Autonomous Self-Backhauled LTE Mesh Network with QoS Guarantee

Romain Favraud, Chia-Yu Chang, Navid Nikaein
Communication Systems Department, EURECOM, France
Email: firstname.name@eurecom.fr

***Abstract*—Public safety communication systems are currently evolving due to emergence of LTE as a mature solution to replace the legacy ones while providing new services. However, LTE is initially designed for commercial cellular network and needs to be furthermore evolved to tackle the substantial requirements of public safety use cases. For instance, opportunistic deployments require modifications to enable the autonomous operation and meshing of moving base stations while satisfying heterogeneous frequency band availability. In this article, we present a novel radio access network infrastructure architecture that enables multi-hop LTE mesh networking for nomadic and autonomous base stations via in-band self-backhauling. Furthermore, we investigate the coordination and orchestration functionality within the proposed architecture and propose a hierarchical resource scheduling algorithm in order to efficiently meet QoS requirements for real-time traffic while maximizing the throughput for elastic flows. To demonstrate the feasibility and reliability of our proposed architecture, we implement the corresponding self-backhauling air-interface based on OpenAirInterface (OAI) platform and compare it with the legacy LTE air-interface. Finally, we evaluate the efficiency and adaptability of our proposed resource scheduling algorithm in various network topology and heterogeneous traffic flows with QoS requirements.**

***Index Terms*—LTE/LTE-A, Self-backhauling, Public safety, Moving Cells, Autonomous network, Relay interface, QoS.**

I. Introduction

As posed in the dominating position in the 4G era, Long Term Evolution (LTE) is still continuing to evolve and extend its features to be compatible with divergent 5G use cases. Among these cases, Public Safety (PS) communications are crucial for the future radio access technologies (RATs) and are highly desirable by ITU [1]. Whereas existing PS solutions (e.g., Terrestrial Trunked Radio (TETRA) [2] and Project 25 (P25) [3]) are mature and can provide reliable mission-critical voice communications, their designs cannot meet the requirement of high-bandwidth applications like real-time video streaming or the exchanges of large amounts of data [4]. Facing similar challenges, military authorities are also evaluating if they can leverage the fast evolution of commercial off-the-shelf (COTS) technologies, particularly LTE, to answer their evolving communication needs [5], [6]. Moreover, major business opportunities are arising in the private sector as LTE-enabled private networks and trending in the industry to provide high data rate communications in specific use cases that cannot be addressed until now, as in the mining industry [7].

However, LTE is a commercial cellular network and was not designed in the initial releases to support various use cases such as providing PS services and the corresponding requirements like reliability, confidentiality and security while being compatible with different topologies. Therefore, the rising question is whether LTE and its evolutions are appropriate solutions for PS services and more generally for diverse 5G use cases. In particular, several specification groups and working items have now emerged to evolve LTE to be compliant with PS communications [8].

Unlike common scenarios in the cellular communication system, the LTE base stations (BSs), called eNodeBs (eNBs), may lose access to the core network (CN) (namely evolved packet core (EPC) in LTE) or to other eNBs due to network outage, network mobility, connection disruption or even the lack of equipment. When it happens, there is no possibility for the network to provide any service to all served user equipments (UEs). To handle such scenario, 3GPP addresses the *Isolated E-UTRAN* concept, originally designed for the PS communications, that allows each eNB to continuously provide minimal services for its served UEs when it only has limited or even no access to the EPC [9], [10]. However, 3GPP studies do not address how such isolated eNBs can communicate with each other; hence, it left a blank space for different technologies to be applied. For instance, the work in [4], [11] aimed to provide connectivity between BSs using different RATs. Furthermore, the re-utilization of the same LTE RAT is feasible and can reduce extra complexity of inter-working with different RATs.

Despite the appeals of utilizing the same LTE RAT, several challenges and design choices remain open including (i) usage of the spectrum between access and backhaul links, namely in-band or out-band backhauling [12], (ii) dynamic meshing of fixed and/or moving BSs, (iii) cross-layer coordination and reconfiguration of BSs to enable autonomous operation, (iv) support of application and service quality of service (QoS) requirements, and (v) retaining service continuity and transparency with respect to common UEs to ease deployment and enable future evolution. These challenges call for additional design considerations in BS architecture [13] such as specific control and management mechanisms, as well as careful implementation and configuration to ensure the architecture feasibility, performance reliability and efficiency.

In this article, we propose a complete solution that comprises a genuine BS architecture as well as the related approaches fulfilling all the above-mentioned challenges to enable autonomous and mobile self-backhauled LTE mesh networks. Such full solution is based on our previous work in [14] that only addressed the considered problem in some specific scenarios. Our contributions are as follows:

- We present all considered use cases, delineate the requirements, and compare potential RAT candidates to show the validity of an LTE-based solution for the PS scenarios (section II and III).
- We propose a genuine BS architecture to enable autonomous self-backhauled LTE mesh network, and detail its physical later interfaces, design elements, operation flows and connection procedures (section IV and V).
- We introduce a hierarchical coordinated resource scheduling algorithm that consider the traffic and characteristics of backhaul links in order to meet per-flow QoS requirements under polynomial-time computation complexity (section VI).
- Using OpenAirInterface platform, we implement our proposal in physical layer aspect and conduct extensive emulations to justify its feasibility in term of acceptable processing time and comparable link-level performance (section VII).
- We investigate the adaptability of the proposed hierarchical approach that can provide the best trade-off among different traffic performance metrics over several network topologies and heterogeneous traffic flows (section VIII, and IX).

II. Use cases

In this section, we introduce the scenarios and derive the high-level requirements for the envisioned self-backhauling mesh network. Furthermore, some essential external constraints are provided to reflect the targets to properly design a solution.

A. *Scenarios*

Normally, a nationwide wireless network relies on a wired network supporting fixed BSs providing planned coverage and bringing services to mobile entities (e.g., hand-held UEs or vehicle integrated devices) that requires a seamless access to the CN. Such a deployed network can support a variety of use cases; however, stringent requirements shall be considered when utilizing it for PS communications, namely robustness, reliability, and non-prone to malfunctions and outages [15]. Despite aforementioned deployment requirements, such fixed BSs may still not survive against unexpected events such as earthquake, hurricane, tsunami, and wildfire. Nevertheless, first responders still need efficient PS communications in all circumstances even in harsh environments that require some large area opportunistic BSs deployments. In that sense, the PS wireless communications cannot rely solely on the planned network and must be able to ensure minimum services and sufficient level of quality when the planned network is not fully available or not possible to deploy [15].

In view of the above limitations, TABLE II captures twelve scenarios that can arise depending on four criteria: (*i*) UE-to-BS connectivity, (*ii*) BS-to-CN connectivity, (*iii*) BS mobility, and (*iv*) BS-to-BS connectivity. In the following, we go through these cases in more details.

- **UE-to-BS connectivity**: In the nominal cases, users are under BS coverage (Scenario 1 to 11). When combined with all other ideal factors, Scenario 1 is the nominal and ideal case with planned and fixed BS coverage and BSs having complete access to the CN that allows to provide services with no intermissions to in-coverage UEs (e.g., continuous

TABLE I: Acronym Table

Abbreviation	Full name
AFR	Average Frequency Reuse
BS	Base Station
CN	Core Network
COE	Coordination and Orchestration Entity
CNS	Centralized Node Scheduler
CSA	Controller Scheduling Algorithm
CQI	Channel Quality Indicator
DCI	Downlink Control Information
DL	Downlink
DLS	Distributed Link Scheduler
DeNB	Donor evolved Node B
DLSS	Downlink StartSymbol
eMBMS	evolved Multimedia Broadcast Multicast Service
eNB	evolved Node B
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
e2NB	enhanced evolved Node B
ESI	End Symbol Index
FDD	Frequency Division Duplexing
HARQ	Hybrid Automatic Repeat reQuest
HSS	Home Subscriber Server
MBSFN	Multicast Broadcast Single Frequency Network
MCS	Modulation and Coding Scheme
MME	Mobility Management Entity
PRB	Physical Resource Block
PDCCH	Physical Downlink Control Channel
PDSCH	Physical Downlink Shared Channel
PUCCH	Physical Uplink Control Channel
PUSCH	Physical Uplink Shared Channel
P-GW	Packet Data Network Gateway
RX	Receiver
RN	Relay Node
R-PDCCH	Relay Physical Downlink Control Channel
R-PDSCH	Relay Physical Downlink Shared Channel
S-GW	Serving Gateway
SF	Subframe
SuF	SuperFrame
TDD	Time Division Duplexing
TTI	Transmission Time Interval
TX	Transmitter
TBS	Transport Block Size
UL	Uplink
UE	User Equipment
vUE	virtual UE

TABLE II: Possible network scenarios

Scenario	UE-to-BS connectivity	BS-to-CN connectivity	BS mobility	BS-to-BS connectivity
1	In BS coverage	Complete	Fixed	Complete
2	In BS coverage	Limited	Fixed	Complete
3	In BS coverage	Limited	Fixed	Limited
4	In BS coverage	None	Fixed	Complete
5	In BS coverage	None	Fixed	Limited
6	In BS coverage	None	Fixed	None
7	In BS coverage	Limited	Moving	Complete
8	In BS coverage	Limited	Moving	Limited
9	In BS coverage	None	Moving	Complete
10	In BS coverage	None	Moving	Limited
11	In BS coverage	None	Moving	None
12	Out-of-coverage	-	-	-

link connectivity with operation center, monitoring, billing). Such scenario happens in the covered cities and (sub)-urban environments where the network deployment has been previously designed and planned. However, users may be out-of-coverage of the service area or fail to maintain any connection to BSs due to their mobility (scenario 12). Hence, users may rely on proximity services (ProSe) based on device-to-device (D2D) communications [16] with nearby users.

- **BS-to-CN connectivity**: When the backhaul link (i.e., links between BS and CN) disruption or failure happens, the CN may not be fully accessible from BSs. If it can only provide some control plane functionalities, i.e., BSs can still accept PS UEs connections but need additional mechanisms to provide data transportations (e.g., local routing [17]), it is referred as limited BS-to-CN connectivity (scenarios 2 and 3). Otherwise, it is referred as unavailable BS-to-CN connectivity (scenarios 4, 5 and 6) and some local CN functionalities at BS is required to serve PS UEs.
- **BS-to-BS connectivity**: We can observe that the BS-to-BS connectivity does not necessarily rely on the backhaul connectivity. For instance, a full connectivity between BSs (scenarios 2 and 4) can allow to form a large network even with limited BS-to-CN connectivity.[1] Note that the limited BS-to-BS connectivity (scenarios 3 and 5) case can only exchange partial information between BSs and may restrict certain features such as inter-BS handover.
- **BS mobility**: Moving BSs can be utilized in a dynamic fashion (e.g., during fight against forest wildfire or using vehicular BS being on land or at sea [13], [18]). In such cases, it is difficult to maintain a good connectivity between BS and CN (scenarios 7 to 11) and it is complicated to interconnect these moving BSs in terms of different areas (coverage region, propagation condition) and embedded equipments (dedicated or shared wireless backhaul). In that sense, the network topology will split and merge dynamically and it shall be maintained properly to provide the widest service area to covered UEs. Furthermore, the interference between BSs will become a performance-limiting factor when two BSs are getting closer and an interference management scheme is mandatory.

To tackle aforementioned non-idealities of PS use cases with different scenarios ranging from 1 to 11, we aim to provide an unified architecture and several enabling approaches. Note that while our initial targeted use cases are related to the PS and military communications requiring emergency or opportunistic deployments, such solution remain applicable to other use cases, e.g., moving vehicular.

B. High level requirements

Based on aforementioned scenarios, an eligible solution shall be capable of dealing with all related non-idealities. Hence, we provide several high-level requirements regarding the system architecture and the required features.

Firstly, each BS should provide service to its local UEs even when it is *isolated*, i.e., not connected to other BSs or to the common CN. This means that each BS shall be an ***autonomous node*** and must at minimum incorporate the following components and functions:

(a) A radio stack to serve local UEs as a BS;
(b) A subset of CN entities to provide policy control, local UEs mobility management, authorization and authentication among the others;
(c) A set of services/applications to allow for the minimum required services (i.e., voice, location, etc.) to be available at served UEs.

Secondly, we aim to efficiently and seamlessly interconnect BSs in order to expand the network and to create an ***autonomous network***. Thus, some additional elements and functions are listed on the top of the autonomous node requirements:

(d) A wireless communication interface to establish inter-BS connections;
(e) Neighboring BSs detection and connection mechanisms to enable network split and merge;
(f) Self-reconfiguration capability to dynamically adapt to network topology;
(g) Interference management schemes to limit interference impact on UEs as well as BSs;
(h) Connections between CNs that are hosted by different BSs to enable seamless inter-CN or inter mobility management entity (MME) handover;
(i) An efficiently inter-BS traffic routing mechanism to route traffic in the network;
(j) Cooperation between services/applications of different network nodes to enable the network-wide service.

To sum up, these requirements are mandatory to build and operate the envisioned ***autonomous network*** that provides the required services with sufficient qualities for PS users.

C. External constraints

On the top of the high-level requirements regarding the required features, the PS authorities as well as other organizations that require similar services will face other constraints when deploying the solution. These constraints are external to the aforementioned scenarios but are essential to enable the solution of **autonomous network**.

Firstly, the constraint on the cost of the solution shall minimize the requirement of capital expenditures (CAPEX) and operating expenses (OPEX). To reduce the CAPEX and OPEX, infrastructure sharing is viewed as a beneficial approach to enable the national PS network deployment [19]. In that sense, a solution that only has limited hardware infrastructure requirement and integrates automatic procedures for their exploitation such as self-configuration, self-organization and self-healing is highly-anticipated.

Secondly, another constraint comes regarding the radio spectrum access and can be severely dimensioning. Due to a high utilization of the available spectrum, nowadays only few frequency bands and narrow bandwidth are available to enable the PS communications. Note such available bands

[1] The performance-enhancing proxy (PEP) in IETF RFC 3135 and RFC 3449 can be adapted to improve performance.

are managed by the state regulation and can be limited and different from country to country. Moreover, the known industrial, scientific and medical (ISM) frequency bands can be utilized freely but suffer from power limitation and high interference. Furthermore, in some use cases such as military operations, new wireless systems shall not disrupt already deployed systems that are still in use which may also limit the available frequency bands even with legally access to several frequency bands.

In summary, a solution shall firstly be designed to cover the use cases in section II-A and fulfill the aforementioned high-level requirement in section II-B. Moreover, its cost shall be minimized in terms of required hardware resource. Lastly, it shall be capable of dealing with heterogeneous frequency band availability and reaches high spectral efficiency.

III. RAT OF AUTONOMOUS NETWORK

Based on the aforementioned scenarios and constraints, we aim to select the applicable RAT in order to deliver the PS service to users. 4G LTE is designed with a number of interesting properties, namely high spectral efficiency, frequency flexibility, large coverage area, native support of variety of IP-based services, and has reached the required maturity level and wide adoption to replace the previous PS communication systems [20]. However, 3GPP has started to address specific PS requirements only from Release 11 due to the growing demand for PS communications [8]. Based on these considerations, we only focus on LTE as the main candidate RAT to provide the service on access link to PS users.

A. *LTE network topologies*

In nominal condition, LTE is used to deploy a nationwide broadband wireless network that relies on fixed eNBs to provide planned coverage to UEs through *Uu* interface as shown in Fig. 1.(1). Such topology can be mapped to the scenario 1 in TABLE II. All these planned eNBs can have full access to the EPC through backhaul links without any interruptions.

Fig. 1.(2) extends the normal one-hop case (i.e., UE ↔ eNB) to the two-hop case utilizing LTE relay node (i.e., UE ↔ relay ↔ eNB) with the new *Un* interface[2] towards eNB. However, relays can be frequency inefficient when handling local communications as the data has to go through the DeNB to reach centralized EPC as shown on Fig. 1.(2). Moreover, these two aforementioned topologies do not handle any mobility of BS or CN and thus cannot cover all scenarios in TABLE II, i.e., no common CN access case (scenario 2 to 6) or mobile BS cases (scenario 7 to 11).

To specifically address some PS scenarios, 3GPP defined the *Isolated E-UTRAN* concept and nomadic eNB as shown in Fig. 1.(3) [9]. It allows specific BSs to host some CN functionalities that can provide minimal services to their served UEs in case of missing or disrupted backhaul connection from BS to the CN. Hence, such topology can be mapped to the scenarios with no common CN connectivity, i.e., scenario 1 to 6 in TABLE II. Nevertheless, it can solely serve local UEs and cannot establish any connection between BSs to form an *autonomous mesh network* as shown in Fig. 1.(4). To realize such topology in Fig. 1.(4), both access links (i.e., UE ↔ eNB) and backhaul links (i.e., eNB ↔ eNB) shall be jointly considered in the design of the architecture.

TABLE III: Comparison of different LTE meshing solutions

BS meshing solution	**Out-band LTE** or **Out-band other RATs**	**In-band LTE**
Backhaul/access frequency bands	Dedicated bands	Shared band
Backhaul/access flexibility	Low	Medium
Scheduling complexity	Medium	Medium to High
Self-backhauling coverage	+ to +++ (depends on RAT)	+++
Hardware cost	+ to +++ (depends on RAT and bands)	+

B. *Backhaul link consideration*

Several RAT solutions can be envisioned for the wireless backhaul links that can realize the mesh to interconnect the mobile BSs. The LTE system can be applied in backhaul links for a higher spectral efficiency; however, its current standard specification and architecture needs to be modified to answer all the considered scenarios shown in TABLE II. While using different RAT seems a straightforward backhauling solution [4], [11], it requires dedicated frequency bands and the associated hardware for backhaul links, and may limit the coverage (e.g., in ISM bands) and network mobility.

To reuse LTE for backhauling, there exist two possible deployments, namely in-band or out-band. Relying on out-band LTE to realize a wireless backhaul is a feasible yet simple solution via using different frequency bands or component carriers between access and backhaul link. However, this approach exhibits the same drawbacks as using different RATs. These observations lead us to consider in-band LTE backhauling as the most promising solution to effectively address all the constraints and requirements at the cost of extra computational complexity and overhead for control and coordination. In addition, it provides the required flexibility to exploit the multiplexing gain by sharing the resources across access and backhaul links (see section IX) as summarized in [21]. To sum up, TABLE III compares all aforementioned backhaul link candidates in several aspects.

IV. ARCHITECTURE OF E2NB

Considering all previous requirements and the selection of LTE RAT for both access and backhaul links, we propose an autonomous BS concept, namely enhanced eNB (e2NB), that covers fixed and moving cell scenarios in order to support new topologies as shown in Fig. 1.(4). Such BS concept relies on LTE to enable UE access and multi-hop self-backhauling capability that allows to create *autonomous mesh network* that enables all scenarios in TABLE II. Moreover, the proposed e2NB is able to utilize a single radio chain, reducing the cost

[2] *Un* interface and relay node is firstly standardized in 3GPP Release 10.

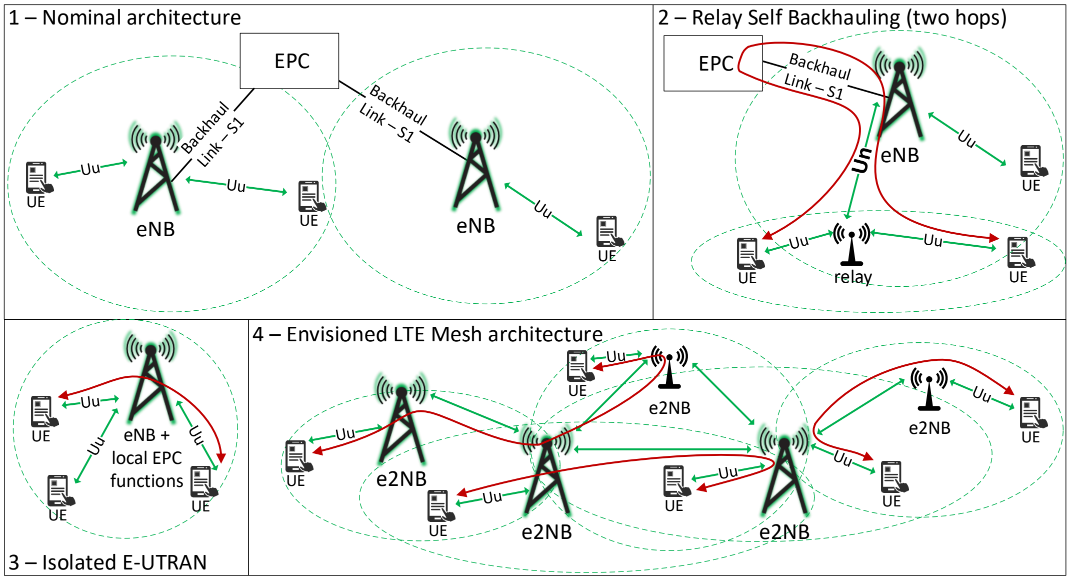


Fig. 1: Topologies based on standard LTE BSs (1,2,3) and the envisioned LTE network (4).

of BSs and allowing to cope with the limited frequency bands availability case. Furthermore, we will show in following sections that it can handle network split and merge mainly due to mobility as well as inter-BS traffic routing.

A. Challenges

Realizing an *autonomous mesh network* based on e2NB with in-band backhauling rises several challenges.

Support of Legacy UEs: Maintaining an in-band mesh network with shared access and backhaul resources while supporting legacy UEs is not straightforward and there is no LTE standard feature to enable the multi-hop wireless network. We have seen in section III-A that the legacy *Uu* interface (cf. Fig. 1.(1)) can only support a single hop and that the *Un* interface can extend to two LTE hops (cf. Fig. 1.(2)). However, there is no clear view on whether such *Un* interface can be used as a complete self-backhauling interface to enable meshing the LTE BSs (see Fig. 1.(4)).

Coordinated scheduling: Because the radio resources are shared between access and backhaul links, scheduling shall guarantee the end-to-end QoS, in particular for real-time traffics [22]. However, due to network mobility, wireless medium characteristics, and multi-hop nature of the traffic flows, legacy scheduling algorithms (i.e. in LTE and ad-hoc mesh network) are not directly and efficiently applicable. Moreover, due to the characteristic of concurrent radio resource accesses from different e2NBs over backhaul and access link, the coordinated scheduling is required to avoid blocking issues. However, a fully centralized scheduling may suffer from scalability issues due to significantly higher overhead and complexity making such approach not applicable.

Autonomous operation: As e2NBs are moving, they can affect the behaviors of other neighboring e2NBs requiring network topology changes and self-reconfiguration as well as interference management. While legacy LTE system is designed with self organization network (SON) capability [23], some extensions are required to continuously detect and resolve configuration conflicts and manage state changes as network nodes are moving.

Security: While LTE already provides specific security features such as authentication and ciphering, some adaptations is needed to maintain the end-to-end security in the time-varying network topology (both e2NB and UEs). Additional considerations have to be taken when inter-networking (e.g., network of e2NBs or eNBs belonging to different authorities) is required.

Service continuity: Maintaining user services and applications when possible as network nodes are moving is very challenging and require tight interactions among different components including topology management, routing, and applications. In addition, some specific service discovery and registry are necessary for the deployed services and applications at the e2NB.

B. Architecture

Fig. 2 presents the proposed e2NB stack, designed to fully satisfy the requirements delineated in section II-B to enable an ***autonomous LTE mesh network***. In corresponding to the requirements (a), (b) and (c) of section II-B, the e2NB shall integrate:

- Legacy LTE eNB protocol stack;
- Transmitter (TX) / Receiver (RX) radio chain;
- LTE MME (Mobility Management Entity);
- LTE HSS (Home Subscriber Server);
- LTE S-GW (Serving Gateway);
- LTE P-GW (Packet Data Network (PDN) Gateway);
- Set of applications and services (voice, data, video, etc.).

Such entities in e2NB allow to obtain an ***autonomous node*** that can serve UEs locally and provide required services. Note that S-GW and P-GW are only required if inter-networking with legacy eNB is required and can be safely omitted from the proposed architecture without any UE service interruption

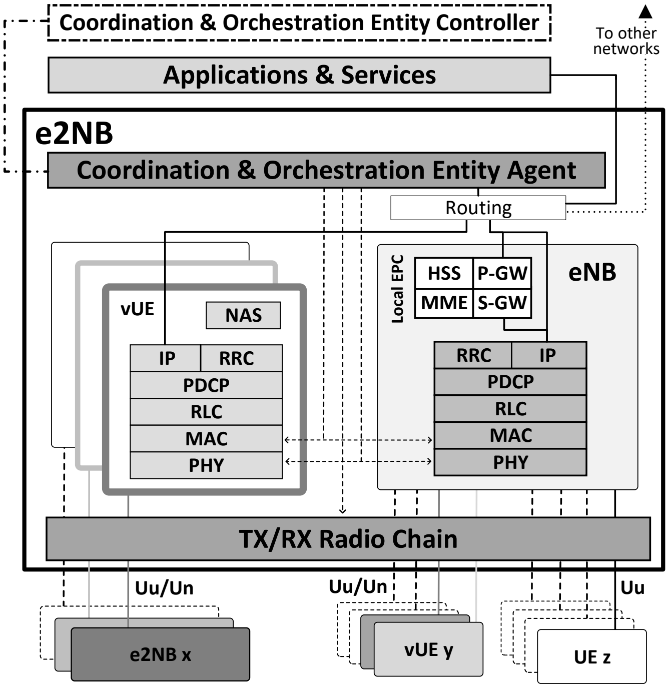


Fig. 2: e2NB stack.

(i.e., X2-based handover, embedded routing module for packet marking and forwarding). In addition, some remaining LTE entities like PCRF (Policy and Charging Rules Function)[3] can also be included in the e2NB if required.

In corresponding to the requirements from (d) to (h) in section II-B to establish a ***autonomous network***, some additional elements are integrated in the e2NB:

- UE stacks as a service, denoted as virtual UEs (vUEs);
- Routing service;
- Coordination and Orchestration Entity (COE) agent.

These included vUEs are intuitively utilized to establish connections with neighboring e2NBs. The routing service marks and forwards traffic between e2NBs while bypassing legacy S/P-GW entity based on the policy rules enforced by COE. The COE agent acts as a local controller in charge of managing routes, network topology in accordance with other COE agents for optimization purposes, and connectivity to coordinate inter-e2NB connections. In the following, we describe the role of each component in a single e2NB.

1) eNB: It provides the same operations as in a legacy LTE eNB in that it communicates with UEs through the legacy *Uu* interface and with MME and optionally S-GW through the legacy *S1* interface [24].

2) MME and HSS: These entities allow the autonomous node functionality at each e2NB. The MME is the key control node for the LTE access network and is connected through *S6a* interface to the HSS. The HSS is responsible for authenticating and authorizing user access via forming the database that contains user subscription and authentication context.

3) S-GW and P-GW: The S-GW and P-GW allows terminating UE data plane bearers as in a legacy LTE network. While they are used to ensure inter-networking of legacy UEs, they are bypassed for communications with vUEs to reduce the latency over the mesh network.

4) vUEs: These vUEs are utilized to establish the inter-e2NB communications. Each single vUE is managed and instructed by the COE and can report real-time radio information (e.g., received signal power, newly detected eNB information) to the COE. Furthermore, each vUE will include the entire protocol stack of a legacy UE required to detect and establish communication with an eNB.

5) TX/Rx radio chain: As the interface towards other network entities (i.e., eNB/vUE of other e2NBs, UE), such radio chain is shared between embedded eNB and vUEs under the control of COE. To cover all the scenarios including the worst case with a single frequency band, the e2NB shall operate with only one radio chain to provide services to UEs through *Uu* and vUEs through *Un* as will be explained in section V. Nevertheless, such constraints can be relaxed for certain type of deployment scenarios to provide the required flexibility in switching Tx/Rx radio chains among eNB/vUEs. Example scenarios include carrier aggregation (CA) techniques or band separation between access and backhaul links in frequency domain.

6) Routing: Such service enables routing and data forwarding both intra-e2NB as well as inter-e2NBs. It is able to transmit and receive packets directly from eNB (e.g., vUE traffic) or from P-GW (e.g., legacy UE traffic). Contrary to the legacy eNBs, an e2NB can act as the an IP service end point (e.g., gateway) and have external interfaces toward other networks. Lastly, the routing path is selected according to the rules provided by the COE over multiple hops within the mesh network.

7) Applications and services: Each e2NB is not only providing services to its local UEs but also to remote UEs (and potentially to other eNBs, e.g., EPC as a service). Furthermore, it relies on the routing service for discovery and cooperation in order to enable network wide UE services. Thus in the envisioned architecture, the e2NB becomes a true service provider by publishing the offered services as well as a service consumer by subscribing to the service of other e2NBs.

8) COE agent: Following the software-define networking (SDN) principles, COE agent is a local controller responsible for self-reconfiguration and self-reorganization of the underlying e2NB as follows:

- Monitor the e2NB connectivity (via embedded vUEs and eNB) and network topology;
- Determine the IP addressing space over the mesh network in cooperation with other e2NBs COE agent;
- Manage the entire life-cycle of each vUE from the initialization (e.g., International Mobile Subscriber Identity (IMEI), International Mobile Subscriber Identity (IMSI) and cryptographic functions), (re-)configuration , runtime management and disposal;
- Control the radio chain access and configure the corresponding network layers of embedded eNB and vUEs;
- Coordinate with the centralized COE and other COE agents to enable the cross-layer control and management among e2NBs.

[3] Some related services and applications can also be included such as IP Multimedia Subsystem (IMS).

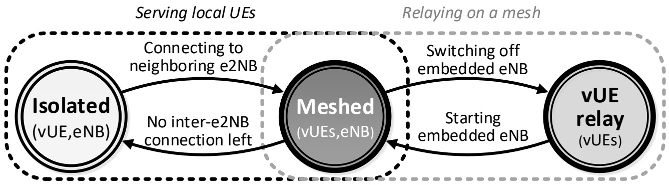


Fig. 3: Main e2NB states.

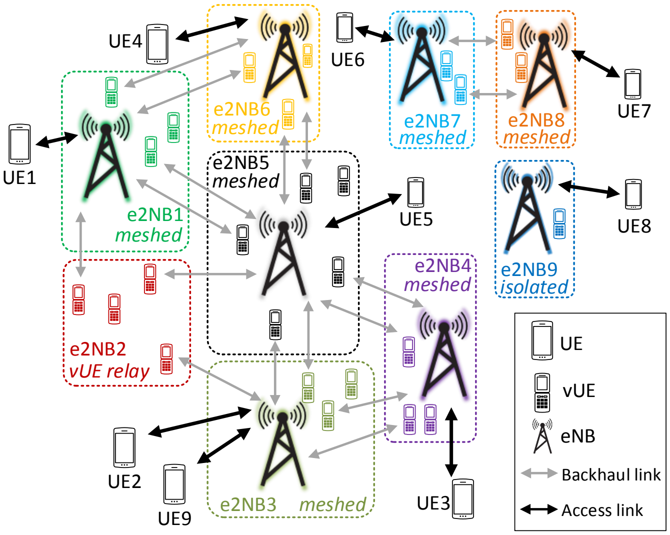


Fig. 4: Example of different mesh networks.

C. Node states and resulted network topology

Under the proposed e2NB architecture, a node can transit between states as shown in Fig. 3. In all states, the e2NB relies on a vUE to periodically scan the network and detect new e2NBs. In the *Isolated* state, the e2NB is not connected to any other e2NBs and provides local service to its served UEs. In the *Meshed* state, the e2NB is connected to at least one neighboring e2NB allowing it to extend the network, give access to its services and gateway connectivity (if any) to the rest of the mesh network, while providing network access to the UEs under its coverage through its eNB stack. Lastly, in the *vUE relay* state, the e2NB is connected to at least two other e2NBs and acts only as a relay between other e2NBs, potentially shares its services and gateway access. However, it does not use its embedded eNB to serve UEs but can keep active the eNB *Un* support. This state is used when e2NBs are close enough to each other such that the UE access can be handled without using all eNB stacks.

A network example is shown with three different states in Fig. 4. Firstly, a mesh network with six e2NBs (e2NB1 to e2NB6) of which five are in the *Meshed* state and serve UEs while one (e2NB2) is in the *vUE relay* state. Connections between *Meshed* state e2NBs have both downlink (DL) and uplink (UL) directions relying on two corresponding vUEs, while connections between *Meshed* e2NB to the *vUE relay* e2NB only rely on the vUE at the *vUE relay* e2NB, i.e., no vUE at the *Meshed* e2NB. Then, e2NB7 and e2NB8 left from the main mesh network but both are still in *Meshed* state as they are still connected together. Finally, e2NB9 is in *Isolated* state as it is not connected to any other e2NB. Last but not least, we can notice that there are one extra instantiated vUE at each e2NB, for instance, 4 vUEs at e2NB2, in order to detect some other neighboring e(2)NBs due to node mobility.

V. Design Elements and Procedures of e2NB

In this section, we detail the design elements and procedures of the proposed e2NB architecture to enable the autonomous network. We firstly tackle several physical layer aspects for the inter-e2NB connectivity. Then, we present the operation flow and connection related procedures of the e2NB, and the associated specific LTE parameters that enable the autonomy, mobility and meshing of the BSs while retaining legacy UE connectivity. Finally, we expand on the CN procedures required to efficiently provide services.

A. Physical layer interfaces

As mentioned before, the in-band deployment comes with highly-anticipated characteristics at the cost of extra design issues that need to be carefully tackled in the physical layer. Hence, we will give some LTE background information and then elaborate the two key *Uu* and *Un* interfaces.

1) Background LTE information: Originally, an eNB is composed of transmitter and receiver chain to communicate with UEs using *Uu* interface in downlink (DL) and uplink (UL) directions, respectively. There are two duplexing modes that can be utilized for DL and UL directions:

- **Frequency Division Duplex (FDD) mode**: A paired frequency band is divided equally to be utilized for DL and UL directions separately.
- **Time Division Duplex (TDD) mode**: The same frequency band is shared between DL and UL direction whereas disjoint time durations are used for different directions. Such time-domain resource partition is based on some pre-defined TDD UL/DL configurations.

Then, we explain the basic resource element unit of LTE air interface. Firstly, an LTE frame lasts for 10ms and a frame comprises 10 subframes (SFs) each with 1ms duration that number from SF 0 to SF 9. Within one SF, we can furthermore decompose the available resource in both time and frequency aspects. In time aspect, there are two slots in each SF and each slot contains several symbols, namely there are fourteen symbols in a SF numbering from symbol 0 to symbol 14 in the normal cyclic prefix and twelve symbols in the extended cyclic prefix case. In frequency aspect, each SF is made up of several subcarriers depending on the frequency bandwidth. Hence, we can come up with the smallest discrete element in LTE with 1 subcarrier and 1 symbol as the resource element (RE). Furthermore, the basic unit that can be allocated to a user is termed as the resource block (RB) with 1 slot in time domain and 12 subcarriers in frequency domain.

2) Uu interface: To fully enable the *Uu* interface, BSs have to broadcast a number of mandatory control messages and synchronization signals to UEs. Firstly, the Primary and Secondary Synchronization Signals (PSS and SSS) must be transmitted in the SF 0 and SF 5 (in FDD) to allow the initial synchronization of UEs. Then, the master block information (MIB) and a number of system information blocks (SIBs) will then be utilized by UEs to retrieve the common system information of the eNB. Furthermore, each eNB transmit the first several symbols (ranges from 1 to 3) in each SF for the control channels, namely Physical Downlink Control Channel (PDCCH), Physical Control Format Indicator Channel (PCFICH), and Physical Hybrid-ARQ Indicator Channel (PHICH). Note the downlink control information (DCI) is contained in the PDCCH with the essential control information of DL and UL resource allocation for each UE. Finally, UEs will use Cell Reference Signals (CRS) for synchronization and channel estimation[4] in order to receive the data channel, namely the Physical Downlink Shared Channel (PDSCH). Note the CRS will take place on specific REs depending on the eNB configuration.

These mandatory transmissions prevent the radio chain to be used for both transmission and reception over *Uu* interface in a time division manner hence prohibits the possibility of in-band *Uu* deployment as we mentioned in section IV. Indeed, it would require an eNB to be able to transmit and the co-located vUEs to receive at the same time which is not possible without full duplex radios. To sum up, the *Uu* interface is not suitable for *in-band* inter-e2NB communication while maintaining *Uu* support for legacy UEs.

3) Un relay interface: As we described in section III-A, 3GPP introduced the *Un* relay interface in Release 10 during its work on LTE relay for network capacity and coverage expansion. It allows to deploy some fixed relay nodes (RNs) in an in-band manner to extend the coverage of standard BSs through one extra hop (cf. Fig. 1.(3)). Each in-band RN needs to receive from its Donor eNB (DeNB) [25], through the *Un* interface and then relay to the UEs through the *Uu* interface using the same frequency band. Such in-band characteristic requires a time division multiplexing (TDM) on the time-domain SFs between *Un* and *Uu* interfaces. To enable such TDM manner while fully support *Uu*, several works aim to define new frame structure [21], [26] for the relay interface. However, these new structures cannot be directly enabled through the LTE system and will request a drastic change that violates aforementioned external constraints introduced in section II-C. In contrast, the *Un* interface can utilize a mechanism that is introduced in LTE eMBMS (enhanced Multimedia Broadcast Multicast Service) that divides SFs into Multicast-Broadcast Single-Frequency Network (MBSFN) SFs,[5] and non-MBSFN SFs. During MBSFN SFs, legacy UEs are expecting the reference signals *only* in the first symbol of a SF (for PCFICH/PDCCH/PHICH decoding). Via utilizing such characteristic, a RN can switch between TX (to served UEs) and RX (from the DeNB) over MBSFN SFs while following the *Uu* interface specification on its access links.

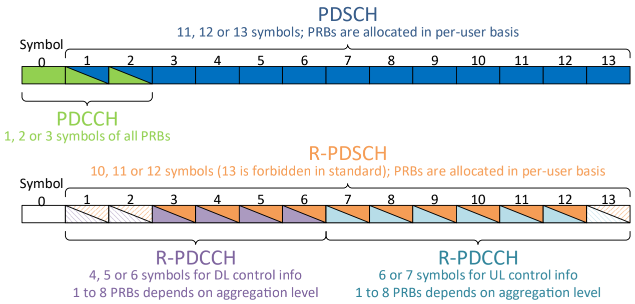

PDSCH
11, 12 or 13 symbols; PRBs are allocated in per-user basis
Symbol
0
1
2
3
4
5
6
7
8
9
10
11
12
13
PDCCH
1, 2 or 3 symbols of all PRBs
R-PDSCH
10, 11 or 12 symbols (13 is forbidden in standard); PRBs are allocated in per-user basis
Symbol
0
1
2
3
4
5
6
7
8
9
10
11
12
13
R-PDCCH
4, 5 or 6 symbols for DL control info
1 to 8 PRBs depends on aggregation level
R-PDCCH
6 or 7 symbols for UL control info
1 to 8 PRBs depends on aggregation level

Fig. 5: Symbol allocation in a SF for *Uu* and *Un* interfaces.

Furthermore, the *Un* interface relies on two new physical channels: relay PDCCH (R-PDCCH) for control information delivery and relay PDSCH (R-PDSCH) for data transportation. Such R-PDCCH can deliver downlink scheduling information (i.e., DeNB to RN) and uplink grant (i.e., for RN to DeNB data transportation) using the same DCI formats as legacy PDCCH. Both R-PDCCH and R-PDSCH (to a specific relay) have the same starting position and ending position in a SF and they span 4 to 6 symbols within the first slot in a SF (i.e., from symbol 0 to symbol 6) and 6 or 7 symbols within the second second in a SF (i.e., from symbol 7 to symbol 13) as shown in Fig. 5. Such flexible symbol duration depends on the higher-layer parameters like *DL_StartSymbol* (DLSS) and end symbol index (ESI) in order to cope with the transmission of the legacy control channels of UEs (PCFICH, PHICH, PDCCH), TX/RX switching time or some nonidealities (e.g., propagation delay, synchronization offset, etc.). In following, we elaborate on more details about the differences between *Un* and *Uu* interface.

a) Relay downlink: Both R-PDCCH and PDCCH format the DCI in the same way, but they are mapped to different time-domain symbols (cf. Fig. 5). PDCCH is mapped over the first symbols of a SF while R-PDCCH takes place into the RBs where PDSCH is usually carried. There are two types of DCIs with different R-PDCCH mapping (see Fig. 5 and 6): (1) DCI format 0, that corresponds to UL allocations, are mapped in the second slot of a SF, and (2) all the other DCIs for DL allocation are mapped on the first slot of a SF. As for the R-PDSCH, one major difference between the PDSCH is in the available symbols within a SF for transpiration (cf. Fig. 5). Since a fewer number of symbols (i.e., down to 11 symbols in Fig. 5) can be used for R-PDSCH transportation, the channel coding rate of R-PDSCH will be increased correspondingly. To summarize, Fig. 6 provides an example of resource allocation in DL for both DeNB and RN[6] of a subframe with 14 symbols. We can see that DeNB needs to allocate resource to transmit to 3 UEs (i.e., UE1, UE2, UE3) through PDCCH/PDSCH and 2 RNs (i.e., RN1, RN2) via R-PDSCH/R-PDCCH in Fig. 6(a). Whereas both transmission (to UEs) and reception (from DeNB) are required at RN2 as depicted in Fig. 6(b) to ensure legacy *Uu* support.

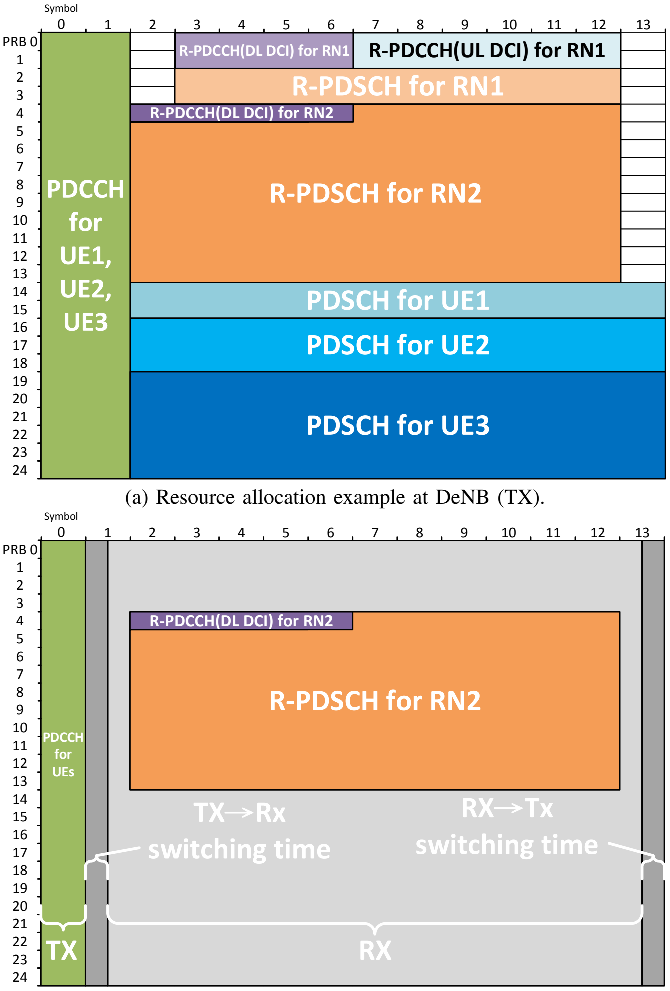
b) Relay uplink: The relay uplink direction of *Un* interface is almost the same as the one of *Uu* interface with following exceptions. Firstly, the last symbol of each SF is reserved for RX/TX switching time hence without any transportation. Secondly, there is no acknowledgment/non-acknowledgement (ACK/NACK) signal to be transported from the DeNB to RN to indicate a successful reception or not[7]. Hence, the RN can only depend on the new data indicator (NDI) information in

[4] Based on the transmission mode, the Demodulation Reference Signal(DMRS) will be used.

[5] In each frame, up to 6 MBSFN SFs are allowed in FDD mode whereas up to 5 MBSFN SFs for TDD mode depending on the TDD UL/DL configuration.

[6] PCFICH/PHICH are omitted for simplicity.

[7] In comparison, PHICH is used for such purpose in *Uu* interface.



(a) Resource allocation example at DeNB (TX).

(b) Resource allocation example at RN2 (TX/RX).

Fig. 6: Resource allocation example of DL at DeNB/RN.

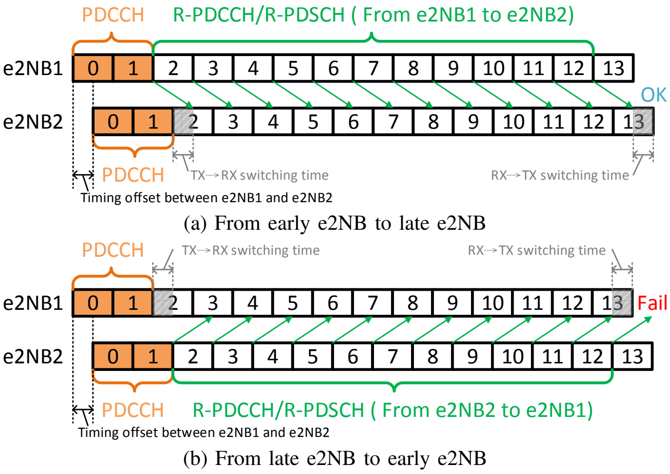
the DCI of R-PDCCH to decide whether a re-transmission is needed or not. Lastly, both TX and RX operations are required for a RN on the uplink channels (PUCCH, PUSCH) to relay the uplink transportation from UEs to its DeNB.

To summarize, special cares shall be taken to enable the *Un* interface. In FDD mode, RN needs to be able to receive and transmit on both DL and UL directions whereas legacy BS only do transmission (on DL) and reception (on UL) in each direction. Within TDD mode, a faster switching than the one of legacy TDD mode is required as there are two switching time between TX and RX in a SF (cf. Fig. 6(b)).

B. Physical layer design issues

We have seen that a combination of *Uu* and *Un* can allow a RN to use one radio chain for both backhaul (to the DeNB) and access (to the UEs) link. Hence, we advocate that the *Un* interface can be leveraged for the in-band inter-e2NB communications to mesh the e2NBs. However, there are still some considerations to be resolved in the physical layer as elaborated in following.

1) Synchronization: Within the backhaul links, synchronization issue has already been studied to minimize disturbance between BSs and to be compliance with the frequency accuracy requirements [27]. Hence, both frequency and time synchronization of e2NBs in the mesh network are crucial for the use of the *Un* interface.



(a) From early e2NB to late e2NB

PDCCH
TX→RX switching time
RX→TX switching time
e2NB1
e2NB2
Fail
PDCCH
R-PDCCH/R-PDSCH (From e2NB2 to e2NB1)
Timing offset between e2NB1 and e2NB2

(b) From late e2NB to early e2NB

Fig. 7: Non-synchronized e2NBs transportation.

a) Time synchronization: Adjacent e2NBs need to be time synchronized; otherwise, some transported physical channels of *Un* interface cannot be properly handled. A simple example with two e2NBs is shown in Fig. 7 without time synchronization where some durations are reserved to TX/RX switching. We can observe that R-PDCCH/R-PDSCH from e2NB1 to e2NB2 of Fig. 7(a) can occupy 11 symbols (i.e., symbol 2 to 12); however, it is not possible to transport from e2NB2 to e2NB1 in Fig. 7(b) since the last symbol (i.e., symbol 12) will overlap with the TX/RX switching duration at e2NB1. Such case violates the possible 11 to 13 symbol duration of R-PDCCH/R-PDSCH as shown in Fig. 5. In that sense, it is not feasible to have a *Un* interface in both directions with a propagation time in the order of one OFDM symbol duration (around 66.67 micro-seconds) that corresponds to around 20km physical distance[8]. Last but not least, 3GPP specified the minimum requirement for TDD time synchronization to be up to 10 (large cell) or 3 (small cell) micro-seconds in TS 36.133.

b) Frequency synchronization: As the in-band characteristic, each e2NB shall transmit at the same carrier frequency of both *Un* and *Uu* interfaces. Hence, frequency synchronization is mandatory to build up a self-backhauling network and avoid extra inter-carrier interference due to the carrier frequency offset (CFO) which is regulated to be within 0.05 ppm to 0.25 ppm by 3GPP (TS 36.104).

To address both synchronization issues, a Global positioning system Disciplined Oscillator (GPSDO) can be utilized as a clock reference for the local oscillator at each e2NB to guarantee extremely high frequency (sub-ppb level) and time (<20 ns) synchronization [28]. However, when the GPS signal is not available (e.g., tunnel), the Rubidium oscillators can provide holdover capability to steer synchronization [29].

2) Range limitation: The time synchronization between e2NBs will limit the maximum distance between e2NBs to have a backhaul link as current standard only allows *Un*

[8] Such spacing can be realistic for high powered maritime and PS use cases.

interface to strip out up to the last symbol as shown in Fig. 5 in order to propagate the *Un* interface channels and transition to the *Uu* interface. In specific, both R-PDCCH and R-PDSCH can either finish at symbol 12 or symbol 13 within a SF. Finishing at symbol 13 makes an e2NB impossible receive all symbols when using perfect time synchronization, while finishing at symbol 12 allows for a theoretical maximum 21.4 kilometer distance between e2NBs without considering the RX→TX switching time at the receiving e2NB. This range is too short for some high power and long range use cases such as naval communications and it calls for modifications in the usable symbol range to further increase the maximum reachable distance. Possible values of DLSS and ESI can be extended to allow using less than 10 symbols for the transmission of R-PDCCH and R-PDSCH.

However, any changes to use less than 10 symbols will impact the available bits to carry the transport block size (TBS) that defines the number of information bits to be transported by the physical layer. Such TBS computation is originally the same for *Uu* and *Un* interface as stated before[9]. The TBS values defined in [30] do not depend on the number of available symbols for PDSCH/R-PDSCH transportation, but the number of bits that can be carried by available REs shall be larger than the value of TBS to guarantee a possibly successful reception, i.e., coding rate shall be less than 1, at the first transmission. Take the case of single antenna transmission mode 1 (TM1) as an example, the maximum value of TBS is 18336 bits shown in [30] when using the largest modulation and coding scheme (MCS) 28 with 25 PRBs (i.e., N_{PRB} = 25). On the other hand, the number of bits that can be carried by available REs can be formulated as:

$$N_{bits} = Q_m * (N_{subcarriers} * N_{symbols} - N_{CRS}) * N_{PRB} \quad (1)$$

where $N_{subcarriers}$ is the number of subcarriers in a PRB which is equal to 12, Q_m is the modulation order, i.e., the number of bits per RE, which equals to 6 when MCS is 28, N_{CRS} is the number of REs used by CRS in the data region within a PRB which equals to 6 in TM 1, $N_{symbols}$ is the number of symbols for PDSCH/R-PDSCH transmission. With $N_{symbols}$ = 11, N_{bits} = 18900 > 18336 in the considered case (MCS 28, 25 PRBs), the coding rate will be 0.97. However, when $N_{symbols}$ becomes 10, then N_{bits} = 17100 < 18336, and the coding rate will be 1.07.

To this end, reducing the number of symbols of R-PDSCH (i.e., $N_{symbols}$) will further prohibit some more combinations of MCS/PRB or either calls for a new TBS table to be defined in order to enable possibly successful decoding at the first transmissions. Both methods are feasible, but the second approach can take the extra advantage to allow for a wider range of code rates.

3) Uneven UL and DL SFs in TDD mode: As described in section V-A3, UL SFs of the *Un* interface for TDD mode are determined through a specific parameters, *SubframeConfigurationTDD* in [31]. Unlike the 1-to-1 mapping between the DL SFs and UL SFs for FDD mode, such ratio can be larger than one for TDD mode, i.e., a single UL SF for the UL transmission from RN to the DeNB can correspond to several DL SFs. For instance, there are one UL SFs (SF 3) and four DL SFs (SF 4, 7, 8, 9) within one frame when *SubframeConfigurationTDD* is 17 [31]. While this is not a problem for a legacy LTE network with RNs, it now becomes one main problem of the proposed e2NB network architecture.

In contrast to a classical LTE RN that is connected only to one DeNB, an e2NB can be connected to several other e2NBs. As it will be presented in section VI, the DL SF allocation of *Un* interface can be done in a dynamic way for efficient utilization in the mesh backhaul. Such dynamics highlight that an e2NB can receive from several different e2NBs over different MBSFN SFs within a single frame. For instance, in *SubframeConfigurationTDD* 17, an e2NB can receive from up to four other e2NBs during a frame as there are four DL available SFs; however, it only has one UL SFs to access. Hence, the PUSCH allocation of such e2NB toward different e2NBs will overlap. To avoid such overlapping, some allocation strategies can be utilized by the *COE controller* and will be discussed in chapter VI.

4) HARQ modifications for Un interface: As the adoption of *Un* interface for the multi-hop in-band backhauling, the mechanism of hybrid automatic repeat request (HARQ) shall be examined correspondingly.

a) FDD mode: The HARQ mechanism in DL direction of *Un* interface is unchanged in FDD mode and there are up to 6 HARQ process in support of at most 6 MBSFN SFs in a frame to enable the retransmission approach. As for UL direction, since there is no PHICH-like channel in *Un* interface to transport the ACK/NACK information; hence, the *Un* interface shall uses the NDI in UL DCI of R-PDCCH to decide if a retransmission is necessary or not relying on a fixed loop around HARQ processes over SFs. However, as it will be presented in section VI, the SF allocation of *Un* interface can be done in a more dynamic way for efficient resource utilization in the mesh backhaul, and the corresponding HARQ process will not follow aforementioned fixed loop over SFs. To tackle with such dynamic HARQ process scenario, we propose to add the *HARQ process* information in UL DCI 0 of R-PDCCH to identify which HARQ process shall be re-transmitted.

b) TDD mode: In the TDD mode, HARQ feedback for DL transmissions is also problematic as the UL PUSCH collisions described in section V-B3 as PUCCH will also collide and prohibit feedback information delivery. To handle such problem, we propose the following mechanism assuming that the *COE controller* is handling which e2NB can use the UL SF allocation as mentioned in section V-B3. Firstly, if the following UL SF is available to deliver the UL feedback on the *Un* interface to the considered e2NB, then the legacy HARQ mechanism is applied. Otherwise, the ACK/NACK feedback of such e2NB will be delayed until the next DL SF allocation to this e2NB. To this end, such delayed ACK/NACK feedback with the channel quality indicator (CQI) report will be transmitted over the message taking place at the UL DCI of the R-PDCCH (cf. Fig. 5). Given the small size of ACK/NACK messages, several ACKs/NACKs can be multiplexed if several DL transmissions have been received from a specific e2NB

[9] Except the case with 1.4MHz radio bandwidth.

since the last DL transmission opportunity. Unfortunately, such method requires large buffers at the e2NB DL queues and might increase latency of transmissions depending on the SF allocation for inter-e2NB transmissions. To cope with such side effect, inter-e2NB transmissions can use more conservative MCS index (e.g., lower modulation order, smaller coding rate) to reduce the probability of retransmission at the cost of lower spectral efficiency.

On the other hand, the HARQ process for DL direction can follow the same mechanism as the one used in FDD with the re-transmission indication relying on the NDI value within the UL DCI.

C. *e2NB procedure and parameters configuration*

In this subsection, we first list the eNB parameters that require specific configuration to enable the backhaul link while supporting legacy UEs at the meantime. Then, we introduce the attach procedure as used by vUEs to establish the connection of inter-e2NB link. Finally, we present the e2NB operation flows that comprises the startup phase, and the three aforementioned major states in Fig. 3.

1) eNB parameters: As the legacy eNB, the eNB stack of e2NB relies on an extensive set of parameters regarding all the spanned network layers. Some of these parameters require specific configuration to allow the e2NB meshing capability. We can separate such parameters into two different groups: (a) parameters that can be configured depending on configuration of adjacent eNBs, and (b) parameters that require specific values to allow the inter-e2NB communication through the *Un* interface while supporting the *Uu* interface to serve local UEs. In the group (a), we find the physical cell identity (PCI), tracking area identity (TAI), radio resource configuration information like random access channel configuration and MBSFN information in SIB2[10], and all parameters related to (further) enhanced inter cell interference coordination (eICIC/FeICIC). Whereas in the group (b), we find parameters related to neighboring cell information in other SIBs, and more generally to the initial connection setup as well as parameters related to uplink scheduling requests.

While some parameters can be updated when the eNB is serving local UEs, others cannot and must be configured before starting the eNB stack otherwise leading to local UEs being disconnected for the update. This is especially the case for the PCI that can be obtained by UEs when receiving the PSS and SSS from eNB. Furthermore, it will impact the generation of a pseudo-random sequence used to scramble specific LTE channels (e.g., PDSCH, PDCCH, etc.) and it defines the position of the REs for CRS. Note the CRS is crucial and used by UEs for synchronization and channel estimation. While the PCI can have 504 different values, the CRS have only 6 different mapping positions. This means that two different PCI values with PCI_1 and PCI_2 such that $PCI_1 \equiv PCI_2 \mod 6$ lead to the same CRS positions. Such same CRS positions will make difficulties for UEs to do any channel estimation and cell detection. Thus, PCI values should be configured carefully among adjacent eNBs.

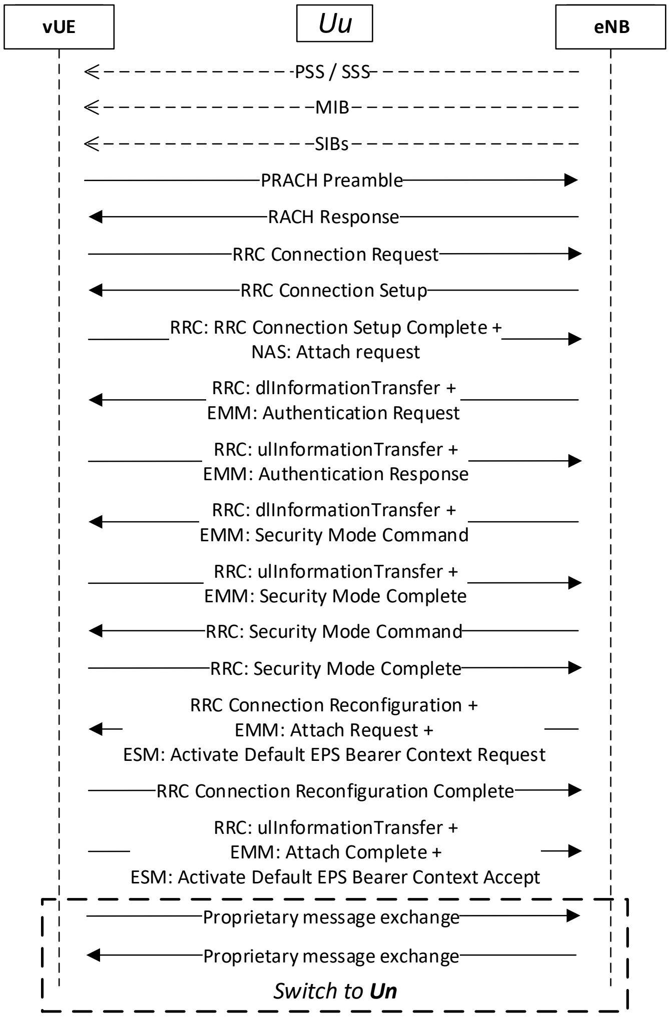

vUE
Uu
eNB
PSS / SSS
MIB
SIBs
PRACH Preamble
RACH Response
RRC Connection Request
RRC Connection Setup
RRC: RRC Connection Setup Complete + NAS: Attach request
RRC: dlInformationTransfer + EMM: Authentication Request
RRC: ulInformationTransfer + EMM: Authentication Response
RRC: dlInformationTransfer + EMM: Security Mode Command
RRC: ulInformationTransfer + EMM: Security Mode Complete
RRC: Security Mode Command
RRC: Security Mode Complete
RRC Connection Reconfiguration + EMM: Attach Request + ESM: Activate Default EPS Bearer Context Request
RRC Connection Reconfiguration Complete
RRC: ulInformationTransfer + EMM: Attach Complete + ESM: Activate Default EPS Bearer Context Accept
Proprietary message exchange
Proprietary message exchange
Switch to Un

Fig. 8: Connection procedure.

2) vUE attach procedure: As introduced beforehand, we utilize both *Uu* and *Un* interfaces for the inter-e2NB backhaul link. Hence, the attach procedure is provided in Fig. 8 and we detail the overall procedure in this subsection. To enable such procedure, we rely on some specific parameter configurations that allows e2NB to connect to new e2NB using its vUEs without disconnecting locally served UEs.

Firstly, a problem comes out as the embedded eNB and vUEs are sharing the radio interface which implies that the vUE cannot naturally receive and detect the PSS/SSS broadcasted by neighboring eNB. Thanks to the time and frequency synchronization previously described, it is simple for a vUE to detect PSS and SSS from adjacent BSs via only listening to some small time duration close to the SF 0 and SF 5 in a frame. Nevertheless, the co-located eNB will need to blank some eNB transmissions to enable such detection and thus impacts the local UE in terms of detection, synchronization and failure in radio link. To better depict such impact, we can first see that the vUE does not need to continuously detect neighboring e2NBs but only periodically to check whether there is a new neighboring eNB. Furthermore, the corresponding parameter in SIB can be configured to make the link more prone to be

[10] And the related prerequisite information to receive SIB2 like MIB, SIB1.

in-sync state. For instance, the eNB can blank SF(s) or even frame(s) without making its served UEs be out-of-synch state if *N310*, *T310* and *N311* in the SIB2 are configured adequately larger. Hence, a controlled blanking (sub-)frames is feasible to allow vUE to detect PSS/SSS of neighboring e2NBs and get their PCIs.

After receiving a new PCI, the vUE will start to receive MIB and SIBs to confirm the PLMN identity as well as the E-UTRAN Cell Identifier (ECI) to uniquely identify the e2NB during the configuration procedure using a predefined hash function. While MIB and SIB1 have fixed position in frames, other SIB positions in time are defined by each eNB. To optimize such reception of SIBs at vUE, the eNBs shall configure the position of SIBs[11] to be always in non-MBSFN SFs, for instance in SF 4 or SF 9 for FDD mode, to limit the vUE access time to the radio chain. Furthermore, the COE agent should properly allocate non-MBSFN SFs to the vUE for SIB decoding.

Based on the e2NB identity, the COE agent can determine from its topology knowledge if such detected e2NB is already part of the mesh to decide whether to establish the connection to such neighboring e2NB.

Then, the random access process will be initiated by vUE to transport the physical random access channel (PRACH) preamble. Note the PRACH configuration index in SIB2 should be set such that the possible SFs for a UE/vUE to transmit the *RACH preamble* duration and position do not overlap with MBSFN SFs as the eNB might not be listening to the UL channel due to the backhaul transportations. Once the eNB successfully receives the RACH preamble, it can transmit the RACH response to vUE in a predefined window size between 2 and 10 ms long. However, the vUE should not continuously listen to a such long duration to limit perturbation of served UE; hence, we can set the eNB to always transmit the response after a fixed amount of time over a non-MBSFN SF and to allocate UL SFs for this procedure over non-MBSFN SFs.

The vUE then follows the legacy attach procedure used by UEs [32] to establish the connection until reaching the RRC Reconfiguration Complete step in Fig. 8. It is noted that similarly to the random access message exchange, the eNB should be configured to transmit messages in a pre-configured timely manner over non-MBSFN SFs. Then, the *Un* interface will be configured and used after finishing the attach complete and activating default EPS bearer procedure in order to exchange between vUE and eNB. Note the RRC connection shall be maintained during message exchange in *Un* interface and it will be released when such connection cannot be maintained, for instance, radio link failure case.

3) e2NB operation flow: In this paragraph, we detail on the operation flows among the startup phase and the running phase that comprises three main states (i.e., Isolated, Meshed and vUE relay) in Fig. 9. Before introducing the operation flow, it is noted that at least one instantiated vUE is required to detect adjacent e(2)NBs.

a) Startup phase: Such startup phase only happens when we just (re-)start the e2NB. It is used to configure eNB parameters before starting the eNB protocol stack with the operation flow in Fig. 9(a). In the legacy LTE system, the initial configuration phase includes choosing the PCI value through the *self-configuration* processes relying on the common CN [23], [33], [34].

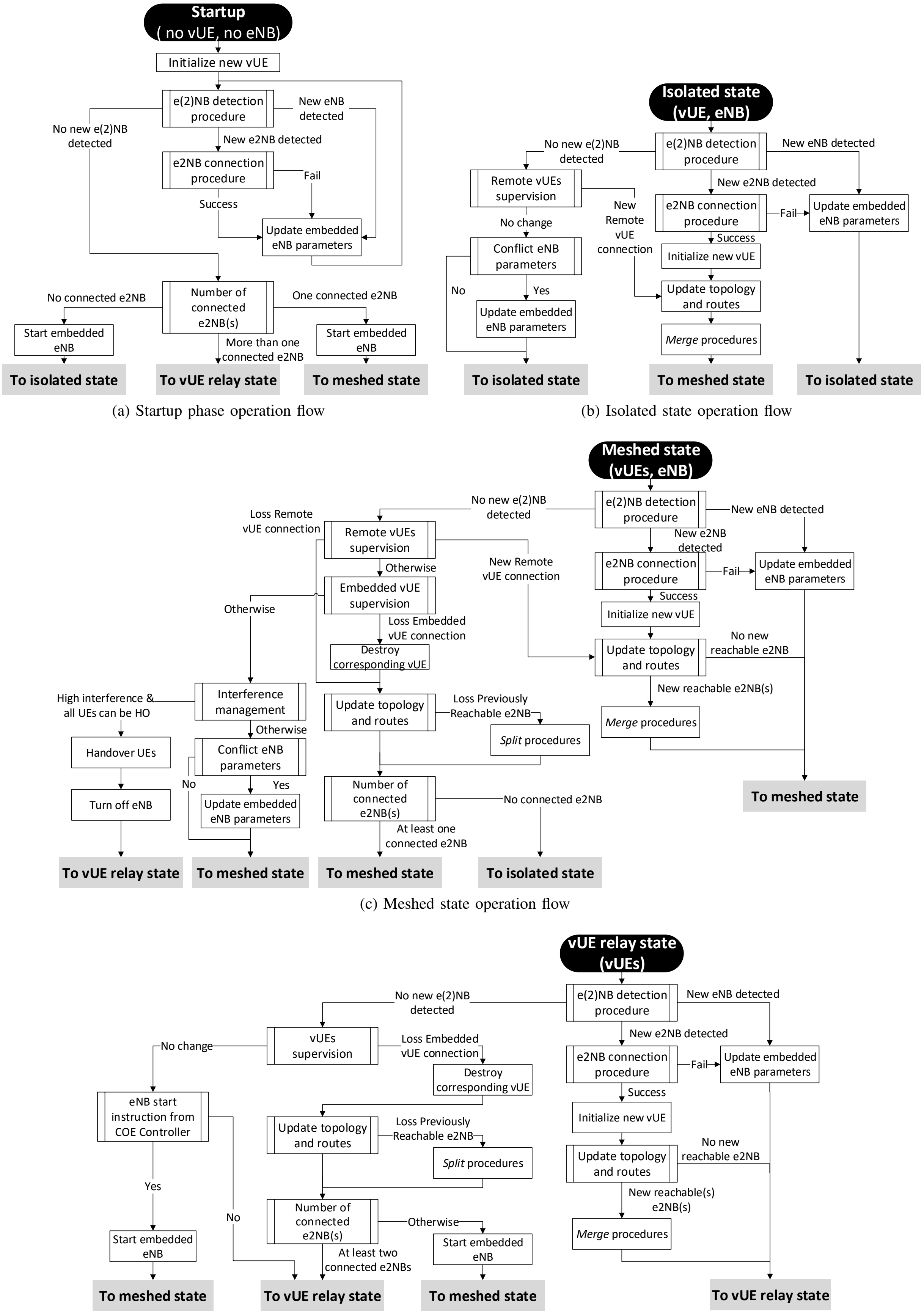
However, in our case, the newly startup e2NB do not have the access to a common CN. Hence, following the e2NB architecture, a specific vUE is firstly orchestrated and instantiated by the local COE agent to detect adjacent e(2)NBs as shown on Fig. 9(a). Such e(2)NB detect mechanism can follow the common cell search scheme of legacy UE defined by 3GPP [35]. When detecting the neighboring e(2)NB, the local COE agent will notify whether a connection shall be established or not based on the broadcasted PLMN identity in SIB1. If yes, the vUE will follow the aforementioned attach procedure shown in Fig. 8 for connection establishment to the e2NB. Such connection between the vUE and e2NB can exchange the higher-level information to agree on the R-PDCCH and R-PDSCH configuration before using *Un*. Then, the e2NB will continue to orchestrate another vUE in order to detect and connect to another neighboring e2NB. Finally, after connecting to all neighboring e2NBs, such newly startup eNB will be into one of the three main states depending on the number of connected e2NBs.

Before stating the operation flow of other states, we detail on how to configure parameter at startup phase. Based on all detected and connected e(2)NBs, the COE agent can derive an adequate PCI value to be used by the embedded eNB. Furthermore, it will also configure parameters related to the eICIC/FeICIC and SIBs as the parameters listed in group (a) of section V-C1. Whereas if no other eNB/e2NB is detected by the vUE, the e2NB parameter will be self-configured by local COE. Such self-configuration can be based on some hash functions relying on a suitable unique identity configured by the vendor or by the responsible authority to statistically reduce parameter colliding probability [36].

b) Isolated state: Such e2NB state is with one instantiated eNB and vUE in order to serve local UE and detect neighboring e2NBs with flow chart in Fig. 9(b). When the embedded vUE connects to a neighboring e2NB or the embedded eNB have a new connection to vUE, the e2NB will then merge in its topology and go into the *Meshed* state.

Otherwise, the e2NB will monitor and update its eNB parameters. In following, we use the PCI as an example to depict how it works. Firstly, the conflicting e2NBs must realize that there is a conflict. Such realization is not a problem when the conflicting e2NBs are meshed due to the maintenance of topology at COE controller. However, it will become a problem when the conflicting e2NBs are isolated or in different meshes. In that case, conflicting e2NBs need to rely on the measurement done by its embedded vUE or served UEs. Even if the local UEs may not be able to report directly such a situation, the monitoring on the inconsistencies between the DL and UL directions can potentially disclose such hidden e2NB problem since only DL direction will be affected by the PCI conflict. Secondly, the e2NB needs to resolve the PCI

[11] Based on the *schedulingInfoList* in SIB1.



(a) Startup phase operation flow

(b) Isolated state operation flow

(c) Meshed state operation flow

(d) vUE relay state operation flow

Fig. 9: Operation flow of startup phase, isolated state, meshed state and vUE relay state.

conflict. A straightforward solution is to reconfigure the PCI and change all related parameters. However, it will disconnect UEs and hence all served UEs need to be firstly handovered to other e2NBs for service continuity. If such solution is not feasible or improper due to high priority traffics, another approach is to reconfigure parameters for a late re-start.

c) Meshed state: Such e2NB state is with one instantiated eNB and several vUEs to serve local UE, connect to neighboring e2NB, and detect neighboring e2NBs with flow chart in Fig. 9(c). Within such state, the e2NB can remain in meshed state or transit to isolated state with possible split or merge if the topology is changed. Note in the specific case that satisfies: (a) all local UEs can be handovered to other e2NBs, (b) no local UE requires direct access to the embedded applications or to the gateway, and (c) high interference to neighboring e2NB; an e2NB can turn off its embedded eNB and work as the vUE relay to mesh backhaul links. Transition from *Meshed* state to *vUE relay* state should be confirmed by the COE controller.

d) vUE relay state: As shown in Fig. 9(d), the e2NB can solely rely on the vUEs to relay traffics. The COE controller handles the re-start of the eNB stack when there is no vUE disconnection based on the network topology and mobility. However, if connections to neighboring e2NBs are lost such that the e2NB is only connected to one neighbor, it may transit to the *Meshed* state by itself[12] and turn on the eNB again. Such eNB re-start can base on parameters decided by the approach stated in the startup phase or instructed by the COE controller.

D. Core Network logical connectivity

1) MME: As legacy LTE access network, MME is a key controlling entity that supports various functionalities to enable logical connectivity and service continuity. Firstly, each MME shall maintain its own TAI, do the tracking area update (TAU) for connected UE, and collaborate with each other to enable the paging mechanism at the corresponding e2NB to reach target vUE/UE. Secondly, the MME shall be changed and the access context shall be exchanged during the handover of UE between e2NBs. Thirdly, the MME will maintain a unique S/P-GW for each UE and manage the bearer for UE to enable the corresponding service.

2) HSS provisioning and cooperation: To allow interconnection between e2NBs, the vUE and the embedded eNB shall be able to authenticate each other. Thus, each HSS must incorporate records of at least one authenticated vUE per e2NB to enable inter-connections. Moreover, mechanisms to update/synchronize the HSS databases from records of HSS belongs to other e2NBs should be integrated to allow the handover of local UEs and potentially the automatic update of the deployed and authorized e2NBs in the mesh network. Security requirements of such procedures should be carefully evaluated, but are deferred to future work.

3) S/P-GW: These two entities may be utilized to allow the IP-based communication of UEs (mainly between legacy eNBs and e2NBs). The S-GW is served as the local mobility anchor which forwards UE data-plane packets during the handover to another e2NB. Based on the network topology and the location of the corresponding P-GW, the S-GW will do data forwarding. Whereas the P-GW is served as the IP anchor of the corresponding service and allocates IP address to the legacy UE.

4) Routing: There exists several routing algorithms that can be deployed on the top of the mesh network leading to different metric optimization, for instance an adaptive distributed mesh routing can be used to cope with different traffic patterns and network topologies [37]. Our proposal does not rely on any hypothesis of a specific routing algorithm but the routes should be updated at each topology change and propagated over the network. Note that network split and merge operation will impact the routing decision, and can potentially trigger the handover and gateway change for UEs.

5) Application and services: Such high-layer applications and services is used to retain the minimal user services regardless of e2NB states and can be used for following purposes:

- Embed applications to enable standalone operation (e.g., voice over IP (VoIP), file transfer to local server, etc.);
- Embed applications to enable cooperative network services (e.g., collaborative map/event, vital information broadcast, service delegation, etc.);
- Enable cross-layer optimization and application-level access control (e.g., flow control policies of real-time traffic flow, load balancing within the meshed network, access to the shared content).

E. Summary

In this section, we detail on the building blocks and approaches of the e2NB to enable autonomous self-backhauling network in a bottom-up approach. All these blocks and e2NB operation flows shall be well managed by the local COE. However, we do not describe in detail how to control the topology, schedule resource and manage interference efficiently to enable the *autonomous mesh network* via utilizing COE. Hence, we continue on introducing our proposed approach to deal with these problems in the next section.

VI. Algorithms for autonomous COE

As outlined in previous sections, realizing an efficient inter-e2NB mesh backhaul requires careful resource scheduling among links. In this section, we firstly give an overview of the considered resource allocation problem for self-backhauling. Then, we introduce the proposed hierarchical approach that comprises centralized and distributed scheduler and the way to utilize the COE architecture. Finally, we detail on the algorithms for both centralized and distributed schedulers.

A. Problem overview

Using the *Un* interface to realize the backhaul links of a LTE network in a mesh fashion is similar to the time-division multiple access (TDMA) based wireless mesh network in order to utilize all MBSFN SFs. However, realizing

[12] Such e2NB will not be in isolated state since it must have two neighboring e2NBs before coming to vuE relay state.

an efficient wireless mesh network on a single frequency band is still an open research problem. As all nodes share the resources of the same frequency band, each transmitter becomes a potential interferer which limits the achievable rate. Furthermore, there exist more related issues that need to be considered in the meantime as summarized in [38], including (a) topology control, (b) routing, (c) link scheduling, (d) interference measurement, and (e) power control. As all these issues are highly inter-dependent across different network layers to provide suitable QoS of each traffic flow, they cannot be solved separately. For instance, in [39], the authors jointly consider resource allocation and relay selection in a multi-hop relay network to maximize total user satisfactions. Hence, we propose a coordinated and cross-layer approach to unleash the performance barriers when meshing e2NBs in order to guarantee the QoS in per-flow basis of the self-backhauling LTE mesh network. Last but not least, such approach will rely on the coordination between COEs of meshed e2NBs that will be described in next subsection.

B. COE role and proposed hierarchical approach

To enable such cross-layer coordination, we rely on the COEs that can serve two different logical roles: *COE controller* and *COE agent*. The COE controller is a logically centralized entity that is connected to a number of COE agents [40], one per e2NB in a typical case (refer to Fig. 2). The COE controller manages and orchestrates the mesh network through *policy enforcement* over the COE agents [8], [13]. The COE agent can either act as a local controller delegated by the centralized controller, or in coordination with other agents and centralized COE controller. The communication protocol between the centralized controller and agents is done through bi-directional message exchange over the backhaul links. In one direction, the COE agent sends measured performance indicators and e2NB status to the centralized controller and other agents, while in the other direction the centralized controller enforces policies that define the operation to be executed by the agents and the underlying eNB and vUEs, as shown on Fig. 10. Such COE coordination provides substantial flexibility to realize the hierarchical approach, and is able to reduce the control overhead by delegating more functions to the COE agent at the cost of less coordination.

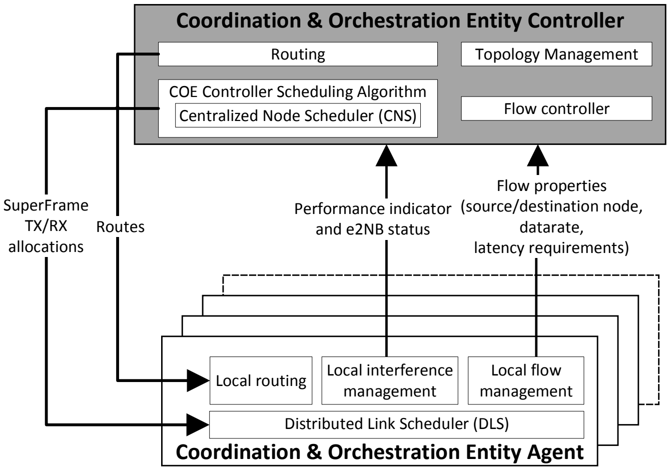

Coordination & Orchestration Entity Controller
Routing
Topology Management
COE Controller Scheduling Algorithm
Centralized Node Scheduler (CNS)
Flow controller
SuperFrame TX/RX allocations
Routes
Performance indicator and e2NB status
Flow properties (source/destination node, datarate, latency requirements)
Local routing
Local interference management
Local flow management
Distributed Link Scheduler (DLS)
Coordination & Orchestration Entity Agent

Fig. 10: Coordination and Orchestration Entity architecture.

Then, all related cross-layer parameters shall be scheduled by the centralized COE controller that include (a) next e2NB hop for backhaul relaying, (b) MBSFN SFs for backhaul transportation, (c) relaying transportation direction (DL/UL), and (d) low-layer transportation resource (e.g., PRBs, MCS) for both access and backhaul links. However, due to the limitation of real deployment and time-scale separation between COE controller and COE agents, the propagation of control messages over the backhaul links cannot be instantaneous. Thus, we can group parameters that shall be scheduled in a real-time manner (i.e., (c), (d)) to be handled in a distributed manner whereas some others are allocated centrally in a larger time-scale benefiting from a whole network view (i.e., (a), (b)). To enable such hierarchical approach, network information shall be abstracted by COE agent, for instance, the signal to power ratio (SINR) is derived from the reference signal received power (RSRP) measurement, in order to provide a simple but sufficient network information to the centralized COE controller.

The topology management unit in the centralized COE controller will enforce policies to e2NBs in the mesh in order to decide to which adjacent e2NBs they should connect to using their embedded vUEs. Note the network topology is denoted via the standard graph notation $\mathcal{G} = (\mathcal{V}_{e2NB}, \mathcal{E}_{link})$. The vertex set $\mathcal{V}_{e2NB}$ comprises e2NBs in the mesh, and the edge set $\mathcal{E}_{link}$ comprises directional edge (u, v) in the mesh where e2NB u acts as an eNB and e2NB v as a vUE. A neighboring vertice set of e2NB u is defined as N_u that comprises all its adjacent e2NBs. Based on the graph formation, we use the Dijkstra's shortest path algorithm in terms of the number of hops to route traffics in the backhaul links. Such algorithm can significantly reduce the per-flow latency generated by extra hops; however, it can also be adapted to different edge weights. After routing, we furthermore compute the link load for real-time traffic over edge (u, v) as $load_{u,v}$ in terms of the number of real-time traffic bits to be transported within a SF. It is based on the flow information available at the flow controller entity at COE controller provided by COE agent. A set $\mathcal{L} = \{load_{u,v} : \forall (u, v) \in \mathcal{E}_{link}\}$ is defined to comprise all link load of edges in the mesh that will be used for resource scheduling of real-time traffic at the centralized node controller (CNS). In contrast, the non-real-time traffic, i.e., elastic traffic, will be served in a best-effort manner.

TABLE IV: Comparison of centralized/distributed schedulers

Characteristic	Centralized NS	Distributed LS
Network view	Central view of mesh e2NBs	Local view of neighboring vUE/UE
Periodicity	Large time-scale (e.g., frame, superframe)	Small time-scale (e.g., subframe)
Considered link	Backhaul link	Backhaul and Access link
Scheduled resource	Time-domain MBSFN SF	Frequency-domain PRB
Legacy compliance	No legacy design	Compliant with *Uu* scheduler
Interference impact	Interference coordination	Link adaptation
Node prioritization	Prioritize e2NB with high real-time demand	Prioritize vUE over UE
Traffic prioritization	Prioritize real-time traffic over elastic traffic	

After gathering all necessary information, the scheduling problem aims to share the time resources (MBSFN SFs) and frequency resources (PRBs) between e2NBs. Hence, a hierarchical scheduling algorithm is proposed that is composed of a controller scheduling algorithm (CSA) that hosts a CNS and a distributed link scheduler (DLS). Note the centralized and distributed schedulers are executed in different time-scales for several purposes: (i) reduce excess control-plane overhead of full centralization, (ii) reuse the legacy SF-based link adaption scheduler, and (iii) flexible network management and orchestration. Then, TABLE IV summarizes these two schedulers in several aspects (detail later). To sum up, the hierarchical approach not only shows its practical usage and implementation but also is compliant with legacy SF-based scheduler.

C. COE Controller Scheduling Algorithm

Firstly, we present the CSA at the centralized COE controller in Algorithm 1. It periodically computes the backhaul SF allocation that will be transmitted to each COE agent.

Here, as the centralized COE controller cannot allocate the e2NBs on per transmission time interval (TTI)[13] basis, we introduce the *superframe* (SuF) concept. The duration of a superframe is denoted as L_{SuF} as computed in Algorithm 1; however, such duration value is not fixed and can be updated after a period of time P_{SuF} or triggered via some events like a newly-added real-time traffic flow ($event()$ in Algorithm 1). Normally, the duration of a superframe lasts for tens of TTIs whereas the update period will be larger as hundreds on TTIs. Then, the goal of CSA is to allocate the SFs (SF_{TX} in Algorithm 1) within the superframe duration to each inter-e2NB link to fulfill the bandwidth requirement of real-time traffic via using the CNS (CNS in Algorithm 1). If that is not possible, a dissatisfaction indicator is used ($rtDisSat$ in Algorithm 1) to enable the flow control operation via rejecting or removing some traffic flows based on their priorities and call admission control policies ($removeFlow()$ in Algorithm 1). The COE controller is responsible to manage all real-time flows via the integrated flow controller. Afterwards, each DLS will base on the outcomes of the CSA (i.e., SF_{TX}) to distributively allocate the transportation resources at each SF (DLS as presented in section VI-C3 and Algorithm 5).

In the following, we detail on the CSA operation.

1) Superframe duration computation (L_{SuF}): As its centralized manner, the CSA can retrieve following information regarding all real-time traffic flows from the *flow controller*:

- $MaxLat$: The maximum acceptable latency for the real-time flows (e.g., 150 ms for VoIP as detailed in section VIII).
- M_{hops}: The expected maximum number of hops of all active real-time traffic flows.
- $offset$: A stretch factor of M_{hops} to deal with the mobility pattern (i.e., vehicular speed) and network topology.

Based on aforementioned information, the CSA computes the *superframe* duration (L_{SuF}) as shown in Algorithm 1. Then, all time-domain MBSFN SFs within such *superframe* duration are considered for the allocation to inter-e2NB links for self-backhauling as we detail below.

Algorithm 1: COE Controller Scheduling Algorithm

```
Input  : $P_{SuF}$ is the SuperFrame update periodicity
         $\mathcal{V}_{e2NB}$ is the set of e2NBs
         $\mathcal{L}$ is the set of link load
Output: $SF_{TX}$ is the backhaul SF allocation for e2NBs
begin
    $SF\_CNT = 0$ ; /* Logical SF index at COE controller */
    $N = 0$ ; /* Logical update index at COE controller */
    while COE controller active do
        $SF\_CNT = SF\_CNT + 1$ ; /* Current SF index */
        $Update = False$ ;
        if $event(flows, topology, ...)$ then
            $Update = True$ ;
        if $SF\_CNT \equiv 0 \pmod{P_{SuF}} \vee Update$ then
            $N = N + 1$ ; /* Current update index */
            $\left[S_u^D, S_u^U\right] = getSat(\mathcal{V}_{e2NB})$ ; (cf. Eq. 2)
            $rtDisSat = 1$ ;
            while $rtDisSat == 1$ do
                $[MaxLat, M_{hops}] = getInfos(flows)$ ;
                $L_{SuF}^N = \lceil MaxLat/(M_{hops} + offset) \rceil$ ;
                $SF_{rt}^D = f(L_{SuF}, \mathcal{V}_{e2NB}, \mathcal{L})$ ; (cf. Alg. 2)
                $SF_e^D = g(L_{SuF}^{N-1}, L_{SuF}^N, SF_{rt}, \mathcal{V}_{e2NB}, \cdots$
                  $\cdots, S_u^D)$ ; (cf. Eq. (3))
                $SF_e^U = h(L_{SuF}^{N-1}, L_{SuF}^N, SF_{rt}, \mathcal{V}_{e2NB}, \cdots$
                  $\cdots, S_u^U)$ ; (cf. Eq. (3))
                $\left[SF_{TX}^N, rtDisSat\right] = CNS(L_{SuF}^N, \mathcal{V}_{\text{e2NB}}, ...$
                  $\cdots, \mathcal{E}_{link}, SF_{rt}^D, SF_e^D, SF_e^U)$ ; (cf. Alg. 3)
                if $rtDisSat == 1$ then
                    /*Too many real-time flows*/
                    $flows = removeFlow(flows)$ ;
```

2) SF allocation for inter-e2NB self-backhauling: After getting the duration of *superframe*, we then use Algorithm 2 to compute the number of DL SFs required by each e2NB to transport real-time flows within this duration as $SF_{rt}^D[u], \forall u \in \mathcal{V}_{e2NB}$. Note the main operation to get the $SF_{rt}^D[u]$ is to divide the number of required DL PRBs (i.e., $rPRB^D[u]$) to the total number of PRBs in a single DL SF (i.e., N_{PRB}^D). In the computation of the number of required PRBs, we firstly multiply the link load within a SF (i.e., $LL_{u,v}$) by the duration of a superframe (i.e., L_{SuF}) to get the number of real-time bits to be transported in the superframe duration. Then, we introduce $PRB_{u,v}^{D/U}(x)$ as the functions that compute the number of required DL/UL PRBs to transport x bits over link (u, v) based on the measured channel quality information reported from COE agents to the COE controller.

Furthermore, to respect the *Uu* and *Un* specifications in FDD mode, an UL SF n will be allocated accordingly if the DL SF $(n-4)$ is allocated. We observe that such correspondingness can be leveraged to allocate the link loads in reverse direction (i.e., (v, u)) to the PRBs of the corresponding UL SFs (i.e., $nPRB^u[u]$). Hence, a portion (i.e., $TransportedRatio$) of the corresponding link load in reverse direction (i.e., $LL_{v,u}$) is reduced and more spare SFs in the superframe duration can be utilized to allocate other traffic flows. Finally, if the CSA is able to allocate $SF_{rt}^D[u]$ DL SFs at each e2NB u over MBSFN SFs in the *superframe* duration, then the real-time traffics can

[13] It corresponds to a single SF duration.

Algorithm 2: $SF_{rt}^{D} = f(L_{SuF}, \mathcal{V}_{e2NB}, \mathcal{L})$

```
Input  : $\mathcal{V}_{e2NB}$ is the set of e2NBs
         $L_{SuF}$ is the superframe duration
         $\mathcal{L}$ is the set of link load with $load_{u,v}$ of edge $u \rightarrow v$
         $N_u$ is the neighboring e2NB set of $u$ from $\mathcal{V}_{e2NB}$
Output: $SF_{rt}^{D}$ is the set of the number of SFs required by each
         e2NB for real-time traffic transportation
begin
    foreach $u \in \mathcal{V}_{e2NB}$ do
        $nPRB^{U}[u] = 0$ ; /* Initialize allocated UL PRBs of $u$ */
        foreach $v \in N_u$ do
            $LL_{u,v} = load_{u,v}$ ; /* Initialize each link load */
    foreach $u \in \mathcal{V}_{e2NB}$ do
        $rPRB^{D}[u] = \sum_{v \in N_u} PRB_{u,v}^{D}(LL_{u,v} \cdot L_{SuF})$ ;
        $SF_{rt}^{D}[u] = \left\lceil \frac{rPRB^{D}[u]}{N_{PRB}^{D}} \right\rceil$ ;
        $nPRB^{U}[u] = SF_{rt}^{D}[u] \cdot N_{PRB}^{U}$ ;
        foreach $v \in N_u$ do
            $LL_{u,v} = 0$ ; /* allocated over DL in $SF_{rt}^{D}[u]$ */
            $rPRB^{U} = PRB_{v,u}^{U}(LL_{v,u} \cdot L_{SuF})$ ;
            if $rPRB^{U} \neq 0$ then
                $tPRB^{U} = \min(rPRB^{U}, nPRB^{U}[u])$ ;
                $nPRB^{U}[u] = nPRB^{U}[u] - tPRB^{U}$ ;
                $TransportedRatio = (1 - \frac{tPRB^{U}}{rPRB^{U}})$ ;
                $LL_{v,u} = LL_{v,u} \cdot TransportedRatio$ ;
```

be transported by the mesh network.

After getting the number of SFs required by the real-time traffics, there might be some remaining MBSFN SFs in the superframe duration that are not utilized. Thus, we aim to carefully allocate these SFs to the elastic traffic in the backhaul links. Firstly, we introduce the "*saturated*" concept to know the bottleneck links of elastic traffics over the mesh network. Here, for each link (u, v), we independently consider DL direction $(DL(u,v))$ and UL direction $(UL(u,v))$. A SF over DL or UL direction is viewed as a "saturated SF" when it can only transport less bits than the queued bits of all aggregated elastic traffic flows after transporting the real-time traffic. Furthermore, a direction ($DL(u,v)$ or $UL(u,v)$) is considered to be "saturated" if the ratio of the number of "saturated SF" among all allocated backhaul SFs in this direction is higher than a threshold value, for instance, 90%. Finally, saturated neighboring sets of e2NB u in DL and UL directions are defined in Eq. (2).

$$S_u^D \triangleq \{v : v \in N_u, DL(u,v) \text{ is saturated}\} \tag{2a}$$

$$S_u^U \triangleq \{v : v \in N_u, UL(v,u) \text{ is saturated}\} \tag{2b}$$

Based on aforementioned definitions, we compute two values, i.e., $SF_e^U[u]$ and $SF_e^D[u]$, to represent the relative needs of the number of UL/DL SFs by e2NB u to transport the elastic traffic to its saturated neighbors in S_u^D and S_u^U as detailed in Eq. (3). As the first step, we estimate the average frequency reuse (AFR) factor from the previous scheduling results (i.e., L_{SuF}^{N-1}, $SF_{TX}^{N-1}[u]$ in Algorithm 1) where $N_{mbsfn}(L_{SuF})$ returns the set of MBSFN SFs during the superframe duration L_{SuF}. Such AFR indicates the level of resource reusing within the whole network and cannot be smaller than 1. Then, as the second step, we get the number of "*free SFs*" in a superframe duration as SF_{free} via excluding SFs that are already reserved for real-time traffic from all MBSFN SFs. Thirdly, we compute $B_e^D[u]$ and $B_e^U[u]$ as the sum of average TBS per PRB of all saturated DL and UL directions from u, respectively. These two summations use $TBS(a,b)$ function which outputs the TBS when applying MCS index a with b PRBs. Here, $MCS_{u,v}^D$ represents the applied MCS index on DL direction $DL(u,v)$ whereas $MCS_{u,v}^U$ is the applied MCS index on UL direction $UL(u,v)$. Finally, the number of SFs that are required by u for elastic traffic of UL/DL directions are derived as $SF_e^U[u]$ and $SF_e^D[u]$.

$$AFR = \frac{\sum_{u \in \mathcal{V}_{e2NB}} SF_{TX}^{N-1}[u]}{\left\| N_{mbsfn}(L_{SuF}^{N-1}) \right\|} \tag{3a}$$

$$SF_{free} = \left(\left\| N_{mbsfn}(L_{SuF}^{N}) \right\| - \sum_{u \in \mathcal{V}_{e2NB}} SF_{rt}^{D}[u] \right) \cdot AFR \tag{3b}$$

$$B_e^D[u] = \sum_{v \in S_u^D} TBS\left(MCS_{u,v}^D, 1\right) \tag{3c}$$

$$SF_e^D[u] = \left\lceil \frac{B_e^D[u] \cdot SF_{free}}{\sum_{v \in \mathcal{V}_{e2NB}} B_e^D[v]} \right\rceil \tag{3d}$$

$$B_e^U[u] = \sum_{v \in S_u^U} TBS\left(MCS_{v,u}^U, 1\right) \tag{3e}$$

$$SF_e^U[u] = \left\lceil \frac{B_e^U[u] \cdot SF_{free}}{\sum_{v \in \mathcal{V}_{e2NB}} B_e^U[v]} \right\rceil \tag{3f}$$

Based on the above derivations, the CNS is shown in Algorithm 3 in order to allocate backhaul SFs for both real-time and elastic traffics. $SF_{TX}[u][v]$ and $SF_{RX}[v][u]$ are the set of SFs used for transmission and reception on edge (u,v), respectively[14]. The main design principle of this algorithm is to allocate SFs based on the prioritization of real-time traffic over elastic traffic[15] such that there is no collisions between e2NBs that would be interfering too much. As the output, SF_{TX} contains all transmitting SFs for all links over a superframe duration and $rtDisSat$ indicates if the scheduler can satisfy all required SFs for real-time traffic or not. Such dissatisfaction indicator is used for aforementioned flow control operation in Algorithm 1.

Last but not least, the interference blocking set $\mathcal{I}_{u,v}$ comprises the e2NBs which shall be blocked due to the transmission on edge (u,v) as shown in Algorithm 4. Note the decision is made based on the received signal power (i.e., $P_{u,v}$ and $P_{w,v}$) and a pre-specified blocking criterion denoted as $criteria(a,b)$. Such $criteria(a,b)$ can be the difference of two input signal power (i.e., like A3 event of LTE handover), the decreasing of mapped MCS index due to the interferer, or other criteria. Finally, the output of such algorithm will be utilized in the CNS of Algorithm 3 for interference coordination at centralized COE controller.

[14] The wildcard character in algorithm 3 represent all possible e2NBs.

[15] $sort\_descend(\mathcal{V}, a, b, \cdots)$ is to sort $\mathcal{V}$ in descending order following the metric ordering from a, b and so on.

Algorithm 3: Centralize Node Scheduler (CNS) $CNS(L_{SuF}^N, \mathcal{V}_{e2NB}, \mathcal{E}_{link}, SF_{rt}^D, SF_e^D, SF_e^U)$

```
Input  : $L_{SuF}^N$ is the superframe duration
         $\mathcal{V}_{e2NB}$, $\mathcal{E}_{link}$, $SF_{rt}^D$, $SF_e^D$, $SF_e^U$.
Output: $SF_{TX}$, $rtDisSat$
$SF_{MBSFN} = N_{mbsfn}\left(L_{SuF}^N\right)$ ; /*Initialize the MBSFN SF set*/
foreach $u \in \mathcal{V}_{e2NB}$ do
    foreach $v \in \mathcal{V}_{e2NB} \wedge (u,v) \in \mathcal{E}_{link}$ do
        $Tx(u,v) = 0$ ; /* Initialize the DL SF temporary number */
        $SF_{TX}[u][v] = SF_{MBSFN}$ ; /* Initialize transmit SFs */
        $SF_{RX}[v][u] = SF_{MBSFN}$ ; /* Initialize receive SFs */
$rtDisSat = 0$ ; /* Initialize dissatisfaction indicator */
foreach $SF \in SF_{MBSFN}$ do
    $sort\_descend(\mathcal{V}_{e2NB}, SF_{rt}^D, SF_e^D, SF_e^U)$ ;
    $\mathcal{A}_{e2NB} = \varnothing$ ; /* Initialize active e2NB set */
    foreach $u \in \mathcal{V}_{e2NB}$ do
        if $SF_{rt}^D[u] + SF_e^D[u] + SF_e^U[u] \geq 1$ then
            if $SF \in SF_{TX}[u][*]$ then
                $Transmit = 0$ ; /* Initialize transmit indicator */
                foreach $v \in \mathcal{V}_{e2NB} \wedge (u,v) \in \mathcal{E}_{link}$ do
                    if $SF \in SF_{RX}[v][u]$ then
                        $\mathcal{I}_{u,v} = genIntf\,(u,v)$ ; (cf. Alg. 4)
                        if $\exists\, w \in \mathcal{A}_{e2NB} \bigcap \mathcal{I}_{u,v}$ then
                            /* Cannot transmit to vUE $v$ that would receive
                               too much interference from e2NBs already
                               activated in $\mathcal{A}_{e2NB}$ */
                            $remove(SF, SF_{RX}[v][u])$ ;
                            $remove(SF, SF_{TX}[u][v])$ ;
                        else
                            $Transmit = 1$ ;
                            $Tx(u,v) = Tx(u,v) + 1$ ;
                            $remove(SF, SF_{RX}[u][*])$ ;
                            $remove(SF, SF_{TX}[v][*])$ ;
                            foreach $w \neq u \in \mathcal{V}_{e2NB}$ and
                             $(v,w) \in \mathcal{E}_{link}$ do
                                $remove(SF, SF_{RX}[v][w])$ ;
                            foreach $w \in \mathcal{I}_{u,v}$ do
                                $remove(SF, SF_{TX}[w][*])$ ;
                    else
                        $remove(SF, SF_{TX}[u][v])$ ;
                if $Transmit == 1$ then
                    $\mathcal{A}_{e2NB} = u \bigcup \mathcal{A}_{e2NB}$ ;
                    if $min(Tx(u,*)) \geq 1$ then
                        foreach $v \in \mathcal{V}_{e2NB} \wedge (u,v) \in \mathcal{E}_{link}$ do
                            $Tx(u,v) = Tx(u,v) - 1$ ;
                        if $SF_{rt}^D[u] \geq 1$ then
                            $SF_{rt}^D[u] = SF_{rt}^D[u] - 1$ ;
                        else if $SF_e^D[u] \geq 1$ then
                            $SF_e^D[u] = SF_e^D[u] - 1$ ;
                        else if $SF_e^U[u] \geq 1$ then
                            $SF_e^U[u] = SF_e^U[u] - 1$ ;
foreach $u \in \mathcal{V}_{e2NB}$ do
    if $SF_{rt}^D\,[u] \neq 0$ then
        $rtDisSat = 1$ ;
```

Algorithm 4: Generate interferer e2NB ($genIntf\,(u,v)$)

```
Input  : $(u,v)$ is the edge of the graph from $u$ to $v$
         $P_{x,w}$ is received signal power at $w$ from $x$
         criteria$\,(a,b)$ is blocking criteria with input $a, b$
Output: $\mathcal{I}_{u,v}$
begin
    $\mathcal{I}_{u,v} = \varnothing$ ;
    foreach $w \in \mathcal{V}_{e2NB} \wedge w \neq u$ do
        if criteria$(P_{u,v}, P_{w,v})$ then
            /* Add interferer when meet blocking criteria */
            $\mathcal{I}_{u,v} = w \bigcup \mathcal{I}_{u,v}$ ;
```

3) Distributed link scheduling: The results of CSA (i.e., SF_{TX}) are transported to each COE agent for a distributed link scheduling. Such distributed link scheduler aims to allocate the frequency domain resource (i.e., PRB) and transport bits (i.e., TBS) of the e2NB u in per-SF basis as shown in Algorithm 5. Here, a local network view is maintained by each e2NB via forming the vUE set (i.e., $\mathcal{V}_{vUE}$) and UE set (i.e., $\mathcal{V}_{UE}$) for its link scheduling purpose. Our designed algorithm is to prioritize backhaul links (i.e., vUEs using Un interface) over access links (i.e., UEs using Uu interface) as the former one can only reach 60% of peak rate (max number of MBSFN SF per frame) compared to 100% for the legacy UE while also prioritizing real-time traffics over elastic ones. Among vUEs/UEs in the same set (i.e., $\mathcal{V}_{vUE}$/$\mathcal{V}_{UE}$), we firstly sort them based on the number of queued real-time traffic bits and then provide PRM_{min} PRBs in a round-robin way. Furthermore, in the PRB provisioning, the number of requested bit (i.e., $ReqBit$) within the corresponding queues is used to derive the allocated PRBs, and thus prevent resource over-provisioning. It has to be noted that Algorithm 5 can be adapt to apply priorities among UEs/vUEs.

D. Relaying direction selection

As mentioned beforehand, both DL and UL directions can be selected during relaying in backhaul links. It means that each packet can go over the DL or UL direction to reach the next hop. However, the selection of direction highly depends on the traffic QoS requirement, for instance, an ultra reliable traffic will select the one with better signal quality whereas the mobile broadband traffic prefers the one with higher throughput. Since our considered real-time traffic is sensitive to the latency, we use the expected waiting time of both UL and DL queues as the metric to decide which queue (DL or UL) is more preferred.

E. Extensions to TDD mode

All aforementioned algorithms are designed for FDD system; however, a joint operation of heterogeneous FDD/TDD network is envisioned as an efficient deployment to utilize available spectrum resources [41]. Moreover, the e2NB solution targets the worst case scenario with limited radio bandwidth to which the TDD case is applicable. Hence, our proposed approach shall also support the TDD mode.

Firstly, the maximum number of MBSFN SFs within a frame is reduced to 5 since SF 0, 1, 2, 5, 6 can not be used for MBSFN purpose in TDD mode. Furthermore, the FDD characteristic cannot be leveraged to allocate resource also

Algorithm 5: Distributed Link scheduler (distributed_LS)

```
Input : $u$ is current e2NB identifier
        $SF$ is current subframe identifier in the *superframe*
        $\mathcal{V}_{UE}$ is set of UEs at $u$ with non-empty queue
        $\mathcal{V}_{vUE}$ is set of vUEs at $u$ with non-empty queue
        $Q[x][p]$ is queue size of (v)UE $x$ with priority $p$
        $N_{PRB}$ is the number of available PRBs in $SF$
        $SF_{TX}$ is transmit SFs from Algorithm 3
        $MCS[x]$ is the applied MCS index of vUE/UE $x$
        $PRB_{min}$ is the minimum number of allocated PRBs
for each user
Output: PRB, TBS
$sort\_descend(\mathcal{V}_{UE}, Q[*][0])$ ; /*sort UE based on queue size*/
$sort\_descend(\mathcal{V}_{vUE}, Q[*][0])$ ; /*sort vUE based on queue size*/
foreach $x \in \mathcal{V}_{UE} \bigcup \mathcal{V}_{vUE}$ do
    $PRB\,[x] = 0$ ; /* Initialize allocated PRB */
    $TBS\,[x] = 0$ ; /* Initialize allocated TBS */
$T_{PRB} = N_{PRB}$ ; /* Initialize available PRBs in a SF */
$priority = 0$ ; /* 0: real-time, 1: elastic */
while $T_{PRB} > 0 \wedge priority < 2$ do
    $satisfy = 1$ ; /* Indicate flow of current priority is satisfied */
    foreach $x \in \mathcal{V}_{vUE}$ do
        if $SF \in SF_{TX}\,[u]\,[x]$ then
            $ReqBit = \sum_{p=0}^{priority} Q\,[x]\,[p]$ ;
            if $T_{PRB} > 0 \wedge TBS[x] < ReqBit$ then
                $PRB\,[x] = PRB\,[x] + PRB_{min}$ ;
                $T_{PRB} = T_{PRB} - PRB_{min}$ ;
                $TBS[x] = TBS\,(MCS[x], PRB[x])$ ;
                $satisfy = 0$ ;
    if $satisfy == 1$ then
        /* Schedule UEs after satisfying all vUEs */
        foreach $x \in \mathcal{V}_{UE}$ do
            $ReqBit = \sum_{p=0}^{priority} Q\,[x]\,[p]$ ;
            if $SF \in SF_{TX}\,[u]\,[*]$ then
                if $T_{PRB} > 0 \wedge TBS[x] < ReqBit$ then
                    $PRB\,[x] = PRB\,[x] + PRB_{min}$ ;
                    $T_{PRB} = T_{PRB} - PRB_{min}$ ;
                    $TBS[x] = TBS\,(MCS[x], PRB[x])$ ;
                    $satisfy = 0$ ;
    if $satisfy == 1$ then
        /* Schedule elastic traffics after fulfilling real-time ones */
        $priority = priority + 1$ ;
```

for reverse link when computing SF_{rt} in Algorithm 2. As fewer MBSFN SFs can be used for backhaul links but with more required SF_{rt} for real-time traffic transportation, we can anticipate a smaller number of free SFs can be used for elastic traffic (i.e., SF_{free} in Eq. (3)).

Secondly, different TDD UL/DL configurations will bring uneven number of UL/DL SFs within a frame as well as different UL/DL relations among SFs. As explained in section V-B3, in some configurations, a single TDD UL SF may have to multiplex several allocation from various e2NBs coming from different DL SFs. A first solution to avoid this problem is to rely only on DL transportation of the *Un* interface over MBSFN SFs with the proposed adaptations for HARQ handling. With guarantee from the above approach that each e2NB can get at least one DL SF per superframe duration, it will provide the full connectivity between e2NBs. However, it may not be the best approach in terms of satisfying different traffic flows as the induced latency for HARQ process and CQI report in a scenario with highly heterogeneous DL SF allocations between e2NBs. The second solution is to enable the *COE controller* to allocate the UL SFs just after the allocation of DL SFs. Here, several strategies can be enforced as to allocate UL SFs to minimize the average waiting time until there is a possible SF can be used for the feedback, no matter it is along DL or UL direction. However, we may not be able to ensure the presence of a fast feedback opportunity for every DL transmission. The last solution is to modify the number of UL SFs in a frame for the self-backhauling to make a more even ratio between DL and UL SFs. This solution is not directly possible using the legacy UL SFs; however, some MBSFN SFs can be selected by the *COE controller* to be used as UL-like SFs, i.e., the allocation of these SFs is managed by the UL DCIs transmitted in the previous DL SFs. Even with fewer DL SFs to be allocated, such method can ensure a fast feedback and leverage the benefit of the proposed UL SF allocation in terms of the computation of SF_{rt} for FDD mode.

F. Extensions to multi-antenna and multi-sector BS

The proposed approach was designed and presented with single antenna BS as a target to match the use case requirements of a low complexity and affordable solution. In general, multi-antenna BS can greatly improve performance in a wireless mesh network as the extra dimension (i.e., number of antennas) is being added to further reuse the available frequency spectrum. For instance, authors of [26] propose to use the massive multiple-input multiple-output (MIMO) system for in-band backhauling based on multi-antenna transceiver.

Apart from the multi-antenna BS that can directly apply the proposed approach without any drastic changes, the multi-sector BS can also be supported but with some necessary changes. As all sectors are unlikely to be isolated sufficiently from each other, it is fair to consider the blocking issue between adjacent sectors, i.e., a sector that is transmitting will disable any receptions at its adjacent sectors in the mean time. Such blocking issue can happen especially on high-power BS. In that sense, we propose two approaches to remedy this issue.

A simpler approach is to consider that all sector on an active BS will always be transmitting when the BS is active. In such a case, the interferer generation (Algorithm 4) does not change and CNS (Algorithm 3) is still valid as it is. However, different values of $SF_{rt}^{D,i}$ can be further computed for antenna i based on the value of SF_{rt}^{D} as the maximum value for all antennas of such BS. The same approach can be applied in the computation of $SF_{e}^{D,i}$ and $SF_{e}^{U,i}$ at antenna i with the value of SF_{e}^{D} and SF_{e}^{U} reflecting the maximum value among all antennas at this BS. Such approach would allow for a better satisfaction to different traffic flows due to a higher number of links being activated at the same time from a BS. However, it leverages only the highest number among antennas in terms of transmission aspect without considering the highest number among antennas in reception aspect.

More changes are required to leverage for both aspects of transmission and reception. Ideally, each sector can be considered equivalently as an e2NB with independent SF_{rt}^{D}, SF_{e}^{U}, SF_{e}^{D} in Algorithm 4 and 3 but with one additional constraint to block adjacent sectors when it is being allocated. Such constraint can lead to a better frequency re-use; however, it will require each BS to know for all its sectors the interference contributed from other sectors of adjacent BSs. Such information shall be reported to the *COE controller* but may be complicated to be measured.

G. Analysis of the approach complexity

We then discuss on the complexity of the proposed approach.

Algorithm 1 is running at the centralized *COE controller* and it loops until there is no more dissatisfaction for real-time flows. The number of loops highly depends on the selected flow control algorithm which is not detailed here as it depends on the traffic patterns and QoS requirement. In each loop, it computes SF_{rt}^{D}, SF_{e}^{D}, SF_{e}^{U} and executes the CNS.

SF_{rt}^{D} computation in Algorithm 2 loops on the e2NB u within e2NB set $\mathcal{V}_{e2NB}$ (containing N e2NBs), where each e2NB u further loops on the number of its adjacent e2NBs (i.e., N_u) in FDD mode. While the set $\mathcal{V}_{e2NB}$ can contain a large number of e2NB, the set of adjacent e2NBs for each e2NB stays limited and will most likely never be higher than a constant value c which is not proportional to N. So, in a topology with few e2NBs where the number of e2NBs is comparable to the number of adjacent neighbors, the complexity of SF_{rt}^{D} computation is equivalent to $O\left(N^2\right)$ while for a larger network with a higher number of nodes, it tends toward $O\left(N\right)$.

The computational complexity of SF_{e}^{D} and SF_{e}^{U} are equivalent to $O\left(N\right)$ due to the summation over the set $\mathcal{V}_{e2NB}$.

The CNS in Algorithm 3 loops firstly on the number of MBSFN SFs which depends on the lowest latency requirements of the real-time flows but is staked in all cases. Inside this first loop, it loops on the e2NBs u in the set $\mathcal{V}_{e2NB}$ (equivalent to $O\left(N\right)$). Inside the second loop, it loops on the edge set of connected neighbors to e2NB u: $v \in \mathcal{V}_{e2NB} \wedge (u,v) \in \mathcal{E}_{link}$ which is equivalent to N_u. Inside this third loop, it computes the interferers of the edge (u,v) through Algorithm 4 which loops on an equivalent of N in the worst case as the considered activated e2NB set ($\mathcal{A}_{e2NB}$) is a subset of $\mathcal{V}_{e2NB}$. However, the set $\mathcal{A}_{e2NB}$ considered in each run of Algorithm 4 can be further restricted in a large network where far away nodes will for sure not be interfering enough to meet the blocking criteria. Hence, it will be fair and practical to consider the case that there will be no interference between two nodes when there are more than x-hops in between, with x ranges for instance from 3 to 10 depending on the selected conservativeness. Finally, the computational complexity of CNS can be summarized as $O\left((N^3\right)$ in its worst cases. However, it is rarely behaving in such condition. Indeed, in small networks where each N_u can be considered equivalent to N, the set $\mathcal{A}_{e2NB}$ will most probably be very small compared to $\mathcal{V}_{e2NB}$ as most links will interfere each other (i.e., low frequency re-use factor). In a larger network, each N_u will be an order of magnitude smaller than N and will most probably be bounded by a constant value c which is not proportional to N, while $\mathcal{A}_{e2NB}$ will tend toward the frequency re-use factor for the last iteration in each second loop. Including these considerations, Algorithm 3 should show a complexity more closer to $O\left(N^2\right)$.

To conclude, the CNS is showing to be the most complex operation that is executed by Algorithm 1. It leads Algorithm 1 to be $O\left(N^3\right)$ when considering the worst case in a very rare condition; however, its computational complexity is mostly equal to $O\left(N^2\right)$ following our previous discussion.

VII. Physical channel performance evaluation

Even with several aforementioned merits when adopting *Un* interface for self-backhauling purpose in section V, there is still no previous work endeavored to compare its performance with the legacy *Uu* interface. In this sense, we evaluate if the *Un* interface is performing sufficiently good compared to the legacy *Uu* interface in terms of spectral efficiency, maximum throughput and computing requirements to ensure its usefulness in this section.

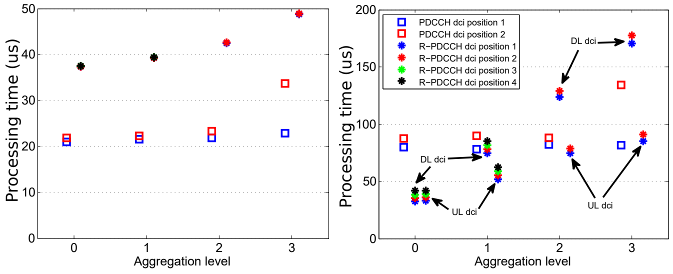
We implement a subset of the R-PDCCH and R-PDSCH of the *Un* interface in OpenAirInterface (OAI) [42], a software-based LTE/LTE-A system implementation spanning the full 3GPP protocol stack. We then experiment this subset in DL direction between a DeNB and a RN. The subset includes R-PDCCH encoding and decoding of DCI for DL allocation in the first slot and UL allocation in the second slot with resource allocation type 0 (see section 7.1.6 in [30] for explanations on resource allocation types). It also includes R-PDSCH encoding and decoding with resource allocation type 0. Both support all DLSS and ESI configurations (see section V-A3). However, one unimplemented feature defined in 3GPP specification [31] is the possible R-PDSCH allocation at the second slot when only the first slot of a SF is used to deliver R-PDCCH.

In our evaluation, extensive experiments have been carried out to analyze the impact of different parameters on link-level performance and computation time [43]. To guarantee the consistency of results, all experiments were conducted in a single thread on a Intel Xeon E5-2640v4 processor running at a fixed 2.4GHz core clock frequency with both HyperThreading and Turbo being disabled.

A. Computation time

We examine the computation time of both control channel (PDCCH/R-PDCCH) and data channel (PDSCH/R-PDSCH) of *Uu* and *Un* interfaces in order to address the hardware constraint mentioned in section II-C. These experiments aim to reflect the extra expenditures to deploy our proposed e2NB architecture with the extra *Un* interface when comparing with legacy eNB.

Firstly, we compare the computation time required to perform DCI encoding/decoding of PDCCH and R-PDCCH with different aggregation levels (from 0 to 3) and different DCI positions under 10 MHz radio bandwidth. On the *Uu* side, a DL DCI (DCI format 1) is generated and the full process from CRC attachment to RE mapping of this DCI is done. We can see in Fig. 11a that the DCI positions affects only marginally the computation time on aggregation level 0, 1



(a) DCI encoding and mapping (b) DCI search and decoding

Fig. 11: R-PDCCH and PDCCH computational time.

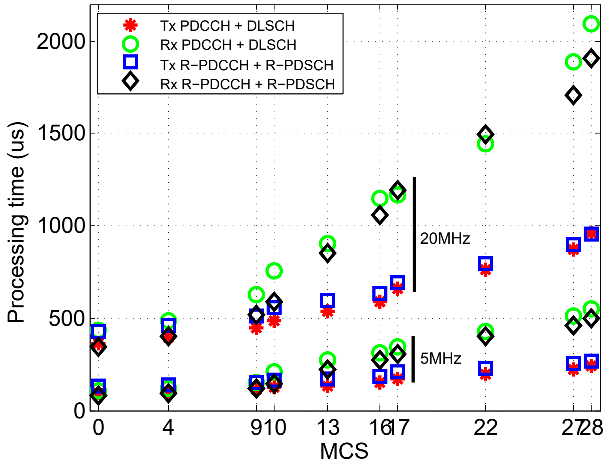


Fig. 12: TX/RX time for PDCCH/PDSCH and R-PDCCH/R-PDSCH.

and 2 while it has an impact on aggregation level 3. This is because when using aggregation level 3, generating a DCI on the second position requires the use of a second symbol for PDCCH allocation. On the *Un* side, two DCIs are generated with one DL DCI format 1 mapped to the first slot of the advertised virtual resource block (VRB) resource set and one UL DCI format 0 mapped to the second slot of the VRB set. It can be observed from Fig. 11a that the computation time of R-PDCCH encoding procedure is almost doubled compared to PDCCH of all aggregation levels. This is expected because we generate only one DCI in the PDCCH case while we generate two DCIs in the R-PDCCH case. We can observe that in both cases, the computation time increases as the aggregation level increases, although R-PDCCH seems to be more affected. Furthermore, in Fig. 11b, the computation time for the DCI decoding is presented. Results for PDCCH and R-PDCCH are quite different as the aggregation level increases due to different decoding strategies. Indeed, for PDCCH, all possible symbols corresponding to the PDCCH indicated by PCHICH are examined, which is the most time-consuming part of the process, before applying the DCI decoding procedures that look for a compatible CRC which are quite fast. This explains the decoding processing time being twice as large in the case of aggregation level 3 and DCI in second position requiring to estimate more symbols. On the other side, we can observe that the R-PDCCH decoding procedure is faster for small aggregation levels but increases with higher aggregation level especially for DL DCI decoding. There are two reasons for that. Firstly, the DCI decoder at RN knows the VRB set, but the possible positions of a potential DCI in this VRB set are multiples. In aggregation level 0, there are at most six possible positions (if the VRB set contains at least six VRBs) occupying a total of six VRBs of which REs are initially evaluated. If nothing is found by the DCI decoder (i.e., no matching CRC), then it moves to the next aggregation level, 1 in this case, which has six possible positions of DCIs spanning over two VRBs, covering twelve VRBs. This requires the estimation of extra REs before looking for DCI at a higher aggregation level. Then, the DL DCI decoding part will search for several DCI formats at each aggregation level while the UL DCI decoding will only look for DCI format 0 which explains the difference between DL and UL DCI decoding processing time.

Then, we vary the MCS value and compare the total processing time of the transmission procedure or reception procedure when using either PDCCH/PDSCH or R-PDCCH/R-PDSCH. Such comparisons are taken under both 5MHz and 20MHz radio bandwidth. A static resource allocation is used, which includes 25 (5MHz) and 100 PRBs (20MHz) for the *Uu* channel, and 24 PRBs (5MHz channel) and 96 PRBs (20MHz) for the relay channel with a R-PDCCH VRB set containing only one (5MHz) or four (20MHz) PRBs. It can be seen from Fig. 12 that TX procedures take slightly longer, in the order of 25 us, for all MCS when using the *Un* interface over the *Uu* interface. This is mainly due to the extra DCI encoding time for R-PDCCH (one for DL and one for UL) compared to PDCCH (only 1 DL DCI) as shown in Fig. 11a. For RX procedures, the *Un* interface shows a shorter computation time as the increasing of bandwidth and MCS when compared with the *Uu* interface due to a lower DCI decoding time for R-PDCCH and to the lower number of PRBs allocated. Last but not least, the results reveal that the HARQ deadlines of the *Un* interface can be met in all cases (including 100 PRBs and MCS 28) as the sum of TX and RX processing remains below 3ms [44].

To conclude this paragraph, we see that at equivalent bandwidth usage, the required processing capability for the *Un* interface is equivalent to the one for the *Uu* interface. It means that mixing *Uu* and *Un* interfaces at e2NB will not require extra processing capabilities than enabling only the *Uu* interface. Such characteristic enables the feasibility of implementing the e2NB architecture in a fully software architecture (i.e., OAI) executed on commodity hardware rather than requiring a specialized hardware infrastructure and additional expenditures.

B. *Link-level performance*

As a link-level performance metric, we examine the minimum SNR level to successfully decode 75% of transport blocks (TBs) for both PDSCH/R-PDSCH in Fig. 13 using aggregation level 0. This metric can reflect the reliability and justify the utilized effectiveness of the interface. Such experiment is taken under different radio bandwidth (5, 10, 20 MHz), different values of DLSS and ESI that modify the number of symbols used by the *Un* interface as explained in section V-A3,

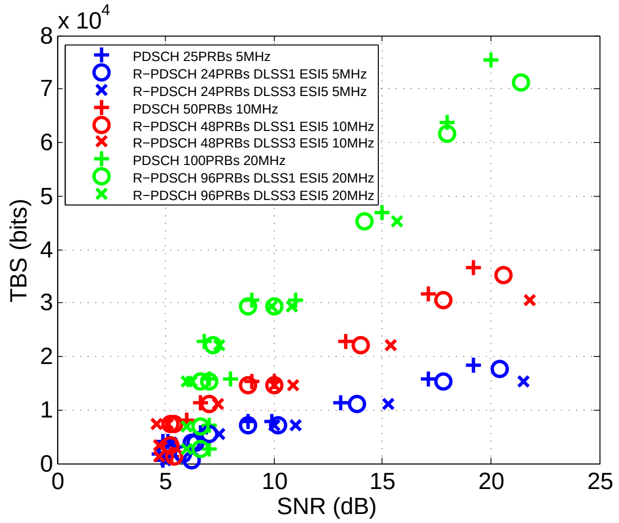


Fig. 13: Minimum SNR level to decode 75% of the TBs.

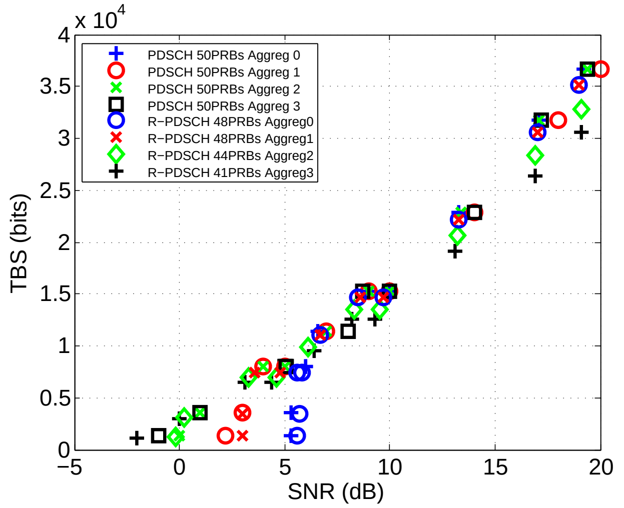


Fig. 14: Minimum SNR level for several aggregation levels.

and varying MCS values (0,4,9,10,16,17,22,27,28). Either 12 symbols ($DLSS = 1$, $ESI = 5$) or 10 symbols ($DLSS = 3$, $ESI = 5$) are available for R-PDSCH. The 10-symbol configuration actually can not be successfully decoded when only receiving the first transmission as stated in section V-A3 when the code rate is larger than 1. Since there are fewer symbols for R-PDSCH transportation, a higher code rate is experienced and a slightly higher SNR level is required to reach 75% of successful decoding. We can see that the difference between the required SNR over *Uu* and *Un* interface increases as the MCS increases, reaching around 1.5dB between PDSCH and R-PDSCH with 12 symbols when using MCS 28, and almost 5dB between PDSCH and R-PDSCH with 10 symbols when using MCS 27. This is due to the required SNR increasing non linearly with the increase of code rate. Moreover, we can see that due to our current implementation that does not allow for the second slot of a PRB to be used for R-PDSCH if only its first slot is used for R-PDCCH, the achieved TBS at equivalent bandwidth usage (including R-PDCCH) is smaller using the *Un* interface. Finally, we can observe that for higher TBS values (resulting from high MCS), no result are available for the R-PDSCH configuration relying on 10 symbols for the transportation. This is due to the code rate being higher than 1, as explained in section V-B2, preventing to transmit and decode.

To sum up, the spectral efficiency of the R-PDCCH/R-PDSCH is slightly lower than the one of PDSCH due to a fewer number of available symbols. R-PDSCH also reaches slightly lower maximum data rate (especially for large MCS that leads to large TBS) due to the lower number of assignable PRBs, however this is due to the current implementation limitations. Moreover, we can see that using fewer than 11 symbols for R-PDSCH leads to impossible MCS values, lowering the maximum achievable throughput. This calls for the use of dedicated TBS tables to optimize these cases or to restrict the usable ranges in the common TBS table.

We also examine the results under different aggregation levels in Fig. 14 under 10MHz radio bandwidth when modifying the used MCS value. It can be observed that the required SNR level to achieve 75% successful transportation is similar in both R-PDCCH/R-PDSCH and PDCCH/PDSCH cases with a small difference on the achieved TBS when using a smaller MCS. However, the gap between the achievable TBS of R-PDCCH/R-PDSCH and PDCCH/PDSCH increases when using larger MCS and higher aggregation level due to the limitations of our implementation (no use of the second slots of PRBs used for R-PDCCH to carry R-PDSCH). However, the complete implementation would see the SNR difference between R-PDSCH and R-PDCCH to increase as it would slightly increase the code rate. Moreover, we can see as expected that starting from 5dB of SNR, aggregation level 0 is sufficient and higher aggregation levels are not useful. In such a case, the differences between R-PDCCH/R-PDSCH and PDCCH/PDSCH efficiency is comparatively small.

To sum up, our examinations prove that the performance of the *Un* interface is close to the legacy *Uu* one, supporting the idea of using it as an efficient and reliable backhaul interface for the mesh network as claimed in section V-A. Furthermore, a software implementation of *Un* interface is feasible that makes our proposed e2NB architecture attractive for the emerging 4G/5G use cases as stated in TABLE II.

VIII. Evaluation of the proposed approach

Based on the aforementioned link-level performance examination on *Un* interface, we further evaluate the system-level performance of the envisioned network in this section. Hence, we compare several resource scheduling approaches on different network topologies experiencing various traffic flows.

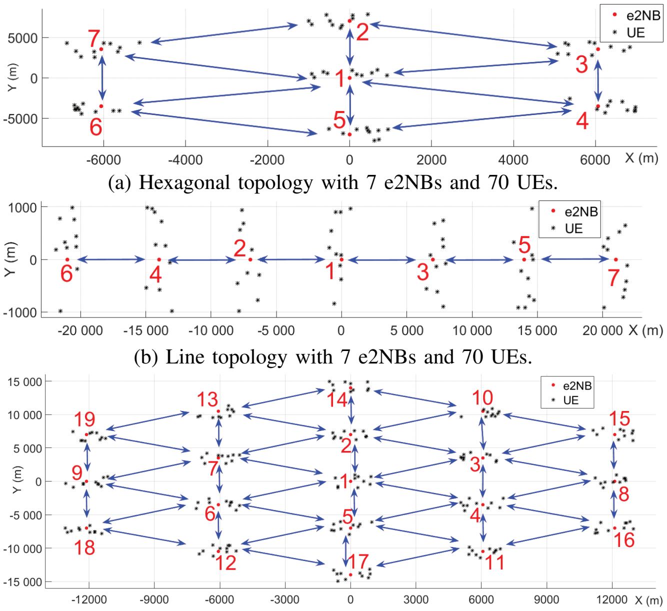
A. Simulation environment

A complete LTE simulator is developed in MATLAB allowing to create a 2D-map representing a desired network topology of e2NBs with their associated UEs and to generate arbitrary flows between nodes (e.g., UE to UE, UE to e2NB, e2NB to e2NB traffic). To model the processing time for each incoming packet at e2NB, we assume that it takes 5ms to finish all processing before pushing it to queue (DL or UL) for relaying to next hop. Such assumption is realistic considering less than 3ms is required for the physical layer as shown in section VII.

1) Simulation parameters: Our simulation parameters applied to UEs and eNBs are mostly taken from 3GPP documents ([45]–[47]) with each e2NB operating in TM 1 using a single omnidirectional antenna. To characterize the in-band characteristic, we use the same carrier frequency (2.1GHz, band 4) through the network with either a 5MHz FDD, 10MHz FDD or 10MHz TDD channel bandwidth. In TDD, UL SFs are not used for the *Un* interface but only MBSFN DL SFs are as described in section VI-E. Furthermore, to evaluate the interference impact, we do not assume any applied interference cancellation scheme at both e2NB and UE. Between e2NBs, a free space path loss model of coefficient 2.1 is applied with Claussen shadow fading and EPA channel type. Between e2NBs and UEs, a rural (from [47]) path loss model is used with Claussen shadow fading and EPA channel type. The above channel model is selected to limit the interference effects between UEs served by adjacent e2NBs as no interference mitigation or coordination method is assumed for the access link (i.e., eICIC) as we are mainly interested in characterizing the behavior of the proposed schemes for the backhaul links. Moreover, in urban scenarios, inter-BS channels are usually of better quality than BS-UE channels due to the position of antennas and the higher line-of-sight (LOS) air propagation probability. Moreover, the HARQ mechanism is not completely implemented in the simulator but work as an ARQ mechanism: ACK/NACK are handled but there is no benefit taken from the previously failed transmission. UEs are only scheduled for UL/DL transmission over non-MBSFN SFs. In Algorithm 4, e2NB w is considered an interferer of link (u, v) if the expected SINR over DL link (u, v) when e2NB w is concurrently activated is expected to reduce the expected MCS by more than 7 (for instance, if reported SINR over link (u, v) is expected to allow for MCS 22 without any interferer, w is considered an interferer and is blocked if the expected SINR at v becomes too small to allow for $MCS > 15$ over link (u, v) when w is transmitting). Moreover, the $offset$ parameter for L_{SuF} computation is set to 1 and P_{SuF} is set to 400ms. Finally, the simulations are performed for a duration of 10000 SFs.

2) Network topologies: Here, we consider three different representative network topologies as shown in Fig. 15(a), 15(b) and 15(c). In each topology, all e2NBs have 10 attached UEs and are connected to adjacent e2NBs as indicated by the bi-directional arrows as shown in Fig. 15.

3) Traffic patterns: Depending on the scenarios, both real-time and elastic traffic flows are continuously generated during the simulations. For real-time traffic flows, we randomly pair all UEs to establish bi-directional VoIP calls. Each call generates 20 bytes payload packets with a 20ms arrival rate in a fixed distribution. Each packet represents a 40 bytes final transport size on the physical layer that includes all the protocol headers from UDP to MAC layer relying on Robust Header Compression (ROHC). For QoS requirement of the real-time traffic, we use the maximum one-way-delay of 150ms for 95-percentile of the packet to ensure a quality call with a MOS of 3.5 using a G.729 codec [48]. Whereas the elastic traffic is set between BSs to represent the inter-site data transfers that often happen in military and public safety scenarios. Each elastic traffic is served in the best effort manner to maximize its data rate. It behaves as a buffer saturating flow, continuously generating packets at the initial node such that the buffer is never empty.



(a) Hexagonal topology with 7 e2NBs and 70 UEs.

(b) Line topology with 7 e2NBs and 70 UEs.

(c) Hexagonal topology with 19 e2NBs and 190 UEs.

Fig. 15: Considered network topologies.

B. Considered Algorithms

In this paper, we base on our two prior works that only show the prototype of fully proposed architecture and algorithm herein. In [14], we compared a first version of the hierarchical approach to a legacy link scheduling algorithm for mesh networks that optimizes the network throughput and frequency reuse but uses only point to point (PTP) transmissions in each SF that is not as the preferred multi-point to point transmission used in LTE thanks to OFDMA characteristic. The results showed that our approach is superior to the legacy one. In a second work, the aforementioned first version approach is enhanced as an interference-aware cross-layer hierarchical approach exploiting the FDD system property. The simulation results showed its advantages in FDD system in terms of meeting real-time QoS requirements as well as providing a high throughput. Hence, in this section, we compare three realistically implementable variant algorithms of our fully proposed approach in this work:

1) A *baseline algorithm* which is unaware of the required SF of both real-time and elastic traffics, i.e., $SF^D_{rt}[u] = 1$ and $SF^D_e[u] = k > 1$, $SF^U_e[u] = k > 1, \forall u \in \mathcal{V}_{e2NB}$. It only aims to allocate the same number of backhaul SFs to each e2NB, and is denoted as *B.*.
2) A *simplified algorithm* which does not leverage the FDD characteristic when computing the required SFs for real-time traffic in Algorithm 2, denoted as *DL.*.
3) The full algorithm proposed in this work that exploits the UL direction of FDD mode, denoted as *UL.*.

However, no dynamic flow control is implemented in our evaluation. Thus, when dissatisfaction happens (i.e., $rtDisSat == 1$ in Algorithm 1), we increase the SuF duration (i.e., L_{SuF}) by 10 SFs rather than rejecting or removing any real-time flow.

Based on these three considered algorithms and the applied traffic patterns, we can summarize all compared scenarios as listed in TABLE V. We can firstly see that each algorithm can be applied to tackle three different types of traffic pattern. The *B.*, *DL.* and *UL.* represents the baseline, simplified and full algorithms when applied to the case with only elastic traffic flows in the network topology, i.e., no UE pairing for real-time flows. Here, as there is no elastic flows, we forcedly set their SuF duration (i.e., L_{SuF}) to be 150ms and $rPRB^D\,[u]$ to be 1 in order to be applicable in Algorithm 1 and 2. In contrast, *B. V*, *DL. V* and *UL. V* denote the case where we apply these three algorithms with both elastic and real-time traffic flows. Note that the UEs can be randomly paired with any others to transport real-time VoIP flows without any restrictions. Lastly, we consider the case that only allows UEs to be paired within a maximum number of hops over the mesh, i.e., N_{hop}. Take *B. 1V* as an example, it means that the baseline algorithm is applied and UEs can only be paired when the number of hops over the mesh is not larger than 1. Hence, VoIP flows can only be established between UEs that are served by the same e2NB (no hop on the mesh as UE2 and UE9 in Fig. 4) or by two adjacent e2NBs (1 hop on the mesh as UE2 and UE3 in Fig. 4).

TABLE V: Compared algorithms with corresponding notations

Algorithm	Traffic	Possible UE pairing	Notation
Baseline	Only elastic	-	B.
	Real-time and elastic	Pair users with $N_{hop} \leq 1/2/3$	B. 1V, B. 2V, B. 3V
		No restriction on pairing	B. V
Simplified	Only elastic	-	DL.
	Real-time and elastic	Pair users with $N_{hop} \leq 1/2/3$	DL. 1V, DL. 2V, DL. 3V
		No restriction on pairing	DL. V
Full	Only elastic	-	UL.
	Real-time and elastic	Pair users with $N_{hop} \leq 1/2/3$	UL. 1V, UL. 2V, UL. 3V
		No restriction on pairing	UL. V

C. Simulation Results

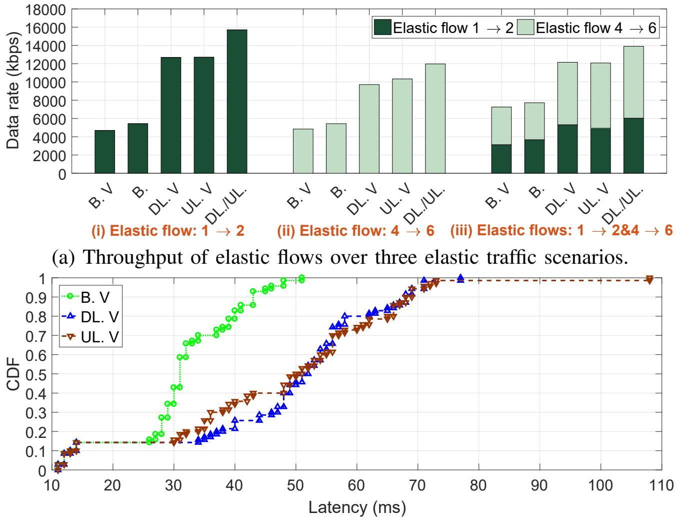
Based on the aforementioned three network topologies in Fig. 15, we compare following two performance metrics belong to different types of traffic:

- Satisfaction ratio in terms of the 95-percentile point of per-flow latency for **real-time traffic**
- Cumulated throughput of all elastic flows for **elastic traffic**

1) Hexagonal topology with 7 e2NBs: Besides the randomly-paired VoIP traffic (35 bi-directional VoIP calls between the 70 UEs), three different flow scenarios are evaluated for elastic traffic: (i) from $e2NB_4$ to $e2NB_6$ ($4 \rightarrow 6$), (ii) from $e2NB_1$ to $e2NB_2$ ($1 \rightarrow 2$) and (iii) both aforementioned two elastic flows ($4 \rightarrow 6$ & $1 \rightarrow 2$). Note that randomness of UE initial positions and randomness of UE pairing for VoIP flows depends on a seed that is fixed when changing of evaluated algorithm, ensuring that the VoIP flow distribution (source and destination, number of hops) stays the same when changing the scheduling algorithm.

a) FDD mode: Performance is firstly evaluated using a 10MHz radio bandwidth in FDD mode (i.e., totally it requires 20MHz bandwidth separated in one 10MHz for DL and another 10MHz for UL). In Fig. 16(a), we observe that the ones without real-time traffic flows can have a larger throughput than the ones with real-time traffic, i.e., throughputs of *B.*, *DL.* and *UL.* are higher than the ones of *B. V*, *DL. V*, and *UL. V*, respectively. An identical result is displayed for *DL.* and *UL.* as their only difference is in the handling of SF_{rt}^D; hence, they are identical when there is no real-time flows. Further, we can see that both full algorithm and simplified algorithm are able to provide a higher throughput than the basic algorithm in all three scenarios, with VoIP traffic being enabled or not. And a similar performance is seen between the full algorithm and simplified algorithm with VoIP traffic enabled. This is because in this topology, the full algorithm is not able to optimize SF_{rt}^D better than the simplified algorithm due to the combination of a low number of maximum hops and a high mesh degree.

Moreover, the throughput of flow $4 \rightarrow 6$ and flow $1 \rightarrow 2$ in their respective scenario are different as the first one goes over two hops on the mesh while the second one has only one hop. In the first place, we could have expected the throughput of the elastic flow from $4 \rightarrow 6$ would be half of the one of $1 \rightarrow 2$ since it requires at least twice the radio resources to provide the same end-to-end throughput; however, it is only around 20% lower rather than the expected 50%. This is because we exploit the UL/DL specificity of FDD mode. In detail, such FDD characteristic is in that the radio of each e2NB will transmit on half of its resources and receive on another half even with the dynamics to modify the number of DL MBSFN SFs to be used for either TX or RX at each e2NB. Indeed, when a DL MBSFN SF is used to receive from another e2NB, the corresponding UL SF (4ms later) is reserved for transmission to that e2NB in case of receiving an UL allocation. In this sense, a similar throughput can be foreseen for a directed flow between two hops or one hop. Specifically, when we consider the flow $4 \rightarrow 6$ that goes through $e2NB_1$ (or either



(a) Throughput of elastic flows over three elastic traffic scenarios.

(b) CDF plot of the 95-th percentile of packet latency among real-time flows with two elastic flows: 1 → 2 and 4 → 6.

Fig. 16: Hexagonal topology with 7 e2NBs and 70 UEs using 10MHz in FDD mode.

$e2NB_5$), $e2NB_1$ can do the transmission over DL direction to $e2NB_6$ at SF n and can also transmit to $e2NB_4$ over the UL direction at SF $(n+4)$. Thus, the COE controller will utilize such characteristic to allocate DL SFs toward one e2NB and exploit the corresponding UL SFs toward another e2NB in order not to use double resources to route the two-hop elastic flow and can achieve a close throughput as the one-hop one. To sum up, $e2NB_1$ will be allocated with roughly the same amount of resources for either one-hop $(1 \rightarrow 2)$ or two-hop $(4 \rightarrow 6)$ to serve these two best effort traffic equally and thus a similar performance on the end-to-end is observed.

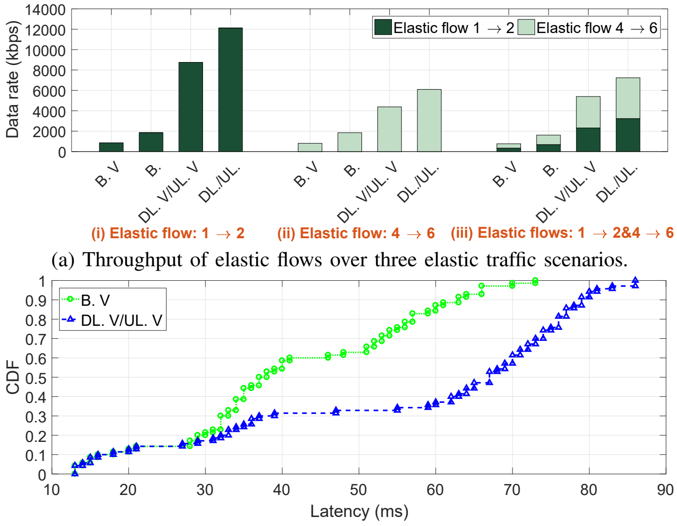
Then, Fig. 16.(b) shows the CDF of the 95-percentile point of per-flow latency over real-time traffic flow under the scenario with two elastic flows $(4 \rightarrow 6$ & $1 \rightarrow 2)$. The bottom left portions of figure correspond to some VoIP flows between UEs that are under the coverage of a same e2NB; hence, they experience very low latency and can be properly allocated by either algorithm as they depend only on the local scheduler. It can be observed that all real-time VoIP flows can satisfy the 150ms requirement for 95% of their packets (i.e., satisfaction ratio is 100% for all three algorithms). Nevertheless, we can see that the baseline algorithm performs better than the others since the maximum 95-percentile delay is around 50ms while it is up to 80ms and even 110ms for some flows when using simplified and full algorithm. This is because of the trade-off between boosting the throughput of elastic traffic shown in Fig. 16.(a) which is achieved by allocating more SFs to e2NBs that delivers elastic traffic while reducing the periodicity of transmissions of other e2NBs, thus the average latency and 95-percentile point is increased.

b) TDD mode: We then evaluate the behavior of the different algorithms over a 10MHz TDD mode using the TDD UL/DL configuration 4 (i.e., 10 SFs in a frame are categorized into 7 DL SFs (with 4 MBSFN SFs), 2 UL SFs, 1 special SF). Using the full algorithm in the TDD mode does not make any difference to the simplified algorithm as we are not relying on the UL SFs for the backhaul links in the current TDD approach, even though such TDD configuration 4 allows for one UL SF to be used for *Un* interface [31].

We can observe in Fig. 17.(a) that both full and simplified algorithms can largely outperforms the baseline one no matter the use of VoIP traffic or not. In contrast to the FDD mode, the throughput of elastic flows of TDD mode is now reduced by 50% for the two-hop flow compared to the one-hop flow. This is due to the fact that there are limited SFs to be allocated for backhaul links as only MBSFN DL SFs for TDD mode are utilized for meshing in the considered approach.

Then, Fig. 17.(b) shows the CDF of the 95-percentile point of per-flow latency over real-time traffic flow under the scenario with two elastic flows $(4 \rightarrow 6$ & $1 \rightarrow 2)$. Like the results shown in Fig. 16.(b), both approaches can satisfy the VoIP latency requirements. But the baseline algorithm provides a lower latency than the other twos as more SFs are provisioned evenly between e2NBs for VoIP flows as the trade-off of much worse throughput in Fig. 17.(a).

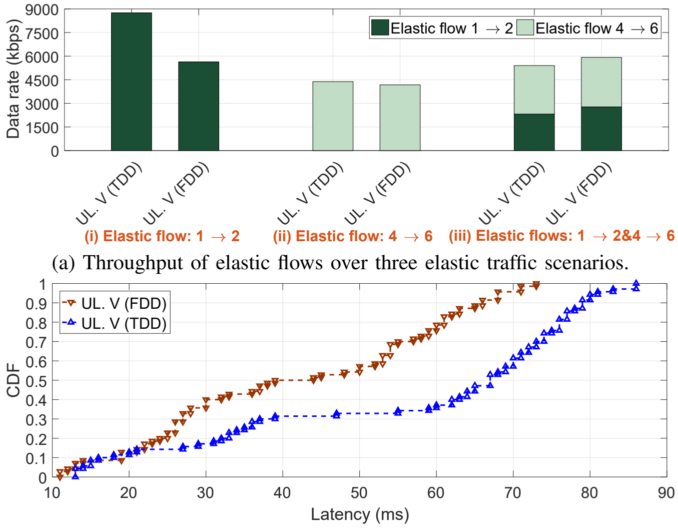
c) Comparison between FDD mode and TDD mode: To have a fair comparison between these two modes, we now compare the case of 10MHz TDD mode with a 5MHz FDD mode as they both require a total 10MHz radio bandwidth. Based on the available MBSFN SFs within a frame, FDD mode can allocate up to 60% of SFs for the backhaul links, while TDD of configuration 4 can only allocate up to 40% of SFs. However, FDD has more design constraints. For a one-hop flow, a maximum of 30% of SFs can be allocated in FDD mode in one direction as there is always a one-to-one mapping between DL SF in one direction and UL SF in another direction. To this end, when the traffic flows over the network topology are highly asymmetric, i.e., only one direction is the bottleneck, such FDD characteristic will limit the performance as we can observe in Fig. 18.(a) where the full algorithm in TDD mode provides a higher throughput for the one-hop flow $(1 \rightarrow 2)$ than the FDD mode. However, with a balancing over available SFs in both DL and UL directions for relaying, a comparable performance is shown between FDD



(a) Throughput of elastic flows over three elastic traffic scenarios.

(b) CDF plot of the 95-th percentile of packet latency among real-time flows with two elastic flows: 1 → 2 and 4 → 6.

Fig. 17: Hexagonal topology with 7 e2NBs and 70 UEs using 10MHz in TDD mode.



(a) Throughput of elastic flows over three elastic traffic scenarios.

(b) CDF plot of the 95-th percentile of packet latency among real-time flows with two elastic flows: 1 → 2 and 4 → 6.

Fig. 18: Hexagonal topology with 7 e2NBs and 70 UEs using 10MHz in TDD mode or 5MHz in FDD mode.

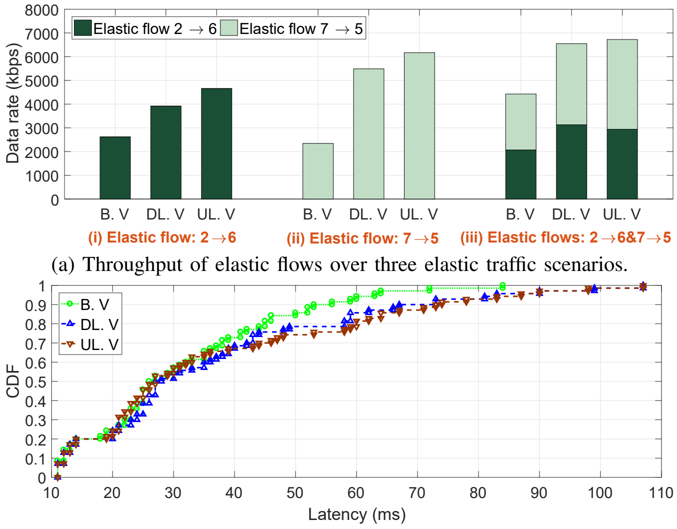
mode and TDD mode in terms of the two-hop elastic flow scenario and a slightly better performance of FDD mode is depicted in the scenario with two concurrent elastic flows.

Moreover, in Fig. 18.(b), we can see that a lower average VoIP traffic latency happens for FDD mode than TDD mode as there is a higher number of MBSFN SFs within a frame (i.e., 60% of SFs are MBSFN SFs in FDD mode as stated beforehand) and a perfect balance between DL and UL SFs to support bi-directional VoIP calls.

2) Line topology with 7 e2NBs: Like the hexagonal topology, 35 bi-directional VoIP calls are set-up between the 70 UEs and we explore three flow scenarios for elastic traffics: (i) from $e2NB_2$ to $e2NB_6$ ($2 \rightarrow 6$), (ii) from $e2NB_7$ to $e2NB_5$ ($7 \rightarrow 5$) and (iii) both aforementioned two elastic flows ($2 \rightarrow 6$ & $7 \rightarrow 5$).

a) FDD mode: Performance over this topology is firstly evaluated in FDD mode with a 5MHz radio bandwidth in both directions (i.e., a total 10MHz radio bandwidth is partitioned into 2 different 5MHz for DL and UL direction separately).

In Fig. 19.(a), we can easily observe that both simplified and full algorithm provide a higher cumulated throughput of elastic flows than the baseline algorithm over all three scenarios of elastic traffic flows. Moreover, the full algorithm outperforms the simplified version since it exploits the full FDD capability and is able to save some SFs for elastic traffic. However, the difference between the full algorithm and baseline algorithm is smaller than the one shown in the hexagonal topology of Fig. 16. A potential reason is due to fewer SFs can be shared by e2NBs to deliver elastic flows due to the topology. Firstly, as the maximum number of hops over the network as increased from 2 to 6, the SuF duration (i.e., L_{SuF}) will be reduced by a factor around 3. However, each e2NB still needs at least one allocated SF (or an UL opportunity to an adjacent node) within this SuF duration to deliver real-time traffic and ensure latency requirement. After allocating SFs for real-time flows within this shorter SuF duration, a fewer number of unallocated SFs can be shared to deliver elastic traffics and thus the throughput difference between several algorithms is reduced. Lastly, we can observe that the difference between the throughput of one-hop flow scenario ($7 \rightarrow 5$) and the throughput of two-hop flow scenario ($2 \rightarrow 6$) is similar to the phenomenon we already explained in the hexagonal topology.



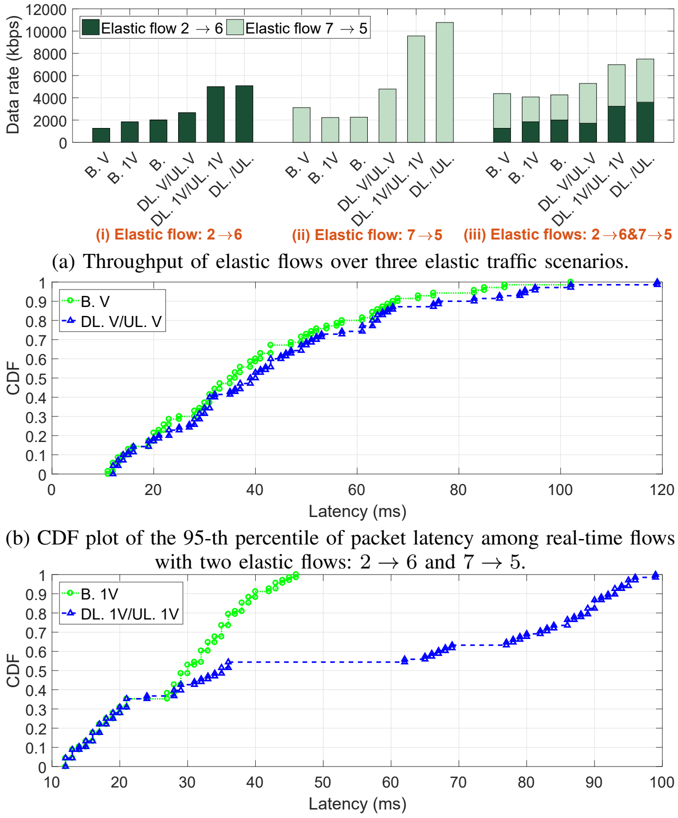
(a) Throughput of elastic flows over three elastic traffic scenarios.

(b) CDF plot of the 95-th percentile of packet latency of real-time flows with two elastic flows: $2 \rightarrow 6$ and $7 \rightarrow 5$.

Fig. 19: Line topology with 7 e2NBs and 70 UEs using 5MHz in FDD mode.

In Fig. 19.(b), we show the performance metric of real-time flows under the two elastic flow scenarios (i.e., $2 \rightarrow 6$ & $7 \rightarrow 5$) in terms of the CDF plot of 95-percentile packet delay among all real-time flows. All real-time flows are satisfied as the maximum 95-percentile delay is about 110ms that is smaller than the requested 150ms, i.e., the satisfaction ratio is 100%. Moreover, we can observe that baseline algorithm provides a slightly lower latency than the others as the trade-off we already explained in the previous hexagonal topology. The latency difference is smaller between the baseline algorithm and the others than the one in the 7 e2NBs hexagonal topology. Such phenomenon is due to the less flexibility on MBSFN SF allocation of this topology as explained in the previous paragraph.

b) TDD mode: Afterward, we evaluate the TDD mode with a 10MHz radio bandwidth. Besides the previous examined cases that allow to randomly pair all users to transport real-time VoIP flows, we further examine the case that only pairs users within a maximum number of e2NB hops, i.e., B. 1V, DL. 1V, UL. 1V for 1 e2NB hop in maximum. Such case is examined to disclose the impact on the performance of our proposed approach under different traffic patterns. If we



(a) Throughput of elastic flows over three elastic traffic scenarios.

(b) CDF plot of the 95-th percentile of packet latency among real-time flows with two elastic flows: $2 \rightarrow 6$ and $7 \rightarrow 5$.

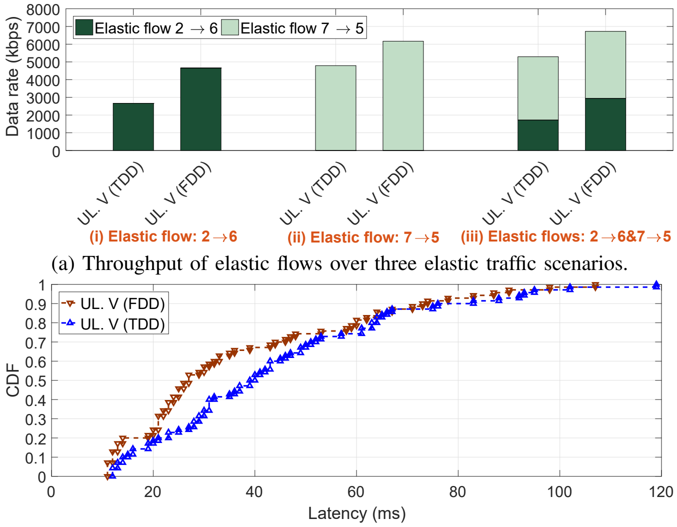
(c) CDF plot of the 95-th percentile of packet latency among real-time flows under maximum 1-hop mesh with two elastic flows: $2 \rightarrow 6$ and $7 \rightarrow 5$.

Fig. 20: Line topology with 7 e2NBs and 70 UEs using 10MHz in TDD mode.

pair UEs with more hops on the mesh in between, the SuF duration (L_{SuF}) will become smaller to ensure the end-to-end latency of such flows (see M_{hops} and Algorithm 1 introduced in section VI-C). Hence, the throughput of elastic flow will be impacted negatively as much fewer SFs are available to be used in a shorter L_{SuF} (as explained in previous FDD paragraph). To quantitatively evaluate such impact, we restrain the maximum number of hops between UEs during the pairing process such that the UEs within a pair are either served by the same e2NB or by one-hop adjacent e2NBs as B. 1V, DL. 1V, UL. 1V shown in Fig. 20.(a). This constraint matches the use cases where users are mainly communicating with other users in the same area or vicinity.

We can observe in Fig. 20.(a) that both full and simplified algorithms provide a higher throughput of elastic flows than the baseline algorithm over all considered scenarios: with non-restricted VoIP flows, with one hop restricted VoIP flows, and without VoIP flow. Moreover, we can observe that the throughput is higher when VoIP flows are restricted to a maximum of one hop over the mesh: double throughput when with a single elastic flow (either 7 → 5 or 2 → 6) and 30% gain when with two elastic flows (close to the one without any VoIP flows, i.e., DL./UL.) Such results confirm our intuition. We can also observe that, as in the TDD modes of the 7 e2NBs hexagonal topology, the throughput of the one-hop elastic flow (7 → 5) is almost two times as the throughput of two-hop elastic flow (2 → 6).

In Fig. 20.(b), without any restrictions on the maximum number of hops among inter-UE VoIP flows, the baseline algorithm only achieves a slightly better latency than the others. We can observe in Fig. 20.(c), where VoIP flows are restricted to a maximum of one hop over the mesh, that the baseline algorithm performs much better that the others as it can distribute the MBSFN SFs evenly between e2NBs and thus reduce the waiting time in queues as well as the latency. Last but not least, all algorithms can provide 100% satisfaction for all VoIP flows among each case, but both simplified and full algorithms can achieve a higher throughput of elastic flows.



(a) Throughput of elastic flows over three elastic traffic scenarios.

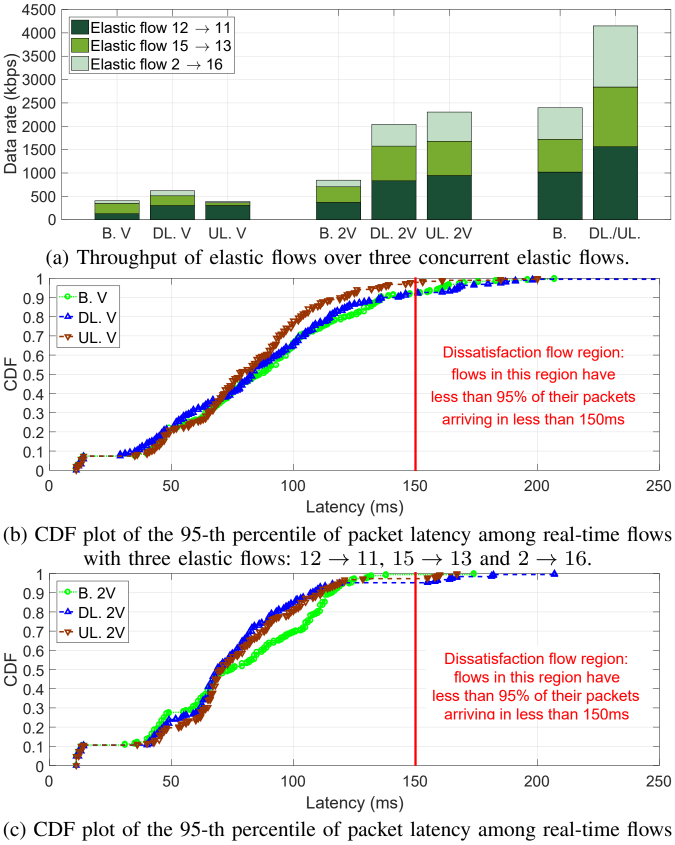
(b) CDF plot of the 95-th percentile of packet latency among real-time flows with two elastic flows: 2 → 6 and 7 → 5.

Fig. 21: Line topology with 7 e2NBs and 70 UEs using 10MHz in TDD mode or 5MHz in FDD mode.

c) Comparison between FDD mode and TDD mode: Here, we regroup our results previously shown in Fig. 19 and Fig. 20 to compare the behavior of FDD (5MHz for both directions) and TDD mode (10MHz) occupying the same aggregated bandwidth on the line topology. However, the comparison here is in favor of FDD mode as the performance limiting factor of such topology is on the fewer SFs of a shorter SuF duration, while the FDD mode can naturally provide more MBSFNs in a frame for backhaul links (i.e., 60% in FDD than 40% in TDD mode). We can observe in Fig. 21.(a) that with non-restricted VoIP flows, the FDD mode always provides a higher data rate than the TDD mode. In the mean time, we can observe in Fig. 21.(b) that FDD mode is slightly outperforming the TDD mode in terms of the latency, although both modes can achieve 100% of satisfaction. This phenomenon is also due to a higher number of MBSFN SFs in the FDD mode.

3) Hexagonal topology with 19 e2NBs: We further explore a more complicate hexagonal topology with 19 e2NBs and consider the scenario with 3 concurrent elastic flows and 95 random-paired VoIP flows: one flow from $e2NB_{12}$ to $e2NB_{11}$ (12 → 11 with two hops), one flow from $e2NB_{15}$ to $e2NB_{13}$ (15 → 13 with three hops), and one from $e2NB_2$ to $e2NB_{16}$ (2 → 16, with three hops).

a) FDD mode: Firstly, we check the performance with 5MHz radio bandwidth of FDD mode. It can be observed in Fig. 22.(a) that the simplified algorithm outperforms the other



(a) Throughput of elastic flows over three concurrent elastic flows.

(b) CDF plot of the 95-th percentile of packet latency among real-time flows with three elastic flows: 12 → 11, 15 → 13 and 2 → 16.

(c) CDF plot of the 95-th percentile of packet latency among real-time flows under maximum 2-hop mesh with three elastic flows: 12 → 11, 15 → 13 and 2 → 16.

Fig. 22: Hexagonal topology with 19 e2NBs and 190 UEs using 5MHz in FDD mode.

two algorithms in the cumulated throughput from all three elastic traffics when there is no limitation on UEs pairing for VoIP flows, i.e., DL. V. However, we can immediately observe in Fig. 22.(b) that neither approach can provide a 100% satisfaction to VoIP flows. Specifically, around 92% of VoIP flows are satisfied when using baseline and simplified algorithm; however, the full algorithm can achieve 98% of satisfaction. The baseline algorithm cannot meet the requirements because it is evenly allocating the SFs between the e2NBs within the SuF duration (around 30ms in this case) but some links cannot be satisfied with such evenly allocated SFs. As for the simplified algorithm, it is unable to satisfy the computed MBSFN SFs requirements in the required SuF duration (30ms in this case for the highest number of hops (4)). This leads to an increase of the SuF duration (i.e., as introduced in section VIII-B) to satisfy the VoIP throughput requirements of every link. But the COE will be incapable of satisfying VoIP flows with a higher number of hops (up to 4 in this topology) due to the larger value on the SuF duration (around 40ms in this case). On the other hand, the full algorithm leverages the FDD characteristic when computing the MBSFN SF DL requirements such that it can fit these requirements into the initial 30ms long SuF duration, leading to satisfy most VoIP flows.

When it comes to restrict the number of maximum hops over the mesh to be 2 (i.e., $N_{hop} = 2$) for VoIP flows, Fig. 22.(a) shows that the full algorithm is the best by around 10% enhancement over the simplified algorithm. In Fig. 22.(c), we still can see that neither approach can provide 100% satisfaction, with around 5% of dissatisfaction for the simplified algorithm, 2% for the full one, and less than 1% for the baseline one. The failure among simplified and full algorithms to meet the requirements in this case is hard to analyze as they should allocate enough SFs within the SuF duration. Hence, we analyze 84 vUE transmission behavior in Fig. 23 that shows the UL transmissions from some vUEs are subject to a higher failure rate. In DL direction, the concurrent DL transmissions on MBSFN SFs are mostly from the same set of e2NBs over several repeated SuF duration allocated by the *COE controller*. These characteristic can easily enable a better measurement on the interferer power as well as channel quality as vUE. Hence, the accurate feedback information from connected vUEs will enable the e2NB to efficiently adapt the applied MCS through adaptive modulation and coding (AMC) scheme . In contrast, UL transmissions over MBSFN SFs may not have the same interference sources over different SuF durations as the DLS at each e2NB will allocate the transmitting vUEs from its connected vUE set. Hence, it makes harder to accurately estimate the concurrent UL transmissions and noise power done by the e2NB from one SF to the corresponding one in next SuF duration. It will lead to a poor AMC decisions for the upcoming UL transmissions depending on the conservativeness or aggressiveness characteristic of the adaptation scheme. To tackle this problem, the DL direction should be more privileged for the real-time traffic even when both DL and UL directions are available to transport traffic from an e2NB to others. If only the UL direction is available, the UL scheduler should be designed to be conservative on AMC decisions (i.e., changing slowly to higher MCS, expecting more interference than reported, etc.).

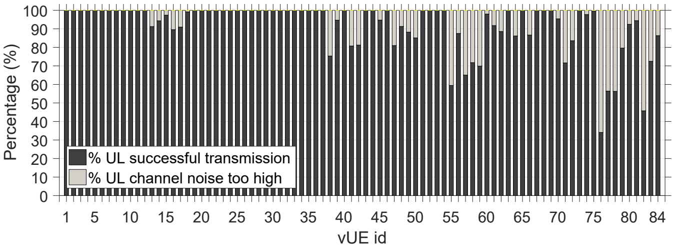
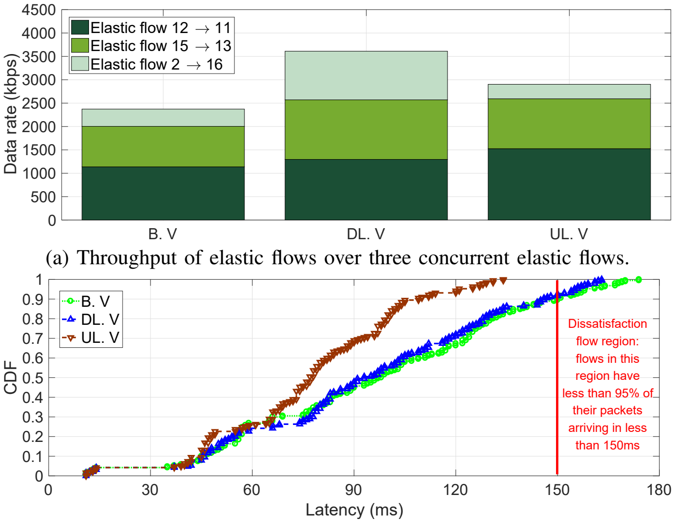


Fig. 23: vUE UL transmission successful rate of hexagonal topology with 19 e2NBs and 190 UEs using 5MHz radio bandwidth of FDD mode under *UL. 2V*.



(a) Throughput of elastic flows over three concurrent elastic flows.

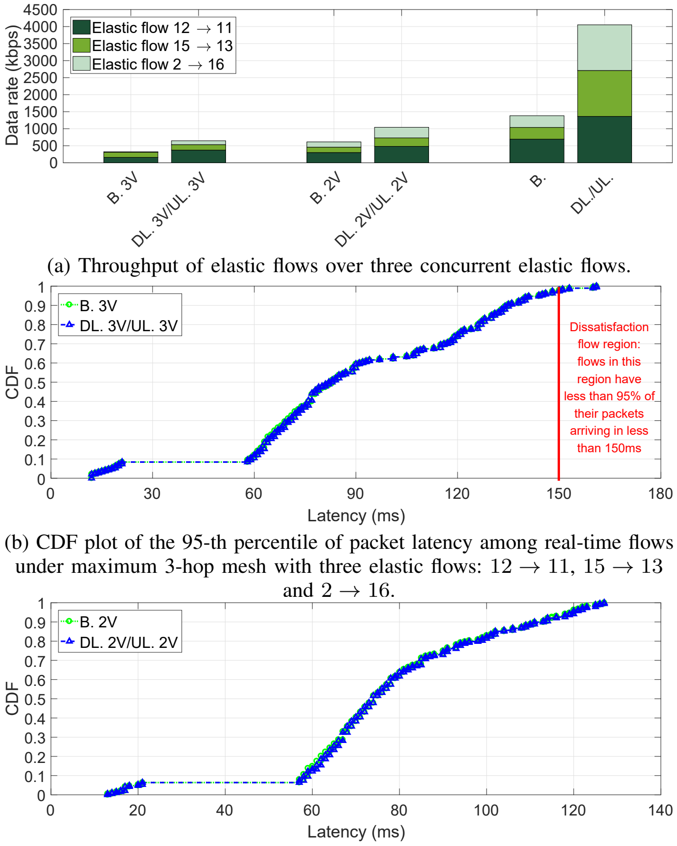
(b) CDF plot of the 95-th percentile of packet latency among real-time flows with three elastic flows: 12 → 11, 15 → 13 and 2 → 16.

Fig. 24: Hexagonal topology with 19 e2NBs and 190 UEs using 10MHz in FDD mode.

We then evaluate the 10MHz in FDD mode without any limitations on user pairing in Fig. 24.(a). The simplified algorithm provides the best throughput for elastic flows, while the full algorithm performs only slightly better than baseline one. However, we can observe in Fig. 24.(b) that full algorithm is the only approach that can meet the latency requirements for all VoIP flows over the network; that is to say, the satisfaction ratio is 100% whereas the satisfaction ratio is only 90% for other two algorithms. As explained in the 5MHz case, this is mainly because these two algorithms are unable to allocate their computed real-time SFs (i.e., SF^{D}_{rt}) over the MBSFN SFs in the SuF duration L_{SuF}. As there is no flow control in the simulation, it leads these two algorithms to increase the duration of SuF; otherwise flows would be rejected in the CSA of centralized *COE controller*. Increasing the duration of the SuF is not desirable as it increases the average time between hops over some links, introducing extra latency for VoIP flows. Both baseline and simplified algorithms do not exploit the FDD characteristics when deriving SF^{D}_{rt} while the full one does and hence reduces the expected total number of required DL SFs for real-time traffic. Once again, the full algorithm shows the best trade-off between throughput and meeting latency requirements.

b) TDD mode: Finally, we examine the TDD mode over the same topology using TDD UL/DL configuration 4 on a 10MHz radio bandwidth. We also evaluated the results when not applying any restrictions on the maximum number of hops over the mesh; however, all algorithms fail to handle so many multi-hop VoIP flows and most VoIP packets are dropped from the queues. Hence, we do not show these results here and apply the restriction on the maximum number of hops over the mesh to pair UEs and generate bi-directional VoIP flows.

The results without VoIP flows and with VoIP flows and such constraint (i.e., $N_{hop} = 2$ or $N_{hop} = 3$) are shown in Fig. 25. We can observe in Fig. 25.(a) that both simplified and full algorithms outperform the baseline one on the throughput over all scenarios, providing twice as much cumulated throughput when applying the restriction to two or three hops and around three times as much when without any VoIP flow. Regarding latency aspect, we can see in Fig. 25.(b) and Fig. 25.(c) that these three algorithms have very close performance; however, none of them can satisfy 100% of the VoIP flows when restricting the maximum hops to 3 over the mesh network. In contrast, 100% satisfaction is achieved with restrictions on the two hops which was not the case when using a 5MHz FDD channel (cf. Fig. 22(c)) as the trade-off of a lower throughput (cf. Fig. 22(a)) following the same discussion done in section VIII-C1. We also notice that the maximum data rate of the full algorithm when without any VoIP flows is similar to what was achieved by the same algorithm when using 5MHz FDD channel (cf. Fig. 22).



(a) Throughput of elastic flows over three concurrent elastic flows.

(b) CDF plot of the 95-th percentile of packet latency among real-time flows under maximum 3-hop mesh with three elastic flows: 12 → 11, 15 → 13 and 2 → 16.

(c) CDF plot of the 95-th percentile of packet latency among real-time flows under maximum 2-hop mesh with three elastic flows: 12 → 11, 15 → 13 and 2 → 16.

Fig. 25: Hexagonal topology with 19 e2NBs and 190 UEs using 10MHz in TDD mode.

D. Summary

To summarize our findings in every scenario, TABLE VI provides several important results in this section. We can observe that the full algorithm using FDD approach can provide the best trade-off between meeting the latency requirement while achieving the highest data rate for the elastic flows in most of the evaluated scenarios. Whereas TDD mode can provide the best trade-off in two specific scenarios: (a) a unique one hop elastic flow over the 7 e2NBs hexagonal topology, and (b) two hops restriction on VoIP flows over the 19 e2NBs hexagonal topology. In the later case, the runner-up is the same algorithm in FDD mode since it does not perform as expected due to a higher interference along the UL directions. Using a more conservative scheduler for UL transmissions will potentially further improve its results and allows the FDD to outperform the TDD mode. Moreover, our proposed full algorithm always achieves the best trade-off as it takes into account the network topology, considered real-time traffic characteristics and it leverages FDD characteristics. Thus, it has a better estimation on the number of required SFs for real-time traffic, i.e., SF^{D}_{rt}. Last but not least, the simplified algorithm can be used in TDD mode as it has the same performance as the full version,

TABLE VI: Summary of the evaluation results

Scenario	**Best VoIP latency**	**Best elastic data rate**	**Best data rate / latency trade-off**
7 e2NBs hexagonal 1 → 2	Baseline Alg. in FDD	Full Alg. in TDD	Full Alg. in TDD
7 e2NBs hexagonal 4 → 6	Baseline Alg. in FDD	Full Alg. in TDD	Full Alg. in FDD
7 e2NBs hexagonal 1 → 2 & 4 → 6	Baseline Alg. in FDD	Full Alg. in FDD	Full Alg. in FDD
7 e2NBs line 2 → 6	Baseline Alg. in FDD	Full Alg. in FDD	Full Alg. in FDD
7 e2NBs line 7 → 5	Baseline Alg. in FDD	Full Alg. in FDD	Full Alg. in FDD
7 e2NBs line 1 → 2 & 7 → 5	Baseline Alg. in FDD	Full Alg. in FDD	Full Alg. in FDD
19 e2NBs hexagonal no hop restriction	Full Alg. in FDD	Simplified Alg. in FDD	Full Alg. in FDD
19 e2NBs hexagonal two hops restriction	Full Alg. in TDD	Full Alg. in FDD	Full Alg. in TDD

We have observed that the FDD mode performs generally better than the TDD mode thanks to a higher number of available MBSFN SFs for the self-backhauling. It is also easier to manage HARQ process and other feedback reports in FDD mode as explained in section V and VI. We have also seen that the conservativeness of the DLS on the applied AMC strategy is of importance to well utilize the UL direction in FDD mode. However, the TDD mode stays of interest as it is the only one able to match the most stringent requirements on frequency resource availability.

E. Potential improvements of the proposed approach

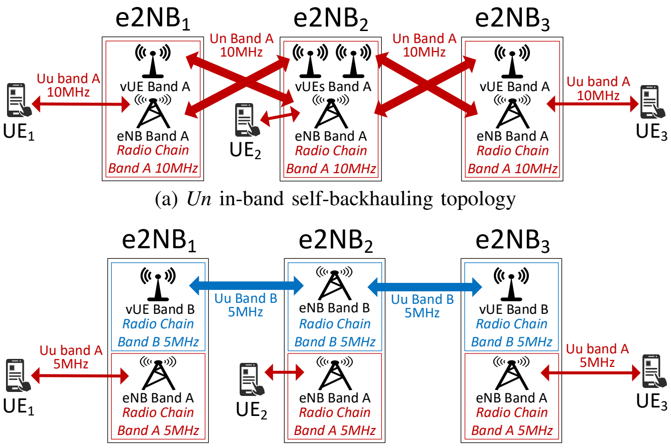
Based on all observations delivered in this section, we can further sum up some potential improvements of our proposed approach to further enhance its performance. Firstly, an inefficient UL channel estimation for the vUE along the backhaul links can degrade the performance in FDD mode. This calls for more conservative UL scheduler or applying more sophisticated interference avoidance schemes.

Moreover, the analysis showed that there is a side effect of the currently proposed computation of SF_e^D and SF_e^U at the *COE controller* that is memory-less. Such memory-less characteristic will make the *COE controller* take a longer durations to re-recognize the saturation condition in the end-to-end data transportation. Since once we try to resolve such saturation phenomena via allocating more resources (i.e., SFs) to these links through SF_e^D and SF_e^U, the *COE controller* will think the saturation condition is resolved immediately. Hence, it will not allocate anymore resources until such link is saturated again; nevertheless, such oscillations behavior has been observed using other types of elastic flows. We can propose for instance two solutions to overcome this issue. Firstly, SF_e^D and SF_e^U can be computed as they are, but the values effectively used by other algorithms can be a moving average over the current value and previous ones. Secondly, instead of directly setting SF_e^D and SF_e^U to a specific value, we can change their formulations to include the maximum change in each value update to avoid any drastic changes. Both approaches can better smooth the variations of the SFs allocation for the backhaul link and prevent the observed oscillations.

We also showed that limiting the number of hops for VoIP flows can ensure a better performance on the network, for both call quality and elastic flow data rate while not restricting it could impact negatively on every traffic flows in the mesh. The flow control algorithm can take such condition into account with other metrics to preserve the expected network behavior. Otherwise, we can include such condition when doing the handover process to make the source UE closer in the number of hops toward its destination.

IX. Performance Comparison of In-Band and Out-Band Deployment

Apart from aforementioned in-band deployment, the out-band deployment is another possible approach for self-backhauling network. We discussed in section III-B that the out-band backhaul was not complying with bandwidth limited scenarios as it requires at least another dedicated frequency band that is separated from the band for access link. Furthermore, we can also show that in-band self-backhauling is more flexible and provides better throughput performance than out-band deployment.



(a) *Un* in-band self-backhauling topology

(b) *Uu* out-band self-backhauling topology

Fig. 26: Network topology for in/out-band comparison.

Hence, we compare these two possible deployments in terms of the cumulated data rate using the full proposed scheduling algorithm. Here, we consider a simple network topology consisting of three BSs in a line topology. In the in-band case shown in Fig. 26(a), each e2NB shares the same frequency band for both *Uu* and *Un* interfaces with 10MHz channel bandwidth in FDD mode. Whereas the e2NBs of the out-band case in Fig. 26(b) requires two radio chains for two different FDD bands each with 5MHz radio bandwidth each: one is used to serve UEs on Band A using the *Uu* interface, and another one for the backhaul links on Band B relying on the *Uu* interface. Due to the out-band characteristic of the second case, there is no interference between the access and backhaul links; hence, two separated link schedulers are applied individually. However, the in-band case can rely on our proposed algorithms in order to allocate SFs for both access link and backhaul link that can span a whole 10MHz radio bandwidth.

In the first part of comparison, we consider three scenarios each with individual elastic traffic flow: (i) From $e2NB_1$ to $e2NB_2$, (ii) From $e2NB_1$ to $e2NB_3$, and (iii) From UE_1 to $e2NB_1$. The data rates among three scenarios are shown in Fig. 27 and we can observe a higher data rate on both backhaul and access links of the in-band case. Such result is due to the inefficiency bandwidth utilization when dividing the whole 10MHz radio bandwidth into two fixed 5MHz ones for both *Uu* interfaces of out-band deployment. Indeed, *Un* interface can utilize at maximum 6 MBSFN SFs per frame for the backhaul links (the actual number of SFs and the use of DL or UL directions is managed by the COE). On a 10MHz channel bandwidth, it gives roughly 20% more throughput than using 10 UL SFs per frame on a 5MHz channel bandwidth as done by the *Uu* backhaul. With a traffic flow transmitted from $e2NB_1$ to $e2NB_3$ over two backhaul hops, the data rate over *Uu* backhaul does not change from the $e2NB_1$ to $e2NB_2$ case, as it is limited by the UL direction from $e2NB_1$ to $e2NB_2$ (PRBs reserved for PUCCH makes it slightly slower than DL). On the other hand, *Un* backhaul links perform slightly worse in $e2NB_1$ to $e2NB_3$ case than in $e2NB_1$ to $e2NB_2$ because it has to rely on UL SFs for half of its transmissions while it mainly use DL SFs in the latter case. When transmitting only a single flow from UE_1 to $e2NB_1$, the *Un* backhaul

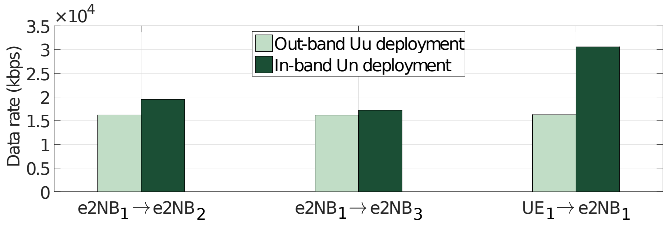
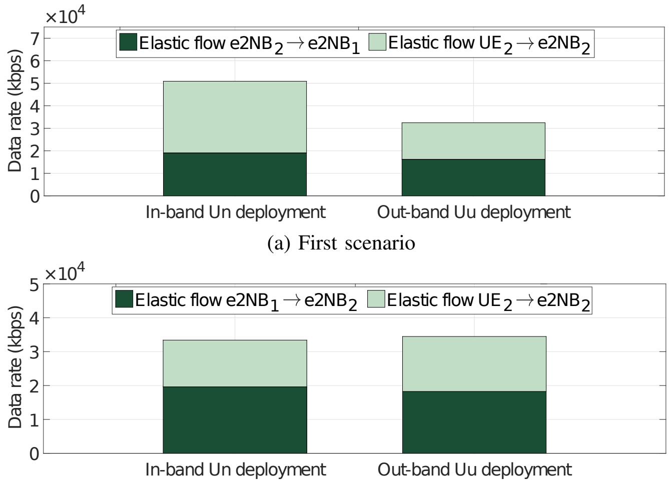


Fig. 27: Data rate of traffic flow of in/out-band deployments.



(a) First scenario

(b) Second scenario

Fig. 28: Cumulated data rate of two traffic flows.

architecture performs much better as almost all the 10MHz bandwidth can be used by the UE_1 for the UL (almost no SF are reserved for backhaul links as there is no traffic) while the *Uu* case is limited to a 5MHz UL channel bandwidth.

Furthermore, we compare two scenarios each with two concurrent traffic flows: (i) from UE_2 to $e2NB_2$ and from $e2NB_2$ to $e2NB_1$, and (ii) from UE_2 to $e2NB_2$ and from $e2NB_1$ to $e2NB_2$. In Fig. 28a, the cumulated data rate is shown for the first scenario and we can observe a higher data rate for the in-band case. In that case, most of the MBSFN SFs are allocated as backhaul DL SFs for $e2NB_2$. As there is no traffic coming from other e2NBs, UE_2 gets the full UL bandwidth (10MHz UL for local UEs), which allows the in-band *Un* deployment to outperform the out-band *Uu* one (5MHz UL for local UEs). Such result shows $e2NB_2$ can properly transmit (i.e., to $e2NB_1$) and receive (i.e., from UE_2) at the same time, and our proposed approach can efficiently schedule all available resources. Moreover, the cumulated date rate of the second scenario is shown in Fig. 28b and we can see that both deployments show close performance. Such phenomena is due to the concurrent reception at $e2NB_2$ from both $e2NB_1$ and UE_2. While the in-band *Un* solution allows more flexibility in sharing the radio resources between backhaul and access links than the out-band *Uu* solution, it is still limited by the FDD characteristic that the radio chain do 50% TX and 50% RX. As most MBSFN SFs are allocated to $e2NB_1$ for backhauling in DL direction, the $e2NB_2$ has on average only 4 UL SFs available for its UE. This reduces the UL data rate and the global throughput on the network slightly behind the *Uu* case that has complete backhaul/access separation.

To sum up, the in-band deployment can properly and more efficiently utilize available resource over access link and backhaul link than the out-band deployment. These results further convince that our proposed algorithm can perform an efficient scheduling for in-band self-backhauling network.

X. Conclusion

In this article, we propose a new BS architecture that evolves from the legacy LTE BS to enable new 5G use cases found in PS, military communication, and more generally in autonomous moving cells/vehicular scenarios. Detailed description of main building blocks from physical layer, procedures and algorithms to applications are provided. To specifically answer the dynamic resource allocation among access and backhaul links, we present a cross-layer hierarchical approach based on the proposed architecture to efficiently allocate shared resources considering QoS requirements. Furthermore, we analyze and show the feasibility and reliability of our proposed architecture via implementations and experiments on OAI platform. We also evaluate the effectiveness of the proposed hierarchical approach to satisfy QoS requirements of real-time traffic flows while improving the network throughput of elastic flows. Finally, the results reveals the relevance and applicability of our proposed approach in creating autonomous mobile LTE mesh networks.

Acknowledgments

Research and development leading to these results have received funding from *Naval Group* and the European Framework Program under H2020 grant agreement 671639 for the COHERENT project.

References

[1] ITU-R, "IMT vision - framework and overall objectives of the future development of IMT for 2020 and beyond," Tech. Rep.

[2] ETSI, "300 392-1, v1. 4.1; Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D)," 2009.

[3] TIA, "Project 25 system and standards definition," *TIA/EIA Telecommunications Systems Bulletin, TSB102-A*, 1995.

[4] K. Gomez *et al.*, "Enabling disaster-resilient 4G mobile communication networks," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 66–73, December 2014.

[5] P. Fraga-Lamas *et al.*, "Evolving military broadband wireless communication systems: Wimax, lte and wlan," in *2016 International Conference on Military Communications and Information Systems (ICMCIS)*, May 2016, pp. 1–8.

[6] B. Farsund *et al.*, "Lte for military communication - business models and vulnerabilities," in *2017 19th International Conference on Advanced Communication Technology (ICACT)*, Feb 2017, pp. 64–71.

[7] L. J. Su and D. Mavrakis, "Private networks for the mining industry," Technical Analysis Report, 2017. [Online]. Available: https://www.abiresearch.com/market-research/product/1028392-private-networks-for-the-mining-industry/

[8] R. Favraud *et al.*, "Toward moving public safety networks," *IEEE Communications Magazine*, vol. 54, no. 3, pp. 14–20, March 2016.

[9] *Isolated Evolved Universal Terrestrial Radio Access Network (E-UTRAN) operation for public safety; Stage 1*, 3GPP Technical Specification 22.346, Rev. 13.0.0, 2014.

[10] *Study on architecture enhancements to support isolated Evolved Universal Terrestrial Radio Access Network (E-UTRAN) operation for public safety*, 3GPP Technical Report 23.797, Rev. 13.0.0, 2015.

[11] A. Al-Hourani and S. Kandeepan, "Cognitive relay nodes for airborne lte emergency networks," in *2013, 7th International Conference on Signal Processing and Communication Systems (ICSPCS)*, Dec 2013, pp. 1–9.

[12] P. Bhat *et al.*, "Lte-advanced: an operator perspective," *IEEE Communications Magazine*, vol. 50, no. 2, pp. 104–114, February 2012.

[13] R. Favraud and N. Nikaein, "Wireless mesh backhauling for LTE/LTE-A networks," in *IEEE MILCOM 2015*, Oct 2015, pp. 695–700.

[14] R. Favraud *et al.*, "QoS Guarantee in Self-Backhauled LTE Mesh Networks," in *2017 IEEE Global Communications Conference (GLOBECOM)*, Dec 2017, pp. 1–7.

[15] D. Câmara and N. Nikaein, *Wireless Public Safety Networks 2: A Systematic Approach*. Elsevier, 2016.

[16] *Study on LTE device to device proximity services; Radio aspects*, 3GPP Technical Report 36.843, Rev. 12.0.1, 2014.

[17] *Feasibility study for Proximity Services (ProSe)*, 3GPP Technical Report 22.803, Rev. 12.2.0, 2013.

[18] Y. Sui *et al.*, "Moving cells: a promising solution to boost performance for vehicular users," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 62–68, June 2013.

[19] R. Fantacci *et al.*, “Public safety networks evolution toward broadband: Sharing infrastructures and spectrum with commercial systems,” *IEEE Communications Magazine*, vol. 54, no. 4, pp. 24–30, 2016.

[20] R. Ferrus *et al.*, “LTE: the technology driver for future public safety communications,” *IEEE Communications Magazine*, vol. 51, no. 10, pp. 154–161, October 2013.

[21] R. A. Pitaval *et al.*, “Full-duplex self-backhauling for small-cell 5g networks,” *IEEE Wireless Communications*, vol. 22, no. 5, pp. 83–89, October 2015.

[22] Nokia, “LTE networks for public safety services,” White Paper, 2014.

[23] *Telecommunication management; Self-Organizing Networks (SON); Concepts and requirements*, 3GPP Technical Specification 32.500, Rev. 8.0.0, 2008.

[24] S. Sesia *et al.*, *LTE-the UMTS long term evolution: from theory to practice*. John Wiley & Sons, 2011.

[25] O. Teyeb *et al.*, “Dynamic relaying in 3gpp lte-advanced networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, no. 1, p. 731317, Sep 2009.

[26] P. Kela *et al.*, “Flexible backhauling with massive mimo for ultra-dense networks,” *IEEE Access*, vol. 4, pp. 9625–9634, 2016.

[27] D. Bladsj *et al.*, “Synchronization aspects in LTE small cells,” *IEEE Communications Magazine*, vol. 51, no. 9, pp. 70–77, September 2013.

[28] M. A. Lombardi *et al.*, “Time and frequency measurements using the global positioning system,” *Cal Lab: International Journal of Metrology*, vol. 8, no. 3, pp. 26–33, 2001.

[29] M. A. Lombardi, “The use of GPS disciplined oscillators as primary frequency standards for calibration and metrology laboratories,” *NCSLI Measure*, vol. 3, no. 3, pp. 56–65, 2008.

[30] *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures*, 3GPP Technical Specification 36.213, Rev. 12.5.0, 2015.

[31] *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer for relaying operation*, 3GPP Technical Specification 36.216, Rev. 10.3.1, 2011.

[32] *Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification*, 3GPP Technical Specification 36.331, Rev. 8.0.0, 2007.

[33] *Telecommunication management; Automatic Neighbour Relation (ANR) management; Concepts and requirements*, 3GPP Technical Specification 32.511, Rev. 8.0.0, 2008.

[34] *Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-configuring and self-optimizing network (SON) use cases and solutions*, 3GPP Technical Report 36.902, Rev. 9.0.0, 2009.

[35] *Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) procedures in idle mode*, 3GPP Technical Specification 36.304, Rev. 8.10.0, 2011.

[36] J. Baliosian and R. Stadler, “Distributed auto-configuration of neighboring cell graphs in radio access networks,” *IEEE Transactions on Network and Service Management*, vol. 7, no. 3, pp. 145–157, September 2010.

[37] I. F. Akyildiz *et al.*, “Wireless mesh networks: A survey,” *Comput. Netw. ISDN Syst.*, vol. 47, no. 4, pp. 445–487, Mar. 2005.

[38] P. H. Pathak and R. Dutta, “A survey of network design problems and joint design approaches in wireless mesh networks,” *IEEE Communications Surveys Tutorials*, vol. 13, no. 3, pp. 396–428, Third 2011.

[39] A. BenMimoune *et al.*, “Dynamic joint resource allocation and relay selection for 5g multi-hop relay systems,” *Telecommunication Systems*, vol. 66, no. 2, pp. 283–294, Oct 2017.

[40] X. Foukas *et al.*, “FlexRAN: A Flexible and Programmable Platform for Software-Defined Radio Access Networks,” in *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT ’16. New York, NY, USA: ACM, 2016, pp. 427–441.

[41] Huawei, “The second phase of LTE-Advanced,” White Paper, 2013.

[42] “OpenAirInterface,” http://www.openairinterface.org.

[43] R. Favraud and N. Nikaein, “Analysis of LTE Relay Interface for Self-backhauling in LTE Mesh Networks,” in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Sept 2017, pp. 1–7.

[44] N. Nikaein, “Processing Radio Access Network Functions in the Cloud: Critical Issues and Modeling,” in *Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services*, ser. MCS ’15. New York, NY, USA: ACM, 2015, pp. 36–43.

[45] *Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects*, 3GPP Technical Report 36.814, Rev. 9.0.0, 2010.

[46] *Radio Frequency (RF) system scenarios*, 3GPP Technical Report 25.942, Rev. 10.1.0, 2012.

[47] *Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Frequency (RF) system scenarios*, 3GPP Technical Report 36.942, Rev. 10.3.0, 2012.

[48] A. F. Ribadeneira, “An analysis of the MOS under conditions of delay, jitter and packet loss and an analysis of the impact of introducing piggybacking and reed solomon FEC for VoIP,” Master’s thesis, Georgia State University, 2007.