing dynamic content in caches with small population,” in *Proc. IEEE Infocom*, 2016.

- [89] S.-E. Elayoubi and J. Roberts, “Performance and cost effectiveness of caching in mobile access networks,” in *Proceedings of the 2Nd ACM Conference on Information-Centric Networking (ICN)*, 2015.
- [90] X. Bao, Y. Lin, U. Lee, I. Rimac, and R. R. Choudhury, “Dataspotting: Exploiting naturally clustered mobile devices to offload cellular traffic,” in *Proc. of IEEE INFOCOM*, 2013.
- [91] H. Cai, I. Koprulu, and N. B. Shroff, “Exploiting double opportunities for deadline based content propagation in wireless networks,” in *Proc. of IEEE INFOCOM*, 2013.
- [92] L. Vigneri, T. Spyropoulos, and C. Barakat, “Storage on Wheels: Offloading Popular Contents Through a Vehicular Cloud,” in *IEEE WoWMoM*, 2016. [Online]. Available: <https://hal.inria.fr/hal-01315370>
- [93] ——, “Low cost video streaming through mobile edge caching: Modelling and optimization,” *IEEE Transactions on Mobile Computing*, pp. 1–1, 2018.
- [94] ——, “Quality of experience-aware mobile edge caching through a vehicular cloud,” *IEEE Transactions on Mobile Computing*, 2018.
- [95] “T-Mobile Music Freedom,” <https://www.t-mobile.com/offer/free-music-streaming.html>, 2017.
- [96] Y. Yiakoumis, S. Katti, and N. McKeown, “Neutral net neutrality,” in *Proc. ACM SIGCOMM*, 2016, pp. 483–496.
- [97] S. Guo, M. Derakhshani, M. Falaki, U. Ismail, R. Luk, E. Oliver, S. U. Rahman, A. Seth, M. Zaharia, and S. Keshav, “Design and implementation of the kiosknet system,” *Computer Networks*, vol. 55, no. 1, pp. 264–281, 2011.
- [98] “Internet.org by Facebook,” <https://info.internet.org/>, 2017.
- [99] “free basics by Facebook,” <https://0.freebasics.com/desktop>, 2017.
- [100] C. Liang and F. R. Yu, “Wireless network virtualization: A survey, some research issues and challenges,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 358–380, 2015.
- [101] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, “Cellslice: Cellular wireless resource slicing for active ran sharing,” in *COMSNETS*, 2014, pp. 1–6.
- [102] T. Spyropoulos and P. Sermpezis, “Soft cache hits and the impact of alternative content recommendations on mobile edge caching,” in *Proc. ACM Workshop on Challenged Networks (CHANTS)*, 2016, pp. 51–56.
- [103] D. Lee, H. Seo, B. Clerckx, E. Hardouin, D. Mazzarese, S. Nagata, and K. Sayana, “Coordinated multipoint transmission and reception in lte-advanced: deployment scenarios and operational challenges,” *IEEE Communications Magazine*, vol. 50, no. 2, pp. 148–155, 2012.

- [104] W. C. Ao and K. Psounis, “Distributed caching and small cell cooperation for fast content delivery,” in *Proc. ACM MobiHoc*, 2015.
- [105] A. Lindgren, A. Doria, and O. Schelen, “Probabilistic routing in intermittently connected networks,” *SIGMOBILE Mobile Computing and Communication Review*, vol. 7, no. 3, 2003.
- [106] A. Lindgren and K. S. Phanse, “Evaluation of queuing policies and forwarding strategies for routing in intermittently connected networks,” in *Proceedings of IEEE COMSWARE*, 2006.
- [107] H. P. Dohyung Kim and I. Yeom, “Minimizing the impact of buffer overflow in dtn,” in *Proceedings International Conference on Future Internet Technologies (CFI)*, 2008.
- [108] H. Lilliefors, “On the kolmogorov-smirnov test for normality with mean and variance unknown,” *Journal of the American Statistical Association*, Vol. 62. pp. 399-402, 1967.
- [109] A. Guerrieri, A. Montresor, I. Carreras, F. D. Pellegrini, and D. Miorandi, “Distributed estimation of global parameters in delay-tolerant networks,” in *in Procedings of Autonomic and Opportunistic Communication (AOC) Workshop (colocated with WOWMOM*, 2009, pp. 1–7.
- [110] DTN Architecture for NS-2. [Online]. Available: \url{http://www-sop.inria.fr/members/Amir.Krifa/DTN}
- [111] Y. Wang, P. Zhang, T. Liu, C. Sadler, and M. Martonosi, “Movement data traces from princeton zebranet deployments,” CRAWDAD Database. <http://crawdad.cs.dartmouth.edu/>, 2007.
- [112] Cabspotting Project. [Online]. Available: \url{http://cabspotting.org/}
- [113] C. Boldrini, M. Conti, and A. Passarella, “Contentplace: social-aware data dissemination in opportunistic networks,” in *Proceedings of ACM MSWiM*, 2008.
- [114] K. Scott and S. Burleigh, “Bundle protocol specification,” *IRTF RFC 5050*, August 2008.
- [115] G. Sandulescu and S. Nadjm-Tehrani, “Opportunistic dtn routing with window-aware adaptive replication,” in *Proc. of AINTEC '08*. ACM, 2008, pp. 103–112. [Online]. Available: <http://doi.acm.org/10.1145/1503370.1503397>
- [116] K. Shin, K. Kim, and S. Kim, “Traffic management strategy for delay-tolerant networks,” *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1762 – 1770, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1084804512001531>

- [117] F. N. dos Santos, B. Ertl, C. Barakat, T. Spyropoulos, and T. Turletti, “CEDO: content-centric dissemination algorithm for delay-tolerant networks,” in *16th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM ’13)*, 2013.
- [118] L. Pelusi, A. Passarella, and M. Conti, “Opportunistic networking: data forwarding in disconnected mobile ad hoc networks,” *Comm. Mag., IEEE*, vol. 44, no. 11, pp. 134–141, Nov. 2006.
- [119] R. Groenevelt, P. Nain, and G. Koole, “The message delay in mobile ad hoc networks,” *Perform. Eval.*, vol. 62, no. 1-4, 2005.
- [120] Z. J. Haas and T. Small, “A new networking model for biological applications of ad hoc sensor networks,” *IEEE/ACM Trans. Netw.*, vol. 14, no. 1, 2006.
- [121] W. Hsu and A. Helmy, “On nodal encounter patterns in wireless LAN traces,” *IEEE Transactions on Mobile Computing*, vol. 9, pp. 1563–1577, 2010.
- [122] W. Gao, Q. Li, B. Zhao, and G. Cao, “Multicasting in delay tolerant networks: a social network perspective,” in *Proc. ACM MobiHoc*, 2009.
- [123] T. Spyropoulos, T. Turletti, and K. Obraczka, “Routing in delay-tolerant networks comprising heterogeneous node populations,” *IEEE Transactions on Mobile Computing*, vol. 8, pp. 1132–1147, 2009.
- [124] C. Boldrini, M. Conti, and A. Passarella, “Less is more: Long paths do not help the convergence of social-oblivious forwarding in opportunistic networks,” in *Proc. ACM MobiOpp*, 2012.
- [125] C.-H. Lee and D. Y. Eun, “On the forwarding performance under heterogeneous contact dynamics in mobile opportunistic networks,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 6, 2013.
- [126] Y. Kim, K. Lee, N. B. Shroff, and I. Rhee, “Providing probabilistic guarantees on the time of information spread in opportunistic networks,” in *IEEE INFOCOM*, 2013.
- [127] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, “CRAWDAD data set epfl/mobility (v. 2009-02-24),” Downloaded from <http://crawdad.cs.dartmouth.edu/epfl/mobility>.
- [128] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, “CRAWDAD data set cambridge/haggle (v. 2009-05-29),” Downloaded from <http://crawdad.cs.dartmouth.edu/cambridge/haggle>, May 2009.
- [129] W.-J. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy, “Modeling spatial and temporal dependencies of user mobility in wireless mobile networks,” *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, 2009.
- [130] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, “Slaw: A new mobility model for human walks,” in *IEEE INFOCOM*, 2009.

- [131] D. Shah, “Gossip Algorithms,” *Foundations and Trends in Networking*, vol. 3, no. 1, pp. 1–125, 2009.
- [132] R. Couillet and M. Debbah, *Random Matrix Methods for Wireless Communications*. New York, NY, USA: Cambridge University Press, 2011.
- [133] T. Hossmann, G. Nomikos, T. Spyropoulos, and F. Legendre, “Collection and analysis of multi-dimensional network data for opportunistic networking research,” *Comput. Communications*, vol. 35, no. 13, 2012.
- [134] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi, “Human mobility, social ties, and link prediction,” in *Proc. ACM KDD*, 2011.
- [135] E. Gilbert and K. Karahalios, “Predicting tie strength with social media,” in *Proc. CHI*, 2009.
- [136] J. Ott, E. Hyttiä, P. Lassila, J. Kangasharju, and S. Santra, “Floating content for probabilistic information sharing,” *Pervasive Mob. Comput.*, vol. 7, no. 6, pp. 671–689, Dec. 2011.
- [137] A.-K. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot, “Mobiclique: middleware for mobile social networking,” in *ACM Proc. WOSN*, 2009.
- [138] A. Passarella, R. I. Dunbar, M. Conti, and F. Pezzoni, “Ego network models for future internet social networking environments,” *Computer Communications*, vol. 35, no. 18, pp. 2201–2217, 2012.
- [139] N. Eagle and A. Pentland, “Inferring social network structure using mobile phone data,” in *Proceedings of the National Academy of Sciences (PNAS)*, vol. Vol 106 (36), 2009, pp. 15 274–15 278.
- [140] M. S. Granovetter, “The Strength of Weak Ties,” *The American Journal of Sociology*, vol. 78, no. 6, pp. 1360–1380, 1973.
- [141] T. Spyropoulos, R. N. Rais, T. Turletti, K. Obraczka, and A. Vasilakos, “Routing for disruption tolerant networks: taxonomy and design,” *Wirel. Netw.*, vol. 16, no. 8, pp. 2349–2370, Nov. 2010.
- [142] B. Han, P. Hui, V. Kumar, M. Marathe, J. Shao, and A. Srinivasan, “Mobile data offloading through opportunistic communications and social participation,” *IEEE Trans. on Mob. Comp.*, vol. 11, no. 5, 2012.
- [143] M. Pitkänen, T. Kärkkäinen, and et al., “SCAMPI: service platform for social aware mobile and pervasive computing,” *ACM Comput. Commun. Rev.*, vol. 42, no. 4, pp. 503–508, Sep. 2012.
- [144] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, “Web caching and zipf-like distributions: evidence and implications,” in *Proc. IEEE INFOCOM*, 1999.

- [145] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci, “Measuring serendipity: Connecting people, locations and interests in a mobile 3g network,” in *Proc. ACM IMC*, 2009.
- [146] P. Hui and J. Crowcroft, “How small labels create big improvements,” in *IEEE PerCom Workshops*, 2007.
- [147] E. Daly and M. Haahr, “Social network analysis for information flow in disconnected delay-tolerant MANETs,” *IEEE Trans. Mob. Comp.*, vol. 8, no. 5, 2009.
- [148] A. Mei and J. Stefa, “Give2get: Forwarding in social mobile wireless networks of selfish individuals,” *IEEE Trans. on Dependable and Secure Computing*, vol. 9, no. 4, pp. 569–582, july-aug. 2012.
- [149] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen, “Smart: A secure multilayer credit-based incentive scheme for delay-tolerant networks,” *IEEE Trans. on Vehicular Technology*, vol. 58, no. 8, pp. 4628–4639, oct. 2009.
- [150] B. B. Chen and M. C. Chan, “Mobicent: a credit-based incentive system for disruption tolerant network,” in *Proc. IEEE INFOCOM*, 2010.
- [151] A. Passarella, M. Conti, C. Boldrini, and R. I. Dunbar, “Modelling inter-contact times in social pervasive networks,” in *Proc. ACM MSWiM’11*.
- [152] S. Ioannidis and A. Chaintreau, “On the Strength of Weak Ties in Mobile Social Networks,” in *Proc. ACM SNS Workshop*, 2009.
- [153] P. Sermpezis and T. Spyropoulos, “Understanding the effects of social selfishness on the performance of heterogeneous opportunistic networks,” Eurecom, Tech. Rep. RR-13-283, 2013.
- [154] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, “MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks,” in *Proc. IEEE INFOCOM*, 2006.
- [155] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.
- [156] P. V. Mieghem, *Graph Spectra for Complex Networks*. New York, NY, USA: Cambridge University Press, 2011.
- [157] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, “Distributed community detection in delay tolerant networks,” in *Proceedings of 2Nd ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture*, ser. MobiArch ’07, 2007.
- [158] A. Scherrer, P. Borgnat, E. Fleury, J. L. Guillaume, and C. Robardet, “Description and simulation of dynamic mobility networks,” *Comput. Netw.*, vol. 52, no. 15, pp. 2842–2858, Oct. 2008. [Online]. Available: <http://dx.doi.org/10.1016/j.comnet.2008.06.007>

- [159] M. Newman and M. Girvan, “Finding and evaluating community structure in networks,” 01 2004.
- [160] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, ser. NIPS’01. Cambridge, MA, USA: MIT Press, 2001, pp. 849–856. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2980539.2980649>
- [161] S. White and P. Smyth, “A spectral clustering approach to finding communities in graphs,” in *SDM*, 2005.
- [162] T. Henderson, D. Kotz, and I. Abyzov, “The changing usage of a mature campus-wide wireless network,” in *Proceedings of ACM MOBICOM*, 2004.
- [163] C. Tudec and T. Gross, “A mobility model based on wlan traces and its validation,” in *Proceedings of IEEE INFOCOM*, 2005.
- [164] N. Eagle and A. Pentland, “Reality mining: sensing complex social systems,” *Personal and Ubiquitous Computing*, May 2006.
- [165] A. Chaintreau, A. Mtibaa, L. Massoulie, and C. Diot, “The diameter of opportunistic mobile networks,” in *Proceedings of ACM CoNEXT*, 2007.
- [166] V. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of Stat. Mech.: Theory and Exp.*, vol. 2008, no. 10, 2008.
- [167] C. Boldrini, M. Conti, and A. Passarella, “Users mobility models for opportunistic networks: the role of physical locations,” in *WRECOM*, 2007.
- [168] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnović, “Power law and exponential decay of inter contact times between mobile devices,” in *Proceedings of ACM MobiCom ’07*, 2007.
- [169] P. Hui, J. Crowcroft, and E. Yoneki, “Bubble rap: social-based forwarding in delay tolerant networks,” *IEEE Trans. Mob. Comp.*, vol. 10, no. 11, 2011.
- [170] A. Picu and T. Spyropoulos, “Distributed stochastic optimization in Opportunistic Nets: The case of optimal relay selection,” in *ACM CHANTS*, 2010.
- [171] N. Eagle and A. Pentland, “Reality mining: sensing complex social systems,” *Pers. Ubiquit. Comput.*, vol. 10, no. 4, 2006.
- [172] V. Lenders, J. Wagner, and M. May, “Measurements from an 802.11b mobile ad hoc network,” in *IEEE WoWMoM*, 2006.
- [173] P. Brémaud, *Markov chains : Gibbs fields, Monte Carlo simulation, and queues*. Springer, 2001.

- [174] K. Fall, “A delay-tolerant network architecture for challenged internets,” in *ACM SIGCOMM*, 2003.
- [175] T. Spyropoulos, K. Psounis, and C. Raghavendra, “Spray and Wait: an efficient routing scheme for intermittently connected mobile networks,” in *ACM WDTN*, 2005.
- [176] A. Balasubramanian, B. Levine, and A. Venkataramani, “Replication routing in DTNs: A resource allocation approach,” *IEEE/ACM Trans. Netw.*, vol. 18, no. 2, 2010.
- [177] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, “Impact of human mobility on opportunistic forwarding algorithms,” *IEEE Trans. Mob. Comp.*, vol. 6, no. 6, 2007.
- [178] H. Cai and D. Y. Eun, “Crossing Over the Bounded Domain: From Exponential to Power-Law Intermeeting Time in Mobile Ad Hoc Networks,” *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, 2009.
- [179] S. L. Albin, “Approximating a Point Process by a Renewal Process, II: Superposition Arrival Processes to Queues,” *Operations Research*, vol. 32, no. 5, 1984.
- [180] I. Meilijson, “Limiting properties of the mean residual lifetime function,” *Ann. Math. Statist.*, vol. 43, no. 1, 1972.
- [181] D. R. Cox, *Renewal Theory*, ser. Methuen’s monographs on applied probability and statistics. Methuen, 1962.
- [182] A. Picu and T. Spyropoulos, “DTN-Meteo: Towards Forecasting the Performance of DTN Protocols under Heterogeneous Mobility,” ETHZ, Tech. Rep. 348, 2012.
- [183] C. Boldrini, M. Conti, and A. Passarella, “Performance modelling of opportunistic forwarding with imprecise knowledge,” in *WiOpt*, 2012.
- [184] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility,” in *IEEE PERCOMW*, 2007.
- [185] D. Ciullo, V. Martina, M. Garetto, and E. Leonardi, “Impact of correlated mobility on delay-throughput performance in mobile ad hoc networks,” *Networking, IEEE/ACM Transactions on*, vol. 19, no. 6, 2011.
- [186] M. Asplund and S. Nadjm-Tehrani, “Analysing delay-tolerant networks with correlated mobility,” in *Ad-hoc, Mobile, and Wireless Networks*, ser. Lecture Notes in Computer Science, 2012, vol. 7363.
- [187] E. Bulut, S. C. Geyik, and B. K. Szymanski, “Utilizing correlated node mobility for efficient {DTN} routing,” *Pervasive and Mobile Computing*, 2013.

- [188] A. Picu and T. Spyropoulos, “Performance of Distributed Algorithms in DTNs: Towards an Analytical Framework for Heterogeneous Mobility,” in *IEEE GLOBECOM*, 2011.
- [189] J. Kemeny and L. Snell, *Finite Markov Chains*. Springer, 1983.
- [190] A. Picu and T. Spyropoulos, “Minimum Expected \*-cast Time in DTNs,” in *ICST BIONETICS*, 2009, pp. 103–116.
- [191] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, “Pocket switched networks and human mobility in conference environments,” in *ACM WDTN*, 2005.
- [192] R. Wolff, “Poisson Arrivals See Time Averages,” *Operations Research*, vol. 30, no. 2, 1982.
- [193] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Performance analysis of mobility-assisted routing,” in *ACM MobiHoc*, 2006.
- [194] “Mobile data offloading through WiFi,” 2010, proxim Wireless.
- [195] K. Lee, I. Rhee, J. Lee, S. Chong, and Y. Yi, “Mobile data offloading: How much can WiFi deliver,” in *Proc. of ACM CoNEXT*, 2010.
- [196] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, “Energy consumption in mobile phones: A measurement study and implications for network applications,” in *Proc. of ACM IMC*, 2009.
- [197] S. Sen, C. Joe-Wong, S. Ha, and M. Chiang, “Incentivizing time-shifting of data: a survey of time-dependent pricing for internet access,” *IEEE Communications Magazine*, vol. 50, no. 11, 2012.
- [198] C. Joe-Wong, S. Ha, and J. Bawa, “When the price is right: Enabling time-dependent pricing for mobile data,” *ACM SIGCHI 2013*.
- [199] S. M. Ross, *Stochastic Processes*, 2nd ed. John Wiley & Sons, 1996.
- [200] D.Y.Barrer, “Queuing with impatient customers and ordered service,” *Operations Research*, 1957.
- [201] R.E.Stanford, “Reneging phenomena in single channel queues,” *Mathematics of Operations Research*, 1979.
- [202] U. Yechiali and P. Naor, “Queueing problems with heterogeneous arrivals and service,” *Operations Research*, 1971.
- [203] N. Perel and U. Yechiali, “Queues with slow servers and impatient customers,” *European Journal of Operations Research*, no. 201, 2010.
- [204] E. Altman and U. Yechiali, “Analysis of customers’ impatience in queues with server vacations,” *Queueing Systems*, vol. 52, 2006.

- [205] A. Sharma, V. Navda, R. Ramjee, V. N. Padmanabhan, and E. M. Belding, “Cool-tether: Energy efficient on-the-fly WiFi hot-spots using mobile phones,” in *Proc. of ACM CoNEXT*, 2009.
- [206] D. Zhang and C. K. Yeo, “Optimal handing-back point in mobile data offloading,” in *Proc. of IEEE VNC*, 2012.
- [207] S. Wietholter, M. Emmelmann, R. Andersson, and A. Wolisz, “Performance evaluation of selection schemes for offloading traffic to IEEE 802.11 hotspots,” in *Proc. of IEEE ICC*, 2012.
- [208] K. Berg and M. Katsigiannis, “Optimal cost-based strategies in mobile network offloading,” in *Proc. of ICST CROWNCOM*, 2012.
- [209] Y. Im, C. J. Wong, S. Ha, S. Sen, T. Kwon, and M. Chiang, “AMUSE: Empowering users for cost-aware offloading with throughput-delay tradeoffs,” in *Proc. of IEEE Infocom Mini-conference*, 2013.
- [210] S. Dimatteo, P. Hui, B. Han, and V. Li, “Cellular traffic offloading through WiFi networks,” in *Proc. of IEEE MASS*, 2011.
- [211] T. Bonald and A. Proutiere, “Wireless downlink data channels: User performance and cell dimensioning,” in *Proc. Mobile Computing and Networking (MobiCom)*, 2003.
- [212] J. G. Andrews, S. Singh, Q. Ye, X. Lin, and H. S. Dhillon, “An overview of load balancing in HetNets: Old myths and open problems,” *IEEE Wireless Communications*, 2014.
- [213] H. Boostanimehr and V. Bhargava, “Unified and distributed QoS-driven cell association algorithms in heterogeneous networks,” *IEEE Transactions on Wireless Communications*, 2015.
- [214] D. Fooladivanda and C. Rosenberg, “Joint resource allocation and user association for heterogeneous wireless cellular networks,” *IEEE Transactions on Wireless Communications*, 2013.
- [215] T. Han and N. Ansari, “Smart grid enabled mobile networks: Jointly optimizing BS operation and power distribution,” in *Proc. IEEE International Conference on Communications*, 2014.
- [216] J. Bartelt, A. Fehske, H. Klessig, G. Fettweis, and J. Voigt, “Joint bandwidth allocation and small cell switching in heterogeneous networks,” in *Proc. IEEE Vehicular Technology Conference*, 2013.
- [217] Y. Wang and K. Pedersen, “Performance analysis of enhanced inter-cell interference coordination in LTE-Advanced heterogeneous networks,” in *Vehicular Technology Conference (VTC Spring)*, 2012.

- [218] S. Deb, P. Monogioudis, J. Miernik, and J. P. Seymour, “Algorithms for enhanced inter-cell interference coordination (eICIC) in lte hetnets,” *IEEE/ACM Transactions on Networking*, 2014.
- [219] J. Lee, Y. Kim, H. Lee, B. L. Ng, D. Mazzarese, J. Liu, W. Xiao, and Y. Zhou, “Coordinated multipoint transmission and reception in LTE-advanced systems,” *IEEE Communications Magazine*, 2012.
- [220] J. Ghimire and C. Rosenberg, “Revisiting scheduling in heterogeneous networks when the backhaul is limited,” *IEEE Journal on Selected Areas in Communications (JSAC)*, 2015.
- [221] M. Shariat, E. Pateromichelakis, A. Quddus, and R. Tafazolli, “Joint TDD backhaul and access optimization in dense small cell networks,” *IEEE Transactions on Vehicular Technology*, 2013.
- [222] O. Somekh, O. Simeone, A. Sanderovich, B. Zaidel, and S. Shamai, “On the impact of limited-capacity backhaul and inter-users links in cooperative multicell networks,” in *Proc. Conference Information Sciences and System (CISS)*, 2008.
- [223] P. Rost, C. Bernardos, A. Domenico, M. Girolamo, M. Lalam, A. Maeder, D. Sabella, and D. Wubben, “Cloud technologies for flexible 5G radio access networks,” *IEEE Communications Magazine*, 2014.
- [224] N. Wang, E. Hossain, and V. K. Bhargava, “Joint downlink cell association and bandwidth allocation for wireless backhauling in two-tier hetnets with large-scale antenna arrays,” *IEEE Transactions on Wireless Communications*, 2016.
- [225] W. C. Liao, M. Hong, and Z. Q. Luo, “Max-min network flow and resource allocation for backhaul constrained heterogeneous wireless networks,” in *Proc. IEEE ICASSP*, 2014.
- [226] H. Beyranvand, W. Lim, M. Maier, C. Verikoukis, and J. A. Salehi, “Backhaul-aware user association in fiwi enhanced lte-a heterogeneous networks,” *IEEE Transactions on Wireless Communications*, 2015.
- [227] A. Mesodiakaki, F. Adelantado, L. Alonso, and C. Verikoukis, “Joint uplink and downlink cell selection in cognitive small cell heterogeneous networks,” in *Proc. IEEE Globecom*, 2014.
- [228] A. D. Domenico, V. Savin, and D. Ktenas, “A backhaul-aware cell selection algorithm for heterogeneous cellular networks,” in *Proc. of IEEE PIMRC*, 2013.
- [229] Z. Cui and R. Adve, “Joint user association and resource allocation in small cell networks with backhaul constraints,” in *Proc. IEEE CISS*, 2014.
- [230] F. Pantisano, M. Bennis, W. Saad, and M. Debbah, “Cache-aware user association in backhaul-constrained small cell networks,” in *in Proc. IEEE WiOpt*, 2014.

- [231] D. Chen, T. Quek, and M. Kountouris, “Backhauling in heterogeneous cellular networks: Modeling and tradeoffs,” *IEEE Transactions on Wireless Communications*, 2015.
- [232] H. Kim, H. Y. Kim, Y. Cho, and S.-H. Lee, “Spectrum breathing and cell load balancing for self organizing wireless networks,” in *Proc. IEEE Communications Workshops*, 2013.
- [233] M. Harchol-Balter, *Performance Modeling and Design of Computer Systems*. Imperial college press, 2010.
- [234] G. Arvanitakis, T. Spyropoulos, and F. Kaltenberger, “An analytical model for flow-level performance in heterogeneous wireless networks,” in *IEEE Trans. on Wireless Communications*, vol. 3, no. 17, pp. 1488 – 1501, 2018.
- [235] T. Bonald, S. Borst, N. Hegde, M. Jonckheere, and A. Proutiere, “Flow-level performance and capacity of wireless networks with user mobility,” 2009.
- [236] H. Elshaer, F. Boccardi, M. Dohler, and R. Irmer, “Downlink and uplink decoupling: A disruptive architectural design for 5G networks,” in *Proc. IEEE Globecom*, 2014.
- [237] “<http://cbnl.com/solutions-mobile-backhaul>.”
- [238] N. Sapountzis, T. Spyropoulos, N. Nikaein, and U. Salim, “Hop: Hierarchize then optimize: A distributed framework for user association and flexible tdd allocation for access and backhaul networks.”
- [239] N. Sapountzis, T. Spyropoulos, N. Nikaein, and U. Salim, “Under-provisioned backhaul: How capacity and topology impacts user and network-wide performance,” Tech Report RR-16-311, Eurecom, 2016.
- [240] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.
- [241] G. 36.300, “Evolved universal terrestrial radio access (E-UTRA); further enhancements to LTE time division duplex (TDD) for downlink-uplink (DL-UL) interference management and traffic adaptation,” 2012.
- [242] Z. G. Raphael T. Haftka, *Elements of Structural Optimization*. Springer Netherlands, 1992.
- [243] A. Mesodiakaki, F. Adelantado, L. Alonso, and C. Verikoukis, “Energy-efficient user association in cognitive heterogeneous networks,” *IEEE Communications Magazine*, 2014.
- [244] D. G. Luenberger, Y. Ye *et al.*, *Linear and nonlinear programming*. Springer, 1984, vol. 2.

- [245] N. Sapountzis, T. Spyropoulos, N. Nikaein, and U. Salim, “User association in over- and under- provisioned backhaul hetnets,” Tech Report RR-16-318, Eurecom, 2016.
- [246] G. 24.312, “Access network discovery and selection function (andsf) management object (mo),” 2016.
- [247] S. Sesia, I. Toufik, and B. M., *LTE - The UMTS Long Term Evolution: From Theory to Practice*, 2nd Edition. Wiley, 2011.
- [248] I. S. 802.16m, “IEEE p802.16m-2007 draft standards for local and metropolitan area networks part 16: Air interface for fixed broadcast wireless access systems,,” 2007.
- [249] T. Bu, L. Li, and R. Ramjee, “Generalized proportional fair scheduling in third generation wireless data networks,” in *Proc. IEEE Infocom*, 2006.
- [250] S. Das, H. Viswanathan, and G. Rittenhouse, “Dynamic load balancing through coordinated scheduling in packet data systems,” in *Proc. IEEE Infocom*, 2003.
- [251] 3GPP, *Technical Report LTE; Evolved Universal Terrestrial Radio Access (E-UTRA)*, TR 136 931, 2011.
- [252] Alcatel-Lucet, “Mobile bakhaul architecture for hetnet, <https://www.alcatel-lucent.com/solutions/mobile-backhaul>,” 2015.
- [253] G. 36.828, “Evolved universal terrestrial radio access (E-UTRA) and radio access network (E-UTRAN); overall description,” 2012.
- [254] N. Liakopoulos, G. Paschos, and T. Spyropoulos, “Robust user association for ultra dense networks,” in *In Proc. of IEEE INFOCOM*, 2018.
- [255] M. A. Maddah-Ali and U. Niesen, “Fundamental limits of caching,” *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.
- [256] P. Sermpezis and T. Spyropoulos, “Offloading on the edge: Performance and cost analysis of local data storage and offloading in HetNets,” in *IEEE WONS 2017*, 2017.
- [257] E. Leonardi and G. Luca, “Least recently used caches under the shot noise model,” in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 2281–2289.
- [258] N. C. Fofack, P. Nain, G. Neglia, and D. Towsley, “Analysis of ttl-based cache networks,” in *6th International ICST Conference on Performance Evaluation Methodologies and Tools*, Oct 2012, pp. 1–10.
- [259] E. G. Coffman and P. J. Denning, *Operating systems theory*. Prentice-Hall Englewood Cliffs, NJ, 1973, vol. 973.

- [260] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, “Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution,” *IEEE Comm. Magazine*, 2013.
- [261] G. Szabo and B. A. Huberman, “Predicting the Popularity of Online Content,” *ACM Comm.*, 2010.
- [262] <http://www.congas-project.eu/youstataalyzer-database>, 2013.
- [263] R. Crane and D. Sornette, “Robust dynamic classes revealed by measuring the response function of a social system,” *Proc. of the National Academy of Sciences*, 2008.
- [264] T. Hossfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen, “Initial delay vs. interruptions: Between the devil and the deep blue sea,” in *QoMEX*, 2012.
- [265] S. Karlin and H. Taylor, *A First Course in Stochastic Processes*. Elsevier Science, 2012. [Online]. Available: <https://books.google.fr/books?id=dSDxjX9nmmMC>
- [266] A. Antoniou and W.-S. Lu, *Practical optimization - algorithms and engineering applications*. Springer, 2007.
- [267] P. Raghavan and C. D. Tompson, “Randomized rounding: A technique for provably good algorithms and algorithmic proofs,” *Combinatorica*, vol. 7, no. 4, pp. 365–374, 1987.
- [268] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, “SLAW: A New Mobility Model for Human Walks,” in *IEEE INFOCOM*, 2009.
- [269] Senza Fili Consulting, “The economics of small cells and wi-fi offload,” 2013.
- [270] “Veniam,” <https://veniam.com>, 2015.
- [271] K. Poularakis, G. Iosifidis, and L. Tassiulas, “Approximation algorithms for mobile data caching in small cell networks,” *IEEE Transactions on Communications*, vol. 62, no. 10, pp. 3665–3677, Oct 2014.
- [272] P. Sermpezis and T. Spyropoulos, “Not all content is created equal: Effect of popularity and availability for content-centric opportunistic networking,” in *Proc. of ACM MobiHoc*, 2014.
- [273] J. Whitbeck, M. Amorim, Y. Lopez, J. Leguay, and V. Conan, “Relieving the wireless infrastructure: When opportunistic networks meet guaranteed delays,” in *Proc. IEEE WoWMoM*, 2011.
- [274] P. Sermpezis, L. Vigneri, and T. Spyropoulos, “Offloading on the edge: Analysis and optimization of local data storage and offloading in hetnets,” <http://arxiv.org/abs/1503.00648>, 2015.

- [275] R. Zhou, S. Khemmarat, and L. Gao, “The impact of youtube recommendation system on video views,” in *In Proc. of ACM IMC 2010*.
- [276] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, “Mobile edge computing a key technology towards 5g,” *ETSI White Paper No. 11*, 2016.
- [277] D. Kahneman. New York: Farrar, Straus and Giroux, 2011.
- [278] D. K. Krishnappa, M. Zink, C. Griwodz, and P. Halvorsen, “Cache-centric video recommendation: an approach to improve the efficiency of youtube caches,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 11, no. 4, p. 48, 2015.
- [279] L. E. Chatzieleftheriou, M. Karaliopoulos, and I. Koutsopoulos, “Caching-aware recommendations: Nudging user preferences towards better caching performance,” in *Proc. IEEE INFOCOM*, 2017.
- [280] P. Sermpezis, T. Spyropoulos, L. Vigneri, and T. Giannakas, “Femto-caching with soft cache hits: Improving performance with related content recommendation,” in *Proc. IEEE GLOBECOM*, 2017.
- [281] X. Su and T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” *Adv. in Artif. Intell.*, pp. 4:2–4:2, 2009.
- [282] G. Linden, B. Smith, and J. York, “Amazon.com recommendations: Item-to-item collaborative filtering,” *IEEE Internet computing*, vol. 7, no. 1, pp. 76–80, 2003.
- [283] M. Dehghan, L. Massoulie, D. Towsley, D. Menasche, and Y. Tay, “A utility optimization approach to network cache design,” in *Proc. IEEE INFOCOM*, 2016.
- [284] P. Sermpezis, T. Giannakas, T. Spyropoulos, and L. Vigneri, “Soft cache hits: Improving performance through recommendation and delivery of related content,” *IEEE Journal on Selected Areas in Communications*, 2018.
- [285] A. Krause and D. Golovin, “Submodular function maximization,” *Tractability: Practical Approaches to Hard Problems*, vol. 3, no. 19, p. 8, 2012.
- [286] G. L. Nemhauser and L. A. Wolsey, “Best algorithms for approximating the maximum of a submodular set function,” *Mathematics of operations research*, vol. 3, no. 3, pp. 177–188, 1978.
- [287] M. Sviridenko, “A note on maximizing a submodular set function subject to a knapsack constraint,” *Operations Research Letters*, vol. 32, no. 1, pp. 41–43, 2004.
- [288] P. Sermpezis, T. Spyropoulos, L. Vigneri, and T. Giannakas, “Femto-caching with soft cache hits: Improving performance through recommendation and delivery of related content,” *available at <https://arxiv.org/abs/1702.04943>*, 2017.

- [289] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák, “Maximizing a monotone submodular function subject to a matroid constraint,” *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1740–1766, 2011.
- [290] Y. Filmus and J. Ward, “Monotone submodular maximization over a matroid via non-oblivious local search,” *SIAM Journal on Computing*, vol. 43, no. 2, pp. 514–542, 2014.
- [291] <http://netsg.cs.sfu.ca/youtubedata/>, 2007.
- [292] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, “Image-based recommendations on styles and substitutes,” in *Proc. ACM SIGIR*, 2015, pp. 43–52.
- [293] F. M. Harper and J. A. Konstan, “The movielens datasets: History and context,” *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 5, no. 4, p. 19, 2016.
- [294] A. Ghosh, N. Mangalvedhe, R. Ratasuk, B. Mondal, M. Cudak, E. Visotsky, T. A. Thomas, J. G. Andrews, P. Xia, H. S. Jo, H. S. Dhillon, and T. D. Novlan, “Heterogeneous cellular networks: From theory to practice,” *IEEE Communications Magazine*, vol. 50, no. 6, pp. 54–64, 2012.
- [295] J. G. Andrews, H. Claussen, M. Dohler, S. Rangan, and M. C. Reed, “Femtocells: Past, present, and future,” *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 3, pp. 497–508, 2012.
- [296] J. G. Andrews, “Seven ways that hetnets are a cellular paradigm shift,” *IEEE Communications Magazine*, vol. 51, no. 3, pp. 136–144, 2013.
- [297] A. Liu and V. K. Lau, “Cache-enabled opportunistic cooperative mimo for video streaming in wireless systems,” *IEEE Trans. Signal Processing*, 2015.
- [298] D. Munaro, C. Delgado, and D. S. Menasché, “Content recommendation and service costs in swarming systems,” in *Proc. IEEE ICC*, 2015, pp. 5878–5883.
- [299] D. K. Krishnappa, M. Zink, and C. Griwodz, “What should you cache?: a global analysis on youtube related video caching,” in *Procc. ACM NOSSDAV Workshop*, 2013, pp. 31–36.
- [300] A. Liu and V. K. N. Lau, “Exploiting Base Station Caching in MIMO Cellular Networks: Opportunistic Cooperation for Video Streaming,” *IEEE Transactions on Signal Processing*, vol. 63, no. 1, pp. 57–69, Jan 2015.
- [301] N. Naderializadeh, M. A. Maddah-Ali, and A. S. Avestimehr, “Cache-aided interference management in wireless cellular networks,” in *2017 IEEE International Conference on Communications (ICC)*, May 2017.
- [302] A. Tuholukova, G. Neglia, and T. Spyropoulos, “Optimal cache allocation for femto helpers with joint transmission capabilities,” in *2017 IEEE International Conference on Communications (ICC)*, 2017.

- [303] S. Khuller, A. Moss, and J. S. Naor, “The budgeted maximum coverage problem,” *Information Processing Letters*, vol. 70, no. 1, pp. 39–45, 1999.