TÉLÉCOM ParisTech - École Nationale Supérieure de Télécommunications de Paris

THÈSE

Présentée pour obtenir le grade de docteur

de l'École Nationale Supérieure des Télécommunications

Spécialité : Informatique et Réseaux

Presente par :

Muhammad Farukh MUNIR

Cross-Layer Optimizations of Wireless Sensor and Sensor-Actuator Networks

Souténu publiquement le 26 Fevrier 2009 devant le jury composé de :

Président :	Elie	NAJM	Telecom ParisTech, France
Rapporteurs :	Michel	DIAZ	CNRS, France
	Congduc	PHAM	Universite de Pau, France
Examinateurs :	Isabelle	GUÉRIN-LASSOUS	Universite de Lyon 1
	Mischa	DOHLER	CTTC, Barcelona, Spain
Directeur de thèse :	Fethi	FILALI	EURECOM, France

(14:30 - EURECOM Sophia-Antipolis)

Optimisation inter-couche de reseaux de capteurs et capteurs-actionneurs sans-fil

Muhammad Farukh MUNIR

Cross-Layer Optimizations of Wireless Sensor and Sensor-Actuator Networks

Dédicaces

À ma femme et ma famille...

Remerciements

Je remercie mon directeur de these Fethi Filali pour avoir accepte mon travail, pour son aide precieuse, technique et morale, et pour ca grande patience durant toutes les phase de cette these.

Je remercie HEC (Higher Education Comission, Pakistan) pour leur support financiere.

Je remercie egalement tous les doctorants et tous les personnel de EURECOM pour leur sympathie et la bonne ambiance qu'ils generent au sein de l'institut.

Enfin, ce travail n'aurait pas pu etre accomplie son l'amour et le soutien de toute ma famille, ma femme, mes parents, mes soeurs, et mon frere.

Muhammad Farukh MUNIR
Sophia-Antipolis
26 February 2009

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General Introduction

Recent advances in micro-electro-mechanical systems (MEMS) technology, wireless communications, and digital electronics have enabled the development of low-cost, low-power, multifunctional sensor nodes that are small in size and communicate untethered in short distances. Wireless Sensor Networks (WSNs) consist of large number of distributed sensor nodes that organize themselves into a multihop wireless network as shown in Figure 1.1. Each node has one or more sensors, embedded processors, and low-power radios, and is normally battery operated. Typically, these nodes coordinate to perform a common task. These tiny sensor nodes, which consist of sensing, data processing, and communicating components, leverage the idea of WSNs based on collaborative effort of a large number of nodes. WSNs represent a significant improvement over traditional sensors, which are deployed in the following two ways [1]:

- Sensors can be positioned far from the actual phenomenon, i.e., something known by sense perception. In this approach, large sensors that use some complex techniques to distinguish the targets from environmental noise are required.
- Several sensors that perform only sensing can be deployed. The positions of the sensors and communications topology can be carefully engineered. They transmit time series of the sensed phenomenon to the central nodes where computations are performed and data are fused. The central entity is shown as sink in Figure 1.1. It can be placed anywhere depending upon the application needs.

A sensor network is composed of a large number of sensor nodes, which are densely deployed either inside the phenomenon or very close to it. The position of sensor nodes need not be engineered or pre-determined. This allows random deployment in inaccessible terrains or disaster relief operations. On the other hand, this also means that sensor network protocols and algorithms must possess *self-organizing* capabilities. Another unique feature of WSNs is the *cooperative* effort of sensor nodes. Sensor nodes are fitted with an on-board processor. Instead of sending the raw data to the nodes responsible for the fusion, sensor nodes use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data.

The above described features ensure a wide range of applications for WSNs. Some of the application areas are health, military, environment, civil, and security. For example, the

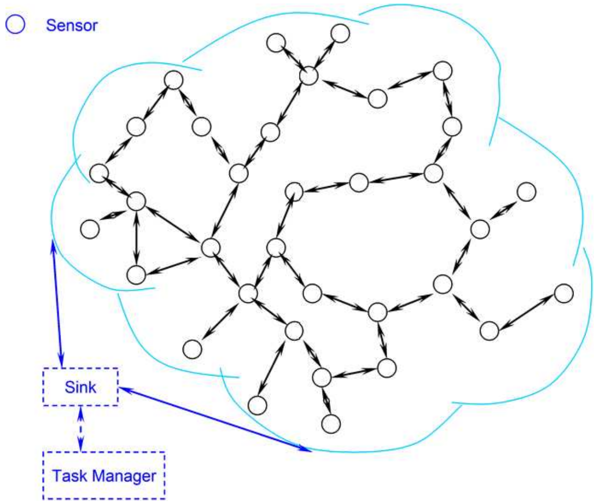

Figure 1.1: A Wireless Sensor Network

physiological data about a patient can be monitored remotely by a doctor. While this is more convenient for the patient, it also allows the doctor to better understand the patient's current condition. WSNs can also be used to detect foreign chemical agents in the air and the water. They can help identify the type, concentration, and location of pollutants. In essence, WSNs can provide the end user with intelligence and a better understanding of the environment. We envision that, in future, WSNs will be an *integral* part of our lives, more so than the present-day personal computers.

These low power and lossy networks (LLNs) are made up of many embedded devices with limited power, memory, and processing resources. They are interconnected by a variety of links, such as IEEE 802.15.4, Bluetooth, Low Power WiFi, wired or other low power PLC (Powerline Communication) links. LLNs are transitioning to an end-to-end IP-based solution to avoid the problem of non-interoperable networks interconnected by protocol translation gateways and proxies. Existing routing protocols such as OSPF, IS-IS, AODV, and OLSR have been evaluated by the IETF ROLL [2] working group and have in their current form been found to *not* satisfy all of the specific WSN routing requirements. The group is currently working on the standardization of routing funtionality for the specific requirements posed by LLNs.

Wireless sensor-actuator networks[1] (SANETs) are among the most addressed research fields in the area of information and communication technologies (ICT) these days, in the US, Europe and Asia. SANETs are composed of possibly a large number of tiny, autonomous sensor devices and actuators[2] equipped with wireless communication capabilities as shown in Figure 1.2. One of the most relevant aspects of this research field stands in its multidisciplinarity and the broad range of skills that are needed to approach their design. Theory of control systems is involved, networking, middleware, application layer issues are relevant, joint consideration of hardware and software aspects is needed, and their use can range from biomedical to industrial or automotive applications, from military to civil environments, etc. Distributed

[1] In related literature, the term WSANs (Wireless Sensor-Actor Networks) is also used to represent the same.

[2] In relevant literature, the term 'actor' is used to represent the same, i.e., a device that has both communication and actuation capabilities.

systems based on networked sensors and actuators with embedded computation capabilities enable an instrumentation of the physical world at an unprecedented scale and density, thus enabling a new generation of monitoring and control applications. SANETs are an emerging technology that has a wide range of potential applications including environment monitoring, medical systems, robotic exploration, and smart spaces. SANETs are becoming increasingly important in recent years due to their ability to detect and convey real-time, in-situ information for many civilian and military applications.

Each sensor node has one or more sensors (including multimedia, e.g., video and audio, or scalar data, e.g., temperature, pressure, light, infrared, and magnetometer), embedded processors, low-power radios, and is normally battery operated. An actuator is a device to convert an electrical control signal to a physical action, and constitutes the mechanism by which an agent acts upon the physical environment. From the perspective considered in this thesis, however, an actuator, besides being able to act on the environment by means of one or several actuators, is also a network entity that performs networking-related functionalities, i.e., receive, transmit, process, and relay data. For example, a robot may interact with the physical environment by means of several motors and servo-mechanisms (actuators). However, from a networking perspective, the robot constitutes a single entity, which is referred to as actuator. Hence, the term actuator embraces heterogeneous devices including robots, unmanned aerial vehicles (UAVs), and networked actuators such as water sprinklers, pan/tilt cameras, robotic arms, etc. Applications of SANETs may include team of mobile robots that perceive the environment from multiple disparate viewpoints based on the data gathered by a sensor network, a smart parking system that redirects drivers to available parking spots, or a distributed heating, ventilating, and air conditioning (HVAC) system based on wireless sensors.

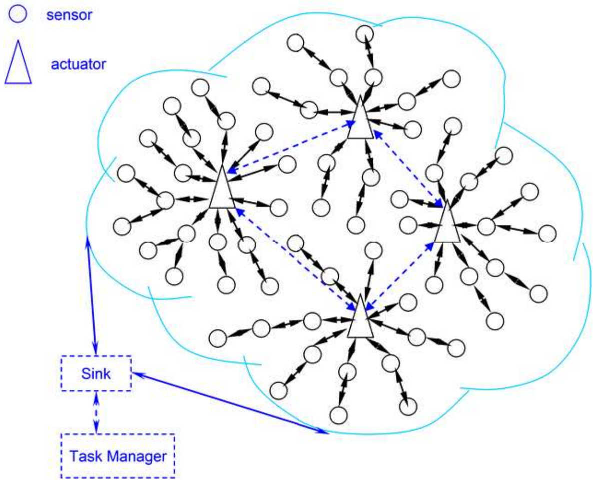

Figure 1.2: A Wireless Sensor-Actuator Network

However, due to the presence of actuators, SANETs have some differences from WSNs as outlined below:

- While sensor nodes are small, inexpensive devices with limited sensing, computation and wireless communication capabilities, actuators are usually resource-rich devices equipped with better processing capabilities, stronger transmission powers and longer battery life.

- In SANETs, depending on the application there may be a need to *rapidly* respond to sensor input. Moreover, to provide right actions, sensor data must still be valid at the time of acting. Therefore, the issue of *real-time* communication is very important in SANETs since actions are performed on the environment after sensing occurs. Examples can be a fire application where actions should be initiated on the event area as soon as possible.

- The number of sensor nodes deployed in studying a phenomenon may be in the order of hundreds or thousands. However, such a dense deployment is not necessary for actuator nodes due to the different coverage requirements and physical interaction methods of acting task. Hence, in SANETs the number of actuators is much lower than the number of sensors.

- In order to provide effective sensing and acting, a distributed local coordination mechanism is necessary among sensors and actuators. In WSNs, the central entity (i.e., sink) performs the functions of data collection and coordination. Whereas, in SANETs, new networking phenomena called sensor-sensor, sensor-actuator, and actuator-actuator coordination may occur. In particular, sensor-sensor coordination deals with local collaboration among neighbors to perform in-network *aggregation* and exploit *correlations* (both spatial and temporal). Sensor-actuator coordination provides the transmission of event features from sensors to actuators. After receiving event information, actuators may need to coordinate (actuator-actuator coordination) with each other (depend on the acting application) in order to make decisions on the most appropriate way to perform the actions.

Many protocols and algorithms have been proposed for WSNs in recent years [3]. However, since the above listed requirements impose *stricter* constraints, they may not be well-suited for the inherent features and application requirements of SANETs. Moreover, although there has been some research effort related to SANETs, to the best of our knowledge, almost none of the existing studies to date investigate research challenges occurring due to the *co-existence* of sensors and actuators.

Ocean bottom sensor nodes are deemed to enable applications for oceanographic data collection, pollution monitoring, offshore exploration, disaster prevention, assisted navigation and tactical surveillance applications. Multiple Unmanned or Autonomous Underwater Vehicles (UUVs, AUVs), equipped with underwater sensors, will also find application in exploration of natural undersea resources and gathering of scientific data in collaborative monitoring missions. To make these applications viable, there is a need to enable underwater communications among underwater devices. Underwater sensor nodes and vehicles must possess *self-configuration* capabilities, i.e., they must be able to coordinate their operation by exchanging configuration, location and movement information, and to relay monitored data to an onshore station.

Wireless underwater acoustic networking is the enabling technology for these applications. Underwater Acoustic Sensor Networks (UASN) consist of a variable number of sensors and vehicles that are deployed to perform collaborative monitoring tasks over a given area. To achieve this objective, sensors and vehicles self-organize in an autonomous network which can adapt to the characteristics of the ocean environment [5].

Underwater networking is a rather unexplored area although underwater communications have been experimented since World War II, when, in 1945, an underwater telephone was developed in the United States to communicate with submarines. Acoustic communications are the typical physical layer technology in underwater networks. In fact, radio waves propagate at long distances through conductive sea water only at extra low frequencies (30-300 Hz), which require large antennae and high transmission power. Optical waves do not suffer from such high *attenuation* but are affected by *scattering*. Moreover, transmission of optical signals requires high *precision* in pointing the narrow laser beams. Thus, links in underwater networks are based on acoustic wireless communications.

The traditional approach for ocean-bottom or ocean column monitoring is to deploy underwater sensors that record data during the monitoring mission, and then recover the instruments. This approach has the following disadvantages:

- Real time monitoring is not possible. This is critical especially in surveillance or in environmental monitoring applications such as seismic monitoring. The recorded data cannot be accessed until the instruments are recovered, which may happen several months after the beginning of the monitoring mission.

- No interaction is possible between onshore control systems and the monitoring instruments. This impedes any adaptive tuning of the instruments, nor is it possible to reconfigure the system after particular events occur.

- If failures or misconfigurations occur, it may not be possible to detect them before the instruments are recovered. This can easily lead to the complete failure of a monitoring mission.

- The amount of data that can be recorded during the monitoring mission by every sensor is limited by the capacity of the onboard storage devices (memories, hard disks, etc.).

Therefore, there is a need to deploy underwater networks that will enable real time monitoring of selected ocean areas, remote configuration and interaction with onshore human operators. This can be obtained by connecting underwater instruments by means of wireless links based on acoustic communication.

Many researchers are currently engaged in developing networking solutions for terrestrial WSNs. Although there exist many recently developed network protocols for WSNs, the unique characteristics of the underwater acoustic communication channel, such as limited bandwidth capacity and variable delays, require for very efficient and reliable new data communication protocols. The quality of the underwater acoustic link is highly unpredictable, since it mainly depends on *fading* and *multipath*, which are not easily modeled *phenomena*. This, in return, severely degrades the performance at higher layers such as extremely long and variable *propagation* delays. In addition, this variation is generally larger in *horizontal* links than in *vertical* ones. Acoustic signaling for wireless digital communications in the sea environment can be a very *attractive* alternative to both radio telemetry and cabled systems. However, time-varying multipath and often harsh ambient noise conditions *characterize* the underwater acoustic channel, often making acoustic communications challenging. Major challenges in the design of UASNs are:

- The channel is severely *impaired*, mainly due to *multipath*.

- Temporary loss of *connectivity* mainly due to *shadowing*.

- The *propagation delay* is five orders of magnitude higher than in radio frequency *terrestrial* channels and is usually variable [4].
- Extremely low available *bandwidth*.
- Limited *battery* energy at disposal.

Since underwater monitoring missions can be extremely expensive due to the high cost involved in underwater devices, it is important that the deployed network be highly reliable, so as to avoid failure of monitoring missions due to failure of single or multiple devices. For example, it is crucial to avoid designing the network topology with single points of failure that could compromise the overall functioning of the network. The network capacity is also influenced by the network topology. Since the capacity of the underwater channel is severely limited, it is very important to organize the network topology such a way that no communication bottlenecks are introduced.

1.2 Applications

The range of applications of WSNs, SANETs, and UASNs are increasing very fast and covering several domains: military, civil, environmental, health, etc. In this section, we will talk more about such applications in each of these domains [6].

1.2.1 Military Applications

- Asset Monitoring: commanders can monitor locations of the troops, weapons and supplies to enhance the control and communication.
- Battlefield Monitoring: vibration and magnetic sensors can locate and track enemy forces in the battlefield.
- Urban Warfare: deploying sensors in cleared buildings can prevent their reoccupation and track the enemy activity inside them.
- Protection: prevention and protection from radiations, biological and chemical weapons can be achieved by the deployment of a WSN, which detects the level of radiation or the presence of toxic products.
- Distributed Tactical Surveillance: AUVs and fixed underwater sensors can collaboratively monitor areas for surveillance, reconnaissance, targeting and intrusion detection systems. For example, in [7], a 3D underwater sensor network is designed for a tactical surveillance system that is able to detect and classify submarines, small delivery vehicles (SDVs) and divers based on the sensed data from mechanical, radiation, magnetic and acoustic microsensors. With respect to traditional radar/sonar systems, UASNs can reach a higher accuracy, and enable detection and classification of low signature targets by also combining measures from different types of sensors.
- Mine Reconnaissance: The simultaneous operation of multiple AUVs with acoustic and optical sensors can be used to perform rapid environmental assessment and detect mine-like objects.

1.2.2 Civil Applications

- Surveillance: a sensor network can detect fire in buildings and give information about its location. It can also detect intrusions and track human activity.
- Disaster Prevention: sensor nodes deployed under water can prevent from disaster like oceanic earthquake or impending tsunami.
- Smart Metering Solutions: smart metering solutions, provided by coronis, based on wavenis [8] wireless technology have been deployed in millions of residential, industrial and commercial installations around the world, linking consumers' gas, water and electricity meters efficiently with operator's back-end information and billing systems. These advanced solutions are used for wireless walk-by, drive-by and fully automated fixed network metering. Wavenis wireless technology provides the ultra-long range and extremely low power consumption that are essential for effective last-mile, outdoor coverage in metering networks that serve entire cities, including dense urban areas as well as sprawling suburban and commercial zones.
- Assisted Navigation: sensors can be used to identify hazards on the seabed, locate dangerous rocks or shoals in shallow waters, mooring positions, submerged wrecks, and to perform bathymetry profiling.
- Disaster Recovery: after an earthquake or a terrorist attack, sensor nodes can detect signs of life inside a damaged building.
- Smart Park: a distributed control system supported by SANET. It improves mobility in the urban area by finding free parking spots for drivers willing to park [9, 10]. It also decreases the risk of possible accidents, pollution, and eliminate road rage.

1.2.3 Environmental Applications

- Environment and Habitat Monitoring: a WSN deployed in a sub-glacial environment [11, 12] can collect information about ice caps and glaciers. WSNs can also be deployed to measure population of birds and other species [13]. Also,WSN can provide a flood warning [14] and monitor coastal erosion [15].
- Disaster Detection: forest fire can be detected and localized by a densely deployed WSN.
- Ocean Sampling Networks: networks of sensors and AUVs, such as the Odyssey-class AUVs [16], can perform synoptic, cooperative adaptive sampling of the 3D coastal ocean environment [17]. Experiments such as the Monterey Bay field experiment [18] demonstrated the advantages of bringing together sophisticated new robotic vehicles with advanced ocean models to improve the ability to observe and predict the characteristics of the oceanic environment.
- Environmental Monitoring: UASNs can perform pollution monitoring (chemical, biological and nuclear). For example, it may be possible to detail the chemical slurry of antibiotics, estrogen-type hormones and insecticides to monitor streams, rivers, lakes and ocean bays (water quality in-situ analysis) [19]. Monitoring of ocean currents and winds, improved weather forecast, detecting climate change, understanding and predicting the effect of human activities on marine ecosystems, biological monitoring such as

tracking of fishes or micro-organisms, are other possible applications. For example, in [20], the design and construction of a simple underwater sensor network is described to detect extreme temperature gradients (thermoclines), which are considered to be a breeding ground for certain marine microorganisms.

- Undersea Explorations: UASNs can help detecting underwater oilfields or reservoirs, determine routes for laying undersea cables, and assist in exploration for valuable minerals.
- Disaster Prevention: WSNs that measure seismic activity from remote locations can provide tsunami warnings to coastal areas [21], or study the effects of submarine earthquakes (seaquakes).
- Forest Fire Detection: a SANET could be deployed to detect a forest fire in its early stages [22]. A number of nodes need to be pre-deployed in a forest. Each node can gather different types of information from sensors, such as temperature, humidity, pressure and position. All sensing data is sent by multi-hop communication to the control centre via a number of actuators (gateway devices) distributed throughout the forest. The actuators will be connected to mobile networks (e.g., Universal Mobile Telecommunications System – UMTS) and will be positioned so as to reduce the number of hops from source of fire detection to the control centre. The actuators will also reduce network congestion in large-scale deployments by extracting data from the network at predetermined points. It may also be possible in this scenario that some mobile forest patron units act as mobile actuator, collecting environmental data as they traverse through the forest. As soon as a fire-related event is detected, such as sudden temperature rise, the control centre will be alarmed immediately. Operators in the control centre can judge if it is a false alarm by either using the data collected from other sensors or dispatching a team to check the situation locally. Then both firefighters and helicopters can be sent to put out the fire before it grows to a severe forest fire.

1.2.4 Medical Applications

- Home Monitoring: home monitoring for chronic and elderly patients [23] allows long-term care and can reduce the length of hospital stay.
- Patient Monitoring: sensor nodes deployed on the body of patients in hospitals [24] allow the collection of periodic or continuous data like temperature, blood pressure, etc.

1.3 Motivations and Objectives

WSNs are similar to *ad hoc* networks in the sense that sensor networks borrow heavily on the *self-organizing* and *routing* technologies developed by the ad-hoc research community. However, a major design *objective* for sensor networks is reducing the cost of each node. For many applications, the desired cost for a wirelessly enable device is less than one dollar.

We, in this thesis, consider a set of sensors spread over a region to perform sensing operation. Each of these sensors has a wireless transceiver that transmits and receives at a single frequency, which is common to all these sensors. Over time, some of these sensors generate/collect information to be sent to some other sensor(s). Owing to the limited battery capacity of these sensors, a sensor may not be able to directly communicate with far away nodes. In such scenarios, one of the possibilities for information transfer between two nodes that cannot communicate directly is to use other sensor nodes in the network. To be precise, the source sensors transmits its information to one of the sensors which is within its transmission range. The intermediate sensor then uses the same procedure so that the information finally reaches its destination (a fusion center, i.e., a common sink[3]).

A set comprising of ordered pair of nodes constitute a *route* that is used to assist communication between any two given pair of nodes (i.e., a sensor and a sink). This is a standard problem of *multihop* routing in WSNs. The problem of optimal routing has been extensively studied in the context of wireline networks where usually a shortest path routing algorithm is used: each link in the network has a *weight* associated with it and the objective of the routing algorithm is to find a path that achieves the minimum weight between two given nodes. Clearly, the outcome of such an algorithm depends on the assignment of weights associated to each link in the network. In wireline context, there are many well-studied criteria to select these weights for links such as the queueing delay. In WSNs, the optimality in the routing algorithm is set to extend network lifetime (where lifetime is defined as the time spanned by the network for some data aggregation till first alive node gets disconnected due to energy outage) in a single sink network. In networks with multiple sinks [25], the flow is splitted and sent to different basestations with the aim of extending the network lifetime of these limited battery WSNs. *However, a complete understanding of the effect of routing on WSNs performance and resource utilization (in particular, the stability of transmit buffers and hence, the end-to-end delay and throughput) has not received much attention.*

After sensors in the sensor/actuator field detect a phenomenon, they either transmit their readings to the resource-rich actuator nodes which can process all incoming data and initiate appropriate actions, or route data back to the sink which issues action commands to actuators. We use the former case in this thesis. The advantage is that the information sensed is conveyed quickly from sensors to actuators, since they are close to each other. Moreover, since event information is only transmitted locally through sensor nodes, only sensors around the event area are involved in the communication process which results in energy and bandwidth savings in SANETs.

If the mapping between a sensor node and one (or more) actuator[4] is given *a priori*, then the problem of finding optimal minimum energy routes to optimize network lifetime has been well investigated in the past [25, 26] for WSNs. But, there is very little research contribution toward finding *optimal delay* routes in SANETs. Further, in cases when there are multiple actuators and mapping between the sensors and actuators is not given, the joint problem of finding a destination actuator and minimum end-to-end delay routes is a challenging and interesting problem. This is because the end-to-end delays are topology dependent; actuator selection based on minimum hop routing alone can not guarantee optimal end-to-end delays.

Further, in order to provide effective sensing and acting tasks, efficient coordination mechanisms are required. We will mainly focus on two *most* constrained coordination levels namely: sensor-actuator coordination, and actuator-actuator coordination. The most important characteristic of sensor-actuator coordination is to provide low communication delay due to the proximity of sensors and actuators. However, since the role of sink does not involve collecting the sensor data and coordinating the activities of the nodes, sensor and actuator nodes should

[3] By a fusion center or a common sink, we mean a logical destination for data. This can be located anywhere in or outside the network topology.

[4] actuators/basestations are considered to have similar semantics for modeling purposes, i.e., sinks for data generated in the network.

locally coordinate with each other so as to provide efficient transmission of sensor readings. *In SANETs, for sensor-actuator coordination there is a need to develop protocols which are able to provide real-time services with given delay bounds, according to application constraints and ensure an energy efficient communication among sensors and actuators.* In SANETs, actuators can communicate with each other in addition to communicating with sensors. Since there are few number of actuator nodes and the power capacities of these nodes are higher than sensor nodes, actuator-actuator communication is similar to the communication in wireless ad-hoc networks. Actuator-actuator coordination can occur in the cases where the actuator receiving sensor data may not act on the event area due to small action range or insufficient energy, where one actuator may not be sufficient to perform the required action, thus other nearby actuators should be triggered, where multiple actuators receive the same event information and there is an action threshold, hence these actuators should "talk" to each other so as to decide which one of them performs the action and where multiple events occur simultaneously. *Thus actuator should coordinate and communicate with each other to perform task allocation efficiently and effectively.*

We also consider a SANET that prolongs *network lifetime* by minimizing the *energy consumption* and, in parallel, takes care of *delay-sensitivity* of the sensed data. Therefore, in cases, where there are multiple actuators and mapping between the sensors and actuators is not given, the problem of finding an optimal actuator and extending network lifetime with minimum end-to-end delay constraints is an interesting problem. This problem is relevant from both the application's and wireless networking perspectives. From an application requirement perspective, for some real-time multimedia sensing applications (e.g., video surveillance), it is necessary to have all the traffic generated from a source sensor to be routed to the same actuator (albeit that it may follow different routes) so that decoding and processing can be properly completed. For multimedia traffic such as video, the information contained in different packets from the same source are highly correlated and dependent. If the packets generated by a source are split and sent to different actuators, any of these receiving actuators may not be able to decode the video packets properly. From a wireless networking perspective, the actuator chosen as a sink could have a significant impact on the end-to-end delays which is a *hard* constraint [27] for sensor-actuator applications. *As a result, there appears to be a compelling need to understand how to perform optimal routing to jointly achieve minimum end-to-end delay routes and optimize network lifetime in delay-energy constrained SANETs.*

Apart from SANETs, we also consider UASNs which are deployed to perform collaborative underwater monitoring tasks. The sensors must be organized in an autonomous network that self-configure according to the varying characteristics of the ocean environment. Most impairments of the underwater acoustic channel are adequately addressed at the physical layer, by designing receivers that are able to deal with high bit error rates, fading, and the inter-symbol interference (ISI) caused by multipath. There were efforts at developing channel equalizers and adaptive spatial processing techniques so that coherent phase modulation can be used to achieve the desired high spectral efficiencies. These techniques are computationally demanding with many parameter adjustments, and requirements that are not especially suitable for applications where autonomy, adaptability, and long-life battery operation are being contemplated. *Therefore, we analyze the factors that influence acoustic communications in order to state the challenges posed by the underwater channels for underwater sensor networking.*

1.4 Thesis Outline and Contributions

In Chapter 2, we consider a WSN in which the sensor nodes are sources of delay sensitive traffic that needs to be transferred in a multi-hop fashion to a common processing center. We consider the following data sampling scheme: the sensor nodes have a sampling process independent (layered architecture) of the transmission scheme as shown in Figure 1.3. This system is like the packet radio network (PRN) for which exact analysis is not available. We also show that the stability condition proposed in the PRN literature is not accurate. First, a correct stability condition for such a system is provided. Then, we proposed a *cross-layered* data sampling scheme in which, the sensor nodes sample new data only when it has a opportunity (cross-layered architecture) of transmitting the data as shown in Figure 1.3. It is also observed that this scheme gives a better performance in terms of delays and is moreover amenable to analysis.

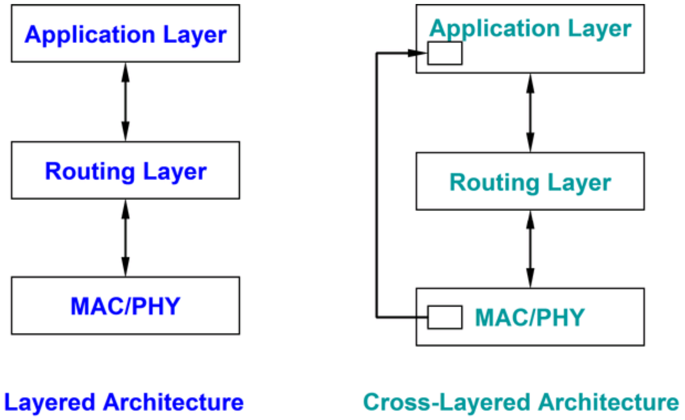

Figure 1.3: A Layered and Cross-Layered Architecture

To provide meaningful service such as disaster and emergency surveillance, meeting real-time and energy constraints and the stability at medium access control (MAC) layer are the basic requirements of communication protocols in such networks. We also propose a cross-layer architecture with two transmit queues at MAC layer, i.e., one for its own generated data, and the other for forwarding traffic as shown in Figure 1.4. We use a probabilistic queueing discipline. Our first main result concerns the stability of the forwarding queues at the nodes. It states that whether or not the forwarding queues can be stabilized, by appropriate choice of weighted fair queueing (WFQ) weights, depends only on *routing* and *channel access rates* of the sensors. Further, the weights of the WFQs play a role in determining the *tradeoff* between the *power* allocated for forwarding and the *delay* of the forwarding traffic.

We then address the problem of *optimal* routing that aims at minimizing the end-to-end delays. Since, we allow for traffic splitting at source nodes, we propose an algorithm that seeks the Wardrop equilibrium instead of a single least delay path. Wardrop equilibria first appeared in the context of transportation networks. Wardrop's first principle states: The journey times in all routes actually used are equal and less than those which would be experienced by a single vehicle on any unused route. Each user *non-cooperatively* seeks to minimize his cost of transportation. The traffic flows that satisfy this principle are usually referred to as "user equilibrium" (UE) flows, since each user chooses the route that is the best. Specifically, a *user-optimized* equilibrium is reached when no user may lower his transportation cost through

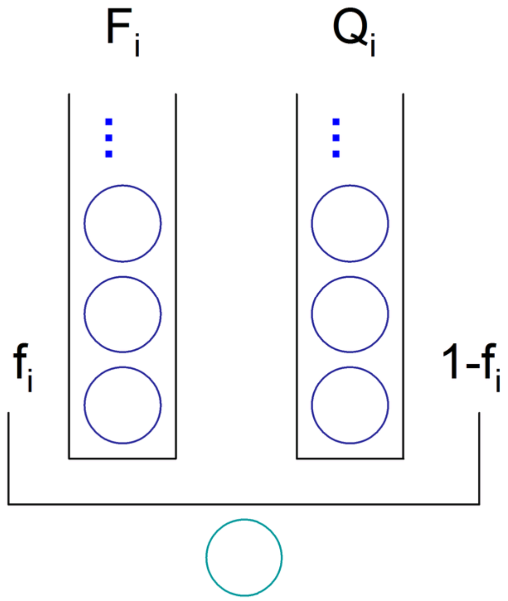

Figure 1.4: A System with Two-Queues at MAC

unilateral action.

The distributed routing scheme is designed for a broad class of WSNs which converges (in the Cesaro sense) to the set of Cesaro-Wardrop equilibria. Each link is assigned a weight and the objective is to route through minimum weight paths using iterative updating scheme. Convergence is established using standard results from the related literature and validated by TinyOS simulation results. Our algorithm can adapt to changes in the network traffic and delays. The scheme is based on the multiple time-scale stochastic approximation algorithms. The algorithm is simulated in TOSSIM and numerical results from the simulations are provided.

In Chapter 3, we consider a two-tier SANET and address the minimum delay problem for data aggregation. We analyze the *average* end-to-end delay in the network. The objective is to minimize the *total* delay in the network. We prove that this objective function is strictly *convex* for the entire network. We then provide a *distributed* optimization framework to achieve the required objective. The approach is based on distributed convex optimization and deterministic distributed algorithm without feedback control. Only local knowledge is used to update the algorithmic steps. Specifically, we formulate the objective as a network level delay minimization function where the constraints are the *reception-capacity* and *service-rate* probabilities. Using the Lagrangian dual composition method, we derive a distributed primal-dual algorithm to minimize the delay in the network. We further develop a *stochastic delay-control* primal-dual algorithm in the presence of noisy conditions. We also present its convergence and rate of convergence properties.

This chapter also investigates a *delay-optimal* actuator-selection problem for SANETs. Each sensor must transmit its locally generated data to only one of the actuators. A polynomial time algorithm is proposed for delay-optimal actuator-selection. We finally propose a distributed mechanism for *actuation control* which covers all the requirements for an effective actuation process.

In Chapter 4, we consider a three-tier SANET and present the design, implementation, and performance evaluation of a novel low-energy, adaptive and distributed (LEAD) *self-organization* framework. This framework provides coordination, routing, and MAC layer protocols for network organization and management. The framework is shown in Figure 1.5. We organize the heterogeneous SANET into *clusters* where each cluster is managed by an actuator. To *maximize* the network lifetime and attain minimum end-to-end delays, it is essential to optimally match each sensor node to an actuator and find an optimal routing scheme. We provide an actuator discovery protocol (ADP) that finds out a destination actuator for each sensor in the network based on the outcome of a *cost* function. Further, once the destination actuators are fixed, we provide an *energy-optimal* routing solution with the aim of maximizing network lifetime. We then propose a delay-energy aware TDMA based MAC protocol in compliance with the routing algorithm. The actuator-selection, optimal routing, and TDMA MAC schemes together guarantees a *near-optimal* lifetime. The proposal is validated by means of analysis and ns-2 simulation results.

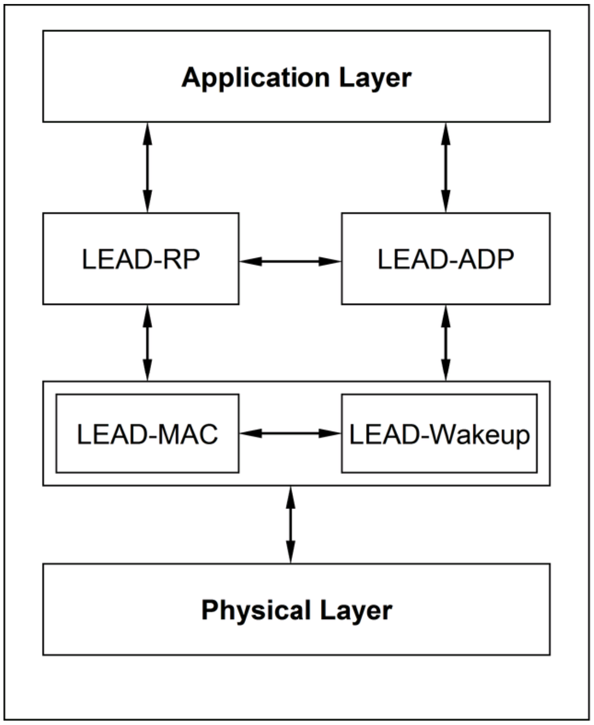

Figure 1.5: The LEAD Framework

Delay and energy constraints have a significant impact on the design and operation of SANETs. Furthermore, preventing sensor nodes from being inactive/isolated is very critical. The problem of sensor inactivity/isolation arises from the *pathloss* and *fading* that degrades the quality of the signals transmitted from actuators to sensors, especially in *anisotropic* deployment areas, e.g., rough and hilly terrains. Sensor data transmission in SANETs heavily relies on the scheduling information that each sensor node receives from its associated actuator. Therefore, if the signal containing scheduling information is received at a very low power due to the *impairments* introduced by the wireless channel, the sensor node might be unable to decode it and consequently it will remain inactive/isolated.

Sensors transmit their readings to the actuators. All actuators cooperate and jointly transmit scheduling information to sensors with the use of beamforming. This results in an important reduction of the number of inactive sensors comparing to single actuator transmission for a given level of transmit power. The reduction is due to the resulting *array gain* and the exploitation of *macro-diversity* that is provided by the actuator cooperation. In order to maximize network lifetime and attain minimum end-to-end delays, it is essential to optimally match each sensor node to an actuator and find an optimal routing solution. A distributed solution for optimal actuator selection subject to energy-delay constraints is also provided.

In Chapter 5, we consider a UASN and first analyze a modulation scheme and associated receiver algorithms. This receiver design take advantage of the *time reversal*[5] (TR) and properties of spread spectrum sequences known as Gold sequences. Furthermore, they are much less complex than receivers using *adaptive* equalizers. This technique improves the signal-to-noise ratio (SNR) at the receiver and reduces the bit error rate (BER). We then applied the phase conjugation to network communication. We show that this approach can give almost *zero* BER for a two-hop communication mode compared to the traditional *direct* communication. This link layer information is used at the network layer to *optimize* routing decisions. We show these improvements by means of analytical analysis and simulations.

In Chapter 6, we present a general summary of the work achieved and the conclusions concerning the results obtained during this thesis. Some perspectives and open questions are given for the continuation of this work in the area of cross-layer optimizations in wireless sensor, sensor-actuator, and underwater acoustic sensor networks.

[5] It is also known as phase conjugation (PC) in the frequency domain

Chapter 2

Cross-Layer Routing in WSNs

In this Chapter, we consider a WSN in which the sensor nodes are sources of delay sensitive traffic that needs to be transferred in a multi-hop fashion to a common processing center. We first consider the layered architecture. This system is like PRNs for which exact analysis is not available in the literature. We also show that the stability condition proposed in the PRN literature is not accurate. First, a correct stability condition for such a system is provided. We then propose a new data sampling scheme: the sensor nodes sample new data only when it has an *opportunity* (cross-layered) of transmitting the data. It is observed that this scheme gives a better performance in terms of *delays* and moreover is amenable to *analysis*.

We also propose a closed (cross-layered) architecture with two transmit queues at each sensor i, i.e., one for its own generated data, and the other for forwarding traffic. Our first main result concerns the stability of the forwarding queues at the nodes. It states that whether or not the forwarding queues can be stabilized (by appropriate choice of WFQ weights) depends only on *routing* and *channel access rates* of the sensors. Further, the weights of the WFQs play a role in determining the *tradeoff* between the *power* allocated for forwarding and the *delay* of the forwarding traffic.

We then address the problem of *optimal* routing that aims at *minimizing* the end-to-end delays. Since we allow for traffic splitting at source nodes, we propose an algorithm that seeks the Wardrop equilibrium (i.e., the delays on the routes that are actually used by the packets from a source are all minimum and equal) instead of a single least delay path. Each link is assigned a weight and the objective is to route through *minimum* weight paths using *iterative* updating scheme. The algorithm is implemented in TinyOS Simulator (TOSSIM) and numerical results from the simulation are provided.

2.1 Introduction

WSNs are an emerging technology that has a wide range of potential applications including environment monitoring, medical systems, robotic exploration, and smart spaces. WSNs are becoming increasingly important in recent years due to their ability to detect and convey real-time, in-situ information for many civilian and military applications. Such networks consist of large number of distributed sensor nodes that organize themselves into a multihop wireless network. Each node has one or more sensors, embedded processors, and low-power radios, and is normally battery operated. Typically, these nodes coordinate to perform a common task.

We propose a closed (cross-layered) architecture for data sampling (application layer) in a wireless sensor network. In this architecture, there is a strong coupling between the sampling process and the channel access scheme as shown in Figure 1.3. The objective in the closed architecture is to provide *sufficient* and *necessary* conditions for the stability region and reducing end-to-end delays. With mathematical analysis and simulations, we show that the closed architecture outperforms the traditional layered scheme, both in terms of stable operating region as well as the end-to-end delays.

We also propose a closed architecture with two transmit queues for data sampling in a wireless sensor network. In this architecture, we consider a new data sampling scheme: Node i, $1 \leq i \leq N$, has two queues associated with it: one queue Q_i contains the data sampled by the sensor node itself and the other queue F_i contains packets that node i has received from any of its neighbors and has to be transmitted to another neighbor as shown in Figure 1.4. In this architecture, there is coupling between the sampling process and the channel access scheme. The objective in the closed architecture is to study the impact of channel access rates, routing, and weights of the WFQs on system performance.

We then propose an adaptive and distributed routing scheme for a general class of WSNs. The objective of our scheme is to achieve Cesaro Wardrop equilibrium, an extension of the notion of Wardrop equilibria that first appeared in [28] in the context of transportation networks. Wardrop's first principle states: The journey times in all routes actually used are equal and less than those which would be experienced by a single vehicle on any unused route. Each user non-cooperatively seeks to minimize his cost of transportation. The traffic flows that satisfy this principle are usually referred to as "user equilibrium" (UE) flows, since each user chooses the route that is the best. Specifically, a user-optimized equilibrium is reached when no user may lower his transportation cost through unilateral action. The notion is defined in (2.1) later in this chapter. Our algorithm is actually an adaptation of the algorithm proposed in [29] to the case of WSNs. In the algorithm of [29], each source uses a two time-scale stochastic approximation algorithm. Differences in the two algorithms are:

1. In WSNs that we consider, each node has an attribute associated with it namely the channel access rate. The delay on a route depends on the attributes of the nodes on the route. However, in order to maintain some long term data transfer rate, each node needs to adapt its attribute to routing.

2. The difference in time scales that we use for various learning/adaptation schemes helps us prove convergence of our algorithm [C-4] (such a proof is not present in [29]).

In this thesis, we consider a static wireless sensor network with n sensor nodes. Given is an $n \times n$ neighborhood relation matrix N that indicates the node pairs for which direct communication is possible. We will assume that N is a symmetric[1] matrix, i.e., if node i can transmit to node j, then j can also transmit to node i. For such node pairs, the $(i,j)^{th}$ entry of the matrix N is unity, i.e., $N_{i,j} = 1$ if node i and j can communicate with each other; we will set $N_{i,j} = 0$ if nodes i and j can not communicate. For any node i, we define

$$N_i = \{j : N_{i,j} = 1\},$$

Which is the set of neighboring nodes of node i. Similarly, the two hop neighbors of node i are defined as

[1] The assumption of symmetry is to only drive the analysis. We consider assymmetric links for conducting simulations.

$$S_i = \{k \notin N_i \cup \{i\} : N_{k,j} = 1 \, for\, some\, j \in N_i\}$$

Note that S_i does not include any of the first-hop neighbors of node i.

Each sensor node is assumed to be sampling (or, sensing) its environment at a predefined rate; we let λ_i denote this sampling rate for node i. The units of λ_i will be packets per second, assuming same packet size for all the nodes in the network. In this work, we will assume that the readings of each of these sensor nodes are statistically independent of each other so that distributed compression techniques are not employed (see [30] for an example where the authors exploit the correlation among readings of different sensors to use distributed Slepian-Wolf Coding [31] to reduce the overall transmission rate of the network).

Each sensor node wants to use the sensor network to forward its sampled data to a *common* fusion center (assumed to be a part of the network[2]). Thus, each sensor node acts as a forwarder of data from other sensor nodes in the network. We will assume that the buffering capacity of each node is infinite[3], so that there is no data loss in the network. We will allow for the possibility that a sensor node *discriminates* between its own packets and the packets to be forwarded (thus allowing for the model of [32] which considers an Ad Hoc network. The nodes in this network *probabalistically* schedule their transmissions to discriminate between the fowarding traffic and the one generated by node itself).

We let ϕ denote the $n \times n$ routing matrix. The $(i, j)^{th}$element of this matrix, denoted $\phi_{i,j}$, takes value in the interval $[0, 1]$. This means a probabilistic flow splitting as in the model of [33], i.e., a fraction $\phi_{i,j}$ of the traffic *transmitted* from node i is forwarded by node j as shown in Figrue 2.1. Clearly, we need that ϕ is a stochastic matrix, i.e., its row elements sum to unity. Also note that $\phi_{i,j} > 0$ is possible only if $N_{i,j} = 1$.

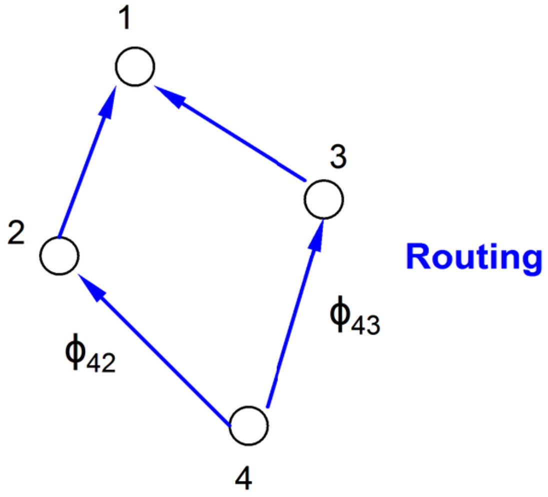

Figure 2.1: Flow Splitting

We assume that the system operates in discrete time, so that the time is divided into

[2] Conceptually, we can assume that this fusion center is also a sensor node, which has 0 sampling rate. A negative sampling rate would mean pushing data from the network towards the fusion center.

[3] We assume infinite buffer size only to keep the analysis simple. Later, we consider fixed buffer sizes and look at various types of data losses.

(conceptually) fixed length slots as shown in Figure 2.2. The system operates on CSMA/CA MAC[4]. Assuming that there is no exponential back-off, the channel access rate of node i (if it has a packet to be transmitted) is $0 \leq \alpha_i \leq 1$. Thus, α_i is the probability that node i, if it has a packet to be transmitted, attempts a transmission in any slot. A node can receive a transmission from its neighbor if it is not transmitting and also no other neighboring node is transmitting. Again, this is a fairly standard assumption for analysis purposes.

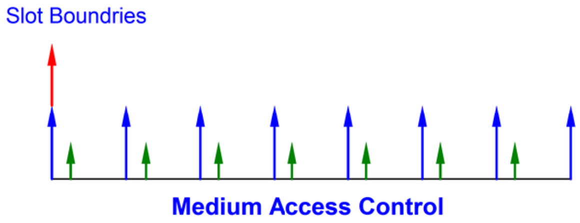

Figure 2.2: Medium Access Control

Under the above model there will be a delay, say $y_{j,i}$ of the packet from node j to be served at node i; this packet could have originated at node j or may have been forwarded by node j. The expected delay of a packet transmitted from node j is thus $\sum_{i \neq j} \phi_{j,i} y_{j,i}$. Since delays are additive over a path, packets from any node will have a delay over any possible route to the fusion center. A route will be denoted by an ordered set of nodes that occur on that route, i.e., the first element will be the source of the route, the last element will be the fusion center and the intermediate elements will be nodes arranged in the order that a packet traverses on this route. Let the total number of possible routes (cycle-free) be R. Let route i, $1 \leq i \leq R$ be denoted by the set $\mathcal{R}_i$ consisting of R_i elements with $\mathcal{R}_{i,j}$ denoting the j^{th}entry of this route. Then, a traffic splitting matrix will correspond to a Wardrop equilibrium iff for any i (see [29] for this definition)

$$\sum_{1 \leq j \leq R : \mathcal{R}_{j,1} = i} \left(\prod_{k=1}^{R_j - 1} \phi \mathcal{R}_{j,k}, \mathcal{R}_{j,k+1} \right) \left(\sum_{k=1}^{R_j - 1} y \mathcal{R}_{j,k}, \mathcal{R}_{j,k+1} \right) = \sum_{k=1}^{R_l - 1} y \mathcal{R}_{l,k} \mathcal{R}_{l,k+1}, \qquad (2.1)$$

for any l with $\mathcal{R}_{l,1} = i$ and such that $\prod_{k=1}^{R_l - 1} \phi \mathcal{R}_{l,k}, \mathcal{R}_{l,k+1} > 0$, i.e., the delays on the routes that are actually used by packets from node i are all equal. In simple terms, eq. (2.1) states that, for any given i, there will be a route that guarantees minimum delay. It is also possible that there is a set of routes that guarantee the same, then delay should be the same on all such routes. Our objective in this thesis is to come up with an algorithm using which any node (say i) is able to converge to the corresponding row of the matrix ϕ corresponding to the Wardrop equilibrium.

The organization of this chapter is as follows. Section 2.2 overviews some interesting related work. In Section 2.3, we formulated the problem. In Section 2.4, we detail the different data

[4] It is important to note that we consider CSMA/CA in order to provide analytical analysis of the system under consideration. Further, CSMA is also being used in IEEE 802.15.4 [34] (Zigbee). The endless list of available MACs for WSNs, is generally categorized into scheduled MACs (e.g. TSMP [35]), protocols with common active period (e.g. SMAC [36]), and preamble sampling based MACs (e.g. 1-HopMAC [37]). We will consider all these categories in the chapters to come. In this section, we focus only on CSMA part of IEEE 802.15.4. Our results can directly enhance the performance of WSNs that use 802.15.4 for multi-hop communciation.

collection mechanisms. Section 2.5 discusses the stability issues. We propose a distributed routing algorithm in Section 2.6. Simulation results from TinyOS simulations are presented in Section 2.7. In Section 2.8, we briefly conclude the chapter and outline the future directions.

2.2 Related Literature

In multihop packet radio networks, packets passed between two packet radio nodes may have to be relayed by intermediate nodes. In [33], Hamilton and Yu developed an optimal routing algorithm for slotted-ALOHA PRNs which minimizes the average packet delay. The packet radio sources serve as sources (and sinks) of traffic as well as repeaters which forward packets to other nodes. The optimal routing algorithm captures the important features of PRNs and avoids routes that result in high levels of interference and delay. The authors provide approximate analysis for multihop PRNs as the exact analysis requires modeling the entire network as a Markovian network of queues. Because of interference among nodes and enormous number of linear equations to be solved, the exact analysis is known to be mathematically intractable and details can be found in the references therein. The algorithm proposed is essentially similar to the minimum delay routing algorithm proposed by Gallagar in [38].

In [30], Cristescu et *al.* exploit the correlation among readings of different sensors to use distributed Slepian-Wolf Coding [31] to reduce the overall transmission rate of the network. They consider a set of correlated sources located at the nodes of a network, and a set of links that are the destinations for some of the sources. For the case of data gathering, the optimal transmission structure is fully characterized and a closed-form solution for the optimal rate allocation is provided. For the general case of an arbitrary traffic matrix, the problem of finding the optimal transmission structure is shown to be NP-complete.

In [32], Kherani et *al.* studied the throughput of multi-hop routes and stability of forwarding queues in a wireless Ad-Hoc network with random access channel. The main results include the characterization of stability condition and the end-to-end throughput using the balance. The impact of routing on end-to-end throughput and stability of intermediate nodes is also investigated. The authors showed that as long as the intermediate queues in the network are stable, the end-to-end throughput of a connection does not depend on the load on intermediate nodes. Some numerical results are also provided to support the results of the analysis.

A routing scheme for a broad class of networks which converges (in the Cesaro sense) with probability one to the set of approximate Cesaro-Wardrop equilibria, an extension of the notion of a Wardrop equilibrium [28] is analyzed in [29]. The routing algorithm is distributed, using only local information about observed delays by the nodes, and is moreover impervious to clock offsets at nodes. The scheme is also fully asynchronous, since different iterates have their own counters and the orders of packets and their acknowledgments may be scrambled. The scheme is adaptive to traffic patterns in the network. The demonstration of convergence in a fully dynamic context involves the treatment of two-time scales [49] distributed asynchronous stochastic iterations. Using an Ordinary Differential Equation (ODE) approach [50], the invariant measures are identified. A direct stochastic analysis shows that the algorithm avoids non-Wardrop equilibria.

The data collected by each sensor is communicated through the network to a single processing center that uses all reported data to determine characteristics of the environment or detect an event. The communication or message passing process must be designed to conserve the limited energy resources of the sensors. Clustering sensors into groups, so that sensors communicate information only to clusterheads and then the clusterheads communicate the aggregated information to the processing center, may save energy. In [39], a distributed, randomized clustering algorithm to organize the sensors in a WSN into clusters. It is then extended to generate a hierarchy of clusterheads and observe that the energy savings increase with the number of levels in the hierarchy. Results in stochastic geometry are used to derive solutions for the values of parameters of the algorithm that minimize the total energy spent in the network when all sensors report data through the clusterheads to the processing center.

Wireless sensor networks consist of small battery powered devices with limited energy resources. Once deployed, the small sensor nodes are usually inaccessible to the user, and thus replacement of the energy source is not feasible. Hence, energy efficiency is a key design issue that needs to be enhanced in order to improve the life span of the network. Several network layer protocols have been proposed to improve the effective lifetime of a network with a limited energy supply. In [40], a centralized routing protocol called Base-Station Controlled Dynamic Clustering Protocol (BCDCP) is proposed, which distributes the energy dissipation evenly among all sensor nodes to improve network lifetime and average energy savings. The performance of BCDCP is then compared to clustering-based schemes such as Low- Energy Adaptive Clustering Hierarchy (LEACH) [41] and Power-Efficient GAthering in Sensor Information Systems (PEGASIS) [42]. Simulation results show that BCDCP reduces overall energy consumption and improves network lifetime over its comparatives.

Clustering has proven to be an effective approach for organizing the network into a connected hierarchy. In [43], the authors highlight the challenges in clustering a WSN, discuss the design rationale of the different clustering approaches, and classify the proposed approaches based on their objectives and design principles. Several key issues that affect the practical deployment of clustering techniques in sensor network applications are also discussed. In [44], the authors propose a novel distributed clustering approach for long-lived ad hoc sensor networks. The approach does not make any assumptions about the presence of infrastructure or about node capabilities, other than the availability of multiple power levels in sensor nodes. A protocol, HEED (Hybrid Energy-Efficient Distributed clustering) is proposed, that periodically selects cluster heads according to a hybrid of the node residual energy and a secondary parameter, such as node proximity to its neighbors or node degree. HEED terminates in O(1) iterations, incurs low message overhead, and achieves fairly uniform cluster head distribution across the network. It is proved that, with appropriate bounds on node density and intracluster and intercluster transmission ranges, HEED can asymptotically almost surely guarantee connectivity of clustered networks. Simulation results demonstrate that our proposed approach is effective in prolonging the network lifetime and supporting scalable data aggregation.

[45] proposes two routing protocols: periodic, event-driven and query-based protocol (PEQ) and its variation CPEQ, two fault-tolerant and low-latency algorithms that meet sensor network requirements for critical conditions supervision in context-aware physical environments. While PEQ can provide low latency for event notification, fast broken path reconfiguration, and high reliability in the delivery of event packets for low-network data traffic, CPEQ is a cluster-based routing protocol that groups sensor nodes to efficiently relay the sensed data to the sink by uniformly distributing energy dissipation among the nodes and reducing latency for high-network data traffic (typical in emergency situations). PEQ and its variant CPEQ use the publish/subscribe paradigm to disseminate requests across the network. Both PEQ and CPEQ protocols are discussed, their implementation, and report on the performance results of several scenarios using NS-2 simulator. The results obtained are

compared with the well-known directed diffusion (DD) protocol [55], and show that the proposed algorithms exhibit a clear indication to meet the constraints and requirements of critical condition supervision in context-aware physical environments. The results indicate that PEQ outperforms DD in the average delay since it uses the shortest path for the delivery of packets and speed up new subscriptions by using the reverse path used for event notification packets. CPEQ also outperforms DD in both the average delay and in the packet delivery ratio when the network scales up.

A newly deployed multi-hop radio network is unstructured and lacks a reliable and efficient communication scheme. In [85], some steps were taken towards analyzing the problems existing during the *initialization* phase of ad hoc and sensor networks. Particularly, the network is modelled as a multi-hop quasi unit disk graph and allow nodes to wake up asynchronously at any time. Further, nodes do not feature a reliable collision detection mechanism, and they have only limited knowledge about the network topology. It is shown that even for this restricted model, a good clustering can be computed efficiently. The algorithm efficiently computes an asymptotically optimal clustering. Based on this algorithm, they describe a protocol for quickly establishing synchronized sleep and listen schedule between nodes within a cluster. Additionally, some simulation results are provided in a variety of settings. In [46], the authors have looked at the impact of clustering on the maintenance of a large-scale WSN, where node numbers are often in the thousands. Using some known scaling laws, they have determined that scalable protocols for flat sensor network topologies cannot exist and that clustering or hierarchical approaches ought to be used instead. They have also identified some approaches which may be useful in determining an optimum cluster size. Thereafter, they have identified crucial research problems at MAC and routing levels, as well as related to *auto-organization* and *self-healing* mechanisms. The concept of *virtual* prototyping has then been introduced: a tool which proves to be very useful in evaluating the reliability of designed protocols.

Geographic-based routing techniques are promising for WSNs, which suffer from severe energy constraints and a low throughput nature. In its simplest form, greedy geographic forwarding faces the problem of a low *delivery* ratio. Other protocols use the *right hand rule* to guarantee delivery. They nevertheless assume nodes know their exact position, whereas positioning systems offer only *limited* accuracy. In [47], the authors propose to use path-recording mechanisms, where nodes append their identifier to the header of the message, together with geographic forwarding. While yielding a comparable number of hops for a message to reach destination than existing routing protocols with guaranteed delivery, this technique offers *guaranteed* delivery regardless of the positioning accuracy. This makes path-recording particularly suitable for real-world WSN implementations. In [48], the authors free themselves from positioning techniques and anchor nodes altogether, and introduce and analyze the concept of virtual coordinates. These coordinates are chosen randomly when a node is switched on, and are updated each time the node relays a packet. As this process goes on, the virtual coordinates of the nodes converge to a near-optimal state. When using a greedy geographic approach on top of these coordinates, they show that the number of hops to reach the destination exceeds the shortest path by a few percent only. Moreover, the approach guarantees delivery even when nodes appear/disappear in the network, and under realistic transmission models.

Clustering techniques offer attractive solutions to routing problems in small scale WSNs. Since, the sensor nodes are limited by *maximum* transmit-power constraint, clustering solutions might not work well in large scale WSNs. This comes from the fact that *cluster heads* are assumed to communicate directly with each other for routing/relaying semantics, which only holds true if the deployment is on a small scale. In this work, we consider a general pupose WSN. we propose a cross-layered architecture for WSNs which outperforms the layered architecture, both in terms of stable operating region as well as end-to-end delays [J-1]. For data gathering, we *do not* use clustering techniques. Our routing algorithm is actually an adaptation of the algorithm proposed in [29] to the case of WSNs. In the algorithm of [29], each source uses a two time-scale stochastic approximation algorithm. In WSNs that we consider, each node has an attribute associated with it namely the channel access rate. The delay on a route depends on the attributes of the nodes on the route. However, in order to maintain some long term data transfer rate, each node needs to adapt its attributes to routing. The difference in time scales that we use for various learning/adaptation schemes helps us prove convergence of our algorithm. We deal with the implications, i.e., early *alive-node* deaths due to load, that arise due to our routing approach, in the later chapters.

2.3 Problem Statement

We consider a set of wireless sensors spread over a region to perform sensing operation. Each of these sensors has a wireless transceiver that transmits and receives at a single frequency which is common to all these sensors. Over time, some of these sensors generate/collect information to be sent to some other sensor(s). Owing to the limited battery capacity of these sensors, a sensor may not be able to directly communicate with far away nodes. In such scenarios, one of the possibilities for information transfer between two nodes that cannot communicate directly is to use other sensor nodes in the network. To be precise, the source sensor transmits its information to one of the sensors which is within its transmission range. The intermediate sensor then uses the same procedure so that the information finally reaches its destination (a fusion center, i.e., a common sink).

A set comprising of ordered pair of nodes constitute a *route* that is used to assist communication between any two given pair of nodes (i.e., a sensor and a sink). This is a standard problem of *multihop* routing in WSNs. The problem of optimal routing has been extensively studied in the context of wireline networks where usually a shortest path routing algorithm is used: each link in the network has a *weight* associated with it and the objective of the routing algorithm is to find a path that achieves the minimum weight between two given nodes. Clearly, the outcome of such an algorithm depends on the assignment of weights associated to each link in the network. In wireline context, there are many well-studied criteria to select these weights for links, e.g., the queueing delay etc. In WSNs, the optimality in the routing algorithm is set to extend network lifetime (where lifetime is defined as the time spanned by the network for some data aggregation till first node death due to energy outage) in a single sink network. In networks with multiple sinks [25], the flow is splitted and sent to different basestations with the aim of extending the network lifetime of these limited battery WSNs. *However, a complete understanding of effect of routing on WSN performance and resource utilization (in particular, the stability of transmit buffers and hence, the end-to-end delay and throughput) has not received much attention.*

2.4 Data Collection Mechanism

We consider three possibilities of data collection mechanism:

2.4.1 Open System (Layered Architecture)

This is the traditional slotted Aloha based system with a layered architecture where the application layer (sampling process in case of WSNs) does not directly interact with the lower layers (the random access MAC in our example).

In Section 2.4.5, we will see the issues with stability in the WSNs that use the slotted Aloha like random access mechanism for channel access with a sampling process without any communication with the MAC layer. Such schemes were extensively used in the PRN literature. The analysis of the model that we consider above is also available in the PRN literature (see for example [33]). The problem of stability that we will see is that for a given *sampling rate*, one needs to *jointly* optimize the *channel access rate* and the *routing* in order to optimize on *delays*. We will also see that the sampling rate at a node may be restricted by the sampling rate of the other *downlink* nodes. Further, in order to maintain stability of a node's transmit buffer, one needs to be operating far from the maximum allowed sampling rate (this is because, under the assumption of Bernoulli sampling process, the average queue length grows exponentially with an increase in the sampling rate). In addition, in this model, the sampling rate is not directly related to the channel access rates (unless it is an outcome of an optimization problem like the one we consider in Section 2.4.5). Thus, there is an extra dimension that needs to be optimally controlled.

2.4.2 Closed System (Cross-Layered Architecture) with Single Transmit Queue

Under this mechanism, there is a *strong* coupling between the channel access process and the sampling process. This approach has the advantage that one does not need to find an optimal sampling rate all over again on changing the channel access rates. The coupling automatically regulates the sampling process for any change in the channel access process.

The combined channel access/data sampling mechanism is as follows: Node i decides to attempt a channel access with probability α_i in any slot (else, it is sensing the channel for any possible transmissions). If decided to attempt a transmission, the node first checks if there is any packet available in its transmit queue. We have following possibilities:

1. No packets waiting in the transmit queue: In this case, the MAC layer of node i will ask the appropriate upper layer to sense data and provide it with a new packet. This packet is then attempted a transmission.

2. At least one packet waiting to be forwarded: In this case, node i will serve the head-of-line packet from its transmit queue.

Note that under this mechanism the transmit queue of node i can have at most one packet[5] in the transmit queue that was generated at node i. It can however have multiple packets in the transmit queue to be forwarded, i.e., those packets that were initially generated at some other node, and have arrived at node i to be forwarded to some other node. Clearly, under this scheme if the transmit queue of node i contains a packet that was generated at node i

[5] It is important to note that the Application layer samples packets at a predefined sampling rate. Therefore, some packets are discarded if there is forwarding traffic available at MAC layer. At this point, one might consider this as cheating since we optimize the delay by simply not producing packets when it is clear that they would need a lot of time to be transmitted. But, we provide an *optimization* framework in Section 2.6 for a node to bring its *transmission rate* as close to the *sampling rate* as possible.

itself, then this packet will be the head-of-line packet till the time it leaves the transmit queue of node i.

2.4.3 Applications for Closed System with Single Transmit Queue

The closed scheme is meant to be used in applications where a sensor network is used to observe the *time* variation of a *random* field over the space on which the network is *deployed*. For such applications, one can think of a *temporal* priority mechanism for transmitting packets so as to reduce the overall *transmissions* in the network. In particular, our sampling scheme amounts to the assumption that a node assigns *highest* priority to the most recent packet generated by the node (this priority is defined over the packets generated by the node, and *does not* include the packets that a node *receives* to forward to some of its neighbors).

2.4.4 Closed System with two Transmit Queues

In this section, we propose a closed architecture with two transmit queues at each sensor i, i.e., one for its own generated data, and the other for forwarding traffic [C-8]. We propose a closed architecture for data sampling (application layer) in a wireless sensor network. We consider a new data sampling scheme: Node i, $1 \leq i \leq N$, has two queues associated with it: one queue Q_i contains the data sampled by the sensor node itself and the other queue F_i contains packets that node i has received from any of its neighbors and has to be transmitted to another neighbor. In this architecture, there is again a coupling between the *sampling* process and the *channel access* scheme. The objective in the closed architecture is to study the impact of channel access rates, routing, and weights of the weighted fair queues on system performance. Furthermore, a distributed routing algorithm (which is allowed to split flows) is proposed that maintains the system at a Wardrop equilibrium and guarantees low delay.

In any slot, a node (provided it has packets to be transmitted) decides with a *fixed* probability to make a transmission attempt. If there is no other transmission by the sensors whose transmission can interfere with the one under consideration, the transmission is successful.

At any instant of time, a sensor may have two types of packets to be transmitted:

1. Packets sensed/generated by the sensor itself.
2. Packets from other neighboring sensors that arrived at this sensor and need to be forwarded.

Clearly, a sensor needs to have some scheduling policy to decide on which type of packet it wants to transmit, if it decided to transmit. A first come first served scheduling is one simple option. Another option that we would be considering in this chapter is to have two separate queues at each sensor node and do a weighted fair queueing for these two queues. In this chapter, we will also study the effect of channel access probability, weights of the weighted fair queueing, and routing on stability and fairness properties of the WSNs.

The closed system presented here is entirely different from the one in previous section. The combined channel access/data sampling mechanism is as follows: Node i decides to attempt a channel access with probability α_i in any slot (else, it is sensing the channel for any possible transmissions). If decided to attempt a transmission, the node first checks the number of packets available to be forwarded, i.e., packets from other nodes that node i is having to be forwarded to some of its neighbors. We have following possibilities:

1. Both F_i and Q_i are empty: In this case, the MAC layer of node i will ask the appropriate upper layer to sense data and provide it with a new packet[6]. This packet is then attempted a transmission.

2. At least one packet waiting to be forwarded: In this case, node i will do the following:

   (a) with probability $1 - f_i$, ask the appropriate upper layer to sense data and provide it with a new packet. This packet is then attempted transmission.

   (b) with probability f_i, forward the head-of-line packet waiting to be forwarded.

We assume that the queue Q_i is always nonempty, i.e., nodes make new measurements as soon as the older ones are transmitted. *Note* that this kind of model with assumption of saturated nodes are intended to provide insights into the performance of the system and also helps study effects of various parameters. This particular setting favours nodes that are far away from the sink as the probability of having an empty queue is larger than the nodes closer to the sink. They probably do not have any or a small of traffic to forward, this mechanism can help these nodes to sense the *extremeties* of the deployment area in a *better* way.

2.4.5 An Example

Consider a 4-node wireless sensor network shown in Figure 2.3. Node 0 is the common destination for all the data generated by the other three sensors, labeled $1, 2, 3$. All the transmission in the network is done only by these sensor nodes; the job of node 0 is to receive data sensed at these sensor nodes. To begin with, we assume that node 3 can not directly transmit to the destination node 0. Node 1 and 2 can communicate with node 0 but not with each other; Node 3 can communicate with both, node 1 and node 2.

The time is slotted and the sensor nodes use CSMA/CA like random access mechanism, supported in 802.15.4 [34], for transmission of their data; if node i has a packet to be transmitted, *it attempts a transmission in a slot with some given probability* α_i. We will assume that the packet *generation* process at node i is Bernoulli with packet generation probability[7] λ_i.

Node 1 and 2 transmit directly to node 0, but one has to decide on the path that packets from node 3 will follow. There are various options for this:

1. either all the packets generated at node 3 will be transmitted to node 1, or to node 2, or

2. for each packet transmitted by node 3, the next hop node is chosen randomly, for example, a packet transmitted from node 3 goes to node 1 with probability 0.3 and to node 2 with probability 0.7.

[6] The Application layer will only provide MAC layer with a packet, if some data has already been sensed that has just not reached MAC layer. Since, the time required to sense a *new packet* and its arrival to MAC queue *largely* exceeds one transmission slot time, it is not possible to *always* have *activity* in the network. For those applications, which require nodes to sleep, this mechanism works optimally in a sense that nodes wake up, *attempt* to transmit even if MAC queue is *empty*, and then go back to sleep.

[7] Such models were frequently used in context of Packet Radio Network (PRN) literature in the $70'$s and $80'$s, see for example [33]. We will see later that for WSNs where MAC layer can be allowed to control the application layer, one can achieve better results compared to those in PRNs where application layer operates independently of the MAC layer.

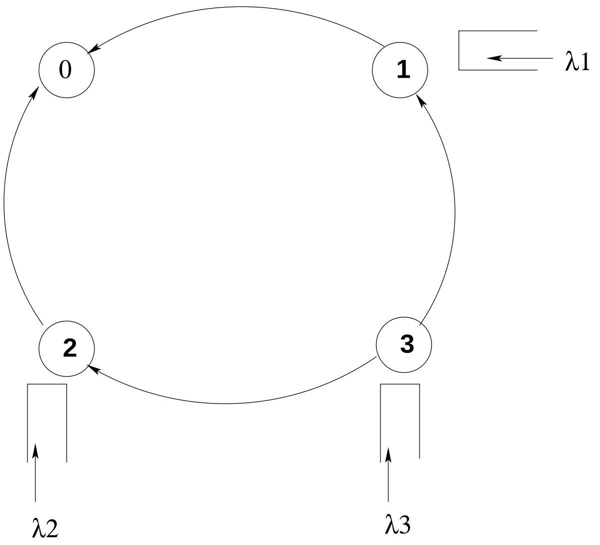

Figure 2.3: Network Configuration

3. for each packet transmitted by node 3, the next hop node is decided using some *cost functions* such as energy state of the receiving node, etc.

For this example, we will assume the first option (of course, it is a special case of the second option); we will allow for the second more general option when we come to optimal routing. Traffic splitting method as provided by the second option were also used in PRNs [33]. We will also assume that each packet from a node is attempted transmission till it is successfully received by the intended destination. A transmission is successfully received by a node if it is seeing no other transmission and the node is not transmitting. For cases where one allows for possibility of dropping a packet after it has incurred some number of collisions will not be considered in this chapter for simplicity; the relevant equations can be found in Section 2.5.3.

Assume that all the packets from node 3 use node 1 to reach node 0. In this case, let π_i denote *the steady-state probability that node i has a packet to be transmitted in a slot.* We can then write down the following *approximate* equations for the stable system (formal derivation of these equations can be found in [33]).

$$\begin{aligned} \pi_1\alpha_1(1-\pi_2\alpha_2) &= \lambda_1+\lambda_3 \\ \pi_2\alpha_2(1-\pi_1\alpha_1) &= \lambda_2 \\ \pi_3\alpha_3(1-\pi_1\alpha_1) &= \lambda_3 \end{aligned} \tag{2.2}$$

These equations are approximate because they are derived under a strong decoupling assumption. For stability of all the queues in the network, we need to choose α_i's such that the above system of equations (in π_i's) gives us a solution $(\pi_1, \pi_2, \pi_3) \in [0, 1)^3$. The stability condition under which above relations are valid are

$$\begin{aligned} \alpha_1(1-\alpha_2) &> \lambda_1+\lambda_3 \\ \alpha_2(1-\alpha_1) &> \lambda_2 \\ \alpha_3(1-\alpha_1) &> \lambda_3 \end{aligned} \tag{2.3}$$

Clearly, for a given sampling rate $\lambda_i, i = 1, 2, 3$, there will exist many possibilities of the channel access rates that give a stable system. These conditions are actually very different from the one proposed in [33]. In fact, a simple counter example can be given under which the conditions of [33] implies stability, while the system is not stable.

This system is not analytically tractable for the queueing delays. Various approximate analysis can be found in [33] and its references. Because of this reason, the extra degree of freedom that one gets in the parameter α_i is hard to utilize properly as the correct dependence of the system performance (for example, the queueing delays at various nodes) is not known. An instance of this difficulty is that the system of rate balance equations (2.2) are not valid for all values of α_i. In fact, the discrepancy between the actual system performance and that obtained from using (2.2) can be as large as 50%. The delay equations provided in [33] and references therein are based on (2.2) and for this reason, these expressions perform poorly for a broad range of parameter[8] α_i.

This is clear from the relations in (2.2), which implies that as long as the system is stable, we can solve the rate balance equations in (2.2). Since these equations depend on π_i and α_i only via $\pi_i\alpha_i$, in the stable region, this product $\pi_i\alpha_i$ will remain unchanged (w.r.t. changes in α_i). Hence we are *tempted* to conclude that there is an extra degree of freedom in α_i that can be employed *without* changing the *end-to-end delays*.

Further, this model was justified in the standard OSI-like model where one did not aim at cross-layer optimization and where the application layer (the sampled voice packets source) was not in control of the MAC layer. If one likes to minimize the expected delay on a node, one way would be to control the arrival of packets from node's own sensing mechanism. One such example that we will be considering (or, proposing) in this thesis is the following:

A sensor node gets a new packet from the application layer only if it decides to transmit in a slot but finds the transmit queue empty. As is the case with random access, sensor node i decides to attempt a transmission with probability α_i. We will call this system the *closed system* and the first system with layered architecture the *open system*.

For the Closed System model, the throughput of nodes 2 and 3 are

$$\begin{aligned} \lambda_2 &= \alpha_2\,(1-\alpha_1) \\ \lambda_3 &= \alpha_3\,(1-\alpha_1) \end{aligned} \tag{2.4}$$

Using these, the throughput of node 1 is

$$\lambda_1 = \alpha_1(1-\alpha_2) - \lambda_3 \tag{2.5}$$

The stability condition is

$$\alpha_1(1-\alpha_2) \geq \alpha_3(1-\alpha_1) \tag{2.6}$$

The expected number of packets at the three nodes are

[8] We remark here that our present observations are not aimed at questioning the significance of [33] and the related work from PRN literature. Most of these studies never aimed at tuning the parameters α_i, and since they assumed relatively small values of α_i which were fixed *a priori*, most of the time in their work the decoupling approximation leading to (2.2) was good. In our work, however we are trying to get the best system performance, hence need to tune the parameters α_i optimally, so that a correct/accurate analytical model is required for all possible values of α_i's .

$$\begin{aligned} T_1 &= \frac{\rho_0}{(1-\rho)(1-\rho+\rho_0)} \\ T_2 &= \alpha_1 \\ T_3 &= \alpha_1 \end{aligned} \tag{2.7}$$

where

$$\rho_0 = \frac{(1-\alpha_1)\alpha_3 + \alpha_1\alpha_2}{\alpha_1(1-\alpha_2)} \tag{2.8}$$

and

$$\rho = \frac{(1-\alpha_1)\alpha_3}{\alpha_1(1-\alpha_2)} \tag{2.9}$$

'The expected delay at each node are easily obtained using Little's Law as $D_i = \frac{T_i}{\lambda_i}$ for $i = 2, 3$ and $D_1 = \frac{T_1}{\lambda_1+\lambda_3}$. It is to be noted that these formula are exact, unlike those in the layered system, where the delay expression available in literature are approximate [33].

The mean node delay at the three nodes in the two systems for $\lambda_i = 0.1$ as obtained from discrete event simulations (for open system) and analysis (for closed system) are shown in Table 2.1 and 2.2. The mean delays for the three flows are thus obtained to be:

Table 2.1: Node level Delays

Node →	1	2	3
Open system	3.52	2.80	1.45
Closed system	0.56	2.30	2.30

Table 2.2: Flow level Delays

Flow →	1	2	3
Open System	3.52	2.80	4.97
Closed System	0.56	2.30	2.86

The mean delays for the closed system were obtained using simple formulae given before in (2.7). For the Open system, since the delay expressions available in literature are approximate, we developed a discrete event simulator to find these delays. The mean delay for the open system was obtained as follows: the simulation was run using different combinations of α_i spanning the stability region of the system. The delay vector provided here is the one which was closest to the origin in terms of Euclidean distance compared to all the other delay vectors obtained by varying α_i.

Observations from the toy example:

1. The values of α_i for Open system that gives the best performance are very large, thus implying waste of resources due to frequent collisions.

2. Clearly, the flow delay is significantly reduced in the Closed system, while using a moderate[9] value of α_i.

[9] These values are obtained using (2.7). By moderate, we mean, a value that is sufficient to cope with the given load on the transmit queue. Its detailed behavior will be presented in Section 2.7.

3. In Open system, one needs to tune the value of α_i in order to get the best delay performance; this may not of much use because the Closed system is giving better results compared to the best result from Open system. Thus, an optimization over α_i in the Open system is not justified. The exact delay expressions are not known. The approximate expressions used in literature are valid only for small values of α_i whereas the optimal point is obtained for large $\alpha_i^{\prime}s$, for which the available approximation has been shown to perform poorly.

For the Open system, we will assume a given set of channel access rates. We will see that the routing algorithm is able to select a good operating point that guarantees stability (as long as such a point *exists* for the given value of channel access probabilities).

2.5 Stability Analysis

2.5.1 Open System

We give the correct stability condition for the Open system (introduced in Section 2.4) where the data sampling process is independent of the transmission scheme.

Lemma 1: The minimum rate at which a node can serve its transmit queue is

$$\mu_i = \alpha_i \sum_{j \in N_i} \phi_{i,j} \left(1 - \alpha_j\right) \prod_{k \in N_j \backslash \{i\}} \left(1 - \alpha_k\right). \tag{2.10}$$

Proof: To prove *Lemma 1*, we consider a simple sensor network as shown in Figure 2.4.

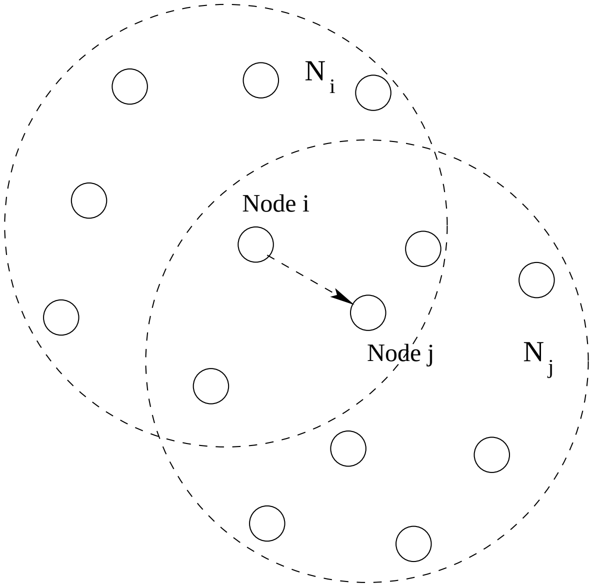

Figure 2.4: An example Network Topology

We define events:
A: node i transmits a packet in a given slot;
A^c: node i does not transmit a packet in a given slot;
B: node j transmits a packet in a given slot;

B^c: node j does not transmit a packet in a given slot;
C: node k transmits a packet in a given slot;
C^c: node k does not transmit a packet in a given slot.

The conditional probability that node i makes a transmission attempt and which is received correctly at node j is given by:

$$P(A \cap B^c \cap C^c)$$

$$= P\left(A\cap[\,B_1^c\left(\cap_k C_k^{c(1)}\right) + B_2^c\left(\cap_k C_k^{c(2)}\right) + \ldots + B_J^c\left(\cap_k C_k^{c(J)}\right)]\right)$$

for $1 \leq j \leq J$, where $C_k^{c(1)}\, means\ for\, node\, j = 1$, all the other k neighbors of j are not transmitting.

$$= P\left(A \cap \sum_j B_j^c\left(\cap_k C_k^{c(j)}\right)\right)$$

$$= P\left(A\right) P\left(\sum_j B_j^c\left(\cap_k C_k^{c(j)}\right)\right)$$

$$= P\left(A\right) \sum_{j\in N_i}\left(P\left(B_j^c\right) \prod_{k\in N_j\backslash\{i\}} P\left(C_k^c\right)\right)$$

$$= \alpha_i \sum_{j\in N_i} \phi_{i,j}\left(1-\alpha_j\right) \prod_{k\in N_j\backslash\{i\}} \left(1-\alpha_k\right).$$

where

$$P\left(A\right) = \alpha_i,$$

$$\sum_{j\in N_i} P\left(B_j^c\right) = \sum_{j\in N_i}\left(1-\alpha_j\right),$$

$$\prod_{k\in N_j\backslash\{i\}} P\left(C_k^c\right) = \prod_{k\in N_j\backslash\{i\}}\left(1-\alpha_k\right).$$

Also, $\phi_{i,j} > 0$, when $N_{i,j} = 1$. In practice, we need to verify that the probabilities $\phi_{i,j}$ are strictly positive for all the feasible routes to ensure that we are able to probe for a change in state of all the available routes.

Lemma 2: The minimum reception capacity of node i is

$$\gamma_i = \left(1-\alpha_i\right) \sum_{j\in N_i} \phi_{j,i}\alpha_j \prod_{k\in N_i\backslash\{j\}} \left(1-\alpha_k\right). \tag{2.11}$$

Proof: The reception capacity of a node i can be identified by the conditional probability

$$P\left(A^c \cap B \cap C^c\right)$$

$$= P\left(A^c\cap[\,B_1\left(\cap_k C_k^{c(i\backslash\{1\})}\right) + B_2\left(\cap_k C_k^{c(i\backslash\{2\})}\right) + \ldots + B_J\left(\cap_k C_k^{c(i\backslash\{J\})}\right)]\right)$$

for $1 \leq j \leq J$, where $C_k^{c(i\backslash\{1\})}\, means\, that\, for\, node\, j = 1$, all the other k neighbors of i are not transmitting.

$$= P\left(A^c \cap \sum_j B_j \left(\cap_k C_k^{c(i\backslash\{j\})}\right)\right)$$

$$= P\left(A^c\right) P\left(\sum_{j\in N_i} B_j \left(\prod_{k\in N_i\backslash\{j\}} C_k^{c(i\backslash\{j\})}\right)\right)$$

$$= P\left(A^c\right) \sum_{j\in N_i} \left(P\left(B_j\right) \prod_{k\in N_i\backslash\{j\}} P\left(C_k^c\right)\right)$$

$$= \left(1-\alpha_i\right) \sum_{j\in N_i} \phi_{j,i}\alpha_j \prod_{k\in N_i\backslash\{j\}} \left(1-\alpha_k\right).$$

where

$$P\left(A^c\right) = \left(1-\alpha_i\right),$$

$$\sum_{j\in N_i} P\left(B_j\right) = \sum_{j\in N_i} \phi_{j,i}\alpha_j,$$

$$\prod_{k\in N_i\backslash\{j\}} P\left(C_k^c\right) = \prod_{k\in N_i\backslash\{j\}} \left(1-\alpha_k\right).$$

The explanation for $\phi_{j,i}$ here is similar as in the proof of *Lemma 1*.

Let the total arrival rate into the transmit buffer of node i be denoted by a_i. If all the transmit queues in the network are *stable*, then the following relation is obtained for $a_i^{\prime}s$

Lemma 3: The arrival rate into nodes are given by the fixed point equation

$$a_i = \lambda_i + \sum_j \phi_{j,i}\left(a_j \wedge \left(\lambda_j + \gamma_j\right) \wedge \mu_j\right) \tag{2.12}$$

where $\wedge$ represents the *minimum* of quantities that appear in (2.12).

Now, we present the stability condition for the system under consideration. This is significantly different from that obtained in the PRN literature [33].

Lemma 4: The transmit queue at node i is stable if

$$\left(\sum_{j\in N_i} \phi_{i,j}\left(\lambda_i + \gamma_i\right)\right) \wedge \mu_i > a_i \tag{2.13}$$

Lemma 5: If all the nodes in the network are stable, then

$$a_i = \lambda_i + \sum_j \phi_{j,i} a_j \tag{2.14}$$

2.5.2 Closed System with Single Transmit Queue

We now consider a sensor network in which there is a strong coupling between the channel access process and the sampling process (introduced in Section 2.4). This approach has the advantage that one does not need to find an optimal sampling rate all over again on changing the channel access rates. The coupling automatically *regulates* the sampling process for any change in the channel access process. Further, we can perform an *exact* stability and delay analysis for this system (as opposed to the *open system* where the available analysis is approximate).

Lemma 6: The stability condition for the transmit queue at node i is

$$\alpha_i \sum_j \phi_{i,j} (1-\alpha_j) \prod_{k \in N_j \backslash \{i\}} (i-\alpha_k) > (1-\alpha_i) \sum_l \phi_{l,i} \alpha_l \prod_{k \in N_i \backslash \{l\}} (1-\alpha_k) \qquad (2.15)$$

Lemma 7: The average data generation rate at node i is

$$\lambda_i = \alpha_i \sum_j \phi_{i,j} (1-\alpha_j) \prod_{k \in N_j \backslash \{i\}} (1-\alpha_k) - (1-\alpha_i) \sum_l \phi_{l,i} \alpha_l \prod_{k \in N_i \backslash \{l\}} (1-\alpha_k) \qquad (2.16)$$

In practice, since we are assuming a pre-specified average data generation node at each node, we will be provided with a vector $\underline{\lambda} = (\lambda_1, ..., \lambda_N)$. For this vector, we need to find values of $\alpha' s$ and *routing* so that the average delays of the flows are *minimized.*

Lemma 8: The average delay at the transmit queue of node i is

1) $\sum_l \phi_{l,i} = 0$:

Delay at node $i = D_i = \frac{1-s_i}{\alpha_i s_i}$, where

$$s_i = \sum_j \phi_{i,j} (1-\alpha_j) \prod_{k \in N_j \backslash \{i\}} (1-\alpha_k) . \qquad (2.17)$$

Proof: If $\sum_l \phi_{l,i} = 0$, then node i has no traffic to be forwarded. The Markov chain of the number of packets in the transmit queue is shown in Figure 2.5.

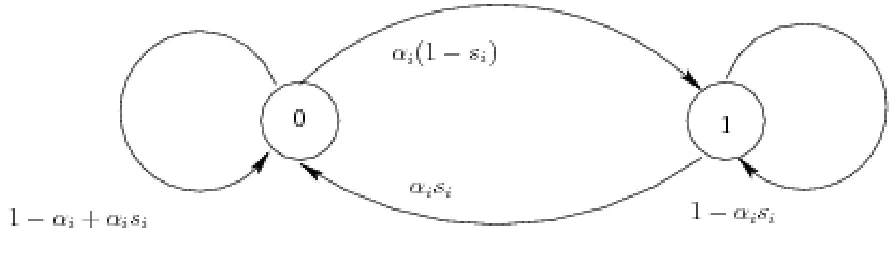

Figure 2.5: Markov chain for the expected number of packets at node i, case 1: $\sum_l \phi_{l,i} = 0$.

so that, we have the following system of equations

$$\pi_0 (1-s_i) = \pi_1 s_i \Rightarrow \pi_1 = \frac{1-s_i}{s_i} \pi_0$$

$$\Rightarrow \pi_0 + \pi_1 = \frac{\pi_0}{s_i} = 1$$

$$\Rightarrow \pi_0 = s_i, \Rightarrow \pi_1 = 1 - s_i$$

Hence expected number of packets in the transmit queue of node i is $1 - s_i$. Using Little's law, the expected delay is

$$D_i = \frac{1 - s_i}{\alpha_i s_i} \tag{2.18}$$

since the effective arrival rate into node i's queue is $\alpha_i s_i$.

2) $\sum_l \phi_{l,i} > 0$:

$$D_i = \frac{\rho_0}{(1-\rho)(1+\rho_0-\rho)(\psi_i+\lambda_i)}, \tag{2.19}$$

where

$$\psi_i = (1-\alpha_i) \sum_l \phi_{l,i} \alpha_l \prod_{k \in N_l \backslash \{i\}} (1-\alpha_k) \, . \tag{2.20}$$

Proof: If $\sum_l \phi_{l,i} > 0$, then the transmit queue of node i can contain more than one packet at a time. The Markov chain of the number of packets in node $i's$ transmit buffer is given in Figure 2.6.

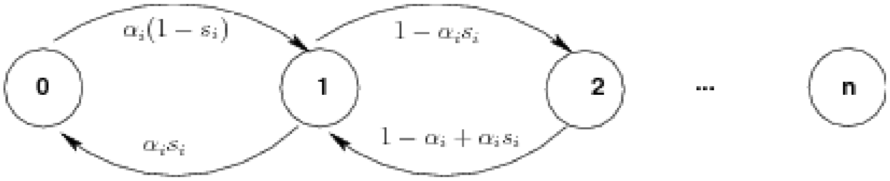

Figure 2.6: Markov chain for the expected number of packets at node i, case2: $\sum_l \phi_{l,i} > 0$.

where

$$p_{0,1} = \alpha_i (1-s_i) + \psi_i$$

$$p_{n,n+1} = \psi_i, \; for \, n \geq 1$$

$$p_{n,n-1} = \alpha_i s_i, \; for \, n \geq 1$$

we define

$$\rho_0 = \frac{\alpha_i (1-s_i) + \psi_i}{\alpha_i s_i},$$

and $\rho = \frac{\alpha_i s_i}{\psi_i}$. Then $\pi_1 = \rho_0 \pi_0$, and $\pi_{n+1} = \rho \pi_n \Rightarrow \rho_0 \rho^n \pi_0$ for $n \geq 1$

$$\Rightarrow \pi_0 + \pi_0 \sum_{n=1}^{\infty} \rho_0 \rho^{n-1} = 1, \Rightarrow \pi_0 \left(\frac{1+\rho_0-\rho}{1-\rho} \right) = 1$$

$$\Rightarrow \pi_0 = \frac{1-\rho}{1+\rho_0-\rho}, \; \pi_n = \rho_0 \rho^{n-1} \pi_0$$

So the expected number of packets in node $i's$ transmit queue is then

$$\sum_{n=1}^{\infty} n \rho_0 \rho^{n-1} \pi_0 = \rho_0 \pi_0 \frac{\partial}{\partial \rho} \sum_{n=1}^{\infty} \rho^n$$

$$= \frac{\rho_0 \pi_0}{(1-\rho)^2} = \frac{\rho_0}{(1-\rho)(1+\rho_0-\rho)}$$

The expected delay at node $i's$ transmit buffer using Little's law is then

$$\frac{\rho_0}{(1-\rho)(1+\rho_0-\rho)(\psi_i+\lambda_i)}$$

2.5.3 Closed System with two Transmit Queues

We, now, give the correct stability condition for our closed system with two transmit queues. It is to be noted that this system can also be thought of as the one in which the sensor node always have a *backlog* of their own sampled data. We fix a node i and look at its forwarding queue F_i. It is clear that if this queue is stable then the output rate from this queue is equal to the input rate into the queue. The only issue to be resolved here is to *properly* define the *output rate*. This is because, owing to the bound M on maximum number of *attempts* for transmission of any packet, not all the packets arriving into F_i may be successfully transmitted. Therefore, the output rate is defined as the rate at which packets are either successfully relayed or dropped due to excessive number of collisions. We start by obtaining the detailed balance equations, i.e., the fact that if the queue F_i is stable, then the output rate from queue F_i is equal to the input rate to this queue. Let

$$s_i = \sum_{j \in \mathcal{N}_i} \phi_{i,j} (1-\alpha_j) \prod_{k \in \mathcal{N}_{j \setminus i}} (1-\alpha_k) \tag{2.21}$$

be the probability that a transmission from node i is *successful*[10]. Also let

$$E_i = \sum_{m=1}^{M} m (1-s_i)^{m-1} s_i + M (1-s_i)^M$$
$$= \tfrac{1-(1-s_i)^M}{s_i} \tag{2.22}$$

be the *expected* number of attempts till success or M consecutive failures of a packet from node i.

Proof: From theory, we have that $\sum_{m=1}^{M} (1-s_i)^{m-1} s_i = 1-(1-s_i)^M$. Taking the derivative of L.H.S and solving, we get

$$= \sum_{m=1}^{M} \left[(1-m)(1-s_i)^{m-2} s_i + (1-s_i)^{m-1} \right]$$

$$= \textstyle\sum_{m=1}^{M} (1-s_i)^{m-1} s_i - \sum_{m=1}^{M} m (1-s_i)^{m-1} s_i + \sum_{m=1}^{M} (1-s_i)^{m-1}$$

Similarly, the R.H.S gives us $M(1-s_i)^{M-1}$. Multiplying both sides by $(1-s_i)$ and solving, we get

$$\textstyle\sum_{m=1}^{M} (1-s_i)^{m-1} s_i - \sum_{m=1}^{M} m (1-s_i)^{m-1} s_i + \sum_{m=1}^{M} (1-s_i)^{m} = M(1-s_i)^M$$

[10] It is to be noted that the s_i in (2.21) is different than the s_i in (2.17). Both represent successful transmission probabilities. In this system, when we refer to s_i, we will be referring to (2.21).

$$\textstyle\sum_{m=1}^{M} m\left(1-s_i\right)^{m-1} s_i + M\left(1-s_i\right)^{M} = \sum_{m=1}^{M}\left(1-s_i\right)^{m-1} s_i + \sum_{m=1}^{M}\left(1-s_i\right)^{m}$$

$$= \frac{s_i\left(1-\left(1-s_i\right)^{M}\right) + \left(1-s_i\right)\left(1-\left(1-s_i\right)^{M}\right)}{s_i}$$

$$= \frac{1-\left(1-s_i\right)^{M}}{s_i}$$

The proof is complete.

Lemma 1: For a given routing, let π_i denote the probability that a node i has packets to forward, then the long term average rate of departure of packets from node $i's$ forwarding queue is

$$\pi_i \alpha_i f_i E_i. \tag{2.23}$$

Proof: Let T_t be an indicator function which is unity if F_i is nonempty. Let I_t be an indicator function that $T_t = 1$ and a transmission is made from F_i (it can be a success or a failure). Then the output rate from F_i of packets is then

$$\lim_{t\to\infty} \frac{1}{t} \sum_{l=1}^{t} I_l = \lim_{t\to\infty} \frac{\sum_{l=1}^{t} T_l}{t} \lim_{t\to\infty} \frac{\sum_{l=1}^{t} I_l}{\sum_{t=1}^{l} T_l}.$$

Since we are working under assumption that node i attempts forwarding of any packet at most M times, we have, with probability one,

$$\lim_{t\to\infty} \frac{\sum_{l=1}^{t} I_l}{\sum_{t=1}^{l} T_l} = \alpha_i f_i E_i.$$

Also, with probability one,

$$\lim_{t\to\infty} \frac{\sum_{l=1}^{t} T_l}{t} = \pi_i.$$

Clearly, the long term output rate from the queue F_i is, with probability one,

$$\lim_{t\to\infty} \frac{\sum_{l=1}^{t} I_l}{t} = \lim_{t\to\infty} \frac{\sum_{l=1}^{t} T_l}{t} \lim_{t\to\infty} \frac{\sum_{l=1}^{t} I_l}{\sum_{t=1}^{l} T_l} = \pi_i \alpha_i f_i E_i.$$

The proof is complete.

Lemma 2: The long term average rate of arrival of packets into F_i is

$$\sum_{j\in\mathcal{N}_i} \phi_{j,i}\left(\alpha_j E_j\right) \tag{2.24}$$

The proof for average rate of arrival is straightforward in the sense that i can only receive packets from $j, j \in \mathcal{N}_i$. $\phi_{j,i}$ is the amount of traffic on link (j,i). α_j is the probability with which j is transmitting and E_j is the expected number of attempts of packet till success or M consecutive failures.

Proposition 1: In the steady state, if all the queues in the network are stable, then for each i

$$\pi_i \alpha_i f_i E_i = \sum_{j \in \mathcal{N}_i} \phi_{j,i} \left(a_j E_j \right) \tag{2.25}$$

Proof: If the queue F_i is stable, then the rate of arrival of packets into the queue is the *same* as the rate at which the packets leave the queue. Let $w_{j,i} = \frac{\sum_j \phi_{j,i}(a_j E_j)}{\alpha_i E_i}$ and $y_i = 1 - \pi_i f_i$ (transmission probability from Q_i). Note that w_i is independent of f_j, $j \in \mathcal{N}_i$, and depends only on the α_j and routing.

In the steady state, if all the queues in the network are stable, then we can write for each i

$$1 - y_i = \sum_{j \in \mathcal{N}_i} w_{j,i} \tag{2.26}$$

The relation of eq. (2.26) has some interesting interpretations. Some of these are:

The Effect of f_i : The quantity $y_i = 1 - \pi_i f_i$ is independent of the choice of f_j, $j \in \mathcal{N}_i$. It only depends on the routing and the value of α_j.

Stability: Since the values of y_i are independent of the values of f_j, $j \in \mathcal{N}_i$, and since we need $\pi_i < 1$ for the forwarding queue of node i to be stable, we see that for any value of $f_i \in (1 - y_i,\, 1)$, the forwarding queue of node i will be stable. Thus we obtain a lower bound on the weights given to the forwarding queues at each node in order to guarantee stability of these queues. To ensure that these lower bounds are all feasible, i.e., are less than 1, we need that $0 < y_i \leq 1$; $y_i = 0$ corresponds to the case where F_i is unstable. Hence, if the routing and $\alpha_j' s$ are such that all the y_i are in the interval $(0, 1]$, then all the forwarding queues in the network can be made stable by appropriate choice of $f_i' s$. Now, since y_i is determined only by routing and the probabilities $\alpha_j' s$, we can then choose f_i (thereby also fixing π_i, hence the forwarding delay) to satisfy some further optimization criteria so that this extra degree of freedom can be exploited effectively.

Throughput: We see that the long term rate at which node i can serve its own queue is $\alpha_i \left(1 - \pi_i f_i\right) = \alpha_i y_i$, which is independent of f_i. Also, the throughput, i.e., the rate at which the packets reach the destination, i.e., $w_i = \sum_j \phi_{j,i} a_j E_j$ is independent of f_j. Similarly, the long term rate at which the packets from the forwarding queue at node i are attempted transmission is $\pi_i \alpha_i f_i = \alpha_i \left(1 - y_i\right)$, which is also independent of the choice f_j, $j \in \mathcal{N}_i$.

Throughput-Stability Tradeoff: In the present case, we can tradeoff throughput with stability and not directly with delay. Let $\pi_i f_i = c$, if $c > 1$, $\forall i$ simultaneously, the system is unstable. We know that the throughput at node i is $1 - \pi_i f_i$. Then, if a node tries to maximize its own throughput, it is actually minimizing c, thus trying to stabilize the system. This is an interesting property in itself.

Choice of f_i: Assume that we restrict ourselves to the case where $f_i = P_f$, $\forall i$. Then, for the stability of all nodes, we need that

$$P_f > 1 - \min_i y_i. \tag{2.27}$$

Since the length of interval that f_i is allowed to take is equal to y_i, we will also refer to y_i as the stability region.

Energy-Delay Tradeoff: For a given set of $\alpha'_i s$ and routing, the throughput obtained on a route $\mathcal{R}_i$ is fixed, independent of f_i. Hence, there is no throughput-delay tradeoff obtained by changing f_i. However, we do obtain energy-delay tradeoff. For a given *stable* routing, we need f_i, which will determine π_i. Clearly, f_i represents the *forwarding-energy*[11] and π_i gives a *direct* measure of *delay*. Therefore, the service rate given to F_i determines the *exact* energy-consumption and delay for relaying, and hence, we can perform an *exact analysis* of the effect of different network parameters on performance in multi-hop WSNs.

2.6 Routing Algorithms for Different Systems Under Consideration

If the traffic split is not allowed, the objective of the distributed routing algorithm would be to find the *shortest delay path* between any given source and the fusion center. However, one may allow for traffic *split* and then try to route the traffic, hoping for a better performance (as the situation without traffic split is a *special* case of traffic splitting). Under this added freedom of traffic splitting, the routing algorithm is expected to put traffic of a node on those routes for which the delays are smallest and equal. This is what is well known as the Wardrop equilibrium. We propose a stochastic approximation algorithm based on a distributed algorithm to converge to a Wardrop equilibrium. This algorithm is actually an adaptation of the algorithm already proposed in [29] to our system for which we can *prove* convergence to Wardrop equilibrium.

2.6.1 Open System

The algorithm here is essentially the same as in [29], i.e., nodes iteratively keep updating the one-hop routing probabilities based on the delays incurred for every possible path.

Let $\phi(n)$ denote the traffic splitting matrix at the beginning of the n^{th} time slot. Node i does some computation to update the i^{th} row of this matrix. Let $Y^k(n)(\mathcal{R}_{k,1} = i)$ be the new value of the delay of a packet sent by sensor i through route $k(i = \mathcal{R}_{k,1})$. Node i keeps an estimate of the average delay on route k.

$$y^k(n+1) = (1-a)y^k(n) + aY^k(n). \tag{2.28}$$

The average delays in (2.28) are calculated to mitigate the effects of route changes based on a very small change in delay. Further, after calculating the expected delays at the start of a time slot, each node adapts its routing probabilities to the new expected delays as follows,

$$\phi_i, \mathcal{R}_{k,2}(n+1) = (1-b)\phi_i, \mathcal{R}_{k,2}(n) + b\left(\textstyle\sum_{1\leq l\leq R:\mathcal{R}_{l,1}=i} y^l(n)\phi_i, \mathcal{R}_{l,2}(n) - y^k(n)\right) \tag{2.29}$$

Proof of Convergence to Wardrop Equilibrium: We will assume that the learning parameters a and b are such that $a \ll b$. This brings us in the two-level stochastic approximation algorithm framework and, following standard results [49], the update of the traffic split will see the average delays y^l as static so that the effect of the second update will be that all the traffic from node i will be directed to the smallest delay route. The algorithm for

[11] Since, this is the exact amount of resource dedicated to the forwarding traffic and represents its direct measure. Also, f_i is indirectly related to delay.

updating the delay estimates over route will thus see no effect of the dynamics of the second update scheme except that the statistical properties of the random variables will come from the splitting vector in which each node directs all its traffic on one of the possible routes from the node to the fusion center; note that in general different nodes will be choosing different routes. Thus, by the standard o.d.e. approach to stochastic approximation algorithms [49], the delay updating algorithm will behave like an autonomous ODE. The convergence of this differential is guaranteed using arguments similar to those used in [51]. Since the point of convergence satisfies the defining condition of the Wardrop equilibrium, the proposed algorithm will *converge*[12] to the Wardrop equilibrium. *Note* that this convergence is for the average of delays, this is what we mean by Cesaro-Wardrop equilibrium.

2.6.2 Closed System with Single Transmit Queue

The updates for this system are going to be the same as that for the Open system. The only new complication here is that one needs to tune the channel access rates, $\alpha_i' s$, also in order to guarantee the long term average data sampling rate. This is easily done because the nodes know (or, can estimate) the statistics of the traffic they are getting from the other nodes and also the success rate of its own transmissions to various neighbors. Using this estimate a node can easily tune its channel access rate to guarantee itself a preset data sampling rate.

In each time slot, i^{th} sensor tries to hold channel for transmission with probability α_i. If the node tries to hold channel in a time slot, it either succeeds in transmitting or fails. If the node succeeds, then if the packet transmitted can be the one which is generated at the current node or it may be the one which the node received from any of the neighboring nodes. Let $n(k)$ be the number of slots in which node has successfully transmitted a packet generated by itself in total k slots. $\hat{\lambda_i^k} = \frac{n(k)}{k}$ is the rate of transmission node is able to provide in the k^{th} slot.

$$\hat{\lambda_i^k} = \frac{n(k)}{k} \tag{2.30}$$

$$\alpha_i^{k+1} = \max\left\{\min\left[\alpha_i^k + c\left(\lambda_i - \hat{\lambda_i^k}\right), 1\right], 0\right\} \tag{2.31}$$

Where c is a positive learning parameter. Delay and routing probability learning will remain as was in the Open System.

2.6.3 Practical Considerations

Here, we will discuss some of the practical aspects of our proposed algorithm. Delay estimation of paths by a node in every slot can be done by having power of the sink so large that it can reach all the sensors in one-hop. Therefore, the sink can acknowledge all the incoming packets such that the sensors will *directly* get estimation of the delay incurred by their packets.

2.7 Simulation Results

We consider a 6-node sensor network shown in Figure 2.7. It can be easily seen that $\phi_{1,0} = \phi_{3,0} = \phi_{5,0} = 1$, node 0 being the common destination for all the packets generated in the

[12] It is, hence, shown that the proposed distributed algorithm provides the *optimal* delay that is achievable based on traffic dynamics.

network. The other available routes could also be used for nodes 1, 3, and 5. But, we were more interested in clearly demonstratign the *traffic split* and *routing probabilities*. The routing algorithm thus has to find appropriate value of $\phi_{2,5}$ and $\phi_{4,3}$ in order that the traffic flow in the network corresponds to a Wardrop equilibrium.

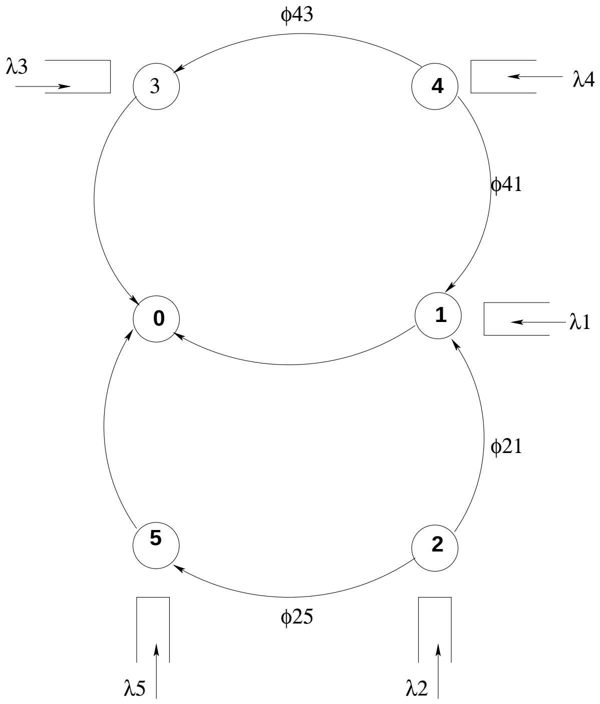

Figure 2.7: Network Simulated for Stability

Apart from a demonstration of the convergence of the proposed algorithm, we will see in this section that the data sampling rates that a network can support using the Open architecture is *very* small. This is essentially because of the *stability* constraints on the channel access rates. On the other hand, the Closed system can support *higher* data sampling rates because of the fact that it is essentially self-regulating, guaranteed to be stable while maintaining large data sampling rates; this is because a node generates a new packet only if it has no other packet in the queue. This however does not mean that the Closed system can support *arbitrary* data sampling rates.

We have implemented the Open and Closed system as an application layer module in TinyOS [52]. TinyOS is an open-source operating system designed for embedded WSNs. It features a component-based architecture which enables rapid innovation and implementation while minimizing code size as required by the severe memory constraints inherent in WSNs. The sensor network model under consideration is shown in Figure 2.8. The sensor nodes sample the data at a predefined rate, $\lambda_i's$. The sampled data is sent to the MAC queue for both open and closed system according to the explanation given earlier in Section 2.4. The transmit queue of node i can have at most one packet in the transmit queue that was generated at node i. It can however have multiple packets in the transmit queue to be forwarded, i.e., those packets that were initially generated at some other node, and have arrived at node i to be forwarded to some other node. Therefore, we need not implement two-queues at the MAC layer for sensor nodes for prioritizing traffic. At simulation start up, the nodes learn the network topology and built routes toward the fusion center (sink, node 0). The fusion center is also a sensor node which has 0 sampling rate. This learning process, which depends on the network topology for the given network in Figure 2.7, can take up to $50 - 70\,seconds$ for larger topologies. The routing layer is initiated with the minimum-hop routing, which is updated during the network lifetime according to the algorithm proposed in Section 2.6. In this section, we present the numerical results once the neighbors are discovered and routes are established toward the fusion center. We have utilized the TOSSIM simulator of TinyOS to validate our proposals. All simulation run for $1000\,seconds$. The results presented in this section are the average over several simulation runs.

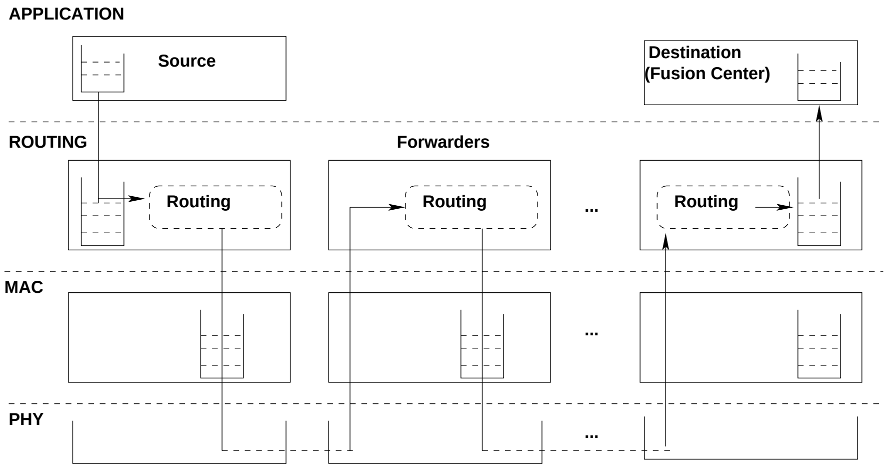

Figure 2.8: Sensor network architecture. → represents the flow of packets from the source to the destination. The forwarding sensor network receives a packet and queues into the forwarding queue at the MAC layer. The routing layer does not buffer the forwarding traffic.

2.7.1 Open System Stability

In Figures. 2.9 and 2.10 we plot, against the slot number, the average delays on the four routes $2 \rightarrow 5 \rightarrow 0$, $2 \rightarrow 1 \rightarrow 0$, $4 \rightarrow 3 \rightarrow 0$, and $4 \rightarrow 1 \rightarrow 0$ for the open system. The data sampling rates were set at $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = 0.2$. Note that the data sampling rates are small. We were forced to select small data rates in order to guarantee stability of the nodes in the network. The channel access rates were set to $\alpha_i \leq 0.2$ for $i = 1, ..., 5$.

Observations

1. The delays on routes $2 \rightarrow 5 \rightarrow 0$ and $2 \rightarrow 1 \rightarrow 0$ are very close to each other, with a very fast convergence. Similarly for routes $4 \rightarrow 3 \rightarrow 0$ and $4 \rightarrow 1 \rightarrow 0$. This shows that the algorithm succeeds in achieving a Wardrop equilibrium.

2. Note the high value of delay on routes $2 \rightarrow 1 \rightarrow 0$ and $4 \rightarrow 1 \rightarrow 0$ even for the moderate (or, very small) load on the system.

3. The delays on different routes are sensitive to the channel access probabilities. Thus, there is a need for carefully tuning the channel access probabilities. In Figure 2.9 and 2.10, we also see the convergence to a *load-balanced* regime (equal delays on all the possible routes from a particular source).

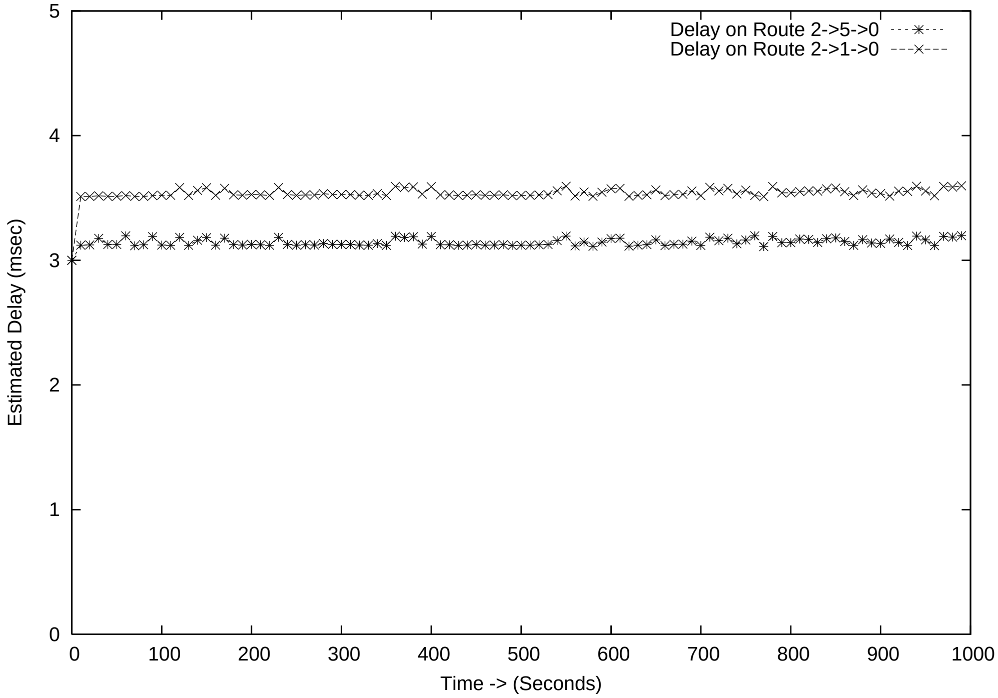

Figure 2.9: Delays incurred on routes 2 → 5 → 0, 2 → 1 → 0 for Open System. Where $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = 0.2$

2.7.2 Closed System Stability

Simulation results for the closed system are presented in Figure 2.11 and 2.12. The data sampling rates were set at $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = 0.2$. Nodes were expected to adapt their channel access probabilities based on the optimal traffic split used by node 2 and 4.

Observations

1. The delays on routes $2 \to 5 \to 0$ and $2 \to 1 \to 0$ are very close to each other, with a fast convergence. This shows that the algorithm succeeds in achieving a Wardrop equilibrium.

2. For routes $4 \to 3 \to 0$ and $4 \to 1 \to 0$, the delays are also close to each other, with a fast convergence. This shows that the algorithm is successful in achieving a Wardrop equilibrium (equal delays on all the possible routes from a particular source).

3. Note the small value of delay on routes $2 \to 5 \to 0$ and $4 \to 3 \to 0$ even for moderate (or, very small) load on the system. This is to be compared with the corresponding values shown under the results for open system where the delays on these routes were higher even though the average data sampling rates were significantly smaller. Thus, in comparison with the open system, the closed system provides better performance.

We simulate another 6-node sensor network shown in Figure 2.13 to demonstrate the results on routing. The only difference with the first network is that we have a different routing setup but its logical representation is the same. It is easily seen that $\phi_{1,0} = \phi_{3,0} = \phi_{5,0} = 1$, node 0 being the common destination for all the packets generated in the network. The routing

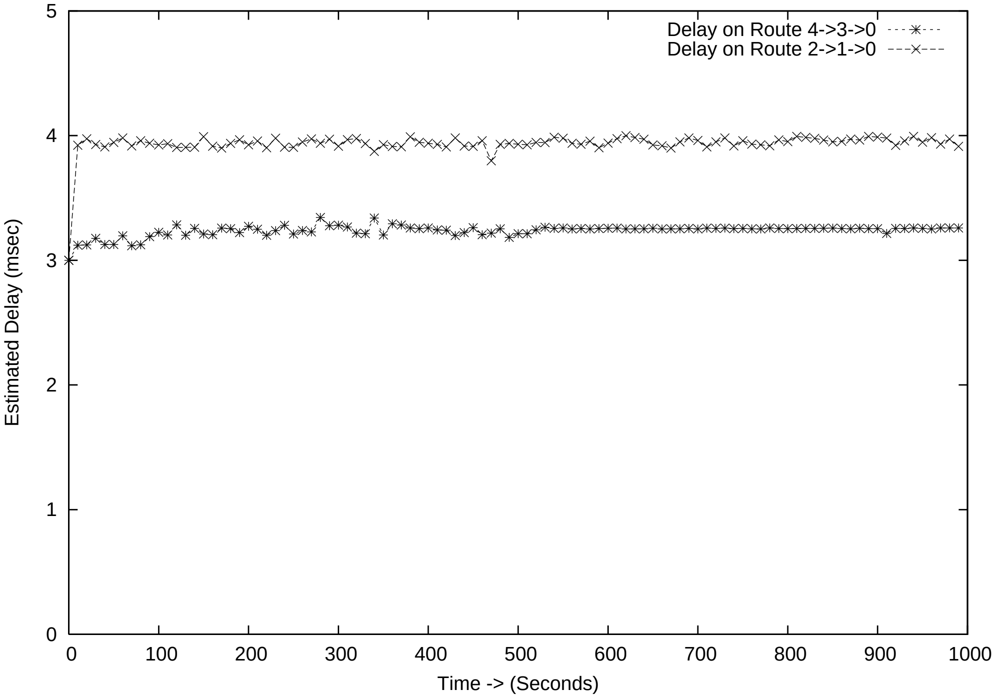

Figure 2.10: Delays incurred on routes 4 → 3 → 0, 4 → 1 → 0 for Open System. Where $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = 0.2$

algorithm thus has to find appropriate value of $\phi_{2,5}$ and $\phi_{4,3}$ in order that the traffic flow in the network corresponds to a Wardrop equilibrium.

2.7.3 Open System Routing

In Figure 2.14 and 2.15 we plot, against the slot number, the average delays on the four routes $3 \rightarrow 2 \rightarrow 0$, $3 \rightarrow 1 \rightarrow 0$, $5 \rightarrow 4 \rightarrow 0$, and $5 \rightarrow 1 \rightarrow 0$ for the open system. The data sampling rates were set at $\lambda_i \leq 0.2$, for $i = 1, ..., 7$. Note that the data sampling rates are small. We were forced to select small data rates in order to guarantee stability of the nodes in the network. The channel access rates were set to $\alpha_i \leq 0.2$ for $i = 1, ..., 7$.

Observations from Open System

1. The delays on routes $3 \rightarrow 1 \rightarrow 0$ and $3 \rightarrow 2 \rightarrow 0$ are very close to each other, with a very fast convergence. Similarly for routes $5 \rightarrow 4 \rightarrow 0$ and $5 \rightarrow 1 \rightarrow 0$. This shows that the algorithm succeeds in achieving a Wardrop equilibrium.

2. Note the high value of delay on routes $3 \rightarrow 1 \rightarrow 0$ and $3 \rightarrow 2 \rightarrow 0$ even for moderate (or, very small) load on the system.

3. Figure 2.15 shows the delay obtained by varying the channel access rates to $\alpha_i = 0.1$ for $i = 1, ..., 5$, and $\lambda' s$ remaining the same as earlier. The estimated delays show the *sensitivity* to channel access probabilities. Thus, there is a need to carefully tune the $\alpha_i' s$. In Figure 2.15, we also see that convergence to a *load-balanced* regime (equal delays on all the possible routes from a particular source) is violated by changing the $\alpha_i' s$. As we will see later, this is not a problem in the closed system because the system adapts

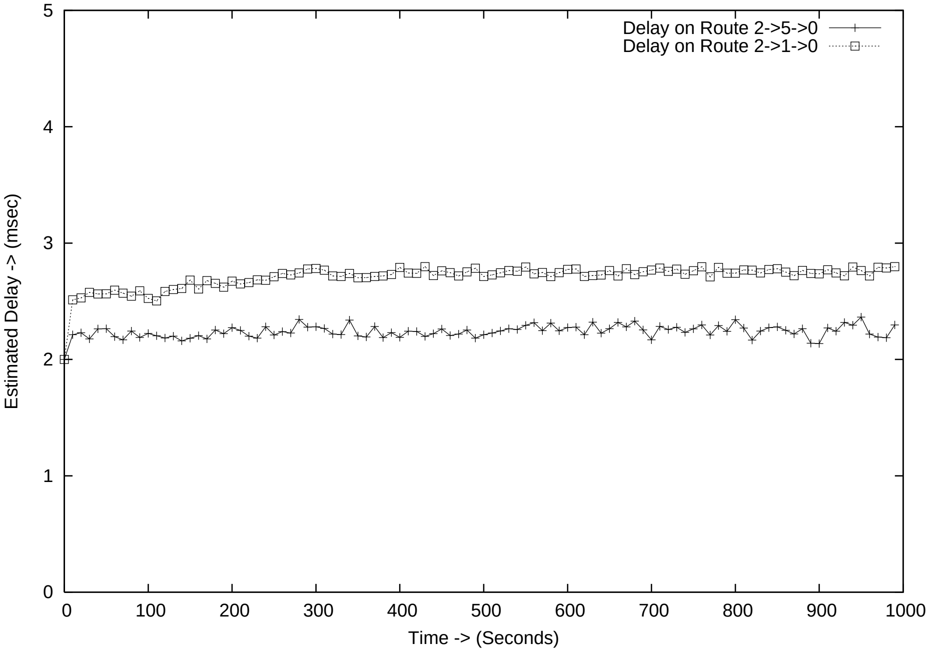

Figure 2.11: Delays incurred on routes 2 → 5 → 0, 2 → 1 → 0 for Closed System. Where $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = 0.2$

its channel access probabilities to meet the target traffic and there is no need of further tuning this parameter.

4. The delays on different routes are also close to each other, with a fast convergence. This is also reflected in the traffic split obtained by the algorithm, as in Figure 2.16 we see that node 3 uses node 2 a little less than the other available route because of smaller delay on $3 \rightarrow 1 \rightarrow 0$. Similarly, node 5 also use $5 \rightarrow 1 \rightarrow 0$ more than $5 \rightarrow 4 \rightarrow 0$ because of smaller delay on the former. It is also interesting to note that the traffic split obtained in this figure is proportional to the delays on different routes in the network, i.e., ϕ_{32} is very close to 0.5 due to a smaller difference in estimated delays on routes $3 \rightarrow 1 \rightarrow 0$ and $3 \rightarrow 2 \rightarrow 0$, whereas, ϕ_{54} is not due to relatively large difference in the estimated delays on routes $5 \rightarrow 1 \rightarrow 0$, $5 \rightarrow 4 \rightarrow 0$. This is Wardrop equilibrium where a slightly higher delay path is less used i.e., the $+ve$ value of traffic on alternate route is imposed by the algorithm to ensure that all the alternatives are probed often enough to cope up with a change in traffic patterns.

2.7.4 Closed System Routing

Simulation results for the closed system are presented in Figure 2.17, 2.18, 2.19, and 2.20. The data sampling rates were set at $\lambda_1 0.1, = \lambda_2 = 0.2, \lambda_3 = 0.1, \lambda_4 = 0.005, \lambda_5 = 0.1, \lambda_6 = 0.1, \lambda_7 = 0.1$. Nodes were expected to adapt their channel access probabilities based on the optimal traffic split used by node 3 and 5.

Observations from Closed System

1. The delays on routes $3 \rightarrow 1 \rightarrow 0$ and $3 \rightarrow 2 \rightarrow 0$ are very close to each other, with

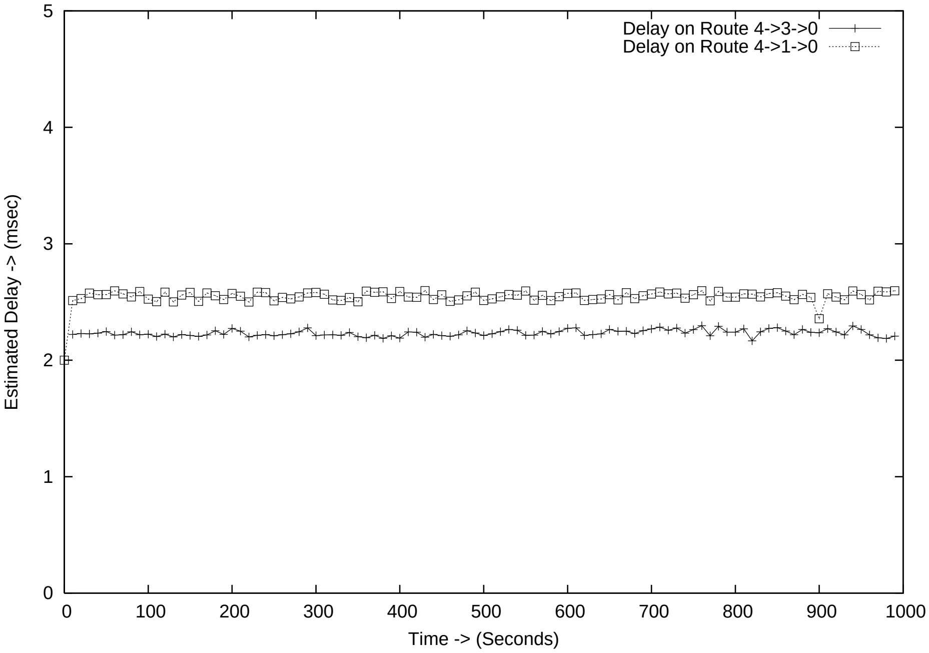

Figure 2.12: Delays incurred on routes 4 → 3 → 0, 4 → 1 → 0 for Closed System. Where $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = 0.2$

a fast convergence. This shows that the algorithm succeeds in achieving a Wardrop equilibrium.

2. For routes $5 \rightarrow 1 \rightarrow 0$ and $5 \rightarrow 4 \rightarrow 0$, the delays are also close to each other, with a fast convergence. This is also reflected in the traffic split obtained by the algorithm, as in Figure 2.19 we see that node 5 uses node 1 for most of its traffic, thus obtaining smaller delay. This is again Wardrop equilibrium where the higher delay path is not used (the small $+ve$ value of traffic on route $5 \rightarrow 4 \rightarrow 0$ is imposed by the algorithm to ensure that all the alternatives are probed often enough to cope up with a change in the network).

3. Note the small value of delay on routes $3 \rightarrow 1 \rightarrow 0$ and $3 \rightarrow 2 \rightarrow 0$ even for moderate (or, very small) load on the system. This is to be compared with the corresponding values shown under the results for open system where the delays on these routes were higher even though the average data sampling rates were significantly smaller. Thus, in comparison with the open system, the closed system provides better performance.

4. Figure 2.20 shows that the algorithm is also able to adapt the channel access rates in a distributed fashion. It can be checked that the values of $\alpha_i' s$ converged-to by the algorithm indeed are just enough to serve the traffic offered to the different nodes.

In the above analysis, we have only presented results on the performance of our distributed routing algorithm for both open and closed systems. We, now, consider a *randomly* deployed sensor network with 50 sensor nodes. There is only corner-sink which is the representative of data collection in the network. Sensors transmit their readings in a multihop fashion towards this sink. The sampling rate of all the nodes is a random variable uniformly distributed between 0 and 0.2. The channel access rate of all the nodes is set according to their sampling

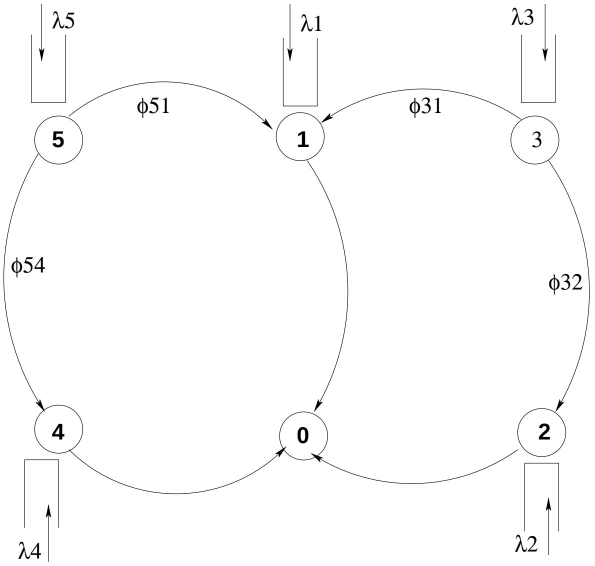

Figure 2.13: Network Simulated for Routing

rates. The simulation runs for 3000s and the sampling vector is changed every 100s to see the impact of *traffic pattern change* on the *performance* of closed system. Figure 2.21 displays individual node delays over time w.r.t change in the sampling rate. It can be easily seen, for each node in the network, that the delay *does not* change much due to a change in the *traffic pattern* over time. The *coupling* in the closed system *automatically regulates* the delay by *adapting* its own sampling process to the change in network dynamics. The last degree of freedom, i.e., the *channel access rates* are also adapted using the proposed optimization criteria. The CDF of the estimated delay in the network is presented in Figure 2.22.

2.7.5 Closed System with Two Transmit Queues

We, again, consider the 6-node sensor network shown in Figure 2.13. *We consider this simple network to clearly demonstrate the stability region in closed system with two transmit queues.* The transmit queue of node i can have multiple packets in the transmit queue (both Q_i,i.e, self generated, and F_i, i.e., those packets that were initially generated at some other node, and have arrived at node i to be forwarded to some other node). Therefore, we need to implement two-queues at the MAC layer for sensor nodes for prioritizing traffic (based on the appropriate weights given to Q_i and F_i). We have implemented the Closed system with two-queues as a *cross-layer* (application-mac) module in TOSSIM [52]. The routing layer is initiated with the minimum-hop routing, which is updated during the network lifetime according to the algorithm proposed in Section 2.6. In this section, we present the simulation results once the neighbors are discovered and routes are established toward the fusion center. All simulation runs for 10^8, seconds.

We present in Table 2.3, the results on stability region and throughput for sensors 1, 2, and 4 as sensors 3 and 5 do not forward any traffic and y_i for $i = 3, 5$ is set to 1.

In order to demonstrate the results on *delay-and-stability* together using a closed-system with two-queues, we have implemented a 50-nodes sensor network with a common sink. In Figure 2.23 we plot, against the slot number, the average delays for our closed-system with

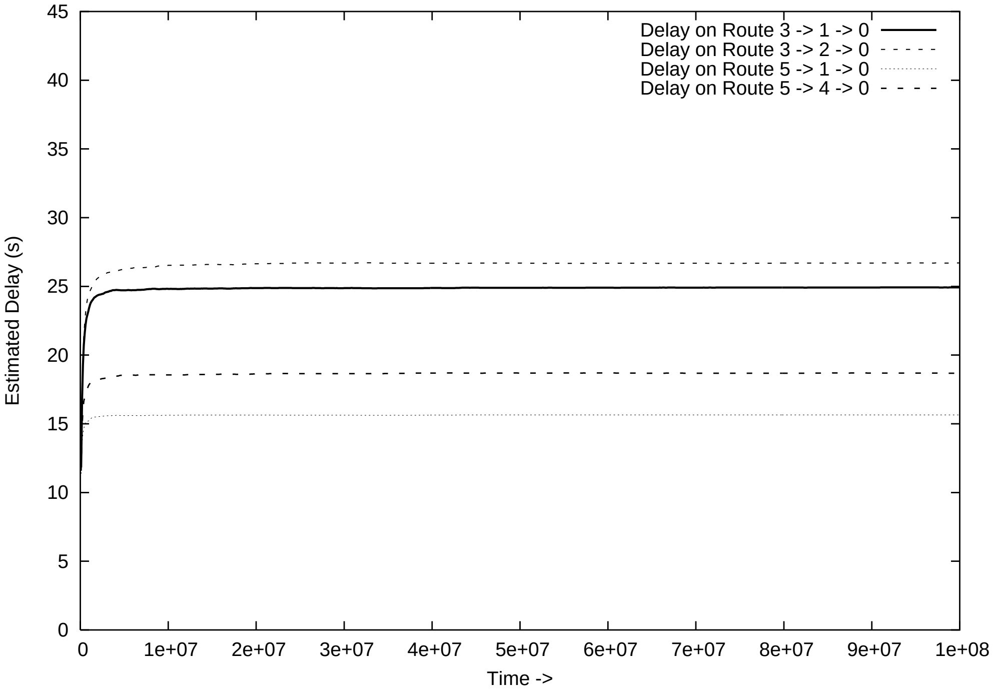

Figure 2.14: *Delays* incurred on routes $3 \rightarrow 1 \rightarrow 0$, $3 \rightarrow 2 \rightarrow 0$, $5 \rightarrow 1 \rightarrow 0$, $5 \rightarrow 4 \rightarrow 0$ for open system. $\alpha_1 = 0.2$, $\alpha_2 = 0.15$, $\alpha_3 = 0.1$, $\alpha_4 = 0.2$, $\alpha_5 = 0.2$, $\lambda_1 = 0.01$, $\lambda_2 = 0.01$, $\lambda_3 = 0.04$, $\lambda_4 = 0.05$, $\lambda_5 = 0.05$.

two-queues and single-queue system. The data sampling rates were set at $\lambda_i \leq 0.1$, $\forall i$. Note that the data sampling rates are small. We were forced to select small data rates in order to guarantee stability of the nodes in the network.

Observations from the Simulations: The average delays on routes in two-queues closed system are very small compared to single-queue system. This is due to the appropriate choice of weights given to both F_i and Q_i (as discussed in Section 2.4) compared to the single queue system where we do not have the service differentiation. The routing schemes (Section 2.6) allows both systems to pick the shortest-delay paths based on delay estimates. These results comply with our motivation that service *differentiation* at MAC layer results in better *over all* performance of the system and can help study the *impact* of different network *parameters* on its performance.

2.8 Conclusions and Future Work

For WSNs with random channel access, we proposed a data sampling approach that guarantees a *long term* data sampling rate while minimizing the end-to-end delays. Simulation and numerical results show that performance of this scheme is better than the traditional *layered* architecture where the channel access mechanism is independent of the data sampling process. We also saw that the proposed scheme does not require *tedious* parameter tuning as is the case for the layered architecture.

We have also obtained some important insights into various *tradeoffs* that can be achieved by varying certain network parameters. Some of them include: 1) Routing can be crucial

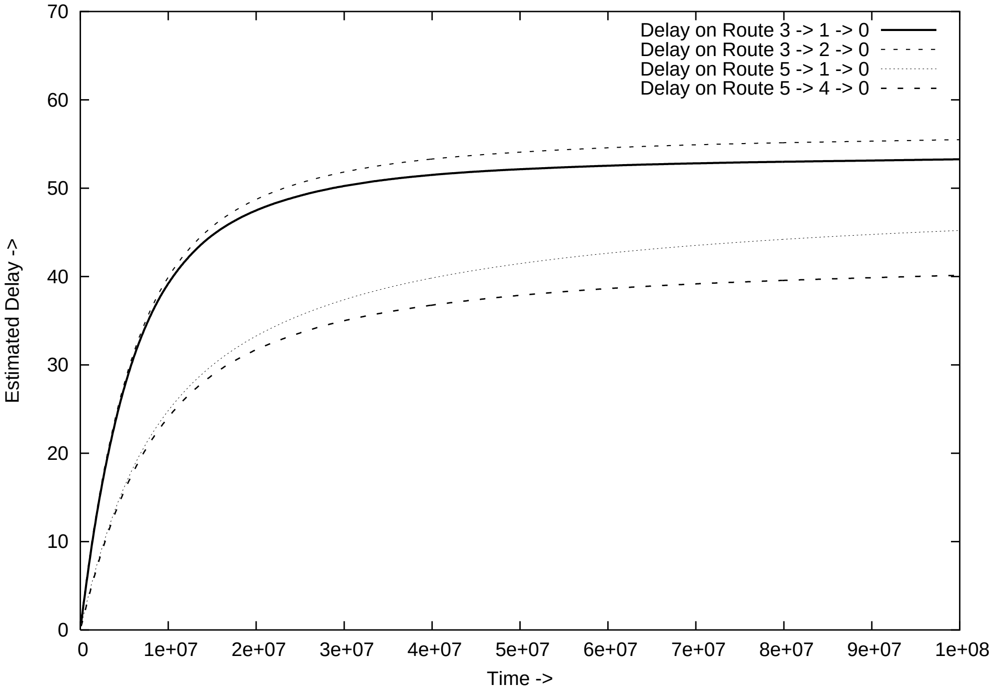

Figure 2.15: *Delays* incurred on routes 3 → 1 → 0, 3 → 2 → 0, 5 → 1 → 0, 5 → 4 → 0 for open system. $\alpha_1 = 0.1$, $\alpha_2 = 0.1$, $\alpha_3 = 0.1$, $\alpha_4 = 0.1$, $\alpha_5 = 0.1$, $\lambda_1 = 0.01$, $\lambda_2 = 0.05$, $\lambda_3 = 0.05$, $\lambda_4 = 0.01$, $\lambda_5 = 0.04$.

in determining the stability properties of the networked sensors. 2) Whether or not the forwarding queues can be stabilized (by appropriate choice of WFQ weights) depends only on routing and channel access rates 3) We have also seen that the end-to-end throughput is independent of the choice of WFQ weights.

We then proposed a *learning* algorithm, applicable to both the open system as well as the closed system, to achieve Wardrop equilibrium for the *end-to-end* delays incurred on different routes from sensor nodes to the fusion center. For the closed system, this algorithm also *adapted* the channel access rates of the sensor nodes. From the simulation results, we have seen a *very* high delay for a single-queue system (provided the system was stable) compared to two-queues system.

Since, the objective of the algorithm was only to converge to a Wardrop equilibrium, at this moment it is not able to make a judicious choice among multiple Wardrop equilibria, if they exist.

In the following chapter, we will see how we can overcome the shortcomings that appeared in the layered architecture.

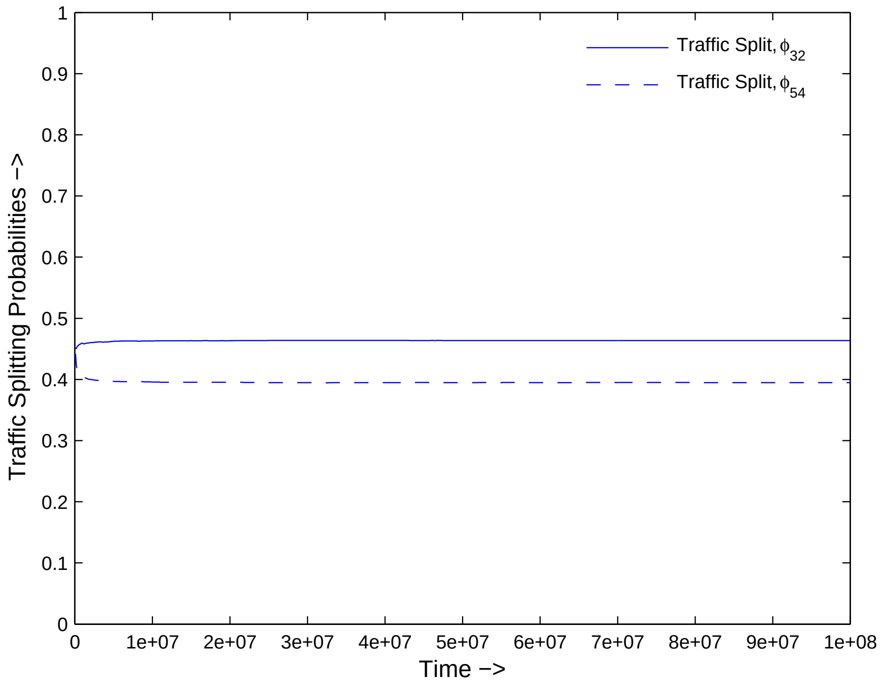

Figure 2.16: Traffic split over the routes 3 → 1 → 0, 3 → 2 → 0, 5 → 1 → 0, 5 → 4 → 0 for open system.

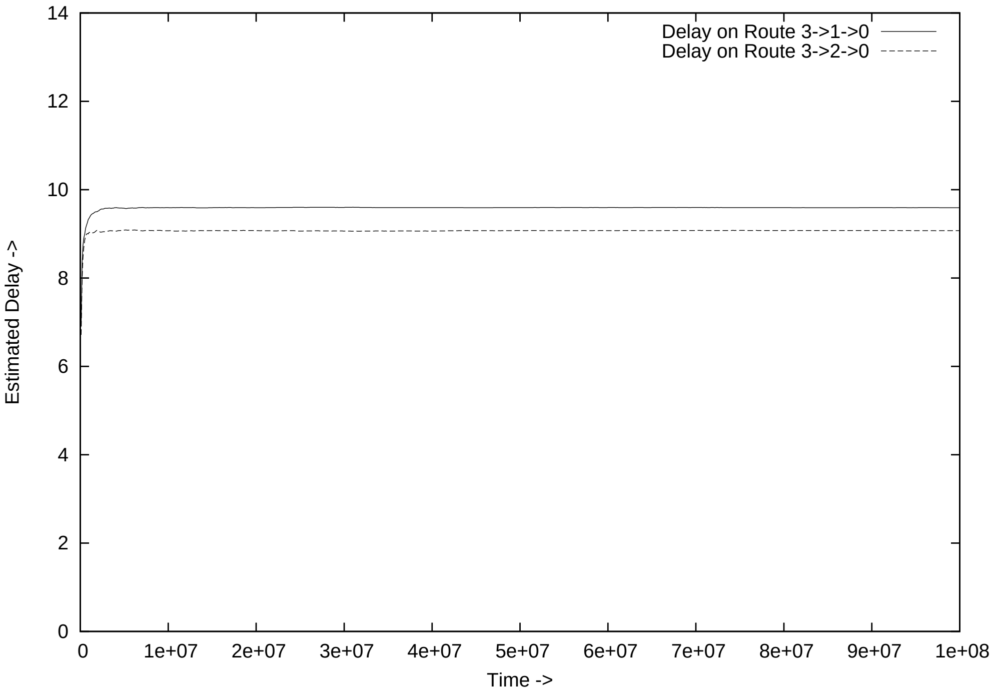

Figure 2.17: *Delays* incurred on routes 3 → 1 → 0, 3 → 2 → 0 for closed system with $\lambda_1 = 0.1$, $\lambda_2 = 0.2$, $\lambda_3 = 0.1$, $\lambda_4 = 0.005$, $\lambda_5 = 0.1$.

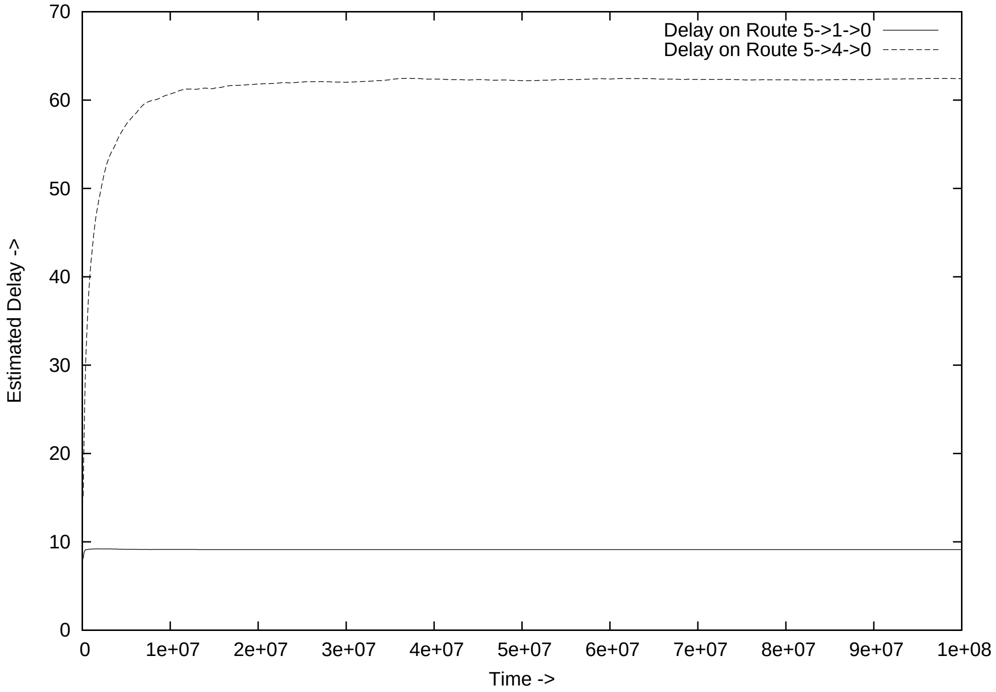

Figure 2.18: *Delays* incurred on routes 5 → 1 → 0, 5 → 4 → 0 for closed system with $\lambda_1 = 0.1$, $\lambda_2 = 0.2$, $\lambda_3 = 0.1$, $\lambda_4 = 0.005$, $\lambda_5 = 0.1$.

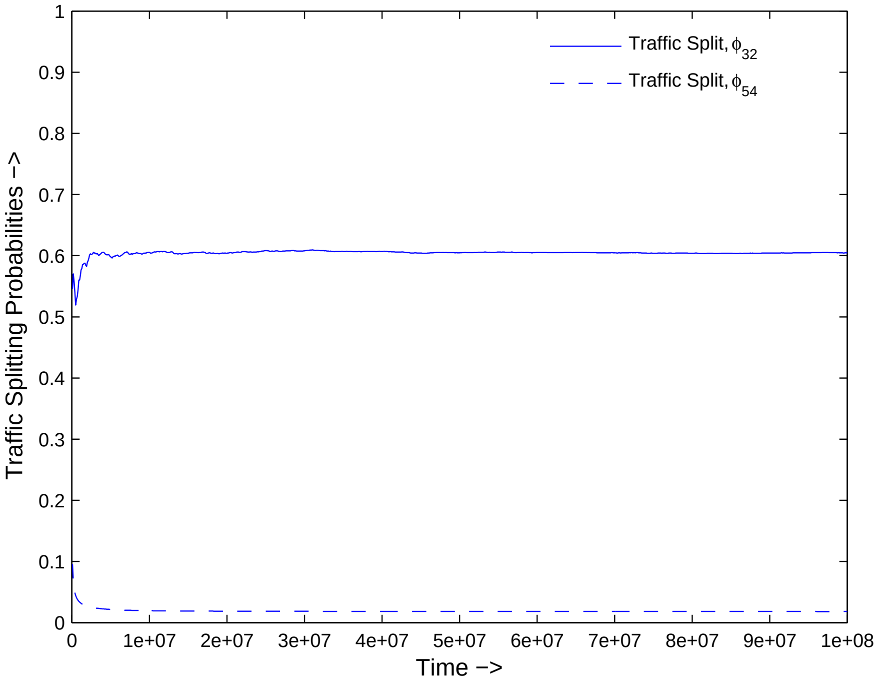

Figure 2.19: Traffic split over the routes 3 → 1 → 0, 3 → 2 → 0, 5 → 1 → 0, 5 → 4 → 0 for closed system.

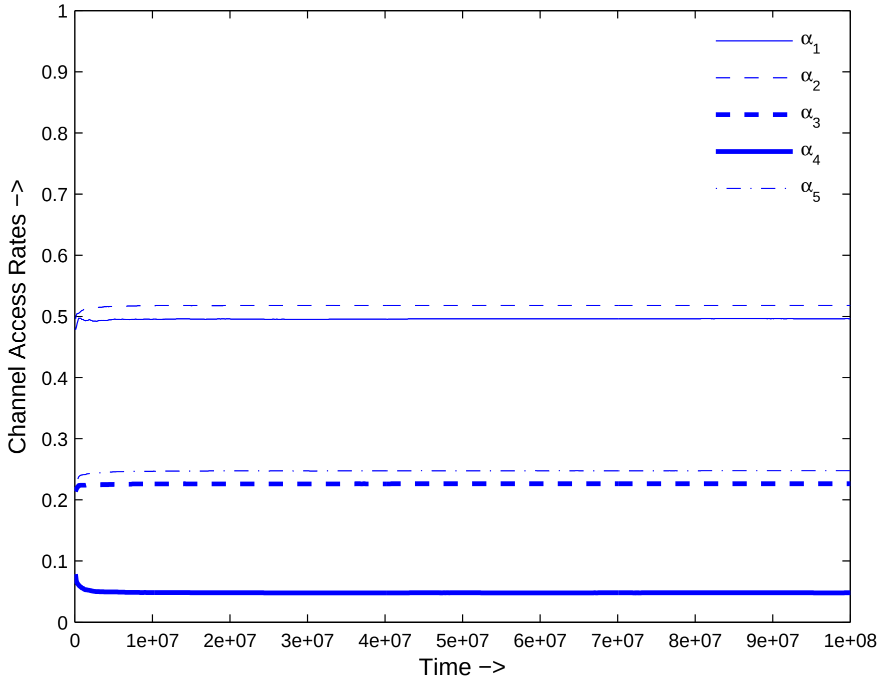

Figure 2.20: Convergence of *channel access rates* for closed system.

Table 2.3: Results on Throughput and Stability Region

$\alpha' s$	Throughput			y		
Nodes→	1	2	4	1	2	4
0.0	0.00	0.00	0.00	1.0	1.0	1.0
0.1	0.50	0.48	0.51	0.88	0.91	0.90
0.2	0.80	0.75	0.79	0.80	0.82	0.85
0.3	0.95	0.92	0.96	0.70	0.71	0.74
0.4	0.85	0.83	0.85	0.72	0.75	0.78
0.5	0.64	0.60	0.62	0.76	0.80	0.82
0.6	0.45	0.45	0.40	0.81	0.83	0.85
0.7	0.32	0.29	0.30	0.85	0.86	0.89
0.8	0.10	0.11	0.11	0.97	0.98	1.00
0.9	0.01	0.02	0.01	1.00	1.00	1.00
1.0	0.00	0.00	0.00	1.00	1.00	1.00

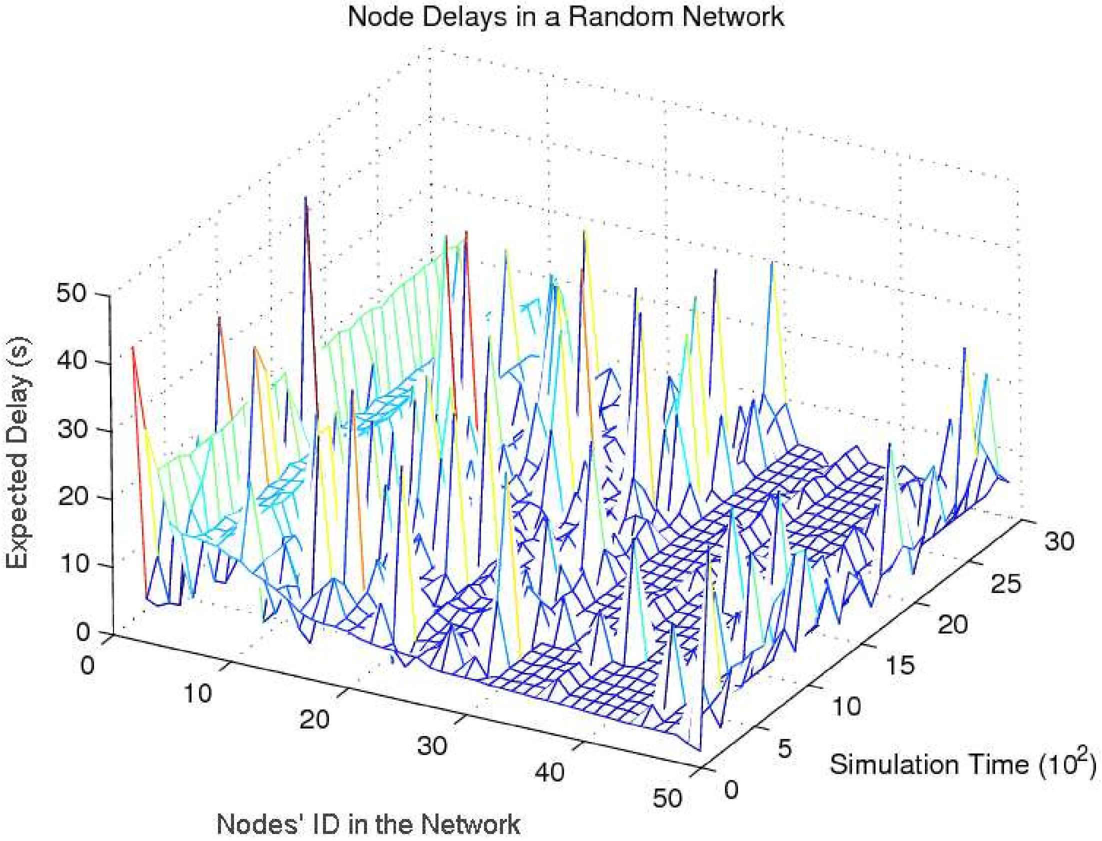

Figure 2.21: Expected Delay in a randomly deployed network over time

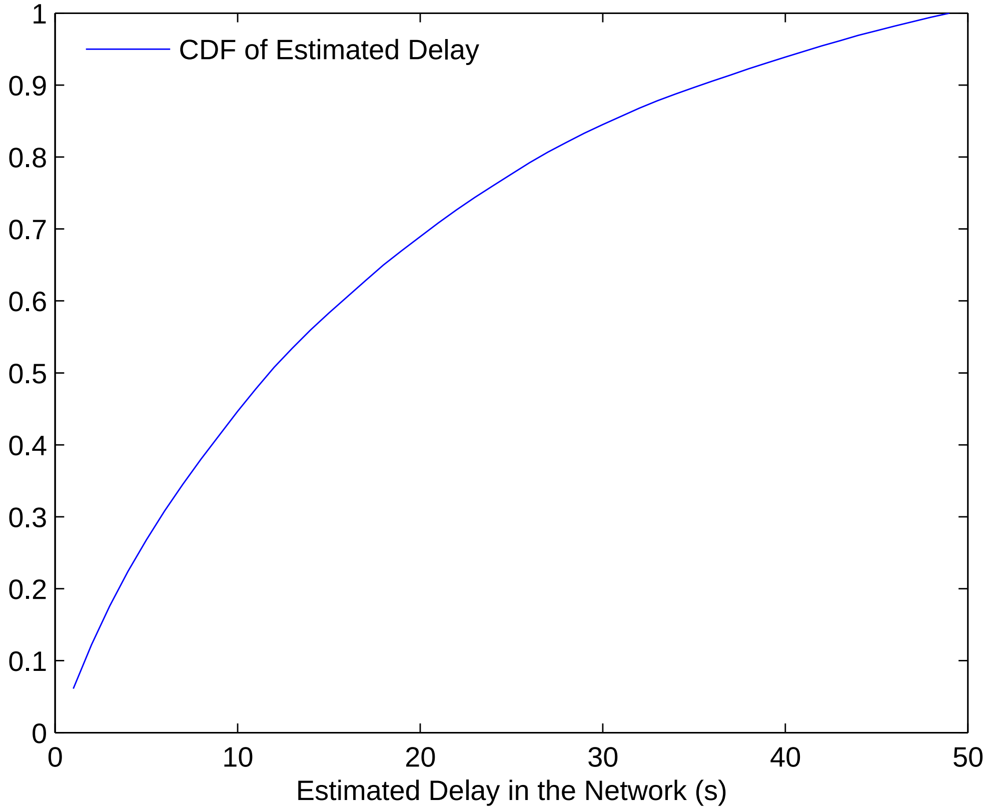

Figure 2.22: CDF of the Estimated Delay in a randomly deployed network

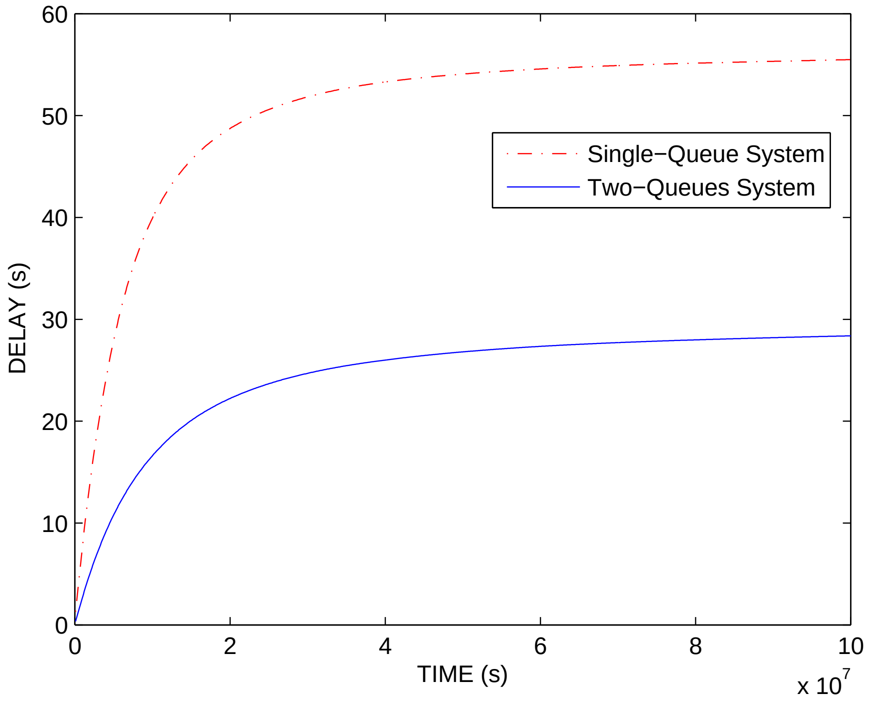

Figure 2.23: Average Delays for Two-Queues Vs. Single Queue System

Chapter 3

Cross-layer Routing in SANETs

SANETs are becoming increasingly important in recent years due to their ability to detect and convey real-time, in-situ information for many military and civilian applications. A fundamental challenge for such networks lies in the *hard* delay constraint, which poses a performance limit on achievable actuation dynamics.

In this chapter, we consider a two-tier wireless sensor-actuator network and address the *minimum* delay problem for data aggregation. We consider the layered architecture introduced in Section 2.4 of Chapter 2. We analyze the *average* delay for this architecture in the network. The *objective* then is to minimize this delay in the network. We prove that the objective function is *strictly* convex for the entire network. We then provide a *distributed* optimization framework to achieve the required objective. The approach is based on distributed *convex* optimization and *deterministic* distributed algorithm without feedback control. Only *local* knowledge is used to update the algorithmic steps. Specifically, we formulate the objective as a *network* level delay *minimization* function where the constraints are the *reception-capacity* and *service-rate* probabilities. Using the Lagrangian *dual* composition method, we derive a distributed *primal-dual* algorithm to minimize the delay in the network. We further develop a *stochastic* delay control primal-dual algorithm in the presence of *noisy* conditions. We also present its *convergence* and *rate* of convergence properties.

This chapter also investigates an optimal actuator *selection* problem for wireless sensor-actuator networks. A sensor can transmit its readings to any near-by actuators. It is proposed that each sensor *must* transmit its locally generated data to only one of the actuators. A polynomial time algorithm is proposed for *optimal* actuator *selection.* We finally propose a distributed mechanism for actuation *control* which covers all the requirements for an *effective* actuation process.

3.1 Introduction

Distributed systems based on networked sensors and actuators with embedded computation capabilities enable an instrumentation of the physical world at an unprecedented scale and density, thus enabling a new generation of monitoring and control applications. SANETs are an emerging technology that has a wide range of potential applications including environment monitoring, medical systems, robotic exploration, and smart spaces. Such networks consist of large number of distributed sensor and few actuator nodes that organize themselves into a multihop wireless network. Each sensor node has one or more sensors (including multimedia, e.g., video and audio, or scalar data, e.g., temperature, pressure, light, infrared, and magnetometer), embedded processors, low-power radios, and is normally battery operated. Typically, these nodes coordinate to perform a common task. Whereas, the actuators gather this information and react accordingly.

SANETs have the following unique characteristics:

- Real-time requirement: Depending on the application there may be a need to rapidly respond to sensor input. Examples can be a fire application where actions should be initiated on the event area as soon as possible.

- Coordination: Unlike WSNs where the central entity (i.e., sink) performs the functions of data collection and coordination, in SANETs, new networking phenomena called sensor-sensor, sensor-actuator and actuator-actuator coordination may occur. In particular, sensor-sensor coordination deals with local collaboration among neighbors to perform in-network aggregation and exploit correlations (both spatial and temporal). Sensor-actuator coordination provides the transmission of event features from sensors to actuators. After receiving event information, actuators may need to coordinate (actuator-actuator coordination) with each other (depend on the acting application) in order to make decisions on the most appropriate way to perform the actions.

From the above requirements, it is evident that delay constraints have a significant impact on the design and operation (i.e., coordination and actuation dynamics) of wireless sensor-actuator networks. The objective is to minimize the total delay in the network. We use the Lagrangian dual decomposition method to devise a distributed primal-dual algorithm to minimize the delay in the network. The deterministic distributed primal-dual algorithm requires no feedback control and therefore converges almost surely to the optimal solution. It is important to pay equal attention to both the observed delay in the network and energy consumption for data transmissions. A fast convergence means that only a small amount of energy is consumed to perform local calculations to achieve the desired optimizations. We also develop a *stochastic* delay control primal-dual algorithm in the presence of *noisy* conditions. We then propose a *selection* algorithm to select an optimal actuator using which each sensor can find an optimal actuator in *polynomial* time. Once the destination actuators are fixed, we use the distributed routing algorithm that achieves a Wardrop equilibrium, proposed in Section 2.6 of Chapter 2. We finally propose a *distributed* actuation control mechanism that takes into account all the possibilities for an efficient *actuation* process suitable to a wide range of sensor-actuator applications.

The organization of this chapter is as follows. Some related literature is presented in Section 3.2. In Section 3.3, we provide the necessary motivation for the problem under consideration and present its formulation. Section 3.4 details the complete network and traffic model. The open system (layered architecture) optimization is presented in Section 3.5. In Section 3.6, we present the stochastic delay control algorithm under noisy conditions. The rate of convergence of this algorithm is presented in Section 3.7. The limitations of sensor-actuator coordination are detailed in Section 3.8. We also present our algorithm for optimal actuator selection and provide the working dynamics of our distributed routing algorithm, which converges to a Wardrop equilibrium. In Section 3.9, we detail the guidelines for efficient actuator-actuator coordination and distributed actuation process for sensor-actuator applications. In Section 3.10, we present some TinyOS simulation results. Section 3.11 concludes the chapter and outlines the future work.

3.2 Related Literature

In [55], the authors proposed an efficient routing protocol for WSNs with global objective set to maximize network lifetime. The constraints are set to minimize the energy consumption for efficient data aggregation. The protocol works by building gradients along an interest propagation. In short, interest propagation sets up state in the network (or parts thereof) to facilitate "pulling down" data toward the sink. The results provided therein have shown significant improvement over traditional routing protocols both in terms of communication and computational load. Whereas in [65], the authors use the same approach as [55] for sensor-actuator networks using anycast routing. A reverse tree-based anycast routing is proposed, which constructs a tree routed at the event source, where sensors can join and leave dynamically. The introduction of actuators in the existing WSNs has opened up a new dimension of "a hard delay constraint" while still looking for near-optimal network lifetime solutions [58]. For example, Targeting an intruder holding a sniper in a surveillance field can be an interesting case to consider. The actuation process has to localize the position of the intruder and actuate the destruction process. The important constraint in this case is the latency of the received data because the sensor data can be no more valid at the time of actuation in case of increased latency.

A well designed application-specific coordination protocol is proposed [66], where cluster formation is triggered by an event so that clusters are created on-the-fly to optimally react to the event itself and provide the reliability with minimum energy expenditure. In order to provide effective sensing and acting, an efficient and distributed coordination mechanism is required for delay-energy aware dissemination of information, and to perform right and timely actions. Therefore, we proposed to establish these clusters once during the initial network deployment and the routing protocol can disseminate the sensed information to the actuators through maximum remaining energy paths. After receiving the event information, actuators may need to coordinate with each other in order to make decisions on the most appropriate way to perform the required action. Depending on the application, there can be multiple actuators interested in some information. Therefore, sensors need to transmit this data toward multiple actuators, which results in excess sensor-energy drain due to multiple transmissions of redundant information [67]. Moreover, the collected and transmitted sensor data must be valid at the time of acting. For example, if sensors detect a malicious person in an area and transmit this information to its optimal actuator; and the act of disposing a tranquilizing gas must find that person in the very same area. Therefore, the issue of real-time communication is very important in SANETs.

Most of the current research on sensor systems is mainly focused toward optimizing the network lifetime (e.g., [129]) and the energy consumption of the sensors bypassing the delay-sensitivity of sensor data for real time applications. In [68], the authors presented a detailed overview of the routing techniques proposed for WSNs. The routing techniques are classified into three categories based on the underlying network structure: flat, hierarchical, and location-based routing. The hierarchical routing schemes have shown a promising improvement for prolonging network lifetime [41]. An enhancement in basic LEACH is proposed in [42], where the network lifetime has been extended by the introduction of closest neighbor communication. In [69, 140], the network lifetime was prolonged on the basis of threshold-sensitive routing schemes. All of these protocols share a common problem: routing semantics binded to application requirements.

In case of multiple basestation, if the mapping between a sensor node and one (or more) basestation is given *a priori*, then the problem of finding optimal minimum energy routes to optimize network lifetime has been investigated in [26, 41] for WSNs. In [25], the authors propose to split the flow from a source sensor and transmit it toward multiple sinks in order to extend the lifetime. But there is very little research contribution toward finding optimal delay routes in wireless sensor-actuator networks.

In this work, we optimize all the three types coordination that occur in sensor-actuator networks namely: sensor-sensor coordination, sensor-actuator coordination, and actuator-actuator coordination. We propose that each sensor should transmit its measurements to only destination actuator in order to conserve energy. We then proposed a closed architecture (using a cross-layering approach) for a broad class of wireless sensor-actuator networks, which gives a stable operating region as well as minimize the end-to-end delays. We also propose an optimal actuator selection algorithm that provides a good mapping between each sensor and a destination actuator for a delay efficient actuation mechanism. Our routing algorithm is an adaptation of the algorithm proposed in 2.6 to the case of SANETs. In this algorithm, each source uses a two time-scale stochastic approximation algorithm. We finally propose a distributed actuation mechanism that covers a wide range of requirements for distributed actuation dynamics in sensor-actuator applications.

3.3 Problem Statement

If the mapping between a sensor node and one (or more) basestation/actuator[1] is given *a priori*, then the problem of finding optimal minimum energy routes to optimize network lifetime has been well investigated in the past [25, 26] for WSNs. But there is very little research contribution toward finding optimal delay routes in wireless sensor-actuator networks. Further, in cases when there are multiple actuators and mapping between the sensors and actuators is not given, the joint problem of finding an optimal actuator and minimum end-to-end delay routes is a challenging and interesting problem. In order to provide effective sensing and acting tasks, efficient coordination mechanisms are required. We will mainly focus on two *most* constrained coordination levels namely: sensor-actuator coordination, and actuator-actuator coordination. Whereas the sensor-sensor coordination, in particular, requires atmost two-hop neighborhood information to exploit correlations and aggregation dynamics (a minor optimization that could be achieved at this coordination level is the optimization of search space over neighborhood sets as flows are directional in sensor-actuator networks). There is a need for an analytic framework in order to characterize the management, coordination and communication issues. Sophisticated distributed coordination algorithms need to be developed for effective sensing and acting tasks. Leveraging a cross-layer approach can provide much more effective sensing, data transmission, and acting in SANETs. Several cross-layer integration issues among the communication layers should be investigated in order to improve the overall efficiency of SANETs.

In this chapter, we consider the layered architecture introduced in Section 2.4 of Chapter 2. We analyze the *average* end-to-end delay in the network. The objective is to minimize the *total* delay in the network. We prove that this objective function is strictly *convex* for the entire network. We then provide a *distributed* optimization framework to achieve the required objective. The approach is based on distributed convex optimization and deterministic

[1] Actuators/basestations have similar semantics for modeling purposes, i.e., sinks for data generated in the network.

distributed algorithm without feedback control. Only local knowledge is used to update the algorithmic steps. Specifically, we formulate the objective as a network level delay minimization function where the constraints are the *reception-capacity* and *service-rate* probabilities. Using the Lagrangian dual composition method, we derive a distributed primal-dual algorithm to minimize the delay in the network. We further develop a *stochastic delay-control* primal-dual algorithm in the presence of noisy conditions. We also present its convergence and rate of convergence properties.

This chapter also investigates a *delay-optimal* actuator-selection problem for SANETs. Each sensor must transmit its locally generated data to only one of the actuators. We propose a polynomial time algorithm for actuator selection. Our algorithm, called "AS: *Actuator Selection*" for optimal routing actuator selection, is based on the conjecture that the optimal actuator for a sensor node should be closely related to the actuator that provides minimum *end-to-end* delay, when there is no constraint on the number of destination actuators. Once, the destination actuators are fixed, we use the adaptive and distributed routing scheme, introduced in Section 2.6 of Chapter 2, for a general class of SANETs. We finally propose a distributed actuation mechanism for a wide range of sensor-actuator applications. Depending upon the acting application, the actuation expectation reflects the most appropriate fashion to carry out the acting tasks.

3.4 The Network Model

We consider a static wireless sensor-actuator network with N sensor and M actuator nodes as shown in Figure 3.1.

3.4.1 Channel Model and Antennas

We assume a simple channel model: a node can decode a transmission successfully iff there is no other interfering transmission. Each sensor node is equipped with an omni-directional antenna. Whereas, the actuators can be equipped with two antennas; one to communicate with sensor network (sensor-actuator coordination), and the other to communicate with actuator network (actuator-actuator coordination) for a fast and effective actuation process.

3.4.2 Frequency

Assume that all sensor nodes share the same frequency band, whereas one of the antenna interface at the actuators share the same frequency band as the sensor network while the other might utilize a different frequency to communicate with the network of actuators. The time is divided into fixed length slots. All the packets are of same length and the length of a time slot corresponds to the time required to transmit a packet over the underlying wireless channel.

3.4.3 Neighborhood Relation Model

Given is an $(N+M)\times(N+M)$ neighborhood relation matrix $\mathcal{N}$ that indicates the node pairs for which direct communication is possible. We will assume that $\mathcal{N}$ is a symmetric matrix, i.e., if node i can transmit to node j, then j can also transmit to node i. For such node pairs, the $(i,j)^{th}$ entry of the matrix $\mathcal{N}$ is unity, i.e., $\mathcal{N}_{i,j}=1$ if node i and j can communicate with each other; we will set $\mathcal{N}_{i,j} = 0$ if nodes i and j can not communicate. For any node i, we define $\mathcal{N}_i = \{j : \mathcal{N}_{i,j} = 1\}$, which is the set of neighboring nodes of node i.

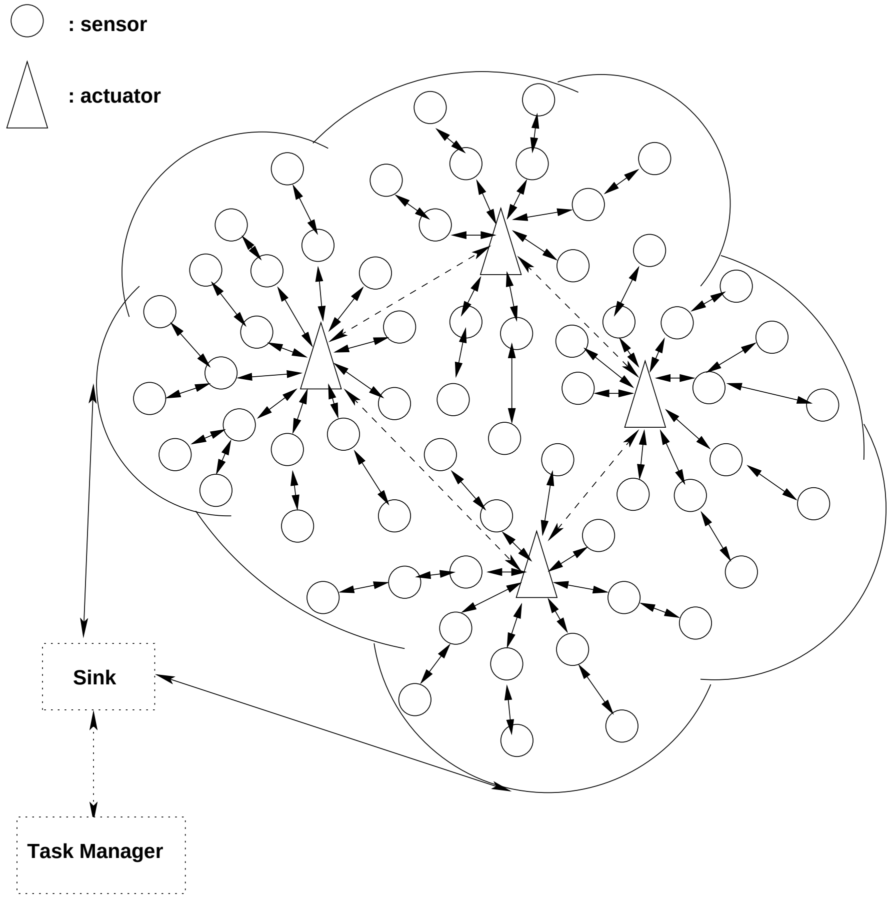

Figure 3.1: Architecture of Sensor-Actuator Networks

3.4.4 Application-layer Sampling-Mechanism

Each sensor node is assumed to be sampling (or, sensing) its environment at a predefined rate; we let λ_i denote this sampling rate for node i. The units of λ_i will be $pkts/s$, assuming same packet size for all the sensors in the network. In this work, we will assume that the readings of each of these sensor nodes are statistically independent of each other so that distributed compression techniques are not employed (see [30] for an example where the authors exploit the correlation among readings of different sensors to use distributed Slepian-Wolf Coding [31] to reduce the overall transmission rate of the network).

3.4.5 Relaying

Each sensor node wants to use the sensor network to forward its sampled data to a *common* fusion center (assumed to be a part of the network (conceptually, we can assume that this fusion center is also a sensor node, which has 0 sampling rate)). Thus, each sensor node acts as a relay for other sensor nodes in the network. We will assume that the buffering capacity of each node is infinite, so that there is no data loss in the network. We will allow for the possibility that a sensor node discriminates between its own packets and the packets to be forwarded.

3.4.6 Traffic Model

We let ϕ denote the $(N+M)\times(N+M)$ routing matrix. The $(i,j)^{th}$element of this matrix, denoted $\phi_{i,j}$, takes value in the interval $[0,1]$. This means a probabilistic flow splitting (flow splitting provides an extra degree of freedom to utilize available routes in a fair manner) as in the model of [33], i.e., a fraction $\phi_{i,j}$ of the traffic *transmitted* from node i is forwarded by node j. Clearly, we need that ϕ is a stochastic matrix, i.e., its row elements sum to unity. Also note that $\phi_{i,j} > 0$ is possible only if $\mathcal{N}_{i,j} = 1$. *Our objective in this chapter is to come up with a distributed algorithm (after fixing the destination actuator) using which any node (say i) is able to converge to the corresponding row of the matrix ϕ corresponding to the Wardrop equilibrium.*

3.4.7 Channel Access Mechanism

We assume that the system operates in discrete time, so that the time is divided into (conceptually) fixed length slots. We also assume that the packet length (or, transmission schedule length) is fixed throughout system operation. The system operates on CSMA/CA like MAC. Assuming that there is no exponential *back-off*, the channel access rate of node i (if it has a packet waiting to be transmitted) is $0 \leq \alpha_i \leq 1$ (to avoid pathological cases). Thus, α_i is the probability that node i, if it has a packet to be transmitted, attempts a transmission in any slot. A node can receive a transmission from its neighbor if it is not transmitting and also no other neighboring node is transmitting, i.e., if the transmission is meant for some node j, $j \in \mathcal{N}_i$, then the transmission from node i to node j is successful iff none of the nodes in the set $j \cup \mathcal{N}_j \backslash i$ transmits.

3.5 Optimization Problem for Open System

We will call a routing matrix feasible if the following constraint is met

$$\sum_{1<j\leq n} \lambda_j = a_1,$$

where, without loss of generality, we have given an index 1 to the fusion center. This requirement says that all the data generated in the network must end-up at the fusion center [C-12]. We have the following consideration now: Minimize the total delay in the network

$$\sum_i w_i \frac{1}{1-\frac{a_i}{\mu_i}}\left(1-\frac{a_i}{2\mu_i}\right) \tag{3.1}$$

where we have used the average delay formula for our system with mean service requirement of unity. Here $w_i > 0$ is a weight given to the node i, for example, the node close to fusion center may be heavily loaded, hence we may want to give more attention to this node. Here $\frac{a_j}{\mu_j}$ is the load on node j. s.t. $\sum_i w_i\,(\mu_i - a_i)$, which says, maximize the difference between the service rate and the arrival rate into any node, while in the stable region. It is important to be noted that we first fix the routing in the network, and thus, fixing the arrival rate at each node. We then look at the optimization criteria assuming the network is operating in the stable region. We thus want to maximize the system performance while in the stable region.

We first consider the delay minimization objective function

$$\min \sum_i w_i \frac{1}{1-\frac{a_i}{\mu_i}}\left(1-\frac{a_i}{2\mu_i}\right) = \sum_i w_i \left(1+\frac{1}{2}a_i x_i\right) \tag{3.2}$$

where $x_i = \frac{1}{\mu_i - a_i}$.

First of all, we will prove that x_i is a convex function. Let $f\left(\mu_i\right)$ be a function of a single variable defined on the interval I, then $f\left(\mu_i\right)$ is convex, if for all $a \in I$, all $b \in I$, and all $t \in [0,1]$, we have

$$f\left(\underline{c}\right) - tf\left(\underline{a}\right) - \overline{t}f\left(\underline{b}\right) \leq 0$$

where $\overline{t} = 1 - t$. Also, we assume the feasible region of x_i is $[a, b]$. We have

$$\underline{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_i \end{bmatrix}, \underline{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_i \end{bmatrix} \Rightarrow \underline{c} = t\underline{a} + \overline{t}\underline{b} = \begin{bmatrix} ta_1 + \overline{t}b_1 \\ \vdots \\ ta_i + \overline{t}b_i \end{bmatrix}$$

Then, we need to prove that

$$f\left(t\underline{a} + \overline{t}\underline{b}\right) - tf\left(\underline{a}\right) - \overline{t}f\left(\underline{b}\right) \leq 0$$

or

$$f\left(\underline{c}\right) - tf\left(\underline{a}\right) - \overline{t}f\left(\underline{b}\right) \leq 0$$

n-Node Example: Let us consider an example of n-nodes and use this method to prove convexity. According to the objective function (3.2), we can rewrite it as

$$f\left(\underline{\mu}\right) = \underline{w}^{\mathrm{T}}\left(\mathrm{I} + \frac{1}{2}\underline{ax}\right)$$

where$\underline{\mu} = \left[\mu_1 \cdots \mu_i\right]^{\mathrm{T}}$, $\underline{x} = \left[\frac{1}{\mu_1 - a_1} \cdots \frac{1}{\mu_i - a_i}\right]^{\mathrm{T}}$, $\underline{w} = \left[w_1 \cdots w_i\right]^{\mathrm{T}}$ and $\underline{a} = \left[a_1 \cdots a_i\right]^{\mathrm{T}}$. For the ease of understanding in the proof, here we use q_i replacing a_i in (3.1). So

$$f\left(\underline{\mu}\right) = \underline{w}^{\mathrm{T}}\left(\mathrm{I} + \frac{1}{2}\underline{q}\underline{x}\right) = \underline{w}^{\mathrm{T}}\left[\mathrm{I} + \frac{1}{2}\underline{q}\left(\underline{\mu} - \underline{q}\right)^{-1}\right]$$

where $\underline{x} = \left[\frac{1}{\mu_1 - q_1} \cdots \frac{1}{\mu_i - q_i}\right]^{\mathrm{T}}$.

Therefore, we can write $f\left(\underline{a}\right)$ and $f\left(\underline{b}\right)$ as follows

$$\begin{aligned} f\left(\underline{a}\right) &= \underline{w}^{\mathrm{T}}\left[\mathrm{I} + \tfrac{1}{2}\underline{q}\left(\underline{a} - \underline{q}\right)^{-1}\right] \\ f\left(\underline{b}\right) &= \underline{w}^{\mathrm{T}}\left[\mathrm{I} + \tfrac{1}{2}\underline{q}\left(\underline{b} - \underline{q}\right)^{-1}\right] \end{aligned} \tag{3.3}$$

Similarly, we can write

$$f\left(\underline{c}\right) = \underline{w}^{\mathrm{T}}\left[\mathrm{I} + \frac{1}{2}\underline{q}\left(t\underline{a} + \bar{t}\underline{b} - \underline{q}\right)^{-1}\right] = \underline{w}^{\mathrm{T}}\left[\mathrm{I} + \frac{1}{2}\underline{q}\left(\underline{b} - \underline{q} + t\left(\underline{a} - \underline{b}\right)\right)^{-1}\right] \tag{3.4}$$

where $\mathrm{I} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$. And for $\underline{a}$ and $\underline{b}$, we have

$$tf\left(\underline{a}\right) + \bar{t}f\left(\underline{b}\right) = \underline{w}^{\mathrm{T}}\mathrm{I} + \frac{1}{2}\underline{w}^{\mathrm{T}}\underline{q}\left[t\left(\underline{a} - \underline{q}\right)^{-1} + \bar{t}\left(\underline{b} - \underline{q}\right)^{-1}\right] \tag{3.5}$$

Therefore, we need to prove that

$$f\left(\underline{c}\right) - tf\left(\underline{a}\right) - \bar{t}f\left(\underline{b}\right) \leq 0 \tag{3.6}$$

By plugging (3.4) and (3.5) into (3.6) and solving, we get

$$\begin{aligned} f\left(\underline{c}\right) - tf\left(\underline{a}\right) - \bar{t}f\left(\underline{b}\right) = \tfrac{1}{2}\underline{w}^{\mathrm{T}}\underline{q}\Big[\left(\underline{b} - \underline{q} + t\left(\underline{a} - \underline{b}\right)\right)^{-1} - \\ t\left(\underline{a} - \underline{q}\right)^{-1} - \bar{t}\left(\underline{b} - \underline{q}\right)^{-1}\Big] \end{aligned} \tag{3.7}$$

Assume: $A = \underline{a} - \underline{q}$, $B = \underline{b} - \underline{q}$. Then

$$\begin{aligned} &\left(\underline{b} - \underline{q} + t\left(\underline{a} - \underline{b}\right)\right)^{-1} - t\left(\underline{a} - \underline{q}\right)^{-1} - \bar{t}\left(\underline{b} - \underline{q}\right)^{-1} \\ &= \tfrac{1}{B+t(A-B)} - \tfrac{t}{A} - \tfrac{\bar{t}}{B} \\ &= \tfrac{1}{tA+\bar{t}B} - \tfrac{\bar{t}A+tB}{AB} \\ &= \tfrac{AB-\left(\bar{t}\cdot A+B\cdot t\right)\left(A\cdot t+\bar{t}\cdot B\right)}{AB\left(t\cdot A+\bar{t}\cdot B\right)} \\ &= \tfrac{Q_1}{Q_2} \end{aligned} \tag{3.8}$$

and

$$\begin{aligned} Q_1 &= AB - \left(\bar{t}\cdot t\left(A^2+B^2\right) + AB\left(\bar{t}^2+t^2\right)\right) \\ &= AB - \left[\left(t-t^2\right)\left(A^2-B^2\right) + \left(1-2t+2t^2\right)AB\right] \\ &= -\left(t-t^2\right)\left[A^2+B^2+2AB\right] \\ &= -\left(t-t^2\right)\left(A+B\right)^2 \end{aligned}$$

therefore $Q_1 \leq 0$. Also, $Q_2 = AB\left(t\cdot A + \bar{t}\cdot B\right) > 0$

Therefore we have proved that

$$f\left(\underline{c}\right) - tf\left(\underline{a}\right) - \bar{t}f\left(\underline{b}\right) \leq 0$$

The proof is complete and is for any number of nodes $n \in [1, N]$. We have shown that the function $\min\sum_i w_i \frac{1}{1-\frac{a_i}{\mu_i}}(1-\frac{a_i}{2\mu_i})$ is a strictly convex function in μ_i. Therefore, we now use the Lagrange method to find the optimal value of μ_i. We want to $\min\sum_i w_i\left(1+\frac{1}{2}a_i x_i\right)$, where $x_i = \frac{1}{\mu_i - a_i}$ s.t. $\mu_i > a_i$. Let

$$f\left(\cdot\right) = \min\sum_i w_i\left(1+\frac{1}{2}a_i x_i\right)$$

Then the Lagrange of $f(\cdot)$ is

$$L = \min \sum_i w_i \left[1 + \frac{1}{2} a_i \left(\frac{1}{\mu_i - a_i} \right) \right] + \sum_i \lambda_i (\mu_i - a_i)$$

$$\frac{\delta L}{\delta \mu_i} = \begin{bmatrix} \frac{\delta L}{\delta \mu_1} \\ \vdots \\ \frac{\delta L}{\delta \mu_n} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} w_1 a_1 \left(\frac{1}{\mu_1 - a_1} \right)^2 + \lambda_1 \\ \vdots \\ -\frac{1}{2} w_n a_n \left(\frac{1}{\mu_n - a_n} \right)^2 + \lambda_n \end{bmatrix} = 0$$

$$\frac{\delta L}{\delta \mu_i} = -\frac{1}{2} w_i a_i \left(\frac{1}{\mu_i - a_i} \right)^2 + \lambda_i = 0$$

$$\frac{1}{2} w_i a_i = \lambda_i (\mu_i - a_i)^2$$

$$\lambda_i \mu_i^2 - 2\lambda_i \mu_i a_i - \frac{1}{2} w_i a_i + \lambda_i a_i^2 = 0$$

$$\mu_i = \frac{2\lambda_i a_i \pm \sqrt{(2\lambda_i a_i)^2 - 4\lambda_i \left(\frac{1}{2} w_i a_i + \lambda_i a_i^2 \right)}}{2\lambda}$$

$$= a_i \pm \frac{1}{2\lambda_i} \sqrt{4\lambda_i^2 a_i^2 + 2\lambda_i w_i a_i - 4\lambda_i^2 a_i^2}$$

$$= a_i \pm \frac{1}{2\lambda_i} \sqrt{2\lambda_i w_i a_i}$$

We suppose that $\lambda_i > 0$, because then the condition $\mu_i > a_i$ is satisfied.

$$\mu_i^* = a_i + \underbrace{\sqrt{\frac{w_i a_i}{2\lambda_i}}}_{> 0}$$

As discussed before, w_i is the weight given to the node i. We want to give a higher weight to those nodes that are heavily loaded so that they can have a higher priority for transmissions over nodes that are not. Here is how we calculate the weight w_i for each node i

$$w_i = \frac{F_i}{N_i} a_i$$

Here, F_i is the set of neighboring nodes that are transmitting data to the node i, N_i is the entire set of one-hop neighbors of node i, and a_i is the total arrival rate into node i. Here we take into consideration the total load on the node i in terms of arrival rate a_i along with the neighborhood of the node. Because, we do not want to *destabilize* the *neighborhood* of a node by assigning it a *high* priority over transmissions. Therefore, we consider both the load a_i and the neighborhood $\frac{F_i}{N_i}$ of node i while assigning weight in order to be fair *locally.*

3.5.1 Lagrange Dual Approach

In what follows, we use the Lagrange dual *decomposition* method to solve the minimization problem. The Lagrangian function with the Lagrange multipliers (λ_i) is given as follows:

$$L(\mu, \lambda) = \sum_i w_i \left(1 + \frac{1}{2} a_i x_i\right) + \sum_i \lambda_i (\mu_i - a_i)$$

where $\mu = \{\mu_i,\ i = 1, \ldots, n\}$ and $\lambda = \{\lambda_i,\ i = 1, \ldots, n\}$.
Then, the Lagrange dual function is

$$Q(\lambda) = \min_{\mu} L(\mu, \lambda)$$

Thus, the dual problem is given by

$$D : \max_{\lambda > 0} Q(\lambda)$$

3.5.2 Deterministic Primal-Dual Algorithm

The delay minimization problem can be solved *via* the following deterministic distributed algorithm

- The $\mu_i' s$ are updated by

$$\mu_i(n+1) = \mu_i(n) - \epsilon_n \nabla_{\mu_i} L(\mu(n), \lambda(n))$$

- The Lagrange multipliers are update by

$$\lambda_i(n+1) = \lambda_i(n) + \epsilon_n \nabla_{\lambda_i} L(\mu(n), \lambda(n))$$

where

1.

$$\begin{aligned}\nabla_{\mu_i} L &= \frac{\delta}{\delta \mu_i}\left[\sum_j w_j \left(1 + \frac{1}{2} a_j x_j\right) + \sum_j \lambda_j (\mu_j - a_j)\right] \\ &= w_i \frac{1}{2} a_i \frac{-1}{(\mu_i - a_i)} + \lambda_i = \frac{-w_i a_i}{2(\mu_i - a_i)^2} + \lambda_i\end{aligned}$$

2.

$$\begin{aligned}\nabla_{\lambda_i} L &= \frac{\delta}{\delta \lambda_i}\left[\sum_j w_j \left(1 + \frac{1}{2} a_j x_j\right) + \sum_j \lambda_j (\mu_j - a_j)\right] \\ &= \mu_i - a_i\end{aligned}$$

We note that in the above algorithm, we have used the same step size $\epsilon_n$for both the primal and the dual algorithms. We can finally write the Primal-Dual algorithm as follows

$$\mu_i(n+1) = \mu_i(n) + \epsilon_n \left(\frac{w_i(n)\, a_i(n)}{2(\mu_i(n) - a_i(n))} - \lambda_i(n)\right)$$

$$\lambda_i(n+1) = \lambda_i(n) + \epsilon_n (\mu_i(n) - a_i(n))$$

3.6 Stochastic Delay Control And Stability Under Noisy Conditions

In this section, we examine the *convergence* performance of the above distributed algorithms under *stochastic* perturbations, due to noisy feedback information.

3.6.1 Stochastic Primal-Dual Algorithm For Delay Control

In the presence of noisy feedback information, the *gradients* are *estimators*. More specifically, the stochastic version of the primal-dual algorithm is given as follows

$$\begin{aligned} \mu_i(n+1) &= \mu_i(n) - \epsilon_n \cdot \hat{L}_{\mu_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) \\ \varphi_i(n+1) &= \varphi_i(n) + \epsilon_n \cdot \hat{L}_{\varphi_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) \end{aligned} \tag{3.9}$$

where $\hat{L}_{\mu_i}$ is an estimator of $\nabla_{\mu_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))$ and $\hat{L}_{\varphi_i}$ is an estimator of $\nabla_{\varphi_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))$.

3.6.2 Probability One Convergence Of Stochastic Delay Control Algorithm

Next, we examine in detail the models for stochastic perturbations. Let $\{\mathcal{F}_n\}$ be a sequence of σ-algebras generated by $\{(\mu_i(m), \varphi_i(m)), \forall m \leq n\}$. For convenience, we use $\mathrm{E}_n[\cdot] = \mathrm{E}[\cdot|\mathcal{F}_n]$ to denote the conditional expectation.

1. Stochastic gradient $\hat{L}_{\mu_i}$

   Observe that

$$\hat{L}_{\mu_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) = \nabla_{\mu_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) + \alpha_i(n) + \zeta_i(n)$$

   where

$$\begin{aligned} \alpha_i(n) &\triangleq \mathrm{E}_n\left[\hat{L}_{\mu_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))\right] - \nabla_{\mu_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) \\ \zeta_i(n) &\triangleq \hat{L}_{\mu_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) - \mathrm{E}_n\left[\hat{L}_{\mu_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))\right] \end{aligned} \tag{3.10}$$

   i.e. $\alpha_i(n)$ is the biased random error of $\nabla_{\mu_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))$ and $\zeta_i(n)$ is a martingale difference noise since $\mathrm{E}_n[\zeta_i(n)] = 0$.

2. Stochastic gradient $\hat{L}_{\varphi_i}$

   Observe that

$$\hat{L}_{\varphi_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) = \nabla_{\varphi_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) + \beta_i(n) + \xi_i(n)$$

   where

$$\begin{aligned} \beta_i(n) &\triangleq \mathrm{E}_n\left[\hat{L}_{\varphi_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))\right] - \nabla_{\varphi_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) \\ \xi_i(n) &\triangleq \hat{L}_{\varphi_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) - \mathrm{E}_n\left[\hat{L}_{\varphi_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))\right] \end{aligned}$$

   i.e. $\beta_i(n)$ is the biased random error of $\nabla_{\varphi_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))$ and $\xi_i(n)$ is a martingale difference noise.

We impose the following standard assumptions in order to examine the *convergence* of the stochastic primal-dual algorithm:

A1. *We assume that the estimator of the gradients are based on the measurements in each iteration only.*

A2. *Condition on the step size:* $\epsilon_n > 0$, $\epsilon_n \to 0$, $\sum_n \epsilon_n \to \infty$ *and* $\sum_n \epsilon_n^2 < \infty$.

A3. *Condition on the biased error:* $\sum_n \epsilon_n |\alpha_i(n)| < \infty$ *and* $\sum_n \epsilon_n |\beta_i(n)| < \infty$, $\forall i$.

A4. *Condition on the martingale difference noise:* $\sum_n \epsilon_n \left[\zeta_i(n)^2\right] < \infty$ *and* $\sum_n \epsilon_n \left[\xi_i(n)^2\right] < \infty$, $\forall i$.

We have the following proposition:

Proposition: Under Conditions **A1** - **A4,** the iterates, generated by stochastic approximation algorithm (3.9), converge with probability one to the optimal solutions of the problem.

Sketch of the proof: The proof consists of two steps. First, using the stochastic Lyapunov Stability Theorem, we establish that the iterates generated by (3.9) return to a neighborhood of the optimal points *infinitely* often. Then, we show that the *recurrent* iterates eventually reside in an arbitrary small neighborhood of the optimal points, and this is proved by using *local analysis.* We use the following example to illustrate how to characterize sufficient conditions for the almost sure convergence of stochastic gradient algorithms.

We assume that the exponential marking technique is used to feedback the price information, φ_i, to the source nodes. Therefore the overall non-marking probability is that

$$p_i = \exp(\varphi_i)$$

To estimate the overall price, source i sends N_i packets during round n and counts the non-marked packets. For example, if K non-marked packets have been counted, then the estimation of the overall price $\hat{p}_i$ can be K/N_i. Therefore

$$\hat{L}_{\mu_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) = -\frac{w_i a_i}{2(\mu_i - a_i)^2} + \log(\hat{p}_i) \tag{3.11}$$

By the definition of (3.10), we have

$$\alpha_i(n) = \mathrm{E}_n\left[\log(\hat{p}_i)\right] - \log(p_i)$$

Note that K is a Binomial random variable with distribution $B(N_i, q)$. When N_s is sufficiently large, it follows that $\hat{p}_i \sim \aleph(p_i, p_i(1-p_i)/N_i)$ and $\hat{p}_i \in \left[P_i - c/\sqrt{N_i}, P_i + c/\sqrt{N_i}\right]$ with high probability, where c is a positive constant. Then the estimation bias of the price information can be upper-bounded as

$$|\alpha_i(n)| \leq \frac{c'}{\sqrt{N_i}}$$

for large N_i, where c' is some positive constant.

To ensure the convergence of primal-dual algorithm, from condition **A3,** it suffices to have that

$$\sum_n \frac{\epsilon_n}{\sqrt{N_i}} < \infty$$

Next, we discuss that the variance condition **A4** is satisfied for $\zeta_i(n)$. By (3.10) and (3.11),

$$\begin{aligned}
\mathrm{E}_n\left[\zeta_i(n)^2\right] &= \mathrm{E}_n\left[\hat{L}^2_{\mu_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))\right] - \mathrm{E}^2_n\left[\hat{L}_{\mu_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))\right] \\
&= \mathrm{E}_n\left[\log^2(\hat{p}_i)\right] - \mathrm{E}^2_n\left[\log(\hat{p}_i)\right] \\
&\leq \mathrm{E}_n\left[\log^2(\hat{p}_i)\right] \\
&\leq \mathrm{E}_n\left[\log^2(p_i + c)\right] \quad \forall N_i \gg 0
\end{aligned}$$

Similar studies can be done for $\beta_i(n)$ and $\xi_i(n)$.

3.7 Rate of Convergence of Stochastic Delay Control Algorithm

The rate of convergence is concerned with the *asymptotic* behavior of *normalized* errors about the optimal points. Our primal-dual algorithm can be rewritten as a general constrained form as follows:

$$\begin{aligned}&\left[\begin{array}{c}\mu_i(n+1)\\ \varphi_i(n+1)\end{array}\right]\\ =&\left[\begin{array}{c}\mu_i(n)\\ \varphi_i(n)\end{array}\right]+\epsilon_n\left[\begin{array}{c}-\nabla_{\mu_i}L(\boldsymbol{\mu}(n),\boldsymbol{\varphi}(n))\\ \nabla_{\varphi_i}L(\boldsymbol{\mu}(n),\boldsymbol{\varphi}(n))\end{array}\right]\\ &+\epsilon_n\left[\begin{array}{c}\alpha_i(n)+\zeta_i(n)\\ \beta_i(n)+\xi_i(n)\end{array}\right]+\epsilon_n\left[\begin{array}{c}Z_n^{\mu_i}\\ Z_n^{\varphi_i}\end{array}\right]\end{aligned}$$

where $\epsilon_n Z_n^{\mu_i}$ and $\epsilon_n Z_n^{\varphi_i}$are the correction term which force μ_i and φ_i to reside inside the constraint set. As is standard in the study on the rate of convergence, we assume that the iterates generated by the stochastic primal-dual algorithm have entered in a small neighborhood of an optimal solution (μ_i^*,φ_i^*).

To characterize the asymptotic properties, we define $U_{\mu_i}(n)\triangleq\left(\mu_i\left(n\right)-\mu_i^*\right)/\sqrt{\epsilon_n}$ and $U_{\varphi_i}(n)\triangleq\left(\varphi_i\left(n\right)-\varphi_i^*\right)/\sqrt{\epsilon_n}$, and we construct $U^n(t)$ to be the piecewise constant interpolation of $U(n)=\{U_{\mu_i}\left(n\right),U_{\varphi_i}\left(n\right)\}$, i.e., $U^n(t)=U_{n+1}$, for $t\in[t_{n+i}-t_n,t_{n+i+1}-t_n]$, where $t_n\triangleq\sum_{i=0}^{n-1}\epsilon_n$.

A5. Let $\theta(n)\triangleq(\mu_i(n),\varphi_i(n))$ and $\phi(n)\triangleq(\zeta(n),\xi(n))$. Suppose for any given small $\rho>0$, there exists a positive definite symmetric matrix $\Sigma=\sigma\sigma'$ such that

$$\mathrm{E_n}\left[\phi_\mathrm{n}\phi_\mathrm{n}^\mathrm{T}-\Sigma\right]I\left\{|\theta(n)-\theta^*|\le\rho\right\}\to 0$$

as $n\to\infty$. Where, the term $I\{...\}$ is actually a conditional *identity* matrix. It means that if $|...|\le p$, the term $I\{...\}=I$ (identity matrix); if not, $I\{...\}=0$.

Define

$$A\triangleq\left[\begin{array}{cc}L_{\mu_i\mu_i}\left(\boldsymbol{\mu}^*,\boldsymbol{\varphi}^*\right) & L_{\varphi_i\mu_i}\left(\boldsymbol{\mu}^*,\boldsymbol{\varphi}^*\right)\\ -L_{\varphi_i\mu_i}\left(\boldsymbol{\mu}^*,\boldsymbol{\varphi}^*\right) & 0\end{array}\right]$$

A6. Let $\epsilon_n=1/n$, and assume $A+I/2$ is a Hurwitz matrix. Note that it can be easily shown that the real parts of the eigenvalues of A are all non-positive (cf. page 449 in [145]).

We have the following proposition.

Proposition:

a) Under Conditions A1 and A3-A6. $U^n\left(\cdot\right)$ converges weakly to the solution (denoted as U) to the Skorohod problem

$$\left(\begin{array}{c}dU_{\mu_i}\\ dU_{\varphi_i}\end{array}\right)=\left(A+\frac{I}{2}\right)\left(\begin{array}{c}U_{\mu_i}\\ U_{\varphi_i}\end{array}\right)dt+\sigma dw(t)+\left(\begin{array}{c}dZ_{\mu_i}\\ dZ_{\varphi_i}\end{array}\right)$$

b) If $(\mu_i^*,\varphi_i^*)$is an interior point in the constraint set, the limiting process U is a stationary Gaussian diffusion process, and $U(n)$ converges in distribution to a normally distributed random variable with mean zero and covariance Σ.

c) If (μ_i^*,φ_i^*) is on the boundary of the constraint set, then the limiting process U is a stationary reflected linear diffusion process.

Proposition can be proved by appealing to a combination of tools used in the proofs of Theorem 5.1 in [56] and Theorem 2.1 in Chapter 6 in [57]. Roughly, we can expand, via a truncated Taylor series, the interpolated process $U^n(t)$ around the chosen saddle point (μ_i^*,φ_i^*). Then, the main new step is to show the tightness of $U^n(t)$. To this end, we can follow part 3 in the proof of Theorem 2.1 in Chapter 6 in [57] to establish that the biased term in the interpolated process diminishes asymptotically. Then, the rest follows from the proof of Theorem 5.1 in [56].

The rate of convergence depends heavily on the smallest eigenvalue of $\left(A + \frac{I}{2}\right)$. The more negative the smallest eigenvalue is, the faster the rate of convergence would be. The reflection terms would help increase the speed of convergence, which *unfortunately* cannot be characterized *exactly*.

3.8 Sensor-Actuator Coordination

This is the most constrained coordination level in SANETs. The coordination between sensors and actuators follows a hierarchical architecture [41], which has been shown to perform better in terms of defined QoS as compared to the flat architecture [55]. To minimize the latency between sensing and acting, the main *goal* of this coordination is transmit the event information to the *appropriate* actuators in the shortest time. The excessive burden of relaying information to the actuators can cause the sensor nodes to die due to limited battery supply. Therefore, we propose that each sensor should transmit event information to only *one* of the available actuators in the network instead of transmitting toward *multiple* actuators. Whereas, the optimal actuator upon receiving the event information can transmit to neighboring actuators, if required. Once the destination actuators are fixed for each sensor in the network, we use our distributed routing algorithm on top of layered architecture to optimize the end-to-end delays at this coordination level.

3.8.1 Optimal Actuator Selection

The first step in optimizing this coordination is the selection of an optimal actuator for each sensor in the network. In the following, we detail the selection criteria and algorithm.

3.8.1.1 Observed Delay on Different Routes

Under the above model there will be a delay, say $y_{j,i}^m$ of the packet from sensor node j to be served at sensor node i for actuator node m; this packet could have originated at sensor node j or may have been forwarded by sensor node j. The Expected delay of a packet transmitted from sensor node j is thus

$$\sum_{i \neq j} \phi_{j,i}^m y_{j,i}^m.$$

Since delays are additive over a path, packets from any sensor node will have a delay over any possible route to the actuator node (fusion center). A route will be denoted by an ordered set of sensor nodes that occur on that route, i.e., the first element will be the source of the route, the last element will be the actuator node and the intermediate elements will be sensor nodes arranged in the order that a packet traverses on this route. Let the total number of possible routes (cycle-free) be R^m. Let route i^m, $1 \leq i^m \leq R^m$ be denoted by the set $\mathcal{R}_i^m$ consisting of R_i^m elements with $\mathcal{R}_{i,j}^m$ denoting the j^{th}entry of this route. Then, a traffic splitting matrix will correspond to a Wardrop equilibrium iff for any i ([29])

$$\begin{gathered}\textstyle\sum_{1\leq j^m\leq R^m:\mathcal{R}^m_{j,1}=i}\left(\prod_{k=1}^{R^m_j-1}\phi_{\mathcal{R}^m_{j,k},\mathcal{R}^m_{j,k+1}}\right)\\ \left(\textstyle\sum_{k=1}^{R^m_j-1}y_{\mathcal{R}^m_{j,k},\mathcal{R}^m_{j,k+1}}\right)=\textstyle\sum_{k=1}^{R^m_l-1}y_{\mathcal{R}^m_{l,k},\mathcal{R}^m_{l,k+1}},\end{gathered}\tag{3.12}$$

for any l with $\mathcal{R}^m_{l,1} = i$ and such that $\prod_{k=1}^{R^m_l-1}\phi_{\mathcal{R}^m_{l,k},\mathcal{R}^m_{l,k+1}} > 0$, i.e., the delays on the routes that are actually used by packets from sensor node i are all equal. As we are exploring all the possible routes to all potential destination actuators, this computation can become very heavy. One way to limit the available number of actuators to choose from is to limit the maximum hop count that packets from sensor i should traverse. Therefore, we limit the actuator selection problem to a max. hopcount. This would just limit the solution space of the problem that we are considering. In theory, probing on paths toward actuators that are far from a given sensor will only result in slow convergence of optimal actuator selection problem, and is hence, avoided in this chapter. In practice, such a probe is meaningless as it results in huge waste of the very constrained limited-energy source in the network.

3.8.1.2 AS: Actuator Selection Algorithm

If the optimal mapping between a sensor and actuator is known, we focus only on optimal delay routing. In our network, such a mapping is not available. Therefore, we develop our algorithmic solution in three steps, 1) minimize the available actuator set, 2) find an optimal mapping between a sensor and actuator, and 3) find an optimal routing for this mapping.

To achieve the first step, we use a distributed topology-learning algorithm proposed in [C-1] to find the subset of actuators which are restricted by hopcount. The presence of actuators is not known during the initial network deployment in [C-1]. A distributed learning algorithm is proposed to discover the near-by actuators (the search is restricted by a hop-count parameter in order to avoid *flooding*). In order to find an optimal mapping between a sensor and actuator in the second step, we define our selection algorithm below.

Algorithm 1 (**AS**)

1. For a sensor i, $\sum_k \phi^m_{i,k} = \theta$, where $\theta = \frac{1}{m}\ (1 \leq m \leq M)$. Also, $\phi^m_{i,k} = \frac{\theta}{\sum_k k^m}$.

2. We estimate for node i, $\frac{\sum_k y^m_{i_k}}{k} = \tau_i\,(m)$, where $y^m_{i_k}$ is the delay for sensor i on route k for actuator m.
   a) We choose a destination actuator based on the following: $\min_m \tau_i\,(m)$, i.e., if there exists an actuator m for which $\frac{\sum_k y^m_{i_k}}{k}$ is minimum, choose m as $i's$ destination actuator.
   b) If there exist two actuators m and n, such that $\tau_i\,(m) = \tau_i\,(n)$ and $d^n_i < d^m_i$, then choose n as $i's$ destination actuator. We also consider a worst-case situation, when $\tau_i\,(m) = \tau_i\,(n)$ and $d^n_i = d^m_i$; in this case, we allow a sensor i to randomly select an actuator as its final destination.

3. Stop, when all mappings are fixed.

Note that θ is the percentage of traffic that is sent to one of the destination actuators. We equally distributed the fraction of total traffic to m actuators in order to be *fair* for actuator selection procedure. Whereas, the percentage θ for each actuator m is further uniformly splitted over k routes from sensor i. There is one subtle detail in the AS algorithm that deserves further consideration. Suppose that in step 2(b), if the average delays on k routes for two actuators m and n are equal. Which actuator should we then choose as the optimal actuator for sensor i? Clearly, the distance factor should be taken into account since doing so would help reduce the overall energy consumption over the route. Also, if $\tau_i\left(m\right) = \tau_i\left(n\right)$ and $d_i^n = d_i^m$, a sensor is allowed to randomly select its destination actuator.

We denote $a\left(i\right)$, the resulting destination actuator for sensor i via the above mapping. Then, we have $\sum_k \phi_{i,k}^{a(i)} = 1$ and $\sum_k \phi_{i,k}^{m} = 0$ for $m \neq a\left(i\right)$. In the third step, we use our distributed routing algorithm that converge to a Wardrop equilibrium for the system under consideration.

3.8.2 A Distributed Routing Algorithm

This algorithm is actually an adaptation of the algorithm already proposed in Section 2.6.

3.8.2.1 Open System (Layered Architecture)

Nodes iteratively keep updating the one-hop routing probabilities based on the delays incurred for every possible path.

Let $\phi(n)$ denote the traffic splitting matrix at the beginning of the n^{th} time slot. Node i does some computation to update the i^{th} row of this matrix. Let $Y_k^{a(i)}(n)(\mathcal{R}_{k,1}^{a(i)} = i)$ be the new value of the delay of a packet sent by sensor i through route $k(i = \mathcal{R}_{k,1}^{a(i)})$. Node i keeps an estimate of the average delay on route k.

$$y_k^{a(i)}(n+1) = (1-\beta)y_k^{a(i)}(n) + \beta Y_k^{a(i)}(n). \tag{3.13}$$

Further, after calculating the expected delays at the start of a time slot, each node adapts its routing probabilities to the new expected delays as follows,

$$\begin{aligned}\phi_{i,\mathcal{R}_{k,2}^{a(i)}}(n+1) = (1-\gamma)\phi_{i,\mathcal{R}_{k,2}^{a(i)}}(n) + \\ \gamma\left(\sum\nolimits_{1 \leq l^{a(i)} \leq R^{a(i)}:\mathcal{R}_{l,1}^{a(i)}=i} y_l^{a(i)}(n)\phi_{i,\mathcal{R}_{l,2}^{a(i)}}(n) - y_k^{a(i)}(n)\right)\end{aligned} \tag{3.14}$$

3.8.2.2 Closed System (Cross-Layer Architecture)

The updates for this system are going to be the same as that for the Open system. Delay and routing probability learning will remain as was in the Open System. The optimization criteria for channel access is also presented earlier in Section 2.6 of Chapter 2.

3.9 Actuator-Actuator Coordination

The coordination between the actuators follows a QoS architecture which can be divided into a number of categories based on application requirements [58]. As this particular coordination level is not constrained by limited resources, therefore, one *can* use AODV [81]/OLSR [82] like routing protocols for an efficient coordination among different actuators. Since, we have only one sink in the network, the network of actuators can form an *aggregation* tree toward this common sink and the flow from an actuator can be splitted and send over multiple routes toward the sink for remote processing requirements. At this coordination level, the optimal flow problem to obtain minimum end-to-end delays can be done in a similar fashion as is optimized for the sensor-actuator coordination level. Note that in this case, all the data gathered at different actuators is sent to a common sink, and hence, the solution is not *repeated* here to conserve length.

3.9.1 Classification of Actuation Process

The actuator coordination is classified into two main types which covers all the requirements for an effective actuator-actuator coordination framework.

3.9.1.1 Distributed Single-Actuator Actuation Process

A sensor transmits/forwards the readings to its optimal actuator. The actuator can process all incoming data and initiate appropriate actions without any involvement of neighboring actuators, e.g., a high alert security application. The actuators can later route this information back to the sink for some remote processing. This approach is referred to as AF (Action First) approach.

3.9.1.2 Distributed Multiple-Actuator Actuation Process

Upon receiving the event information, the actuator route it to neighboring actuators in order to best decide the optimal actuation strategy, e.g., in case of fire, the actuators need to efficiently collaborate so that the fire can easily be extinguished before it becomes uncontrollable. In this fashion, an energy constrained sensor do not need to transmit its readings to multiple actuators. Instead, the first actuator to receive this event information will *relay* it to its neighboring actuators to come up with an optimal actuation plan. This approach is referred to as DF (Decision First) approach. This actuation expectation can be expressed as follows:

$$D_m^{a(x,y)} = \eta d\left(m, (x,y)\right) + \zeta p\left(a\left(x,y\right)\right) \tag{3.15}$$

where $D_m^{a(x,y)}$ is the expectation for actuator m, $(1 \leq m \leq M)$ to join the actuation process $a\left(x,y\right)$, where x, y determine the coordinates of the actuation area. $d\left(m,(x,y)\right)$ is the distance of actuator m from the actuation area x, y. $p\left(a\left(x,y\right)\right)$ is the priority of of actuation process $a\left(x,y\right)$. η and ζ are application dependent adjustable parameters. Depending on the application, we can set a threshold ϵ. If $D_m^{a(x,y)} > \epsilon$, then the actuator m will participate in the actuation process. For this study, we do not take into account the energy consumption issues for actuation expectation $D_m^{a(x,y)}$ because the energy of the actuators is assumed to be infinity (or rechargeable energy source). For the case, where a finite source of energy is available at the actuators, an additional energy constraint and its own adjustment parameters could also added.

3.9.2 Data Collection and Distributed Routing

At any instant of time, an actuator may have two types of packets to be transmitted:

1. Packets received by the assigned sensor network.
2. Packets from neighboring actuators that arrived at this actuator and need to be forwarded.

Clearly, an actuator needs to have some scheduling policy to decide on which type of packet it wants to transmit, if it decided to transmit. A first come first served scheduling is one simple option. Yet another option is to have two separate queues for these two types of packets and do a weighted fair queueing for these two queues. In this chapter, we consider the second option. Under this mechanism, an actuator node i has two queues (introduced in Section 2.4of Chapter 2) associated with it: one queue (denoted Q_i) contains the packets that i has received from its assigned sensor network and the other (denoted F_i) contains packets that i has received from one of its neighboring actuators and has to be relayed. The combined channel access/data sampling mechanism is already detailed Section 2.4. We assume that the queue Q_i is always nonempty, i.e., sensor nodes make new measurements and continuously transmit packets to their assigned actuators. A detailed stability analysis of this scheme without power control is presented in Section 2.5.

3.9.3 Stability Analysis with Power Control

Let there be a finite set of power levels that an actuator node is allowed to use; denote this set by $\{l_1, ..., l_L\}$ *assume* $(l_k < l_{k+1})$. A actuator has to decide on the next hop actuator (thus requiring appropriate power for transmission). Let $\mathcal{N}_i(k)$ be the set of actuators that can receive $i's$ transmission when actuator i is using power level l_k. Actuator i accesses channel with probability α_i and we are in the scenario where actuator i always have data to transmit (coming from its assigned sensor network).

The routing now gives the power level used for transmission; assume that $m_{i,j}$ is such that actuator i needs power $l_{m_{i,j}}$ to communicate with actuator j (this is assumed to be symmetric, i.e., $m_{i,j} = m_{j,i}$). Clearly, the routing will now change the neighbors of actuator, i.e., since routing determines the transmission power, the actuators which can use receive transmissions from i will also change. Since j_i denotes the next hop of actuator i, $l_{m_{i,j_i}}$ will denote the power used by actuator i for any transmission.

Lemma 1: The probability of success of a transmission from actuator i is then

$$s_i = \sum_{j_i \in \mathcal{N}_i} \phi_{i,j_i} (1 - \alpha_{j_i}) \prod_{k: j_i \in \mathcal{N}_k(m_{k,j_k}) \backslash i} (1 - \alpha_k) \tag{3.16}$$

Lemma 2: The throughput of data of actuator i is thus

$$\lambda_i = \alpha_i (1 - \pi_i + \pi_i (1 - f_i)) s_i \Rightarrow \alpha_i (1 - \pi_i f_i) s_i. \tag{3.17}$$

where π_i is the probability that the forwarding queue of actuator i is not empty.

Let H be a matrix with entries 0 or 1 so that $H_{i,j} = 1$ if $\sum_{n=1}^{\infty} (\phi^n)_{i,j} > 0$, i.e., data originated at actuator j is forwarded by actuator i. Then, the stability condition for the forwarding queues in the actuator network is

$$\alpha_i f_i s_i \geq \sum_{j \in \mathcal{N}_i} H_{i,j} \lambda_j \tag{3.18}$$

The idea, in this case, is that an actuator may be using large power for transmissions, thus reducing the end-to-end delay, however at the same time it interferes with more neighboring actuators (note that large transmission power of an actuator does not imply that it sees large amount of interference; it merely means that this actuators causes more interference). Hence, an actuator using large transmission power may be causing local inefficiency.

We are mainly interested in the throughput of the actuator nodes. Hence, we want to provide a fair throughput to all of the actuators. Recall that we are in a cooperative framework so that all the actuators in the network can be persuaded to compromise on their performance in order to have better overall performance. For this objective, we would like to be fair among the users, as well as, as efficient as possible. Further, when considering power control, we would like to have long term power constraint which will have the form

$$\alpha_i l_{m_{i,j_i}} \leq q_i \tag{3.19}$$

where q_i is an upper bound on the power consumption by an actuator i. The impact of power control and choice of modulation index on the performance of a multi hop CSMA/CA system has been investigated in [59]. An analytical model for the spectral throughput per user is presented with emphasis on the number of nodes dwelling in the area covered by a given transmission power and on the number of hops. Unlike other works, a closed form solution is derived for the spectral throughput performance of a multi hop CSMA/CA system as a function of the offered load, the nodes density, transmission power and frame error rate. This facilitates the PHY and MAC layers to be jointly optimised. The optimal transmit power for every actuator node can be calculated in a centralized fashion similar to one presented in [60], but this solution do not work well for mobile scenarios. Therefore, we present a distributed approach based on heuristics that adaptively adjusts each actuators transmit power in response to topological changes and attempt to maintain a connected topology using minimum power in Section 3.9.4.

3.9.4 Dynamic Actuator Cooperation

As detailed in Section 3.4, we have a SANET with N static sensors and M mobile actuators. In mobile scenarios, the topology is constantly changing. The solution must, therefore, continually re-adjust the transmit powers of actuators to maintain the desired topology [C-10]. Further, the solution must use only local or already available information since updating global information such as positions of all the actuator nodes require prohibitive control overhead. Thus, the centralized solutions are not viable in the mobile context. Due to these constraints, the mechanism presented here is necessarily a heuristic algorithm and offer no guarantee on worst-case performance. In particular, power control is done using a cross-layer approach between MAC-PHY layers and is at best a *poor* approximation to an optimal solution.

PC: A Heuristic Algorithm

1. Every actuator is configured with three parameters, namely: the *desired* node degree A_d (for an application specific actuation process), a high threshold on node degree A_h, and a low threshold A_l . Periodically, an actuator checks its degree (the current node degree A_c) in its neighborhood set $\mathcal{N}_i$ (provided by routing). If $A_c \geq A_h$, then an actuator reduces its transmit power. If $A_c \leq A_l$, then an actuator increases its transmit power. If none of the above is true, no action is taken. The minimum and maximum transmit powers are l_1 and l_L, respectively (Section 3.9.3). Further, the magnitude of power *change* is a function of A_d and A_c.

2. Let p_d and p_c be the desired and current transmit power levels, respectively. Then, the desired power level (A similar derivation of this desired power level calculation is

provided in [60]. Therefore, we do not repeat it here to conserve space.) is given by

$$p_d = p_c - 5.m.log_{10}\frac{A_d}{A_c}. \tag{3.20}$$

A node knows its current transmit power level p_c and its current neighborhood node degree A_c (given by l_k and $\mathcal{N}_i\left(k\right)$, respectively) and A_d is a configured value. Also, m is the path loss index and it takes values $2 \leq m \leq 5$. In our work, we take the value of $m = 4$ as mentioned in Section 4.6. Then, (3.20) can be used to calculate the required power periodically, iff

$$s_i\left(A_d\right) \geq s_i\left(A_c\right). \tag{3.21}$$

where the calculation of $s_i\left(A_d\right)$ and $s_i\left(A_c\right)$ can be easily performed at the MAC layer using (3.16) with associated parameters.

We are interested in power control if and only if it improves the success probability s_i, which is a function of $\mathcal{N}_i\left(k\right)$ (3.16). Further, it plays and important role in determining the throughput of an actuator (3.17). It is also seen in Section 3.9.3, that the routing with power control changes the neighborhood set $\mathcal{N}_i\left(k\right)$ of actuator i. Therefore, the desired power level in (3.20) is practically applied if and only if (3.21) is valid.

In addition to power control, the mobility of actuator nodes results in network disconnectivity with its assigned sensor-network. Therefore, if an actuator node is expected to move from its current location, it broadcasts a packet informing all the sensors in its cluster of a change in position. This change is typically broadcasted to neighboring actuators as well. Thanks to the distributed learning approach proposed in [C-1], after initial network learning each sensor has multiple paths available to possibly different destination actuators, which can be verified by sending a 'Hello' message. Hence, a new actuator attachment is obtained in a fairly delay-energy efficient manner for the constrained sensor nodes using *dynamic* actuator cooperation. This cooperation is dynamic in a sense that it is event based where the event is characterized by actuator mobility.

3.10 Implementation Results

We consider a 9-node sensor-actuator network shown in Figure 3.2. It is easily seen that $\phi_{6,8} = \phi_{7,8} = \phi_{2,0} = \phi_{4,0} = 1$, node 0 and 8 being the destination actuators for all the packets generated in the network. Node 3 can transmit to 1 and 2 for 0, and to 1 and 7 for 8, respectively. Node 5 can transmit to 1 and 4 for 0, and to 1 and 6 for 8, respectively. Our selection algorithm has to find the optimal actuator for node 1, 3, and 5. We consider this simple network to clearly demonstrate the effect of actuator selection, estimated delay, and routing learning probabilities. Where as, the optimal actuator selection selection algorithm is able to provide an optimal destination actuator to each sensor even in large scale deployments as well. The distributed routing algorithm thus has to find appropriate value for $\phi_{3,2}, \phi_{3,7}$, and $\phi_{5,4}, \phi_{5,6}$ in order that the traffic flow in the network corresponds to a Wardrop equilibrium. In general, the distributed routing algorithm [C-7] is also able to converge to a Wardrop equilibrium for *any-scale* random deployment of SANETs and WSNs as well.

We first demonstrate the working of AS (actuator-selection) algorithm in the following. Each sensor starts sensing the environment and samples packets for transmission. We need to find an optimal destination actuator that corresponds to minimum average end-to-end delay

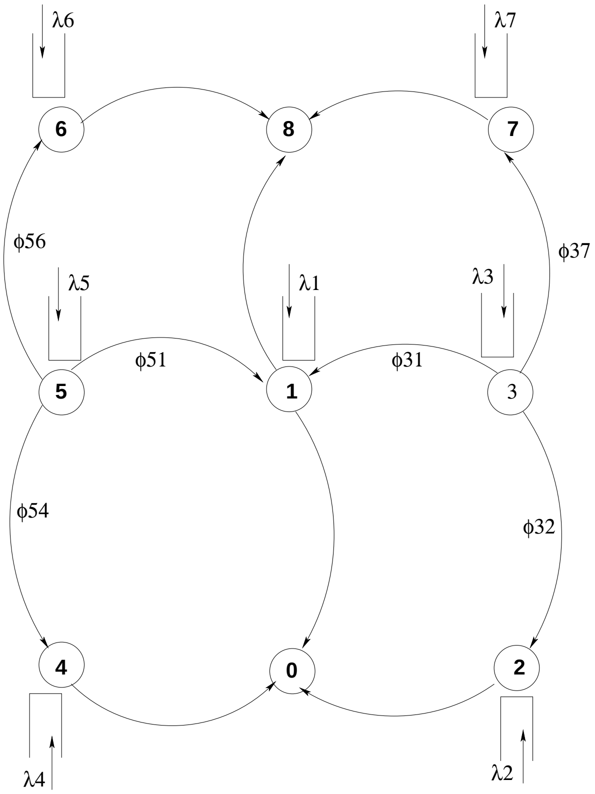

Figure 3.2: The Simulated Network consisting of 7 sensors and 2 actuators.

over all available routes for nodes 1, 3, and 5, given the load on the system $\underline{\lambda} = (\lambda_1, ..., \lambda_n)$. The execution of the AS algorithm is as follows:

- $\sum_k \phi^0_{3,k} = \sum_j \phi^8_{3,j} = \theta = 0.5$, $\left(m = 2, \theta = \frac{1}{2}\right)$. Similarly, $\sum_k \phi^0_{5,k} = \sum_j \phi^8_{5,j} = \theta$, $\left(\theta = \frac{1}{2}\right) = 0.5$. Also, $\phi^0_{3,2} = \phi^0_{3,1} = \frac{\theta}{2}$, $(k = 2) = 0.25$, $\phi^8_{3,7} = \phi^8_{3,1} = \frac{\theta}{2}$, $(k = 2) = 0.25$, $\phi^0_{5,4} = \phi^0_{5,1} = \frac{\theta}{2}$, $(k = 2) = 0.25$, $\phi^8_{5,6} = \phi^0_{5,1} = \frac{\theta}{2}$, $(k = 2) = 0.25$. Where as, $\phi^0_{1,0} = \phi^8_{1,8} = \theta$, $\left(\theta = \frac{1}{2}, m = 2\right) = 0.5$.

- The sampling rate λ_i at each sensor i is uniformly distributed between $(0.1, 0.2)$. The channel access rates were set to $\alpha_i \leq 0.2$ for $i = 1, ..., 7$. The sampling rates obtained are as follows: $\lambda_1 = 0.12$, $\lambda_2 = 0.10$, $\lambda_3 = 0.10$, $\lambda_4 = 0.14$, $\lambda_5 = 0.13$, $\lambda_6 = 0.2$, and $\lambda_8 = 0.18$. We can now measure the following terms: For node 3, $\tau_3(0) = \frac{y^0_{3_{3\to2\to0}} + y^0_{3_{3\to1\to0}}}{2}$ and $\tau_3(8) = \frac{y^8_{3_{3\to7\to8}} + y^8_{3_{3\to1\to8}}}{2}$. For node 5, $\tau_5(0) = \frac{y^0_{5_{5\to4\to0}} + y^0_{5_{5\to1\to0}}}{2}$ and $\tau_5(8) = \frac{y^8_{5_{5\to6\to8}} + y^8_{5_{5\to1\to8}}}{2}$. For node 1, $\tau_1(0) = y^0_{1_{1\to0}}, \tau_1(8) = y^8_{1_{1\to8}}$. For one set of sampling rates $\lambda_i^{\prime}s$ and $\alpha_i^{\prime}s$, we have run the simulation 10 times, and obtained the following results: For node 3, $\tau_3(0) = 2s$,$\tau_3(8) = 3.5s$, therefore node 3 chooses actuator 0 as its final destination as the average delay over all available routes toward actuator 0 is minimum compared to the average delay over existing routes toward actuator 8. For node 5, $\tau_5(0) = 3s$,$\tau_5(8) = 5s$, therefore node 5 also chooses actuator 0 as its final destination as the average delay over all available routes toward actuator 0 is minimum compared to the average delay over existing routes toward actuator 8. For node 1,

$\tau_1(0) = \tau_1(8) = 1.1s$, and $d_1^0 = d_1^8$. The actuator selection in this case is random as is defined in step 2(b) of Section 3.8.1. In our simulation settings, the outcome of this case is based on the expectation of a random variable, as actuator selection is random. Therefore, we need not mention the actuator selection in this particular case. *Note* that, in practice, this scenario is very rare, and hence, the actuator selection for this particular case does not matter. Now, that the destination for all the sensors in the network are fixed, we look at the routing. The results of our distributed routing algorithm, which converges to a Wardrop equilibrium, look similar to the ones presented in Section 2.7 of Chapter 2, and, are hence, not repeated here.

In the following analysis, we have only presented results on the performance of our distributed AS algorithm.The performance results on the distributed actuator-actuator coordination are not demonstrated in this work due to the lack of appropriate infrastructure availability with the current network simulators [53, 52, 54] as they can not simulate hybrid networked-nodes with dual communication capabilities i.e., nodes with multiple transmission interfaces operating at different physical layer frequencies, dual-MAC operating in parallel at two different frequencies, etc.. Also, we do not provide any comparisons with the related literature available on SANETs as there is *no* simulation setup *available* for one such comparison. Although, we did compare a general layered approach (open system) with the proposed cross-layer approach (closed system) under standard settings to demonstrate the benefits one can get from the cross-layered approach.

The proposals presented in this chapter are implemented in ns-2 [53]. Since, it is *hard* to simulate heterogeneous networks (like the one we are considering here), we modified the tcl-based ns-2 scripts in order to simulate the wireless sensor-actuator network. By hard, we mean that one can not simulate a network consisting of hybrid devices with different communication and networking capabilities. These scripts, in particular, modifies the communication capabilities of actuator nodes at run-time. The MAC used in simulations is a CSMA/CA and routing is performed as explained in Section 3.8 for sensor-actuator coordination level and in Section 3.9 for actuator-actuator coordination. From Figure 3.3, it can be seen that the throughput is maximum when an actuator only has one neighbor to route its data to the remote sink. This is only due to the presence of less *interference* in an actuators *neighborhood.* This can also be verified from (3.16), where an increase in the power level results in a increase in the neighborhood degree (minimizes the channel access due to more *contending* neighbors) and also changes the *routing* matrix. We could not present detailed results on the average-power used for transmissions and average-delay for the throughput results given in Figure 3.3 due to simulator restrictions. We believe that there is still a need to do large amount of experimentation with different networking scenarios in order to provide a good insight into the working of PC heuristic algorithm. Figure 3.4 shows the energy consumption due to routing control overhead both in the case of static and mobile topologies to perform power control. The results shown here are for 2-connectivity (at actuator-actuator coordination level) and 0.5 throughput. The updates are *event* based and require only one-hop message exchange among neighboring actuators. It also includes *broadcast* message transmissions to sensors in case of mobility.

3.10.1 Optimization in Open System

We now implement the proposed deterministic distributed primal-dual algorithm. Specifically, we consider a simple 8-node wireless sensor network as shown in Figure 3.5. All the sensors

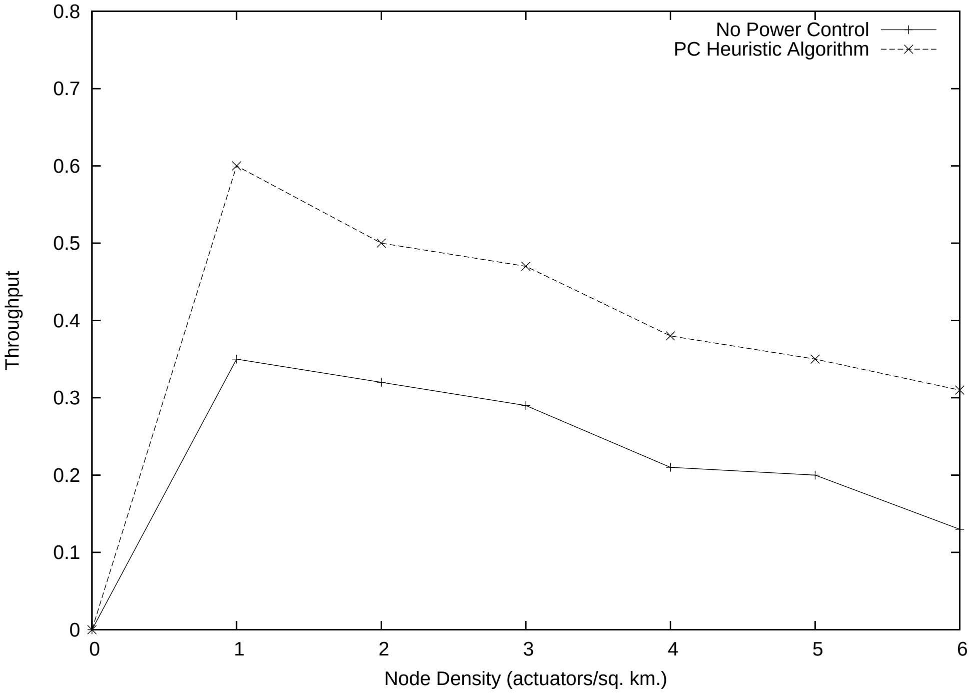

Figure 3.3: Throughput vs. Actuator Density

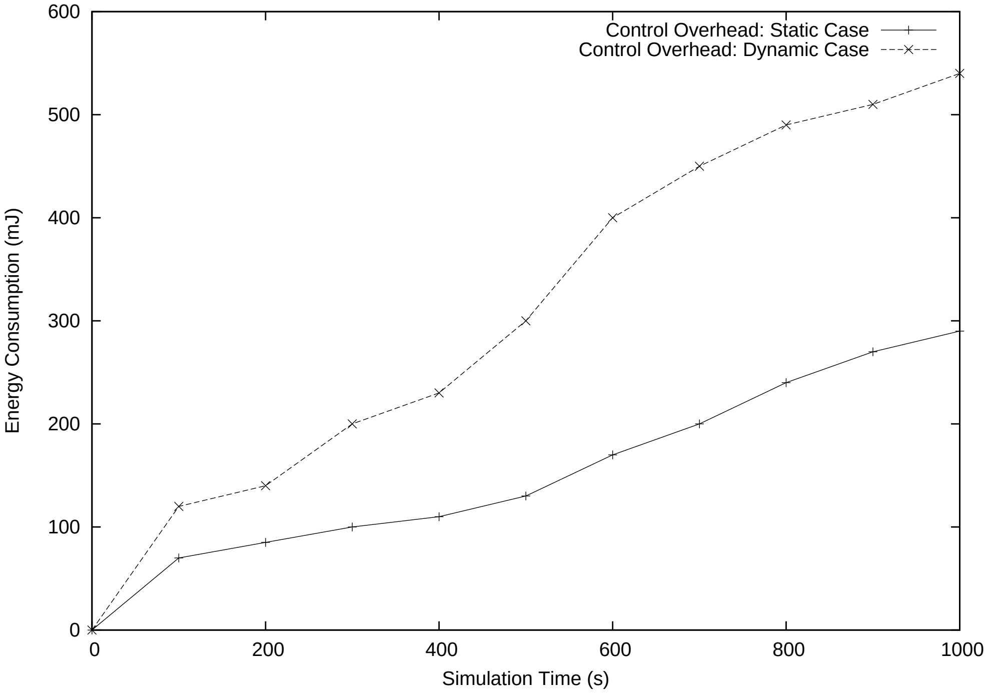

Figure 3.4: Energy Consumption for Control Overhead

sample data with $\tau_i = 0.1$. We use a random access CSMA/CA like MAC without backoff. We first fix the routing in the network, and thus, fixing the arrival rate at each node. We then look at the convergence of primal-dual algorithm. The results obtained by the proposed primal-dual algorithm, together with the theoretical optimal solution, are presented in Table 3.1. It can be easily seen that the results obtained from the primal-dual algorithm is very close to the optimal solution.

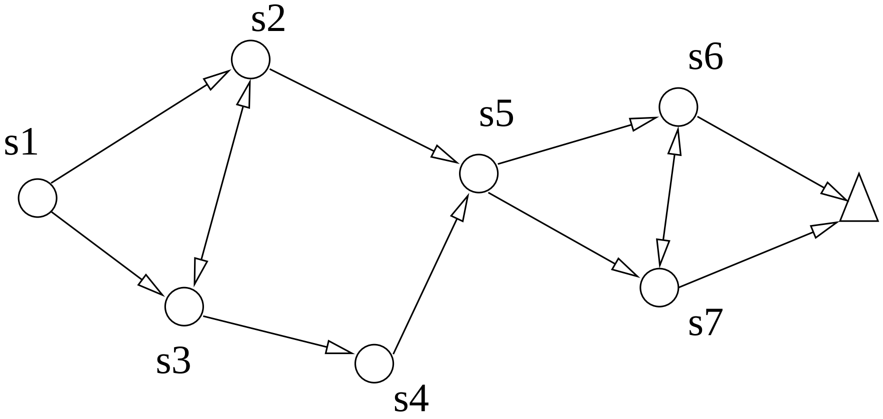

Figure 3.5: A Simple Network Topology

Table 3.1: Comparison between the results of the proposed primal-dual algorithm and the theoretical optimal solution

Node	a_i	μ_{i-opt}	$\mu_{i-primal-dual}$
1	0.1	0.102	0.121
2	0.2	0.208	0.225
3	0.1	0.1220	0.125
4	0.2	0.241	0.256
5	0.35	0.383	0.412
6	0.7	0.719	0.743
7	1.05	1.058	1.072

We now look at the convergence of the distributed primal-dual algorithm for some nodes in the network w.r.t time. Figure 3.6, 3.7, 3.8, and 3.9 shows the convergence of distributed primal-dual algorithm for node 3, 4, 5, and 6 in the network. It can be seen that the optimal value of μ_3 is obtained by the distributed primal-dual algorithm in less than 100 iterations of the algorithm. This shows a very fast convergence of the distributed primal-dual algorithm.

3.11 Conclusions and Future Work

For wireless sensor-actuator networks with random channel access, we propose that each sensor must transmit its readings toward one actuator only in order to take the burden of relaying, toward different actuators, away from energy-constrained sensors in a straight forward fashion. The objective for the open system was to minimize the total delay in the network where the constraints are the arrival-rate and service-rate of a node. Particularly, we have shown that the

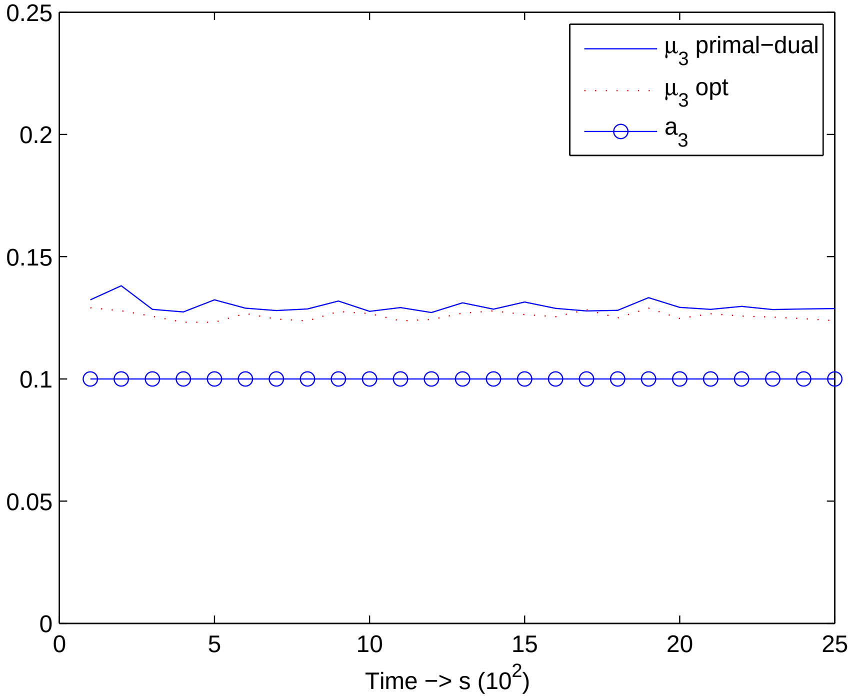

Figure 3.6: Convergence of μ_3 using distributed primal-dual algorithm

objective function is strictly convex for the entire network. We then use the Lagrangian dual decomposition method to devise a distributed primal-dual algorithm to minimize the delay in the network. The deterministic distributed primal-dual algorithm requires no feedback control and therefore converges almost surely to the optimal solution. The results show that the required optimal value of *service rate* is achieved for every node in the network by the distributed primal-dual algorithm. It is important to pay equal attention to both the observed delay in the network and energy consumption for data transmissions. A fast convergence means that only a little extra energy is consumed to perform local calculations to achieve the desired optimizations. Only energy-efficient routing might not serve any purpose for some sensor network applications. Similarly for the stochastic delay control algorithm, we have shown a probability one convergence and its rate of convergence which is entirely distributed in nature.

We then proposed an algorithm for an optimal actuator selection that provides a good mapping between any sensor and an actuator in the network. The selection algorithm finds a delay *optimal* actuator for each sensor in polynomial time. We finally propose a *distributed* actuation control mechanism for SANETs that is responsible for an efficient actuation process. The actuators can *dynamically* coordinate and perform power control to maintain a defined level of *connectivity* subject to *throughput* constraints. The control overhead for static and mobile actuator scenarios is analyzed using ns-2 simulations. The PC heuristic algorithm is applicable to multihop SANETs to increase throughput, battery life, and connectivity.

In future, we will present a detailed simulation based study of PC heuristic algorithm in different networking scenarios with some application specific actuation requirements and practical evaluation of distributed multiple-actuator actuation process. We will also work on

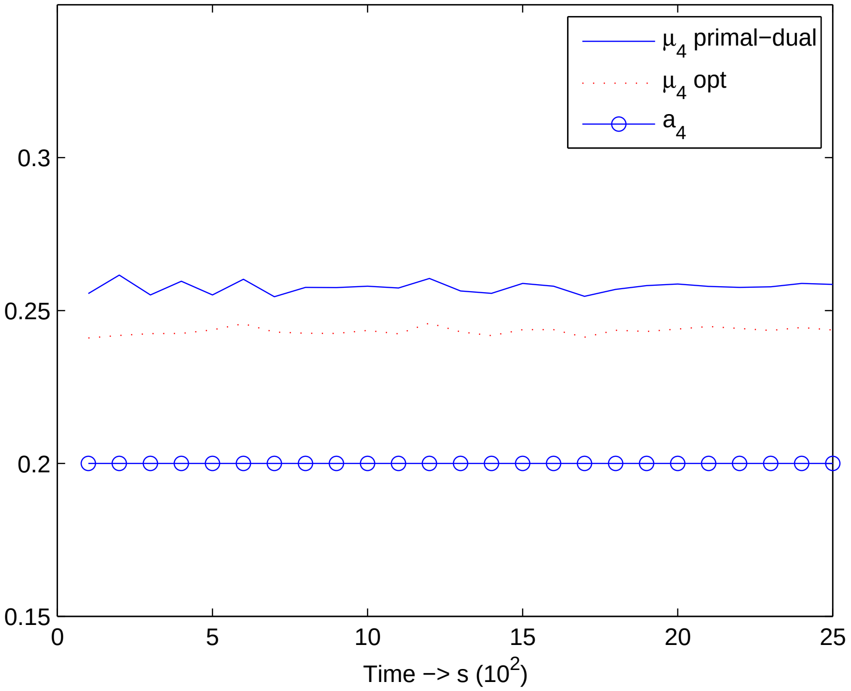

Figure 3.7: Convergence of μ_4 using distributed primal-dual algorithm

the development of PC heuristic algorithm to improve some MAC layer performance metrics using a cross-layer approach. As a consequence of a very fast convergence to Wardrop equilibrium, we are also tempted to perform the energy-analysis of the proposed learning and routing scheme in the context of network lifetime.

In the following chapter, we will look at the energy *efficiency* issues for data *aggregation* in SANETs.

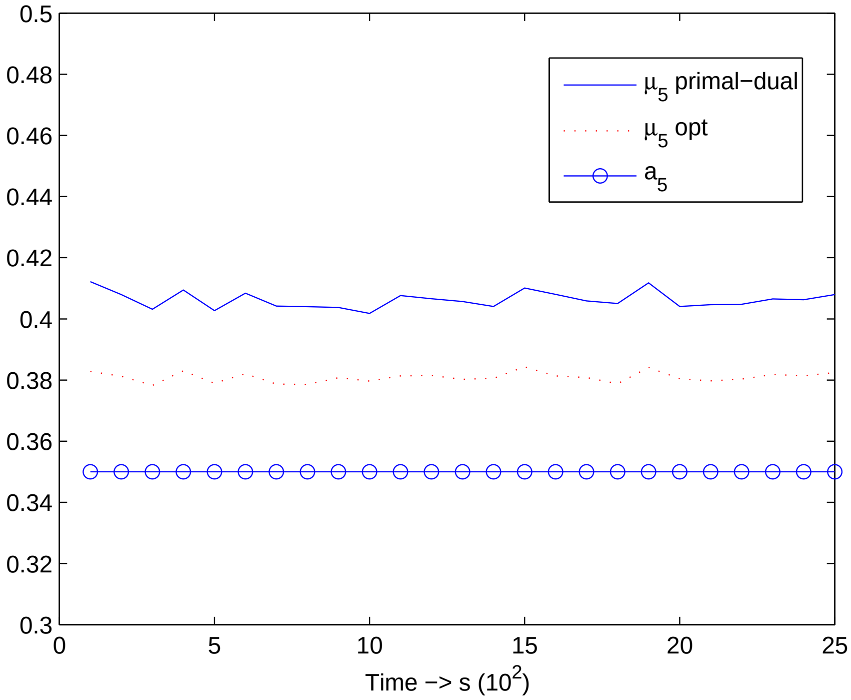

Figure 3.8: Convergence of μ_5 using distributed primal-dual algorithm

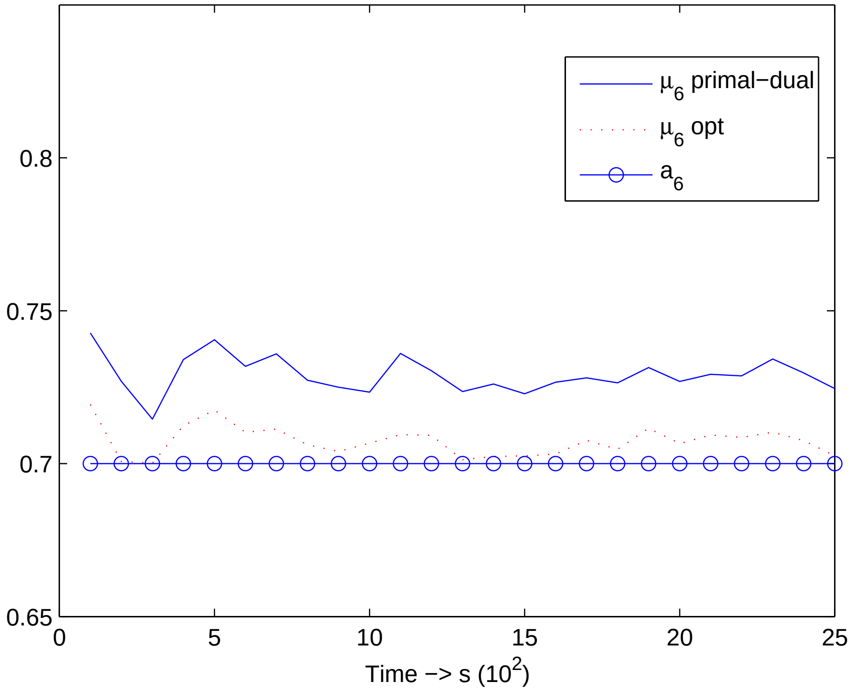

Figure 3.9: Convergence of μ_6 using distributed primal-dual algorithm

Chapter 4

The LEAD Cross-Layer Architecture for SANETs

SANETs are composed of sensors and actuators linked together by wireless medium to perform distributed *sensing* and *acting* tasks. Delay and energy constraints have a significant impact on the design and operation of SANETs. We consider a sensor-actuator network in which both energy and delay are *hard* constraints and must be jointly optimized.

In this chapter, we present the design, implementation, and performance evaluation of a novel low-energy, adaptive and distributed (LEAD) *self-organization* framework. This framework provides coordination, routing, and MAC layer protocols for network organization and management. We organize the *heterogeneous* sensor-actuator network into clusters where each cluster is managed by an actuator. To maximize the network *lifetime* and attain minimum end-to-end delays, it is essential to optimally match each sensor node to an actuator and find an optimal routing scheme. We provide an actuator discovery protocol that finds out a destination actuator for each sensor in the network based on the outcome of a *cost* function. Further, once the destination actuators are fixed, we provide an optimal flow routing solution with the aim of *maximizing* network lifetime. We then propose a delay-energy aware TDMA based MAC protocol in *compliance* with the routing algorithm. The actuator-selection, optimal routing, and TDMA MAC schemes together guarantees a *near-optimal* lifetime. The proposal is validated by means of analysis and ns-2 simulation results.

Furthermore, preventing sensor nodes from being *isolated* is very critical. The problem of sensor inactivity arises from the *pathloss* and *fading* that degrades the quality of the signals transmitted from actuators to sensors, especially in *anisotropic* deployment areas, e.g., rough and hilly terrains. Sensor data transmission in SANETs heavily relies on the *scheduling* information that each sensor node receives from its associated actuator. Therefore if the signal containing scheduling information is received at a very low power due to the impairments introduced by the wireless channel, the sensor node might be unable to decode it and consequently it will remain isolated.

Each sensor node transmits its data to only one of the actuators. However, all actuators cooperate and jointly transmit scheduling information to sensors with the use of beamforming. This results in an important reduction in the number of *isolated* sensors comparing to single actuator transmission for a given level of transmit power. The reduction is due to the resulting *array gain* and the exploitation of *macro diversity* that is provided by the actuator *cooperation.*

4.1 Introduction

Distributed systems based on networked sensors and actuators with embedded computation capabilities enable an instrumentation of the physical world at an unprecedented scale and density, thus enabling a new generation of monitoring and control applications. SANETs are an emerging technology that has a wide range of potential applications including environment monitoring, medical systems, robotic exploration, and smart spaces. Such networks consist of a large number of distributed sensor and few actuator nodes that organize themselves into a multihop wireless network. Each sensor node has one or more sensors (including multimedia, e.g., video and audio, or scalar data, e.g., temperature, pressure, light, infrared, and magnetometer), embedded processors, and low-power radios, and is normally battery operated. Typically, these nodes coordinate to perform a common task. Whereas, the actuators gather this information and react accordingly.

Sensor-actuator networks have the following unique characteristics:

- Real-time requirement: Depending on the application there may be a need to rapidly respond to sensor input. Examples can be a fire application where actions should be initiated on the even area as soon as possible.

- Coordination: Unlike WSNs where the central entity (i.e., sink) performs the functions of data collection and coordination, in SANETs, new networking phenomena called sensor-actuator and actuator-actuator coordination may occur. In particular, sensor-actuator coordination provides the transmission of event features from sensors to actuators. After receiving event information, actuators may need to coordinate with each other (depend on the acting application) in order to make decisions on the most appropriate way to perform the actions.

In this chapter, we investigate a new *self organizing* framework for SANETs. We consider a heterogeneous network that consists of sensors and actuator nodes randomly deployed in the network. Each sensor must transmit its data to only one of the actuators to conserve the *scarce* energy resource. This arises the problem of actuator *selection* for each sensor in the network. In the last chapter, we discussed a delay optimal actuator selection algorithm. Whereas, in this chapter, we look at the energy issues while selecting an actuator. In particular, we propose an optimal actuator selection and flow routing protocol (LEAD-RP) with the aim of *maximizing* the network lifetime. We show that the actuator-selection and flow routing problem with *energy* constraints can be modeled as a mixed integer non-linear programming optimization problem (MINLP) [61]. Since MINLP is NP-hard in general, we develop a distributed approach which provides a good *approximation* of the optimal solution. We use a *relaxation* technique in order to decide on the optimal actuator and then optimize the flow routing toward this actuator to extend network lifetime. For optimal actuator selection, we propose an Actuator Discovery Protocol (LEAD-ADP) that collects information about *neighboring* actuators for each sensor node in the network. The destination actuator is decided as outcome of a *cost* function. Once the destination actuators are fixed, we find out an *optimal* flow routing to maximize the network lifetime. Both of these steps are carried out at the network layer. At the MAC layer, we propose an adaptive TDMA like MAC (LEAD-MAC) with minimized *awake* periods (LEAD-Wakeup) to avoid the problem of *synchronization* during flow splitting and to meet the *delay* constraints in SANETs. The actuator selection, optimal flow routing, and TDMA MAC solution together guarantee a near-optimal *lifetime* for SANETs.

Depending on the application there may be a need to *rapidly* respond to sensor input. Moreover, to provide right actions, sensor data must still be valid at the time of acting. Therefore, the issue of real-time communication is very important in SANETs since actions are performed on the environment after sensing occurs. Examples can be a *fire* application where actions should be initiated on the event area as soon as possible. Unlike WSNs where the central entity (i.e., sink) performs the functions of data collection and coordination, in SANETs, new networking phenomena called *sensor-actuator* and *actuator-actuator* coordination may occur. In particular, sensor-actuator coordination provides the transmission of event features from sensors to actuators. After receiving event information, actuators may need to coordinate with each other in order to make decisions on the most appropriate way to perform the actions. Each sensor node is associated with an actuator which is the destination of the sensor data. In order to prevent sensor data collisions, actuators transmit *time schedules* which coordinate sensor multi-hop transmission. Therefore each sensor after receiving the scheduling information from its associated actuator transmits its data at the right time slot. If the signal containing the scheduling information is received at a very low power due to channel impairments, the sensor node might be unable to decode it and consequently it will remain *isolated*.

To the best of our knowledge the potential problem of isolated sensor nodes in a SANET has not been investigated. Actuators receive sensor data in a multi-hop fashion and transmit the scheduling information to them in a single hop fashion. A sensor node needs to decode the received scheduling information from the actuator that it is *associated* with. This is in order to know its assigned time slot in which it should transmit its sensed data. However due to the impairments introduced by the wireless channel (signal degradation due to *pathloss* and *fading*), it is very likely that some sensor nodes, more likely the ones that are distant from the actuator, would not be able to decode their *scheduling* information. This is because some sensor nodes would probably receive the signal containing scheduling information at a very *low* Signal-to-Noise Ratio (SNR). Consequently they will remain isolated, a fact that could create some isolated zones in the sensing field. This would result to incomplete information reception, a situation that needs to be overcome for the uniform monitoring of the sensing field. A potential solution to this would be the use of positive and/or negative acknowledgments (ACKs and/or NACKs) with respect to the reception of scheduling information. In this fashion, for the sensor nodes that cannot decode their scheduling information, multi-hop transmission of their schedules can be employed. However this would result to a significant overhead burden in terms of time and energy waste of the sensor nodes, that can reduce their lifetime. Furthermore that type of solution would increase the complexity of the employed protocols.

For a sensor network with multiple sinks (sinks/actuators can be thought of similar entities for design purposes), the traffic generated by sensor nodes may be split and sent to different sinks [62, 25]. In the presence of multiple sinks, the problem of optimal sink selection with the aim of extending lifetime using anycast routing is studied in [63]. The authors propose a heuristic solution based on traffic volumes sent to different base stations to select an optimal base station. The proposed solution is based on flow splitting which follows different routes from a source to its selected destination. The provided solution is elegant in the essence of extending lifetime at routing layer. The only issue with this solution is the synchronization (MAC layer) among different nodes to which a source (sensor) directs its flow. They do not address this synchronization problem in the paper. Simulation results show better performance based on numerical data and the issues related to MAC and synchronization were elevated.

In cases, when there are multiple actuators and mapping between the sensors and actuators is not given, the joint problem of finding an optimal actuator and extending network lifetime with minimum end-to-end delay constraints is a challenging and interesting problem. This problem is relevant from both the application's and wireless networking perspectives. From an application requirement perspective, some real-time multimedia sensing applications (e.g., video surveillance) require to have all the traffic generated from a source sensor to be routed to the same actuator (it may follow different routes) so that decoding and processing can be properly completed because the information from the same source is highly correlated and dependent. From a wireless networking perspective, the actuator chosen as a sink could have a significant impact on the end-to-end delays which is a *hard* constraint [C-3] for sensor-actuator applications. This is because the end-to-end delays are topology dependent; actuator selection simply based on energy constraints can not guarantee optimal end-to-end delays, and therefore, it should be based on both delay-energy constraints. As a result, there appears to be a vital need to understand how to perform optimal routing to jointly achieve minimum end-to-end delay routes and optimize network lifetime in delay-energy constrained sensor-actuator networks.

In this chapter, we propose a PHY, Routing and MAC solution with the aim of eliminating isolated zones in the sensing field, maximizing the network lifetime, and attaining minimum end-to-end delays. The problem of sensor inactivity can be effectively faced on the physical layer without increasing the protocol complexity and dissipating extra energy from sensor nodes. Actuators can cooperate and form a distributed antenna array, a concept that has been proposed for cellular communications [64]. The array jointly performs adaptive *beamforming* and distributes the time schedule to each sensor node. Sensors receive the schedule information at a much higher power due to the array gain that results from beamforming and to the exploitation of *macro-diversity* which is inherent to the distributed nature of a SANET. This results to a significant reduction in the number of isolated sensors for a given transmit *power level*. The cost is the need of Channel State Information at the transmitter (CSIT). It is shown by Matlab simulations that this effectively faces the problem of isolated zones. It is then proposed that each sensor node transmits its data to only one actuator. A sensor selects an actuator which is *minimum* number of hops away. *Note* that this actuator selection is just to decide a terminal point for sensor data transmissions and *multi-path* routing is actually used to transmit data between a sensor and its associated actuator. An advantage of setting min. hop criteria for actuator selection is that the lower-tier (sensor-actuator coordination level) of our heterogeneous network can be organized into clusters, where each cluster is centrally managed by an actuator. It is also shown that the flow routing with energy constraints can be modeled as a non-linear programming optimization problem (NLP). We use a relaxation to optimize the flow routing towards this actuator to extend network lifetime. We then propose to use an adaptive TDMA like MAC (that corresponds to the routing solution) to avoid the problem of synchronization during flow splitting and to meet the delay criteria for SANETs.

The organization of this chapter is as follows. Section 4.2 highlights some interesting related literature. The problem formulation is presented in Section 4.3. In Section 4.4, we provide the network model under consideration in detail. Section 4.5 focuses on the coordination framework. Section 4.6 details the design criteria of our proposed actuator-selection, optimal-routing scheme and optimization algorithm. In Section 4.7, we present a distributed network learning framework to solve the actuator-selection problem. In Section 4.8, we present a primal-dual algorithm for lifetime maximization. The medium access scheme is discussed in Section 4.9. In Section 4.10, we present our LEAD-wakeup protocol. Three different actuator-to-sensor transmission schemes are given in Section 4.11. The simulation results are presented in Section 4.12. In Section 4.13, we conclude the chapter and outline the future directions.

4.2 Related Literature

To our knowledge, sensor-actuator networks have not been extensively studied in the networking literature. However, our work in this direction has been informed and influenced by a variety of previous research efforts in the domain of WSNs, which we now describe.

TSMP (Time Synchronized Mesh Protocol) [70] is a networking protocol that forms the foundation of reliable, ultra low-power wireless sensor networking. TSMP provides redundancy and fail-over in time, frequency and space to ensure very high reliability even in the most challenging radio environments. TSMP also provides the intelligence required for self-organizing, self-healing mesh routing. The result is a network that installs easily with no specialized wireless expertise, automatically adapts to unforeseen challenges, and can be extended as needed without sophisticated planning. An intracluster communication bit-map-assisted (BMA) MAC protocol is proposed in [71]. BMA is intended for event-driven applications. The scheduling of BMA can change dynamically according to the unpredictable variations of sensor networks. In terms of energy efficiency, BMA reduces energy consumption due to idle listening and collisions. In this study, two different analytic energy models for BMA, conventional Time Division Multiple Access (TDMA) and energy efficient TDMA (E-TDMA) were developed, when used as intra-cluster MAC schemes. Simulation experiments are constructed to validate the analytic models. Both analytic and simulation results show that in terms of energy efficiency, BMA performance heavily depends on the sensor node traffic offer load, the number of sensor nodes within a cluster, the data packet size and, in some cases, the number of sessions per round. BMA is superior for the cases of low and medium traffic loads, relatively few sensor nodes per cluster, and relatively large data packet sizes. In addition, BMA outperforms the TDMAbased MAC schemes in terms of average packet latency. LEAD-MAC provides better results because of a *hybrid* scheduling scheme besides providing all the features supported in BMA MAC.

The energy efficiency at the MAC layer has recently received attention, especially with the increasing interest in the applications of unattended sensor networks. The S-MAC [36] enables low-duty-cycle operation in a multi-hop sensor network. Nodes form virtual clusters based on *fixed common sleep schedules* to reduce control overhead and enable traffic-adaptive wake-up. T-MAC [96] extends S-MAC by adjusting the length of time sensors stay awake between sleep intervals based on the communication of neighboring sensors. To achieve low power operation, B-MAC [97] employs an *adaptive preamble sampling scheme* to reduce duty cycle and minimize idle listening which is a basic source of energy drain. Whereas, the Z-MAC proposal [98] combines the strengths of TDMA and CSMA while offsetting their weaknesses by switching the MAC to CSMA and TDMA at low and high contention periods, respectively. The performance of Z-MAC falls even below B-MAC in the case of low contention, so it is a more suited protocol for medium to high data rate applications.

A schedule based MAC protocol is more difficult to implement because accurate time synchronization among neighbouring nodes is required. Each node uses a dedicated time slot to transmit messages. As fixed time slots are used, guarantees regarding bandwidth and message delay can be given. The main problem of such a MAC protocol is the complexity introduced by time synchronization. Especially in highly constrained sensor networks the synchronization *overhead* might not be acceptable. In [74], the authors present an f-MAC protocol which overcomes the aforemen tioned restrictions. The f-MAC uses a *framelet* approach: fixed sized frames are retransmitted a fixed number of times with a specific frequency. The protocol provides bandwidth-delay guarantees and time-synchronization among nodes is not necessary. In [75], a MAC protocol is proposed which uses separate wakeup slots for each sensor node in sensor networks. Most MAC protocols proposed for sensor network are inefficient under heavy traffic loads, in particular in high density network topology because of frequent collisions and long sleep latency. They describe a MAC protocol in which each node has a different wakeup schedule in the same duty cycle, and it joins the competition only for its own short wakeup slot when the receiver is ready to receive its data. Simulation results indicate that this scheme can reduce energy consumption and minimize idle listening which increases the power efficiency. The hybrid schedule of LEAD MAC provides the same results without using a seperate wakeup slot. Because, we take both sender and reciever into account while scheduling a transmission.

Asynchronous power efficient communication protocols are crucial to the success of WSNs as a distributed computing paradigm. An improved asynchronous duty-cycled MAC protocol is proposed in [76] for WSNs. It adopts a novel dual preamble sampling (DPS) approach by combining low power listening (LPL) with short strobed preambles to significantly reduce idle listening in existing protocols and improves the performace compared to previously related MAC [97] protocols. Energy efficiency and reliable data delivery are the two most important parameters for designing wireless sensor network protocols. Wireless communication is inherently unpredictable and error-prone. Hence, reliability is required to ensure that a packet reaches its desired destination; otherwise the energy expended in forwarding the packet is wasted. On the other hand, energy-saving mechanisms are required for battery-constrained sensor networks. In [77], the authors have designed an on-demand energy-efficient and reliable MAC protocol that enables reliable data delivery and energy conservation without affecting packet latency. [78] proposes MH-MAC, a new MAC protocol for wireless sensor networks capable of handling applications that generate infrequent huge peaks of traffic. Existing protocols are not adapted to this kind of applications. Asynchronous protocols are energy efficient for the long inactive periods, but fail to cope with the bandwidth and latency requirements of the traffic peaks when more than two nodes are sending data to a common sink. Synchronous protocols that support contention free slots provide good throughput for handling the load peaks, but consume unnecessary energy maintaining clocks synchronized for very long idle periods. MH-MAC is a multimode hybrid protocol that can be configured by the application to run in asynchronous mode or in synchronous mode, with or without contention, providing the best possible trade-off. MH-MAC is a single-hop MAC, which supports multi-hop applications through a cross-layering API.

In [79], the authors present the design of a new low duty-cycle MAC layer protocol called Convergent MAC (CMAC). CMAC avoids synchronization overhead while supporting low latency. By using zero communication when there is no traffic, CMAC allows operation at very low duty cycles. When carrying traffic, CMAC first uses anycast to wake up forwarding nodes, and then converges from route-suboptimal anycast with unsynchronized duty cycling to route-optimal unicast with synchronized scheduling. To validate their design and provide a usable module for the community, they implement CMAC in TinyOS and evaluate it on the Kansei testbed consisting of 105 XSM nodes. The results show that CMAC at 1% duty cycle significantly outperforms BMAC [97] at 1% in terms of latency, throughput and energy efficiency. They also compare CMAC with other protocols using simulations. The results show for 1% duty cycle, CMAC exhibits similar throughput and latency as CSMA/CA using much less energy, and outperforms SMAC [36]in all aspects. LEAD MAC provides better results as it does the same at initial network deployment. This procedure is not repeated everytime traffic is transported from sensors to their destination unless converged routes are changed. [80] introduces Crankshaft, a MAC protocol specifically targeted at dense wireless sensor networks. Crankshaft employs node synchronisation and offset wake-up schedules to combat the main cause of inefficiency in dense networks: overhearing by neighbouring nodes. Further energy savings are gained by using efficient channel polling and contention resolution techniques. Simulations show that Crankshaft achieves high delivery ratios at low power consumption under the common convergecast traffic pattern in dense networks. This performance is achieved by trading broadcast bandwidth for energy efficiency. Finally, tests with a TinyOS implementation demonstrate the real-world feasibility of the protocol.

Mobility in wireless sensor networks poses unique challenges to the MAC protocol design. Generally, MAC protocols for sensor networks assume static sensor nodes and focus on energy efficiency. In [72], the authors present a mobility adaptive, collision-free medium access control protocol (MMAC) for mobile sensor networks. MMAC caters for both weak mobility (e.g., topology changes, node joins, and node failures) and strong mobility (e.g., concurrent node joins and failures, and physical mobility of nodes). MMAC is a scheduling-based protocol and thus it guarantees collision avoidance. MMAC allows nodes the transmission rights at particular timeslots based on the traffic information and mobility pattern of the nodes. By modeling several popular MAC layer protocols, the authors in [73] derive bounds on performance for receiver efficiency. In particular, they analyze four abstract models, Synchronous Blinking (e.g. T-MAC [96], S-MAC [36]), Long Preamble (e.g. B-MAC [97]), Structured Time-Spreading (also called Asynchronous Wake-Up), and Random Time Spreading. These results strongly suggest that scheduling the receiver so as to minimize (or eliminate) the potential for interference (or collisions) could be from 10 fold to 100 fold more efficient than current practice. They provide two new receiver scheduling methods, Staggered On and Pseudorandom Staggered On, both of which are designed to exploit the *untapped* opportunity for greater receiver efficiency.

In [1], the authors proposed an efficient routing protocol for WSNs with global objective set to maximize network lifetime. The constraints are set to minimize the energy consumption for efficient data aggregation. The protocol works by building gradients along an interest propagation. In short, interest propagation sets up state in the network (or parts thereof) to facilitate "pulling down" data toward the sink. The results provided therein have shown significant improvement over traditional routing protocols both in terms of communication and computational load. Whereas in [65], the authors use the same approach as [1] for SANETs using anycast routing. A reverse tree-based anycast routing is proposed, which constructs a tree routed at the event source, where sensors can join and leave dynamically. The introduction of actuators in the existing WSNs has opened up a new dimension of "a hard delay constraint" while still looking for near-optimal network lifetime solutions [58]. For example, targeting an intruder holding a sniper in a surveillance field can be an interesting case to consider. The actuation process has to localize the position of the intruder and actuate the destruction process. The important constraint in this case is the latency of the received data because the sensor data can be no more valid at the time of actuation in case of increased latency.

A well designed application-specific coordination protocol is proposed [66], where cluster formation is triggered by an event so that clusters are created on-the-fly to optimally react to the event itself and provide the reliability with minimum energy expenditure. In order to provide effective sensing and acting, an efficient and distributed coordination mechanism is required for delay-energy aware dissemination of information, and to perform right and timely actions. Therefore, we proposed to establish these clusters once during the initial network deployment and the routing protocol can disseminate the sensed information to the actuators through maximum remaining energy paths. After receiving the event information, actuators may need to coordinate with each other in order to make decisions on the most appropriate way to perform the required action. Depending on the application, there can be multiple actuators interested in some information. Therefore, sensors need to transmit this data toward multiple actuators, which results in excess sensor-energy drain due to multiple transmissions of redundant information [67]. Moreover, the collected and transmitted sensor data must be valid at the time of acting. For example, if sensors detect a malicious person in an area and transmit this information to its optimal actuator; and the act of disposing a tranquilizing gas must find that person in the very same area. Therefore, the issue of real-time communication is very important in SANETs.

Most of the current research on sensor systems is mainly focused toward optimizing the network lifetime (e.g., [26, 25]) and the energy consumption of the sensors bypassing the delay-sensitivity of sensor data for real time applications. In [68], the authors presented a detailed overview of the routing techniques proposed for WSNs. The routing techniques are classified into three categories based on the underlying network structure: flat, hierarchical, and location-based routing. The hierarchical routing schemes have shown a promising improvement for prolonging network lifetime [41]. An enhancement in basic LEACH is proposed in [42], where the network lifetime has been extended by the introduction of closest neighbor communication. In [69], the network lifetime was prolonged on the basis of threshold-sensitive routing schemes.

If the mapping between a sensor node and a base station/actuator[1] is given a priori, then the problem of finding optimal flow strategies to extend network lifetime has been well investigated in the past [26, 41]. For a sensor network with multiple sinks, the traffic generated by sensor nodes may be split and sent to different basestations [25]. In the presence of multiple basestations, the problem of optimal basestation selection with the aim of extending lifetime using anycast routing is studied in [63]. The authors proposed a heuristic solution based on traffic volumes sent to different base stations to select an optimal one. The proposed solution is based on flow splitting which follows different routes from a source to its selected destination. The provided solution is elegant in the essence of extending lifetime at routing layer. Unfortunately, in wireless networking, the routing layer operates on top of a MAC layer and the only MAC scheme that can be used in this case is the CSMA like MAC. Then the limitation with this solution is the synchronization among different nodes to which a source (sensor) directs its flow. The synchronization problem has not been addressed in [63]. Simulation results show better performance for their proposal as they were only based on numerical data and the issues related to MAC and synchronization were elevated. Further, achieving this synchronization is not trivial from a technical point of view in ad hoc manner and is NP-hard in general.

The multi-actuator architecture raises many interesting issues such as cluster formation, cluster-based sensor organization, network management and task allocation among the actuators. In this chapter, we only focus on the issues of network management within the clusters, particularly energy-aware MAC-layer protocol and inter-cluster interference issues.

In [102], the authors presented two scheduling schemes (breadth-first and depth-first as-

[1] Actuators/base stations have similar semantics for modeling purposes, i.e., sinks for data generated in the network.

signment) for a *cluster based* sensor network. The proposed TDMA-MAC is shown to perform well in terms of energy-efficiency and end-to-end delay depending on the choice of scheduling scheme. The gateway nodes transmit the schedule in their cluster using larger transmission power. This introduces a new problem of schedule *interference* among neighboring clusters and is not discussed in the paper. PEDAMACS [103] proposal for sensor networks has utilized the presence of a powerful access point (AP) among sensors which takes the transmission load from the constrained sensors and is further responsible for the reliable delivery of sensor data toward the sinks. In case of multiple APs, the neighboring APs should not transmit their coordination packets at the same time to avoid *inter-cluster interference.* The APs should take into account the sensors that are outside their largest range while generating the schedule. The power levels of the APs are adjusted so that the schedule can reach all the sensors in the cluster. If all the sensors cannot be reached by the schedule, they can still be scheduled at the cost of an increased synchronization overhead apart from the increased delay and energy consumption.

4.3 Problem Statement

We consider a heterogeneous SANET, where data is transmitted in a multihop fashion from sensors to the actuators. We have to decide on the following objectives: destination actuator for each sensor in the network, an optimal routing protocol, and a medium access scheme that together prolongs network-lifetime by minimizing the energy consumption and also takes care of delay-sensitivity of the sensed data. Therefore, in cases, where there are multiple actuators and mapping between the sensors and actuators is not given, the joint problem of finding an optimal actuator and extending network lifetime with minimum end-to-end delay constraints is a challenging and interesting problem. This problem is relevant from both the application's and wireless networking perspectives. From an application requirement perspective, for some real-time multimedia sensing applications (e.g., video surveillance), it is necessary to have all the traffic generated from a source sensor to be routed to the same actuator (albeit that it may follow different routes) so that decoding and processing can be properly completed. For multimedia traffic such as video, the information contained in different packets from the same source are highly correlated and dependent. If the packets generated by a source are split and sent to different actuators, any of these receiving actuators may not be able to decode the video packets properly. From a wireless networking perspective, the actuator chosen as a sink could have a significant impact on the end-to-end delays which is a *hard* constraint [58] for sensor-actuator applications. This is because the end-to-end delays are topology dependent; actuator selection simply based on energy constraints can not guarantee optimal end-to-end delays, and therefore, it should be based on both delay-energy constraints. As a result, there appears to be a compelling need to understand how to perform optimal routing to jointly achieve minimum end-to-end delay routes and optimize network lifetime in delay-energy constrained SANETs.

In this chapter, we propose LEAD-RP to *maximize* network lifetime. For optimal actuator selection, we propose LEAD-ADP that collects information about *neighboring* actuators for each sensor node in the network. The outcome of a *cost* function decides an optimal actuator and then we can find out an *optimal* flow routing to maximize the network lifetime. These steps are carried out at the network layer. At the MAC layer, we propose an adaptive LEAD-MAC with minimized *awake* periods to avoid the problem of *synchronization* during flow splitting and to meet the *delay* constraints in SANETs. The actuator selection, optimal flow routing, and TDMA MAC solution together guarantee a near-optimal *lifetime* for SANETs.

4.4 Network Model

Consider a static 3-tier SANET with N sensor nodes, M actuator[2] nodes, and B basestations as shown in Figure 4.1. A static 3-tier wireless sensor-actuator network with N sensor nodes, M actuators nodes, and B Base Stations is considered as shown in Figure 4.1. Each sensor and actuator is equipped with an omnidirectional antenna. Actuators are inter-connected via a backhaul network (wireline or wireless). It is assumed that an equal number of sensors K is assigned to each actuator, so as $M \times K = N$.

4.4.1 Channel Model

A sensor node can decode a transmission from a neighboring sensor successfully if the experienced SNR or SINR (in the case of CCI) is above a certain threshold. The channel between the i^{th} sensor node and the j^{th} actuator is

$$h_{ij} = \Gamma_{ij}\sqrt{\beta d_{ij}^{-\alpha}\gamma_{ij}} \tag{4.1}$$

where d_{ij} is the distance in m of the i^{th} sensor and the j^{th} actuator. α is the path-loss exponent and β the path-loss constant. γ_{ij} is the corresponding log-normal coefficient which models the large-scale fading (shadowing), $\gamma_{dB} \sim \mathcal{N}\left(0\, dB, 8\, dB\right)$, and Γ_{ij} is the complex Gaussian fading coefficient which models the small-scale fading, $\Gamma \sim \mathcal{NC}\left(0, 1\right)$. The pathloss constant and exponent are chosen according to the COST-231 model, where actuator height is assumed to be $10\, m$ and sensor node height $10\, cm$.

4.4.2 Neighborhood Relation Model

Given an $(N + M + B) \times (N + M + B)$ neighborhood relation matrix $\mathcal{N}$ that indicates the node pairs for which direct communication is possible. We will assume that R is a symmetric matrix, i.e., if node i can transmit to node j, then j can also transmit to node i. For such node pairs, the $(i, j)^{th}$entry of the matrix $\mathcal{N}$ is unity, i.e., $N_{ij} = 1$ if node i and j can communicate; we will set $N_{ij} = 0$ if nodes i and j can not communicate. For any node i, we define $\mathcal{N}_i = \{j : N_{i,j} = 1\}$, which is the set of neighboring nodes of node i. Similarly, a set of interference nodes (cannot be reached by one-hop) for node i (from where the transmissions can be heard at node i, and is defined as $S_i = \{K \notin \mathcal{N}_i \cup \{i\} : \mathcal{N}_{k,j} = 1\ forsome\ j \in \mathcal{N}_i\}$. Note that S_i does not include any of the first-hop neighbors of node i.

4.4.3 Forwarding (Relaying)

The sensor-actuator network is deployed in a remote location. The sensors do the application dependent sensing and transmit their readings to the actuators. The actuators react on the environment based on the readings from the sensors and also forward (relay) this information to the basestations (using long-haul communication). Some in-network aggregation techniques could be applied at this stage if the data is correlated. Since this discussion is application dependent, and therefore, we do not go in its detail. The basestations are further responsible

[2] Conceptually, we can assume that the actuator is also a sensor node, which does not sense the environment.

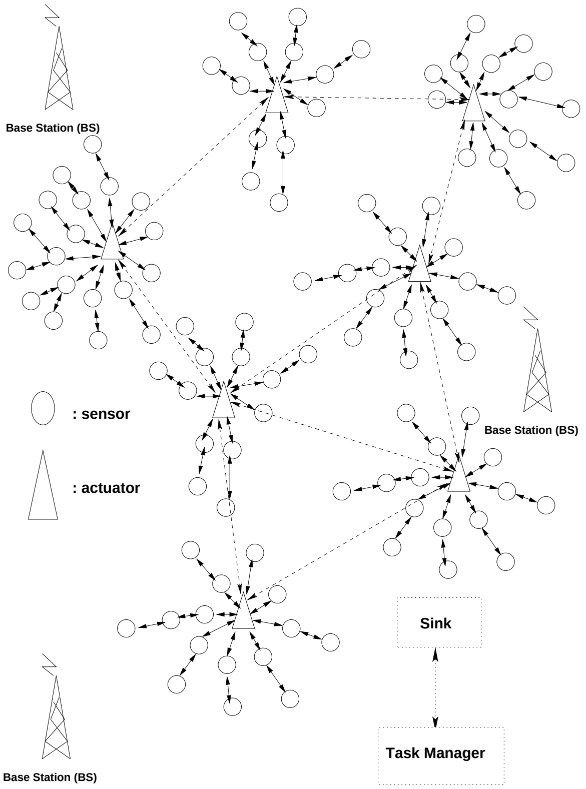

Figure 4.1: Architecture of SANETs

for forwarding (relaying) this information to a sink (this communication can be over satellite links) for remote analysis. Since, there are multiple actuators and basestations in our heterogeneous network, we divide the problem of optimal flow (from a sensor to a sink) at three distinct levels. At level one (sensor-actuator coordination), we investigate the actuator-selection problem and optimal flow routing in order to maximize the network lifetime at this level. At level two (actuator-actuator/basestation coordination), we study a similar problem of base-station selection and optimal flow routing to maximize network lifetime at level 2. Finally at level three (basestation-sink coordination), we study the problem of optimal flow from basestations to the sink. In this study, we assume that there is sufficient energy available at the sink, and thus, there is no energy constraint for the sink.

4.4.4 Channel Model and Antennas

We assume a simple channel model: a node can decode a transmission successfully iff there is no other interfering transmission. Each sensor node is supported by an omni-directional antenna. Each actuator is provided with two omni-directional antennas; one to communicate with the sensor network, and the other to communicate with the network of neighboring actuators/basestations using long-range communications. Similarly, each basestation is also provided with two omni-directional antennas; one to communicate with the network of actuators/basestations and the other to communicate with the sink (as it might be using a satellite link).

4.4.5 Frequency and MAC

Assume that all nodes share the same frequency band at their respective operating level. Time is assumed to be divided into fixed length slots. All the packets (depending on their operating level) are of same length and the length of a time slot corresponds to the time required to transmit a packet over the underlying wireless channel.

4.5 The Three-Level Coordination Framework For SANETs

In SANETs, sensors acquire information such as light, temperature, noise, and humidity from the surroundings, while actuators take decisions based on the information received and perform relevant actions. An integrated support for data aggregation in such networks works flexibly well with all major aggregation proposals: diffusion algorithms [1], streaming queries [69], and event graphs [65]. The three approaches differ in the way they influence the energy utilization and delay constraints, so it is left as an application and requirements specific concern to be monitored by the actuators in the network. In general, the network may support a variety of task types. We now focus on the three coordination levels of our self-organizing framework.

- **Sensor-sensor coordination level:** In WSNs, in-network aggregation [1] and negotiation based routing schemes are shown to work in the absence of any architecture. Therefore, we consider the sensor-sensor coordination level as flat structured. The main problem with the flat architectures is its scalability to large deployments. In our considered architecture, this flat structure is locally applied for sensor coordination in order to facilitate data aggregation functions. Further, the sensors need not to know about all the sensors that belong to the same cluster due to the existence of multi-hop paths and only neighbor knowledge is sufficient for effective coordination [C-3].

- **Sensor-actuator coordination level:** The coordination between sensors and actuators follows a hierarchical architecture, which has been shown to perform better [41, 42, 69] in terms of defined QoS as compared to the flat architecture. To minimize the latency between sensing and acting, the main goal of this coordination is transmit the event information to the appropriate actuator in the shortest time. The excessive burden of relaying information to the actuators can cause the sensor nodes to die due to limited battery supply. Therefore, we optimize the network-lifetime in Section 4.6 using an energy aware routing scheme at this coordination level.

- **Actuator-actuator coordination level:** The coordination between the actuators follows a QoS architecture which can be divided into a number of categories based on application requirements [58]. As this particular coordination level is not constrained by limited resources, we can use AODV [81]/OLSR [82] like routing protocols for an efficient coordination among different actuators.

In the following, we present the overall LEAD Architecture [J-2], which comprises four main components: 1) LEAD-RP, 2) LEAD-ADP, 3) LEAD-MAC, and 4) LEAD-Wakeup introduced in Section 4.1 and shown in Figure 4.2.

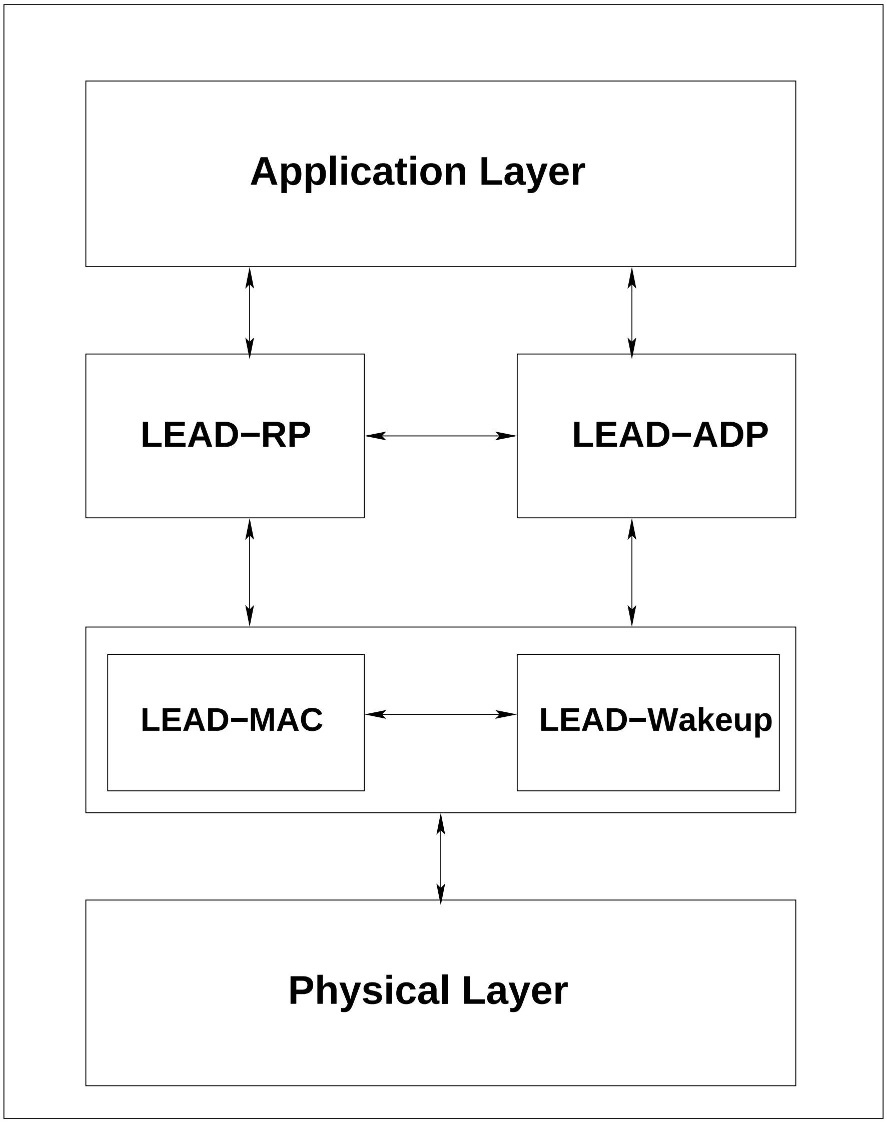

Figure 4.2: The LEAD Architecture

4.6 LEAD-RP: The LEAD Routing Protocol

In this section, we present the theoretical study of the problem under consideration. The sensor-actuator coordination level is the most constrained coordination level in SANETs. In the following, we detail several components of our proposed actuator-selection problem and optimal flow routing protocol for SANETs.

4.6.1 Power Consumption Model

For a sensor node, the energy consumption due to wireless communication (i.e receiving and transmitting) is considered the dominant source in power consumption at the routing layer. The issue of idle listening is delt with at the MAC layer in later sections.

The power consumed by a sensor node i in (J/bit) in reception can be modeled as

$$P_r^i = P_{rx} \sum_{j \in \mathcal{N}_i} f_{j,i} \tag{4.2}$$

where $f_{j,i}$ is the rate $(bits/s)$ at which node j is transmitting packets toward node i. A typical value for the parameter P_{rx} is 50 nJ/bit.

If the power consumed to send a packet is given by P_{tx} (a typical value for this parameter is 50 nJ/bit [41]), then the power consumed by a sensor node i in transmitting its data (both locally originated and forwarded packets) is

$$P_t(i,j) = c_{i,j}.f_{i,j} \tag{4.3}$$

where $c_{i,j}$ is the power consumption coefficient for data transmission between sensor i and j and $f_{i,j}$ is the total flow from sensor i to sensor j in $bits/s$. Also

$$c_{i,j} = \alpha + \beta.d_{i,j}^m \tag{4.4}$$

where α and β are constants, $d_{i,j}$ is the distance between the sensors i and j, and m is the path loss index. Typical values of α and β are $50nJ/bit$ and $0.0013\,pJ/bit/m^4$ (for $m = 4$), respectively [41].

4.6.2 Actuator-Selection and Optimal flow Routing

The joint problem of finding an actuator and flow routing to maximize network lifetime at the level 1 (s.t. energy constraints) is non-trivial and interesting for sensor-actuator networks. We define (for details, see Table 4.1)

$$F_{s,s} = \left\{ f_{s_i,s_j}^{s_k,A_l} : (1 \le i,j,k \le N, i \ne j, j \ne k, 1 \le l \le M) \right\}$$

$$F_{s,A} = \left\{ f_{s_i,A_l}^{s_k,A_l} : (1 \le i,k \le N, 1 \le l \le M) \right\}$$

$$F_{s,s_i} = \left\{ f_{s_m,s_i}^{s_k,A_l} : (1 \le m,k \le N, m \ne i, k \ne i, 1 \le l \le M) \right\}$$

$$F_{s_i,s} = \left\{ f_{s_i,s_r}^{s_k,A_l} : (1 \le r,k \le N, r \ne k, r \ne i, 1 \le l \le M) \right\}$$

$$F_{s_i,A} = \left\{ f_{s_i,A_l}^{s_k,A_l} : (1 \leq k \leq N, 1 \leq l \leq M) \right\}$$

We denote T_{l1} as network lifetime at level one (sensor-actuator coordination level), which (in this work) is defined as the time until an *alive* sensor drains its energy [C-6]. Then, we maximize lifetime T_{l1}, s.t.

$$\sum_{r \neq i} f_{s_i,s_r}^{s_i,A_l} + f_{s_i,A_l}^{s_i,A_l} - g_i \lambda^{si,A_l} = 0 \tag{4.5}$$

$$\sum_{r \neq i,k} f_{s_i,s_r}^{s_k,A_l} + f_{s_i,A_l}^{s_k,A_l} - \sum_{m \neq i} f_{s_m,s_i}^{s_k,A_l} = 0 \tag{4.6}$$

$$\begin{aligned} &\Big(\textstyle\sum_{f_{s_i,s_r}^{s_k,A_l} \in F_{s_i,s}} c_{s_i,s_r} f_{s_i,s_r}^{s_k,A_l} + \sum_{f_{s_i,A_l}^{s_k,A_l} \in F_{s_i,A}} c_{s_i,A_l} f_{s_i,A_l}^{s_k,A_l} \\ &+ \textstyle\sum_{f_{s_m,s_i}^{s_k,A_l} \in F_{s,s_i}} P_{rx} f_{s_m,s_i}^{s_k,A_l} \Big) T_{l1} \leq e_i \, for \, (1 \leq i \leq N) \end{aligned} \tag{4.7}$$

$$\sum_{1 \leq l \leq M} \lambda^{s_i,A_l} = 1 \quad (1 \leq i \leq N) \tag{4.8}$$

$$\begin{gathered} T_{l1},\, f_{s_i,s_j}^{s_k,A_l},\, f_{s_i,A_l}^{s_k,A_l} \geq 0,\, \lambda^{s_i,A_l} = 0 \, or \, 1 \\ f_{s_i,s_j}^{s_k,A_l} \in F_{s,s}, f_{s_i,A_l}^{s_k,A_l} \in F_{s,A} \\ 1 \leq i,j,k \leq N,\, i \neq j,\, k \neq j,\, 1 \leq l \leq M \end{gathered}$$

Note that $\lambda^{s_i,,A_l}$ is a binary variable used for Actuator selection: if the data stream generated by a sensor i will be transmitted to actuator l, then $\lambda^{s_i,,A_l} = 1$; otherwise $\lambda^{s_i,,A_l} = 0$. The set of constraints in (4.5) to (4.8) can be interpreted as follows. The set of constraints in (4.5) focuses on traffic flow generated locally at each sensor i. They state that, for each sensor i, if actuator l is the destination, then the locally generated bit rate (*i.e.*, g_i) will be equal to the outgoing data flows from sensor i to actuator l via a single hop $\left(i.e.,\ f_{s_i,A_l}^{s_i,A_l}\right)$ or multihop $\left(i.e.,\ f_{s_i,s_r}^{s_i,A_l}\right)$; otherwise, all flows corresponding to the source-destination pair (s_i, A_l) must be zero. The set of constraints in (4.6) focus on the traffic that uses sensor i as a relay node. They state that at each relay sensor i, the total amount of incoming traffic $\left(i.e.,\ \sum_{m \neq i} f_{s_m,s_i}^{s_k,A_l}\right)$ should be the same as the total amount of outgoing traffic $\left(i.e.,\ \sum_{r \neq i,k} f_{s_i,s_r}^{s_k,A_l} + f_{s_i,A_l}^{s_k,A_l}\right)$ for each source-destination pair (s_i, A_l). The set of constraints in (4.7) concerns energy consumption at sensor i. They state that, for each sensor i, the energy consumption due to transmitting and receiving [see (4.2) and (4.3)] over the course of network lifetime should not exceed the initial energy supply e_i. Note that in (4.7) both flows generated locally at sensor i and those flows that use sensor i as a relay node are included. Finally the remaining set of constraints enforce that sensor i can only transmit all of its data to one actuator under any routing protocol, along with the logical restriction on the optimization variables λ^{s_i,A_l}, $f_{s_i,s_j}^{s_k,A_l}$, $and \, f_{s_i,A_l}^{s_k,A_l}$. Note that P_{rx}, g_i, e_i, c_{s_i,s_r}, $and \, c_{s_i,A_l}$ are all constants in this optimization problem.

The formulation of optimal flow routing and actuator selection is a mixed-integer non-linear programming (MINLP) problem, which is, unfortunately, NP-hard in general. We develop an upper bound for our optimal flow routing and actuator selection problem by studying a closely related problem that can be formulated and solved via linear programming. The non-linearity component in the flow routing problem can be removed by multiplying the equations (4.5)-(4.8) by T_{l1} and then use the linear substitutes $\left(V_{s_i,s_j}^{s_k,A_l} = T_{l1} \, . \, f_{s_i,s_j}^{s_k,A_l}\right)$, $\left(V_{s_i,A_l}^{s_k,A_l} = T_{l1} \, . \, f_{s_i,A_l}^{s_k,A_l}\right)$, and $\left(\mu^{s_i,A_l} = T_{l1} \, . \, \lambda^{s_i,A_l}\right)$.

Therefore, we maximize lifetime T_{l1}, s.t.

$$\sum_{r \neq i} V_{s_i,s_r}^{s_i,A_l} + V_{s_i,A_l}^{s_i,A_l} - g_i \mu^{si,A_l} = 0 \tag{4.9}$$

$$\sum_{r \neq i,k} V_{s_i,s_r}^{s_k,A_l} + V_{s_i,A_l}^{s_k,A_l} - \sum_{m \neq i} V_{s_m,s_i}^{s_k,A_l} = 0 \tag{4.10}$$

$$\begin{gathered} \textstyle\sum_{V_{s_i,s_r}^{s_k,A_l} \in \upsilon_{s_i,s}} c_{s_i,s_r} V_{s_i,s_r}^{s_k,A_l} + \textstyle\sum_{V_{s_i,A_l}^{s_k,A_l} \in \upsilon_{s_i,A}} c_{s_i,A_l} V_{s_i,A_l}^{s_k,A_l} \\ + \textstyle\sum_{V_{s_m,s_i}^{s_k,A_l} \in \upsilon_{s,s_i}} P_{rx} V_{s_m,s_i}^{s_k,A_l} \leq e_i \, for \, (1 \leq i \leq N) \end{gathered} \tag{4.11}$$

$$\sum_{1 \leq l \leq M} \mu^{s_i,A_l} - T_{l1} = 0 \quad (1 \leq i \leq N) \tag{4.12}$$

$$\begin{gathered} T_{l1}, \, V_{s_i,s_j}^{s_k,A_l}, \, V_{s_i,A_l}^{s_k,A_l}, \, \mu^{s_i,A_l} \geq 0 \\ V_{s_i,s_j}^{s_k,A_l} \in \upsilon_{s,s}, V_{s_i,A_l}^{s_k,A_l} \in \upsilon_{s,A} \\ 1 \leq i,j,k \leq N, \, i \neq j, \, k \neq j, \, 1 \leq l \leq M \end{gathered}$$

We now have a standard MILP formulation that was transformed directly from the MINLP. By their equivalence, the solution of this MILP problem yields an upper bound to the basic flow routing problem. We will use this bound as a reference point for comparisons. In the following, we will demonstrate the calculation for maximum lifetime

Maximize lifetime T_{l1}

Subject to

$$\sum_{r \neq i} f_{s_i,s_r}^{s_i,A_l} + f_{s_i,A_l}^{s_i,A_l} - g_i \lambda^{s_i,A_l} = 0 \tag{4.13}$$

$$\sum_{r \neq i,k} f_{s_i,s_r}^{s_k,A_l} + f_{s_i,A_l}^{s_k,A_l} - \sum_{m \neq i} f_{s_m,s_i}^{s_k,A_l} = 0 \tag{4.14}$$

$$\begin{gathered} \left(\sum_{f_{s_i,s_r}^{s_k,A_l} \in F_{s_i,s}} c_{s_i,s_r} f_{s_i,s_r}^{s_k,A_l} + \sum_{f_{s_i,A_l}^{s_k,A_l} \in F_{s_i,A}} c_{s_i,A_l} f_{s_i,A_l}^{s_k,A_l} \right. \\ \left. + \sum_{f_{s_m,s_i}^{s_k,A_l} \in F_{s,s_i}} P_{rx} f_{s_m,s_i}^{s_k,A_l} \right) T_{l1} \leq e_i \end{gathered} \tag{4.15}$$

$$\sum_{1 \leq l \leq M} \lambda^{s_i,A_l} = 1 \tag{4.16}$$

1. equation (4.13), (4.14) and (4.16) multiply T_{l1}, then we have the Lagrangian

$$
\begin{aligned}
L = T_{l1} + \lambda_1 & \left(\sum_{r \neq i} V_{s_i,s_r}^{s_i,A_l} + V_{s_i,A_l}^{s_i,A_l} - g_i \mu^{s_i,A_l} \right) \\
+\lambda_2 & \left(\sum_{r \neq i,k} V_{s_i,s_r}^{s_k,A_l} + V_{s_i,A_l}^{s_k,A_l} - \sum_{m \neq i} V_{sm,s_i}^{s_k,A_l} \right) \\
+\lambda_3 & \left(\sum_{V_{s_i,s_r}^{s_k,A_l} \in v s_i,s} c_{s_{i,}s_r} V_{s_i,s_r}^{s_k,A_l} + \sum_{V_{s_i,A_l}^{s_k,A_l} \in v s_i,A} c_{s_i,A_l} V_{s_i,A_l}^{s_k,A_l} \right. \\
& \left. \sum_{V_{sm,s_i}^{s_k,A_l} \in v s,s_i} P_{rx} V_{sm,s_i}^{s_k,A_l} - e_i \right) + \lambda_4 \left(\sum_{1 \leq l \leq M} \mu^{s_i,A_l} - T_{l1} \right)
\end{aligned}
$$

2. derivations

$$
\begin{aligned}
\nabla_{T_{l1}} L = 1 + \lambda_1 & \left(\sum_{r \neq i} f_{s_i,s_r}^{s_i,A_l} + f_{s_i,A_l}^{s_i,A_l} - g_i \lambda^{s_i,A_l} \right) \\
+\lambda_2 & \left(\sum_{r \neq i,k} f_{s_i,s_r}^{s_k,A_l} + f_{s_i,A_l}^{s_k,A_l} - \sum_{m \neq i} f_{sm,s_i}^{s_k,A_l} \right) \\
+\lambda_3 & \left(\sum_{f_{s_i,s_r}^{s_k,A_l} \in F s_i,s} c_{s_{i,}s_r} f_{s_i,s_r}^{s_k,A_l} + \sum_{f_{s_i,A_l}^{s_k,A_l} \in F s_i,A} c_{s_i,A_l} f_{s_i,A_l}^{s_k,A_l} \right. \\
& \left. + \sum_{f_{sm,s_i}^{s_k,A_l} \in F s,s_i} P_{rx} f_{sm,s_i}^{s_k,A_l} \right) + \lambda_4 \left(\sum_{1 \leq l \leq M} \lambda^{s_i,A_l} - 1 \right) = 0
\end{aligned}
$$

$$
\nabla_{\lambda_j} L = 0, \qquad 1 \leq j \leq 4
$$

then we have

$$
1 + \lambda_3 \frac{e_i}{T_{l1}} = 0
$$

and

$$
\begin{aligned}
\nabla_{\lambda_3} L = & \sum_{V_{s_i,s_r}^{s_k,A_l} \in v s_i,s} c_{s_{i,}s_r} V_{s_i,s_r}^{s_k,A_l} + \sum_{V_{s_i,A_l}^{s_k,A_l} \in v s_i,A} c_{s_i,A_l} V_{s_i,A_l}^{s_k,A_l} \\
& + \sum_{V_{sm,s_i}^{s_k,A_l} \in v s,s_i} P_{rx} V_{sm,s_i}^{s_k,A_l} - e_i = 0
\end{aligned}
\tag{4.17}
$$

According to the definition of the sets of volume, eq. (4.17) can be written as

$$\begin{aligned}&\sum_{l=1}^{M}\sum_{k=1}^{N}\sum_{r\neq i,k} c_{s_i,s_r} f_{s_i,s_r}^{s_k,A_l} + \sum_{l=1}^{M}\sum_{k=1}^{N} c_{s_i,A_l} f_{s_i,A_l}^{s_k,A_l}\\ &+\sum_{l=1}^{M}\sum_{k=1,k\neq i}^{N}\sum_{m\neq i} P_{rx} f_{sm,s_i}^{s_k,A_l} = \tfrac{e_i}{T_{l1}}\end{aligned} \tag{4.18}$$

It is necessary to discuss the summation of the k. The constraints (4.13) and (4.14) are satisfied in the case of $k = i$ and $k \neq i$, respectively. Therefore, eq. (4.18) can be written as

$$\begin{aligned}&\sum_{l=1}^{M}\left[\sum_{k\neq i}\left(\sum_{r\neq i,k} c_{s_i,s_r} f_{s_i,s_r}^{s_k,A_l} + c_{s_i,A_l} f_{s_i,A_l}^{s_k,A_l} + \sum_{m\neq i} P_{rx} f_{sm,s_i}^{s_k,A_l}\right)\right.\\ &\left.+\sum_{r\neq i} c_{s_i,s_r} f_{s_i,s_r}^{s_i,A_l} + c_{s_i,A_l} f_{s_i,A_l}^{s_i,A_l}\right] = \tfrac{e_i}{T_{l1}}\end{aligned} \tag{4.19}$$

From (4.14), we obtain:

$$\sum_{m\neq i} f_{sm,s_i}^{s_k,A_l} = \sum_{r\neq i,k} f_{s_i,s_r}^{s_k,A_l} + f_{s_i,A_l}^{s_k,A_l} \tag{4.20}$$

and from (4.13), we have

$$\sum_{r\neq i} f_{s_i,s_r}^{s_i,A_l} = g_i \lambda^{s_i,A_l} - f_{s_i,A_l}^{s_i,A_l} \tag{4.21}$$

Assume that c_{s_i,s_r} is constant, then substitute (4.20) and (4.21) into (4.19)

$$\begin{aligned}&\sum_{l=1}^{M}\left[\sum_{k\neq i}\left(\sum_{r\neq i,k} (c_{s_i,s_r} + P_{rx}) f_{s_i,s_r}^{s_k,A_l} + \left(c_{s_i,A_l} + P_{rx}\right) f_{s_i,A_l}^{s_k,A_l}\right.\right.\\ &\left.\left. c_{s_i,s_r} g_i \lambda^{s_i,A_l} + \left(c_{s_i,A_l} - c_{s_i,s_r}\right) f_{s_i,A_l}^{s_i,A_l}\right]\right. = \tfrac{e_i}{T_{l1}}\end{aligned} \tag{4.22}$$

Therefore, (4.22) becomes

$$\begin{aligned}&\sum_{l=1}^{M}\sum_{k\neq i}\sum_{r\neq i,k} (c_{s_i,s_r} + P_{rx}) f_{s_i,s_r}^{s_k,A_l} + \sum_{l=1}^{M}\sum_{k\neq i}\left(c_{s_i,A_l} + P_{rx}\right) f_{s_i,A_l}^{s_k,A_l}\\ &+\sum_{l=1}^{M} c_{s_i,s_r} g_i \lambda^{s_i,A_l} + \sum_{l=1}^{M}\left(c_{s_i,A_l} - c_{s_i,s_r}\right) f_{s_i,A_l}^{s_i,A_l} = \tfrac{e_i}{T_{l1}}\end{aligned}$$

then

$$\begin{aligned}&T_{l1}\left[\sum_{f_{s_i,s_r}^{s_k,A_l}\in F_{s_i,s},k\neq i} (c_{s_i,s_r} + P_{rx}) f_{s_i,s_r}^{s_k,A_l} + \sum_{f_{s_i,A_l}^{s_k,A_l}\in F_{s_i,A},k\neq i}\left(c_{s_i,A_l} + P_{rx}\right) f_{s_i,A_l}^{s_k,A_l}\right.\\ &\left. + c_{s_i,s_r} g_i + \sum_{l=1}^{M}\left(c_{s_i,A_l} - c_{s_i,s_r}\right) f_{s_i,A_l}^{s_i,A_l}\right] = e_i\end{aligned} \tag{4.23}$$

Hence, the optimal lifetime for a node i the under given set of constraints is given by (4.23). As we have found only one solution, we can assume that this is the optimal lifetime for the network.

Similarly, the joint problem of finding an optimal basestation and flow routing (to maximize network lifetime at level two, actuator-actuator coordination) can be modeled the same way as done for level one. We define

$$F_{A,A} = \left\{ f_{A_i,A_j}^{A_k,B_l} : (1 \leq i,j,k \leq M, i \neq j, j \neq k, 1 \leq l \leq B) \right\}$$

$$F_{A,B} = \left\{ f_{A_i,B_l}^{A_k,B_l} : (1 \leq i,k \leq M, 1 \leq l \leq B) \right\}$$

$$F_{A,A_i} = \left\{ f_{A_m,A_i}^{A_k,B_l} : (1 \leq m,k \leq M, m \neq i, k \neq i, 1 \leq l \leq B) \right\}$$

$$F_{A_i,A} = \left\{ f_{A_i,A_r}^{A_k,B_l} : (1 \leq r,k \leq M, r \neq k, r \neq i, 1 \leq l \leq B) \right\}$$

$$F_{A_i,B} = \left\{ f_{A_i,B_l}^{A_k,B_l} : (1 \leq k \leq M, 1 \leq l \leq B) \right\}$$

We maximize the lifetime T_{l2}(Lifetime at network level two), s.t.

$$\sum_{r \neq i} f_{A_i,A_r}^{A_i,B_l} + f_{A_i,B_l}^{A_i,B_l} - G_i \lambda^{A_i,B_l} = 0 \tag{4.24}$$

$$\sum_{r \neq i,k} f_{A_i,A_r}^{A_k,B_l} + f_{A_i,B_l}^{A_k,B_l} - \sum_{m \neq i} f_{A_m,A_i}^{A_k,B_l} = 0 \tag{4.25}$$

$$\begin{aligned} &\Big(\textstyle\sum_{f_{A_i,A_r}^{A_k,B_l} \in F_{A_i,A}} c_{A_i,A_r} f_{A_i,A_r}^{A_k,B_l} + \sum_{f_{A_i,B_l}^{A_k,B_l} \in F_{A_i,B}} c_{A_i,B_l} f_{A_i,B_l}^{A_k,B_l} \\ &+ \textstyle\sum_{f_{A_m,A_i}^{A_k,B_l} \in F_{A,A_i}} P_{rx} f_{A_m,A_i}^{A_k,B_l} \Big) T_{l2} \leq E_i \, for \, (1 \leq i \leq M) \end{aligned} \tag{4.26}$$

$$\sum_{1 \leq l \leq B} \lambda^{A_i,B_l} = 1 \quad (1 \leq i \leq M) \tag{4.27}$$

$$\begin{gathered} T_{l2},\, f_{A_i,A_j}^{A_k,B_l},\, f_{A_i,B_l}^{A_k,B_l} \geq 0,\, \lambda^{A_i,B_l} = 0 \, or \, 1 \\ f_{A_i,A_j}^{A_k,B_l} \in F_{A_i,A},\, f_{A_i,B_l}^{A_k,B_l} \in F_{A_i,B},\, 1 \leq i,j,k \leq M \\ i \neq j,\, k \neq j,\, 1 \leq l \leq B \end{gathered}$$

The set of constraints from (4.24) to (4.27) can be interpreted in the same way as (4.5) to (4.8). The formulation of optimal flow routing and basestation selection is again a MINLP problem. We develop a similar upper bound for flow routing and basestation selection problem that can be formulated and solved via linear programming (similar to the formulation as level one). Here, we only present the modeling of optimal flow routing and basestation selection. An optimization criteria similar to level one can be opted here to solve the system of equations. The non-linearity component in the flow routing problem can be removed by multiplying the equations (4.24)-(4.27) by T_{l2} and then use the linear substitutes $\left(V_{A_i,A_j}^{A_k,B_l} = T_{l2} . f_{A_i,A_j}^{A_k,B_l} \right)$,

$\left(V^{A_k,B_l}_{A_i,B_l} = T_{l2}.\ f^{A_k,B_l}_{A_i,B_l}\right)$, and $\left(\mu^{A_i,B_l} = T_{l2}.\lambda^{A_i,B_l}\right)$. Then, the MINLP problem can be reformulated into the equivalent MILP problem as shown below. We maximize lifetime T_{l2}, s.t. ((4.24)-(4.27)).T_{l2} with T_{l2}, $V^{A_k,B_l}_{A_i,A_j}$, $V^{A_k,B_l}_{A_i,B_l} \geq 0$, $V^{A_k,B_l}_{A_i,A_j} \in \upsilon_{A,A}, V^{A_k,B_l}_{A_i,B_l} \in \upsilon_{A,B}$, and $1 \leq i,j,k \leq M$, $i \neq j$, $k \neq j$, $1 \leq l \leq B$.

Since, we have only one sink in the network, the network of basestations can form an *aggregation* tree toward this common sink. The flow from a basestation can be splitted and send over multiple routes toward the sink. The flow problem to extend network lifetime at level three can be written in similar fashion as (4.9) to (4.12) with their appropriate subscripts (note that in this case, all the data gathered at different basestations is sent to a common sink and hence, the optimal flow solution formulation will results in an NLP formulation (which can be relaxed using same technique as presented earlier to an equivalent LP formulation), and is therefore, not presented here.

4.7 LEAD-ADP: The LEAD Actuator Discovery Protocol

In order to remove the mixed-integer (MI) component from the MILP, we consider the following *distributed* learning mechanism that helps in selecting actuators for each sensor in the network. We propose a framework which is tailored toward a standard behavior for most deployment scenarios, aiming to satiate the time-stringent requirements and energy efficient resource utilization in a purely distributed fashion [C-1]. The proposal consists of three phases: the *learning* phase, the *coordination* phase, and the *failure-and-recovery* phase. In the following, we detail the three phases.

4.7.1 The Learning-phase

The learning-phase starts during the initial *deployment* stage when the sensors locate the neighboring actuators using a one-hop broadcast. The finding of the "optimal-actuator attachment" for each sensor node is done through a novel protocol called ADP.

4.7.1.1 Actuator-discovery Protocol (ADP)

When a sensor node is turned on, it should first determine an actuator node as the final destination. For this end, a sensor node transmits a broadcast message named

$$AttachRequest\,(cost, M_j, C)$$

to its one-hop neighbors as shown in Figure 4.3. A neighboring node upon receiving an attach-request message checks that it has sent an attach-request in the period T_n (application specific), if it has already sent a broadcast to its neighbors, it will wait for a reply until timeout. Otherwise, it repeats this procedure unless the probe reaches an actuator. The reply message named

$$AttachReply_i\,(cost,\ M_j, A_i)$$

from the actuator follows the probe and terminates at its origin, defining a *discrete* path to the sensor node. If a node receives multiple replies, it chooses a destination actuator based on the outcome of a *cost function.*

In Algorithm 4.1, we have induced a control procedure to obtain a promised QoS in terms of delay and energy consumption. We have assigned the hop-count to this function to restrict the

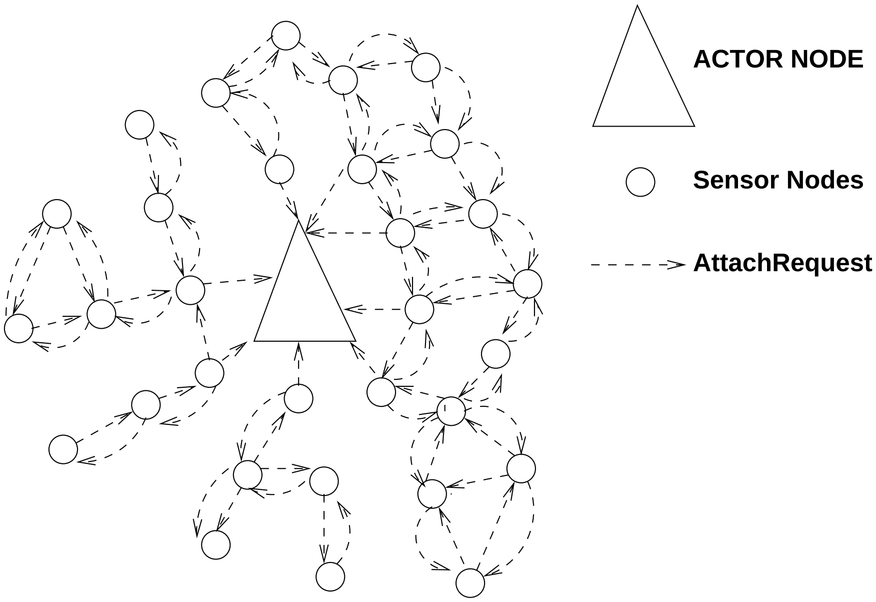

Figure 4.3: AttachRequest by sensors at the start of ADP

search probe (referred as 'C'). This hop-count can be treated as a function of sensor-actuator node ratio in the network to limit unnecessary broadcast and also keeping the chances of actuator-discovery well alive (leaving the issue as implementation concern). We don't take into account the distance between a sensor and its associated neighbors because the energy required to transmit to a node in its sensing radius is a constant (no power control assumed for transmissions). ADP produces loop-free paths to the actuator nodes, as stated below.

LEMMA 1. *The next-hop selected by a sensor with ADP has a defined optimal path to the actuator node, Algorithm. 4.1.*

As depicted by Figure 4.4, now a sensor node has some defined paths to route its sensed data to the actuator nodes by simply forwarding it to one of its one-hop neighbors (immediate next node in the path to the actuator), and the actuator also keeps the defined path to the node (building its *tree* structure for the localized cluster). In a similar fashion all the nodes reserve an optimal path to their nearest actors as shown in Figure 4.5, forming a local *cluster*, thus giving us the initial deployment in the form of *distributed* clusters. The cluster information available at the actuator will be used for scheduling in a later section.

4.7.1.2 Correlation Trees

Once all the nodes have defined paths to their attached actuator, the actuator rearranges all the paths to exploit correlation properties of the SANETs. As shown in Figure 4.6, the actuator rearranges all the paths in the depth-first arrangement order. In this way, we have all the one-hop sensor nodes as the first children of the actuator node, so on and so forth. This gives a depth-first search tree structure. All the sensor-nodes have defined identities (names, address, etc). But when a cluster is created and organized into the tree form by the actuator, it assigns temporary addresses to the sensor nodes and keeps the mapping with itself. As depicted in Algorithm 4.2 once the tree structure is maintained we define the temporary addresses of nodes by addressing all the nodes on the same hop-count first, following their

Algorithm 4.1 LEAD-ADP

```
Pseudo-code executed by all the sensor nodes $N_i$ during initial deployment-phase.
Initially:
 cost = $\infty$
 attached-actuator = $\infty$
 C = constant (the trade-off is explained in Section 4.7.1.1).
 $A_i$ = Identity of the Actuator.
 For any sensor node $N_i$
 do $ActorDiscovery$() {
   if cost ($N_i, A_i$) = $\infty$ then
     for each neighbor $M_j$ of $N_i$ do
       Send $AttachRequest(cost, M_j, C)$
       Receive $AttachReply_i(cost, M_j, A_i)$
#Determine optimal Actuator, and the next-hop among the neighbors to reach
it.
        for each $AttachReply$ do
          if path($cost, M_j$) < path($cost, M_{j-1}$) then
            for $N_i$ MinCost = $path$(cost, $M_j$)
            AttachedActuator = $A_i$
            next_ho_to_actuator = $M_j$
          end-if
        end-for
     end-for
   end-if
}
After deciding the actuator, each node sends a "JoinRequest" to its actuator.
  send $JoinRequest(A_i)$
The actuator sends a "JoinAck" back to the sensor node confirming cluster
joining.
  send $JoinAck(M_j, N_i)$
The procedure attach-request is implemented recursively as follows.
  $AttachRequest(cost, M_j, C)${
    if ($cost$ != $\infty$)
      return ($UpdateCost(cost)$, $M_j$, $A_i$ )
    else if (C != 0) then
      for all neighbors $M_j$ of $N_i$
        do $AttachRequest(cost, M_j, C-1)$
      end-for
    end-if
}
Actuator Reply to the broadcast messages from the one-hop away nodes contains the
following.
  $AttachRequest(cost, M_j, C) \leftarrow ActorReply$(cost = 1, $A_i$)
$UpdateCost()$ is the part of the control semantics, and for this specific case,
it is chosen to be hop-count
$UpdateCost(cost)$ {
  return cost + 1
}
```

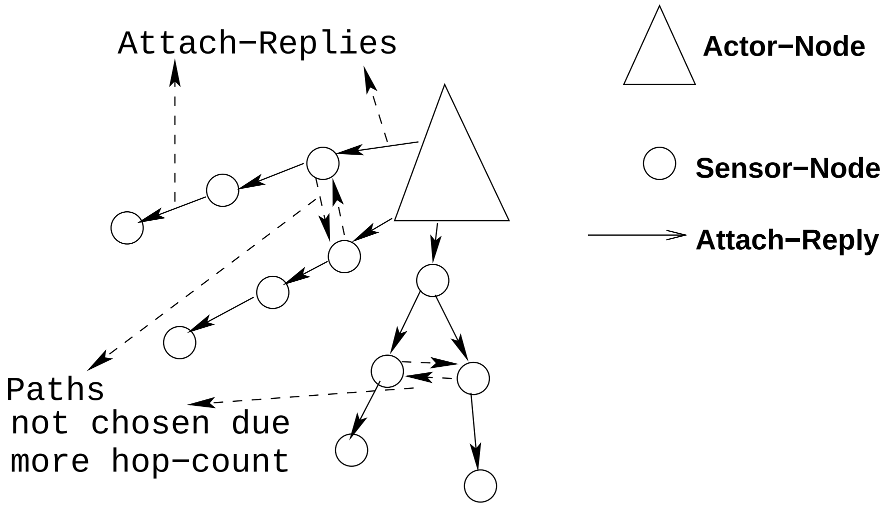

Figure 4.4: Actuator-replies (AttachReply) for corresponding AttachRequest messages

descendants aiming toward a breadth-first addressing scheme. The mapping between the actual node-address and temporary-address is managed by the actuator ($N_{add}^{(i)} \rightarrow T_{add}^{(i)}$) in every cluster. This strategy helps in optimizing the search to the attached neighbors in case of node *mobility* and *failure*, and exploiting the *correlation* properties (see [C-1] for details).

LEMMA 2. *All the sensor-nodes are attached to the actuator with increasing hop-count in a depth-first order, Algorithm. 4.2.*

4.7.2 The Coordination-phase

The deployed sensor nodes start sensing the distributed environment, and transmit their data through the defined path to the attached actuator. For the sensor-actuator coordination, the actuator-attachment and the paths obtained to route data to the actors provide effective energy optimization for the sensor nodes. There can be two deployment configurations for the SANETs:

Static Deployment: In this case, both sensor and actuator nodes are static and the gain is maximum due to efficient routing of data to the acquired actuators.

Mobile Deployment: For mobile deployment, we have four different types of configurations (detailed in Section 4.7.3). The learning phase for mobile-case is essentially the same as for the static-deployment. But at any point in time, the discrete path to the actuator nodes may change due to the mobility of the nodes. The purpose behind organizing the cluster in the above-explained behavior is to exploit the correlation properties (see [C-1]) of the SANETs not only at the data-centric level but also at the node-centric level (direct-addressing).

4.7.3 Failure and Recovery-phase

We assumed that every sensor node has a pre-defined maximum battery life-time with a minimum threshold indicating failure in near future. The Failure and Recovery-phase monitors this time line and inform the actuator before the actual failure to take a few precautionary measures which includes: (i) exploiting the local cluster for an alternate path to nodes that lost their routes to the actuator. (ii) do nothing if there was no further attached node. (iii)

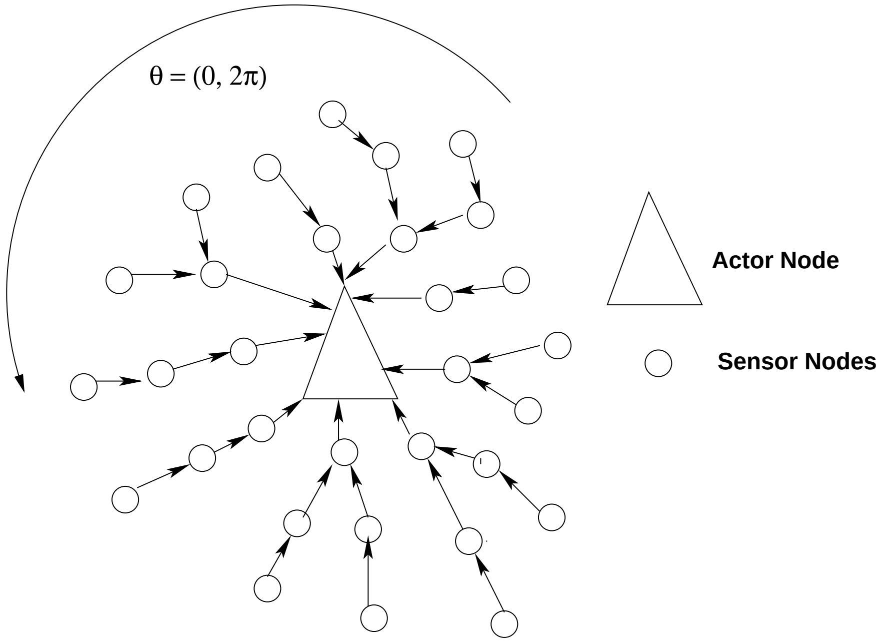

Figure 4.5: The Local Cluster formulated at the termination of ADP

update the cluster information of the attached-actuator for local management as shown in Figure 4.7. Further details can be found in [C-1].

To decide on the optimal actuator, we consider that the cost function is set to *min. hop* count and the actuators chosen by sensors are optimal in min. hop sense. *An advantage of setting the cost-function to min. hop routing is that the lower-tier (level one) of our heterogeneous network can be organized into clusters, where each cluster is centrally managed by an actuator.* It will also result in the disappearance of the mixed-integer (MI) component from the optimization problem and the resultant is a *relaxed* linear optimization problem (LP) which is comparatively easier to solve [94]. In this fashion, a sensor can receive its scheduling information (detailed in Section 4.9) by its *mapped* destination-actuator that corresponds to the *optimal* (s.t. energy constraints) routing solution, and hence, can result in the *realization* of optimal network lifetime in *practice*. We denote the resulting destination for a sensor via the above *mapping* as $d(i)$. Therefore we have, $\mu^{s_i,d(i)} = T$, and $\mu^{s_i,A_l} = 0$ for $A_l \neq d(i)$. Then, we can find a routing solution by replacing A_l in (4.9) to (4.12) by $d(i)$ (destination actuator for sensor i, and is thus, not repeated here).

4.8 Deterministic Lifetime Maximization

In this section, we develop, using the Lagrangian *dual* decomposition method, a *distributed* algorithm to maximize the network *lifetime* with no *feedback* control.

4.8.1 Lagrange Dual Approach

In what follows, we use the Lagrange dual decomposition method to solve the minimization problem. The Lagrangian function with the Lagrange multipliers (λ_i) is given as follows:

Algorithm 4.2 SOT

```
Pseudo-code executed by the actuator-node
scan the local-cluster $\Theta$= [0,2$\pi$)for all nodes
    do $SOT$(node n)
      $visit$(n)
      for each child (next-hop) w of n
        do $SOT$(w) (Initially, first scanned node one-hop away from actuator)
        add edge nw to the Tree $SOT$
      end for
A temporary-address is assigned to each node by the actuator.
Unmark all the vertices
    choose some starting vertex n (actuator-node)
      $mark$(n)
      list $L$ = n
      Tree $SOT$ = n
        while $L$ non-empty
choose some vertex v from the front of list
        $visit$(v)
        $AssignAdd$(v)
        for each unmarked neighbor w
          $mark$(w)
          $AssignAdd$(W)
          add it to the end of list
          add edge vw to $SOT$.
        end-for
```

$$L\left(T_{li},\lambda_1,\lambda_2,\lambda_3,\lambda_4\right)=$$

$$T_{li}+\lambda_1 T_{li}\left(\sum_{r\neq i} f^{s_i,d(i)}_{s_i,s_r}+f^{s_i,d(i)}_{s_i,d(i)}-g_i\right)+$$

$$\lambda_2 T_{li}\left(\sum_{k\neq i}\sum_{r\neq i,k} f^{s_k,d(i)}_{s_i,s_r}+f^{s_k,d(i)}_{s_i,d(i)}-\sum_{k\neq i}\sum_{m\neq i} f^{s_k,d(i)}_{s_m,s_i}\right)+$$

$$\lambda_3\left(T_{li}\left(\sum_{f^{s_k,d(i)}_{s_i,s_r}\in F_{s_i,s}} c_{s_i,s_r} f^{s_k,d(i)}_{s_i,s_r}+\sum_{f^{s_k,d(i)}_{s_i,d(i)}\in F_{s_i,A}}\right.\right.$$

$$\left.\left.c_{s_i,A_l} f^{s_k,d(i)}_{s_i,d(i)}+\sum_{f^{s_k,d(i)}_{s_m,s_i}\in F_{s,s_i}} p_{rx} f^{s_k,d(i)}_{s_m,s_i}\right)-e_i\right)$$

Then, the Lagrange dual function is

$$Q\left(\lambda_1,\lambda_2,\lambda_3\right)=\max_{T_{li}} L$$

Thus, the dual problem is given by

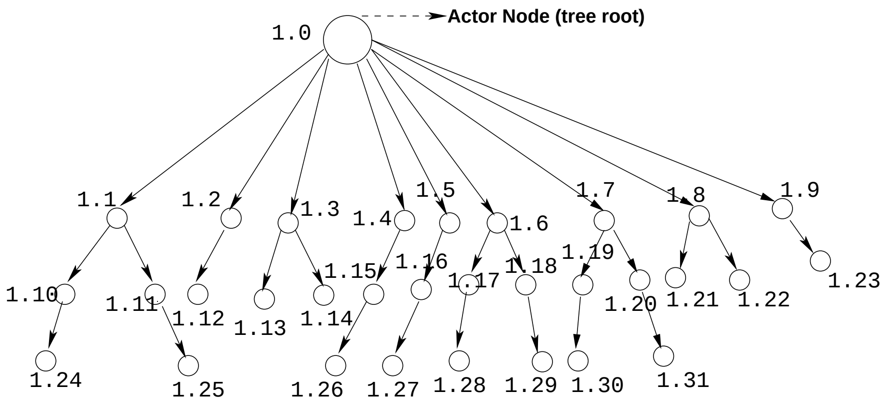

Figure 4.6: A Self-Organized Tree (SOT)

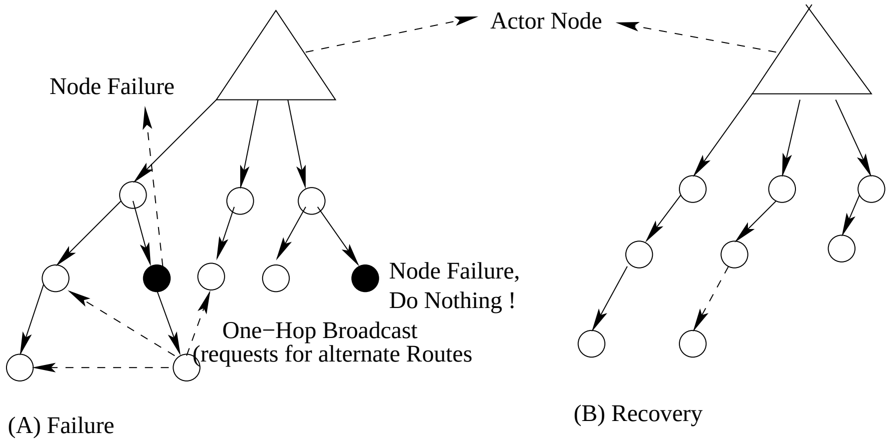

Figure 4.7: The occurrence of failure and steps required for recovery procedure.

$$D : \min_{\lambda_1, \lambda_2, \lambda_3 \geq 0} Q\left(\lambda_1, \lambda_2, \lambda_3\right)$$

4.8.2 Deterministic Primal-Dual Algorithm

The lifetime maximization problem can be solved via the following deterministic distributed algorithm

- The $T_{li}'s$ are updated as

$$T_{li}\left(n+1\right) = T_{li}\left(n\right) + \epsilon_n \nabla_{T_{li}} L$$

$$T_{li}\left(n\right) + \epsilon_n \left[\lambda_1 \left(\sum_{r \neq i} f_{s_i, s_r}^{s_i, d(i)} + f_{s_i, d(i)}^{s_i, d(i)} - g_i \right) + \right.$$

$$\lambda_2 \left(\sum_{k \neq i} \sum_{r \neq i, k} f_{s_i, s_r}^{s_k, d(i)} + f_{s_i, d(i)}^{s_k, d(i)} - \sum_{k \neq i} \sum_{m \neq i} f_{s_m, s_i}^{s_k, d(i)} \right) +$$

$$\lambda_3 \left(\sum_{f_{s_i,s_r}^{s_k,d(i)} \in F_{s_i,s}} c_{s_i,s_r} f_{s_i,s_r}^{s_k,d(i)} + \sum_{f_{s_i,d(i)}^{s_k,d(i)} \in F_{s_i,A}} \right.$$

$$\left. c_{s_i,A_l} f_{s_i,d(i)}^{s_k,d(i)} + \sum_{f_{s_m,s_i}^{s_k,d(i)} \in F_{s,s_i}} p_{rx} f_{s_m,s_i}^{s_k,d(i)} \right) \Bigg]$$

- The Lagrange multipliers are updated by

$$\lambda_1 (n+1) = \lambda_1 (n) - \epsilon_n \nabla_{\lambda_1} L$$

$$= \lambda_{1(n)} - \epsilon_n T_{li} \left(\sum_{r \neq i} f_{s_i,s_r}^{s_i,d(i)} + f_{s_i,d(i)}^{s_i,d(i)} - g_i \right)$$

$$\lambda_2 (n+1) = \lambda_2 (n) - \epsilon_n \nabla_{\lambda 2} L$$

$$= \lambda_2 (n) - \epsilon_n T_{li} \left(\sum_{k \neq i} \sum_{r \neq i,k} f_{s_i,s_r}^{s_k,d(i)} + f_{s_i,d(i)}^{s_k,d(i)} \right.$$

$$\left. - \sum_{k \neq i} \sum_{m \neq i} f_{s_m,s_i}^{s_k,d(i)} \right)$$

$$\lambda_3 (n+1) = \lambda_3 (n) - \epsilon_n \nabla_{\lambda_3} L$$

$$= \lambda_3 (n) - \epsilon_n \left[T_{li} \left(\sum_{f_{s_i,s_r}^{s_k,d(i)} \in F_{s_i,s}} c_{s_i,s_r} f_{s_i,s_r}^{s_k,d(i)} \right. \right.$$

$$+ \sum_{f_{s_i,d(i)}^{s_k,d(i)} \in F_{s_i,A}} c_{s_i,A_l} f_{s_i,d(i)}^{s_k,d(i)} +$$

$$\left. \left. \sum_{f_{s_m,s_i}^{s_k,d(i)} \in F_{s,s_i}} p_{rx} f_{s_m,s_i}^{s_k,d(i)} \right) - e_i \right]$$

We note that in the above algorithm, we have used the same *step* size ϵ_n for both the *primal* and the *dual* algorithms.

4.9 LEAD-MAC: The LEAD Medium Access Control

Once the optimal actuators are decided for each sensor in the network and optimal flow routing is formulated, then the sensors can be scheduled using a TDMA like MAC protocol that corresponds to the flow solution. The actuators explicitly schedule all the sensors based on their knowledge of the cluster. If a random access scheme is used at the MAC layer, then the *experienced* network lifetime can not correspond to the optimal routing solution (4.23) as the optimal routing solution is only based on the flow coming into-and-out of a sensor node subject to energy-constraints. This scheme does not take into account the access energy wasted due to *collisions* and *successive retransmissions*. The detailed power consumption model at the MAC layer can be seen in [C-5]. For the considered model, we optimize the system performance using a TDMA-MAC protocol by minimizing the *awake periods* and power loss due to *interference*. LEAD-MAC has three operational phases: (i) network learning phase, (ii) scheduling phase, and (iii) adjustment phase. The following discussion covers the different protocol phases in detail.

4.9.1 Network Learning Phase

A sensor node finds an optimal actuator using the proposed ADP (Actuator Discovery Protocol, a controlled flooding mechanism in Section 4.7), during the initial deployment phase. The sensors start the learning phase by transmitting a one hop broadcast *search_ request*. When a broadcast reaches an actuator, it is replied with the actuator identity. A random access scheme is used in the topology learning phase, because the sensors do not yet have a transmission schedule. The scheme is designed so that, at the end of this phase, almost all nodes are attached (based on the outcome of an objective function) to an actuator and correctly determine their neighbors and interferers with high probability. We adopt a carrier sense multiple access (CSMA) mechanism similar to IEEE 802.11 [95]. The sensors listen for a random time before transmitting, and transmit if the channel is idle. A random delay is added before carrier sensing to further reduce collisions. However, because a collision will lead to incomplete cluster information at the actuators, the CSMA scheme itself cannot guarantee that an actuator will receive the full cluster information. Therefore, an acknowledgment from the actuator is sent when a sensor transmits a packet to join a particular cluster.

4.9.2 Scheduling Phase

The actuator explicitly schedules all the sensors, based on its knowledge of the cluster. An actuator schedules the sensors in the depth-first order for end-to-end routes, and in a breadth-first order for any given parent node i, to capture forwarded data from all of its downlink sensors. At the end of the network learning phase, the network may be represented by $G = (V, E)$, in which V is the set of nodes, including the actuator node A_i. The undirected edges $E \in V \times V$ are the (transmission) links to be scheduled. The graph forms a *tree*, rooted at actuator A_i. All the traffic from any given tree is destined for the actuator, so every data packet at a node is forwarded to the node's parent. A node may interfere with another node, so these nodes should not transmit simultaneously. The interference graph $C = (V, I)$ is assumed known at the end of network learning phase. Here $I \in V \times V$ is the set of edges such that $(u, v) \in I$ if either u or v can hear each other or one can interfere with the signal of the other. The conflict graph corresponding to $G = (V, E)$ and $C = (V, I)$ is the graph $GC = (V, EC)$ in which EC comprises the edges between node pairs that should not transmit at the same time.

The Scheduling Problem: Each node of G (except the actuator) generates packets. Given the interference graph C, the scheduling problem is to find a minimum length frame during which all nodes can send their packets to the actuator using minimum energy. The problem is NP-complete [101]. We reduce the NP-complete problem of finding the chromatic number of a graph to the scheduling problem[3]. If the original tree network has depth N, the linear network $GL = (VL, EL)$ has nodes $VL = \{A_i, v_1, \ldots, v_N\}$ with node v_l corresponding to level l in the original network. The interference graph $CL = (VL, IL)$ includes edges (v_j, v_l) if there is an interference between a node at level j and a node at level l in the original network. We can now color the linear network using the same approach used in [101]. Then, the scheduling algorithm is given by 4.3. For all the basic proofs on NP-completeness and coloring of network, please refer to [101], as we do not repeat them here in order to conserve space.

Algorithm 4.3 The Scheduling Algorithm

```
Input: Graph $G = (V, E)$of the original tree network, with interference $C = (V, I)$ , color assignment of the corresponding linear network, using $K$ colors, such that each color corresponds to a maximum non-conflicting set.
Output: Transmission schedule for nodes of $G$.
    begin
        while (atleast one packet has not reached the Actuator)
            for $s = 1$ to $K$
                sets = set of levels corresponding to color $s$
                    T=0
                        while (nodes of color $s$ has atleast one packet)
                            for $j = 1$ to $|sets|$
                                $T = T \cup T\{maximum\, non-conflicting\, set\, of\, nodes\, from\, level\, j \in sets\, with\, atleast\, one\, packet\}$
                                if $T \neq 0$
                                    assign this slot to set $T$
                                    update the place of packets
                        end
            end
        end
```

The scheduling frame duration T is divided into slots (a single slot-duration depends on the packet size, available transmission rate, and is typically application dependent). A slot extends the packet duration by a guard interval to compensate for synchronization errors. At the beginning of this phase, an actuator broadcasts the *scheduling packet* using maximum transmit power. Since the actuator reaches all the sensors at the same time, the error in synchronization from the delay between time-stamping and sending the packet at the transmitter is eliminated. Since the range of an actuator is on the order of kilometers, the propagation delay is also negligible (few μsec). Based on the assumption that all the nodes run the same software, all of them will time-stamp the packet at the same time. Therefore, the only error of synchronization in this application comes from clock skew, the difference in the clock tick rates of the nodes. Typical clock drifts of a sensor node in 1sec is 10 μsec [97]. If the packet generation period of each node is around 30 sec, the maximum clock drift will be 0.3 msec compared to approx. 20 msec (the duration of the packet transmission of a packet of 50 byte at 50kbps). The total time-frame duration of this schedule is given by T, which depends on the number of sensors in the cluster. Therefore, the frame duration T is different for every cluster in the network.

[3] The chromatic number of a graph G is the smallest number k such that G is $k-colorable$ if its vertices can be colored using k different colors in such a way that adjacent vertices have different colors.

The minimum duration for a sensor to stay awake T_a is $T_a = T_{rx} + T_{tx} + T_g$, where T_{rx} is the time required to receive a packet, T_{tx} is the time required to transmit one packet to the parent node, and T_g is the guard interval for synchronization errors. The interval T_g is assumed to be a small percentage of the total slot duration. The maximum duration for a sensor to stay awake depends on its sub-tree and can be calculated as a multiple of T_a depending on the application, e.g., if the application allows for *data aggregation*: a sensor can receive forwarded data from its sub-tree, aggregate its own packet and transmit the resultant packet requiring only one *time slot*.

The first transmitted packet to contain the CDMA code is the *collision-free* TDMA schedule by each actuator in the network, so that the sensors receive the schedule from their attached actuator only once. The schedule packet contains a *current-time* field in order for all the sensors in one cluster to *synchronize* to a common clock before starting the transmissions and a *next-time* field, where all the sensors wakeup once in order to *resynchronize* to the common clock.

4.9.3 Adjustment Phase

If a new node is added to the network or a *link level* failure is detected in the network, a sensor will try to attach itself by transmitting a *one-hop* broadcast request in its neighborhood. All the sensors in the network wakeup at *next-time* to resynchronize to the network. At this time, a sensor which receives the *actuator-search* broadcast replies to the sensor with its actuator *id* and *cost* to reach the actuator. Upon receiving the reply to its broadcast (there can be multiple replies), a sensor decides its optimal actuator and transmits an *attachment* request to the actuator. The new sensor is added to the transmission schedule and also acquires the same CDMA code as its *cluster*.

4.10 LEAD-Wakeup

The main idea of LEAD-Wakeup protocol is to extend the scheduling for event-driven sensing applications, where the slots assigned to the nodes do not have to be used. According to the adopted scheduling scheme, all the nodes of one routing path remain active only for a small duration T_a to check the possibility of arrival of forwarded data.

4.10.1 Adaptivity to Network Conditions

A sensor wakes up at the scheduled time to see if it has any new sensed data in its transmit queue. If it has no data to transmit and also, no data arrives from its children sensors during a defined interval (which is equivalent to the reception time for one packet and a guard interval), it immediately goes back to the sleep mode and saves considerable amount of energy (*adaptive duty cycle*). The duration of this adaptive validation period is at least equivalent to $T_{adapt} = T_{rx} + T_g$.

4.10.2 Analysis of LEAD Wakeup

In [102], the authors have shown that the depth-first scheduling works better than the breadth-first scheme for end-to-end delay, throughput and forwarding queue size at the sensors, but fails to perform well in energy consumption compared to the breadth-first scheduling. In this work, we will show that the *hybrid scheduling* scheme with an adaptive duty cycle achieves a good trade-off between the sensor energy consumption and end-to-end delay for sensor-actuator applications.

Energy Consumption: For a wireless sensor, typical states are "active", "idle", and "sleep". The energy saved for a sensor by not sending the sensors directly to *active* state is shown in Figure 4.8. The only form of overhead seen in this power management is the time spent in settling from one state to another and is given by

$$E_{overhead} = T_{si}.\left(\frac{P_i - P_s}{2}\right) + T_{ia}.\left(\frac{P_a - P_i}{2}\right) \tag{4.28}$$

where T_{si} and T_{ia} is the time required to change the state from sleep-to-idle and idle-to-awake, respectively. The appearance of 2 in denominator is not due to the fact that the flanks are triangular, but simply because, it represents exactly half the energy to what could be spent otherwise. The gain in energy using such an adaptive power management can be seen as

$$\begin{aligned} E_{saved} = (\delta t - T_{si}).(P_a - P_i) + \\ T_{si}.\left(P_a - \left(\tfrac{P_i + P_s}{2}\right)\right) + T_{ia}\left(\tfrac{P_a - P_i}{2}\right) \end{aligned} \tag{4.29}$$

where $\delta t = T_{event} - T_1$, T_1= start of the adaptive awake period T_{adapt} and T_{event} = time of arrival of an event (transmission or reception).

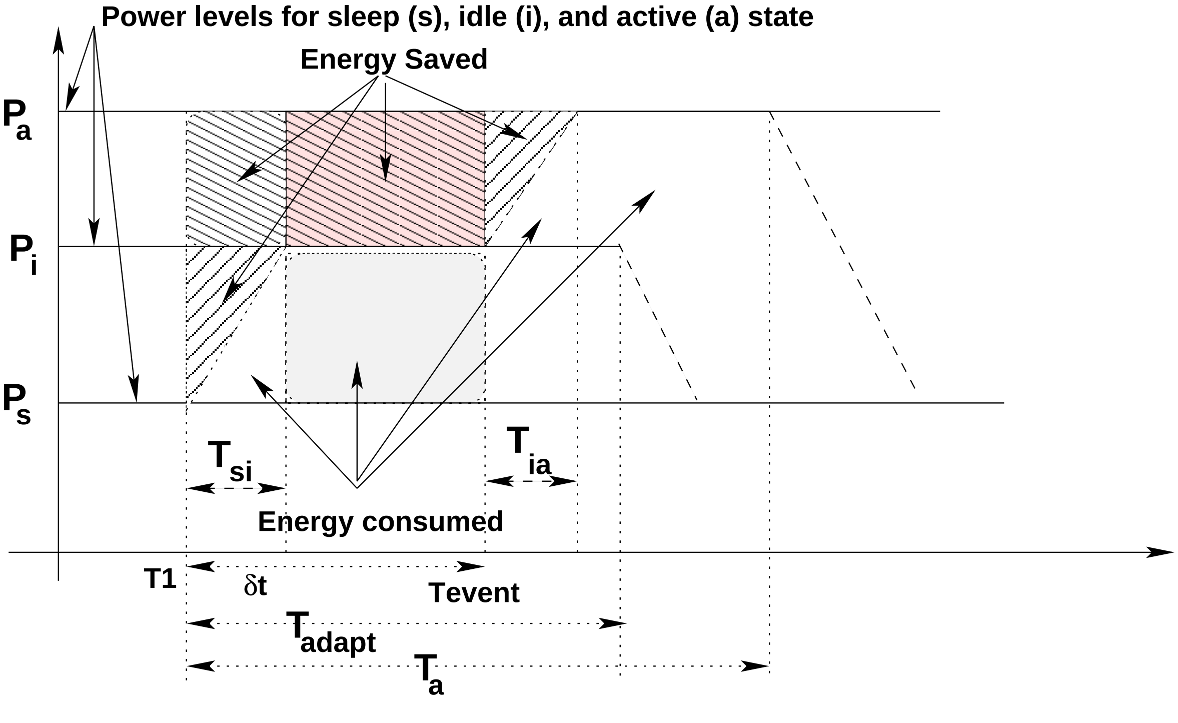

Figure 4.8: Energy Savings through adaptive duty cycle

If we see the case with Intel strong ARM (Table 4.2), it is sensing the environment in S_1. At scheduled time, it will change its sleep state from S_1 to S_2. The sensor node will stay in this state until the arrival of event for T_{adapt}, if it do not receive a data packet, and itself has no packet to transmit then it will go back to sleep. Otherwise, it will jump to S_4 to transmit a packet. The minimum energy consumed by a sensor during one time frame T is given by

$$MinE_i = E_i^{s1}.(T - T_{adapt}) + E_i^{s2}.T_{adapt} \tag{4.30}$$

Similarly, maximum energy consumed by a sensor during T is given by

$$MaxE_i = E_i^{s1}.(T - T_a) + E_i^{s2}.\delta t + E_i^{s4}.(T_a - \delta t) \tag{4.31}$$

The sensors only wakeup when a transmission or reception is expected, therefore, we save the *expected* energy drain due to interference from two-hop neighbors. Due to an adaptive sleep schedule, a sensor saves energy by configuring its transmitter state to sleep. Therefore, the complex lifetime equation in [C-5] is reduced to

$$T_{life}^{i} = \frac{E_i}{\left(P_{rx}\sum_{j\in N_i}\alpha_{j,i} + P_{tx}\sum_{j\in N_i}\alpha_{i,j} + P_{sense}\lambda_i\right)} \tag{4.32}$$

Observed Latency: The average latency seen by a packet from $node_i$ is

$$delay_i = \sum_{K}(\delta_{s_2-s_4} + T_{data}) \tag{4.33}$$

where T_{data} is the time required to actually transmit a packet and K is the number of hops toward the actuator of $node_i$. And the worst case latency seen by a packet from $node_i$

$$delay_i = \sum_{K}(\delta_{s_2-s_4} + T_{data}) + T \tag{4.34}$$

which can happen if an event arrival in the current awake duration T_a does not reach the actuator due to long paths.

4.11 Actuator to Sensor Transmission Schemes

In this section, three different actuator-to-sensor transmission schemes are presented together with their analysis and performance evaluation [C-9]. Actuators can all transmit at the same frequency and therefore interfere with each other. They can also transmit at different frequencies in order to avoid interfering with each other at the cost of higher frequency reuse factor.

4.11.1 Transmission at a single frequency (Reuse Factor 1)

In this case, each actuator communicates with the sensor nodes that are assigned to it. The actuator broadcasts a packet containing scheduling information, for the sensors nodes attached to it, at the same frequency. Each sensor node receives together with useful scheduling information, co-channel interference (CCI) from other actuators. The received signal of the sensor node i is

$$y_i = h_{ij}\sqrt{P_j}x_j + \sum_{k\neq j} h_{ik}\sqrt{P_k}x_k + n \tag{4.35}$$

where $i = 1, 2, ..., K$, h_{ij} is defined in (4.1), j is the actuator that the sensor i is assigned to, P_j is the transmit power of each actuator, n is the additive white Gaussian noise (AWGN) component with power σ^2, and x_j is the transmitted scheduling information of actuator j.

Throughout this chapter it is assumed that all actuators transmit on the same power level. It is also assumed that $\mathrm{E}\left\|\mathrm{x_m}\right\|^2 = 1$. The packet x_j contains the schedules of all sensor nodes attached to actuator j. $\sum_{k \neq j} h_{ik}\sqrt{P_k}x_k$ represents the detrimental CCI term. Therefore, the Signal-to-Interference-Noise Ratio (SINR) of a sensor node i is

$$SINR_i = \frac{\left\|h_{ij}\right\|^2 P_j}{\sum_{k \neq j}\left\|h_{ij}\right\|^2 P_k + \sigma^2} \tag{4.36}$$

If $SINR_i$ is below a certain threshold T ($SINR_i < T$), sensor node i is unable to decode its scheduling information and therefore it is unable to resolve when to transmit its sensed data. Thus, it will remain isolated.

The advantage of this scheme is that each actuator, in order to distribute sensor scheduling information it broadcasts a packet that contains all sensor schedules. Therefore, in one time slot, all schedules are distributed. However, each sensor needs to go through all the contents of the scheduling packet in order to find its own schedule, a fact that increases decoding complexity. The main disadvantage is that some sensor nodes might remain isolated as described above.

4.11.2 Transmissions at different frequencies (Higher Reuse Factor)

In this case also, each actuator communicates with the sensor nodes that are associated with it. Each actuator broadcasts its scheduling information at a different frequency. This eliminates CCI at the cost of a higher frequency reuse factor (RF). The received signal at the sensor i is then

$$y_i = h_{ij}\sqrt{P_j}x_j + n \tag{4.37}$$

where $i = 1, 2, ..., K$. The Signal-to-Noise Ratio (SNR) of a sensor node i is

$$SNR_i = \frac{\left\|h_{ij}\right\|^2 P_j}{\sigma^2} \tag{4.38}$$

The advantage of this scheme comparing to the frequency reuse factor 1 is the elimination of CCI. CCI degrades the received SNR and therefore increases the probability of sensor inactivity. By using different frequencies for each actuator, the number of isolated sensors is decreased for a given level of transmit power.

4.11.3 Actuator Cooperation (Joint Beamforming)

In this scenario, the actuators are assumed to be interconnected via high speed backhaul links (wireline or wireless). After an initial handshake between a sensor and its associated actuator (min. hop fashion, more details on this assignment are provided in Section 4.6), each actuator transmits a training sequence. Then each sensor estimates the channel between itself and all the actuators, and it transmits this set of channel coefficients to its associated actuator in a multi-hop fashion. Therefore, the Transmitter Channel State Information (CSIT) is obtained. Furthermore, each actuator determines the schedules for its associated sensors. Actuators exchange their local CSIT and their scheduling information via the backhaul links, and jointly perform Maximal Ratio Combining (MRC) beamforming in order to transmit the scheduling information to each sensor. Hence, actuators form a distributed antenna array.

The transmission of the scheduling information is done in a Round-Robin fashion and at the same frequency. Each sensor has a channel vector $\mathbf{h}_i = [h_{i1}, h_{i2}, ..., h_{iM}]$. In order for the per-actuator power constraint to be satisfied, each actuator j transmits to sensor i

$$A_{ij} = \frac{h_{ij}^*}{\|h_{ij}\|}\sqrt{P_j}s_i \tag{4.39}$$

The received signal of the sensor node i is then

$$\begin{aligned} y_i &= \textstyle\sum_{j=1}^{M} h_{ij}A_{ij} + n \Rightarrow \\ y_i &= \textstyle\sum_{j=1}^{M} \|h_{ij}\| \sqrt{P_j}s_i + n \end{aligned} \tag{4.40}$$

where $i = 1, 2, ..., N$ and s_i is the schedule assigned to sensor node i. It is assumed that $E\|s_i\|^2 = 1$. Thus the SNR of the sensor node i in the case of equal power transmission is

$$SNR_i = \frac{P\left(\sum_{j=1}^{M}\|h_{ij}\|\right)^2}{\sigma^2} \tag{4.41}$$

Joint beamforming enhances the received SNR due to the array gain and the exploitation of macro-diversity which is inherent in a SANET. Therefore, this scheme provides a robust way of minimizing sensor inactivity. This is achieved at the cost of CSIT at the actuators. Furthermore, multiple time slots are needed in order to deliver the schedule to all sensor nodes, since actuators transmit to one sensor node at a time.

4.12 Simulation Results

In this section, we present our ns-2 [53] simulation results demonstrating the performance of our actuator-selection, optimal flow routing, and TDMA MAC solution. As our analysis is different from the related literature presented in this work, we only compare the results given by the upper bound in (4.23) (optimal flow solution for the relaxed problem which is independent of MAC) and the simulations performed in ns-2 on top of a TDMA like MAC. In our simulations, we consider different network sizes (varying the number of sensors and actuators) with randomly deployed topologies. Also, we evaluate the lifetime only at level one as the optimal lifetime solution at other levels can be interpreted in a similar fashion. Further, in the existing network simulators e.g. [53, 52], there are no available means of simulating a heterogeneous network consisting of sensors and actuators (as actuator have different transmission and processing capabilities). Therefore, we post-process our ns-2 based *tcl-scripts* in order to simulate a heterogeneous sensor-actuator network. Some simulation parameters are listed in Table 4.3. The simulations are run several times for each network setting and the results presented are averaged over these runs.

For each network setting, we calculate the upper bound on lifetime provided by (4.23) through MILP relax. We denote this lifetime as 'Lifetime from Analytical Bound'. We denote the lifetime obtained with actual ns-2 simulation as 'Lifetime from simulations'. The initial energy at sensor i is randomly generated following a uniform distribution with $e_i \in [300, 500]$ (kJ). The data generation rate at each sensor i, g_i, is also uniformly distributed within $[5, 10]$ (kb/s). The sensor-actuator routing model under consideration is the same as in Section 2.8. At simulation start up, the nodes learn the network topology and built routes toward the destination actuators (based on the outcome of a cost-function). In this simulation-analysis, actuators are also sensor nodes which have 0 sampling rate[4]. This learning process, which depends on the network topology, can take upto $50 - 70$ seconds.

We simulated the ADP, LEAD-RP and LEAD-MAC in ns-2 [53]. For sensor-sensor coordination, the sensors only require one-hop neighbor identity through which it can reach the actuator with lower cost as compared to its own. For sensor-actuator coordination, we simulated topologies of various sizes (50-400 sensors). The considered packet size is 50 bytes and the transmission rate is 50kpbs. The average depth of the resulting routing trees is 4.4, 5.2, and 7 for 20, 30, and 60 sensors per cluster, respectively; correspondingly the average number of neighbors is 4.6, 5.0, and 5.5. The results obtained from (4.23) and simulations using ns-2 are presented in Figure 4.9. It can be easily seen that our approach (optimal routing through LEAD-RP, actuator search through ADP, and a TDMA MAC through LEAD-MAC) can provide a network lifetime very close to the optimal solution. The slight difference in the lifetime obtained from the simulations is due to the energy expenditure during initial network learning and route discovery toward actuators. The simulated lifetime lies *exceptionally* close to the analytical bound (for relaxed flow-problem) due to the following reasons: 1) we have built an aggregation tree toward each actuator in the network and calculate the optimal flow routing solution. 2) The scheduling information is sent to the sensors by their mapped-actuator nodes which corresponds to the optimal flow solution. 3) The problem of *synchronization* is easily solved as the *transmission schedule* is calculated by the actuator in each cluster. 4) There is no extra energy expenditure as a result of collisions and successive retransmissions. 5) The nodes are sent to sleep mode, when not transmitting, and also no information is expected to arrive from a sensor's downlink tree.

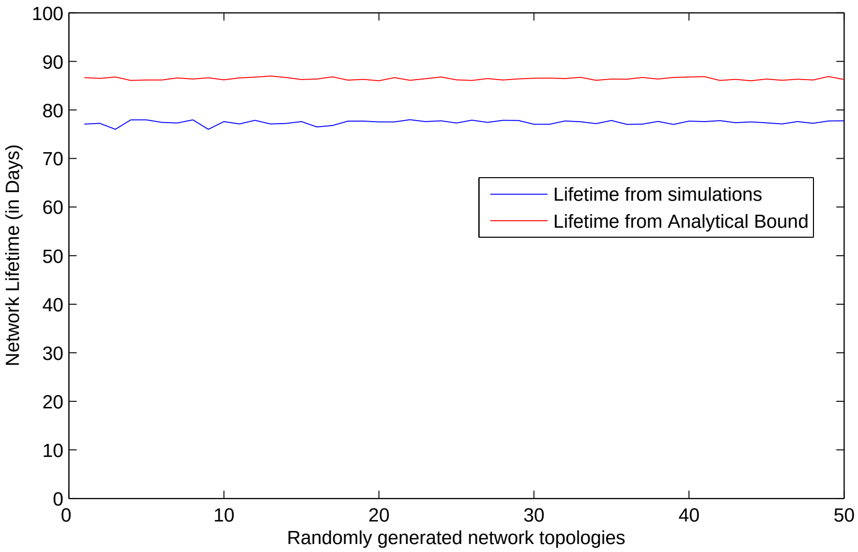

Figure 4.9: Network lifetime under analytical and simulation results

Directed Diffusion [1] and anycast [65] is chosen as the routing protocol for comparison.

[4] At run-time, these nodes are modified to actuators (i.e., different communication capabilities compared to sensors) so that an aggregation tree could be built toward these actuators for each cluster.

Figure 4.10 shows the end-to-end latency as a function of network size. The delay increases with the increase in the network size, but the increase is significantly less for ADP. This gradual increase is the result of smaller mean-path length for ADP as the cost-function is set to min-hop routing and forwarding queues at the sensors are not saturated at the given load. Figure 4.11 show the mean energy consumption as a function of time. ADP energy savings are more significant due to the existence of multiple defined routing paths toward optimal actuators, where depending on the remaining energy of the forwarding sensors, a source sensor can choose between several available paths to efficiently route its data. In Figure 4.12, the mean path length is shown as a function of network size. Again the mean path length (which is related to the end-to-end latency) increases with the network size. However, the increase is more gradual with the ADP as compared to anycast and directed diffusion. Using ADP, sensors always transmit their data to the nearest actuator (because we set the cost-function to min-hop routing for actuator discovery during initial deployment).

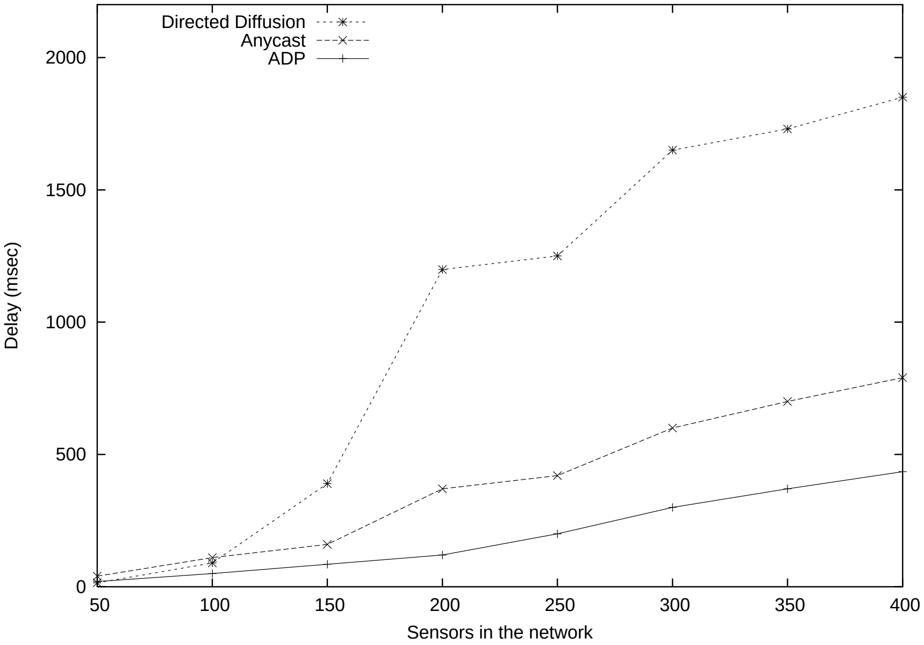

Figure 4.10: Mean end-to-end transmission delays

A comparison with the analytical model of PEDAMACS [103] is presented for delay-energy consumption analysis at MAC layer. We compare LEAD-MAC only with PEDAMACS, because this work has already been shown to perform better compared to other listed MAC proposals for WSNs. The maximum delay observed by a network can be seen in Figure 4.13. The end-to-end delay is less due to the depth-first scheduling policy of end-to-end routes in the hybrid-schedule. Finally, we present a comparison for the energy consumption in Figure 4.14, where sensors consumes less energy due to an adaptive duty cycle and longer sleep periods.

The performance of the aforementioned transmission schemes is evaluated in terms of the number of isolated sensors that results from each transmission scheme. A number of sensors is deployed uniformly in a hexagon with a radius of 1 km. Three actuators are assumed at the three vertices of the hexagon separated by an angle of 120°. Actuator antennas are consider to have a gain of 12 dB, whereas, sensor node antennas have a gain of 0 dB. Through Monte-

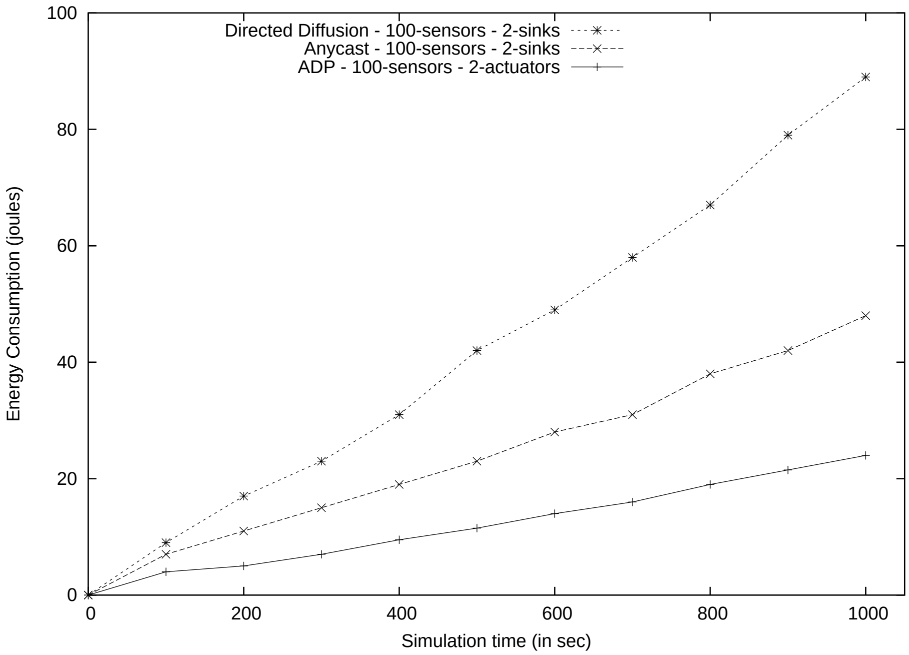

Figure 4.11: Mean energy consumption as a function of time for a network of 100 sensors

Carlo simulation the average number of isolated sensors is calculated for each transmission scheme as a function of the actuator transmit power. Averaging is performed over sensor node positions and channel realizations. A sensor is assumed to be isolated if its received SNIR or SNR is below the threshold of 1 Watt. In Figure 4.15 it is plotted the average number of isolated sensors versus the actuator transmit power for 1200 deployed sensors. It can be seen that for the power of -12 dBw isolated sensor zones are almost completely eliminated in the case of MRC beamforming. In the case of Reuse Factor 3 (RF3), isolated zones are eliminated when the transmit power is approximately 0 dBw and in the case of Reuse Factor 1 (RF1) the average number of isolated sensors saturates approximately at 0 dBw.

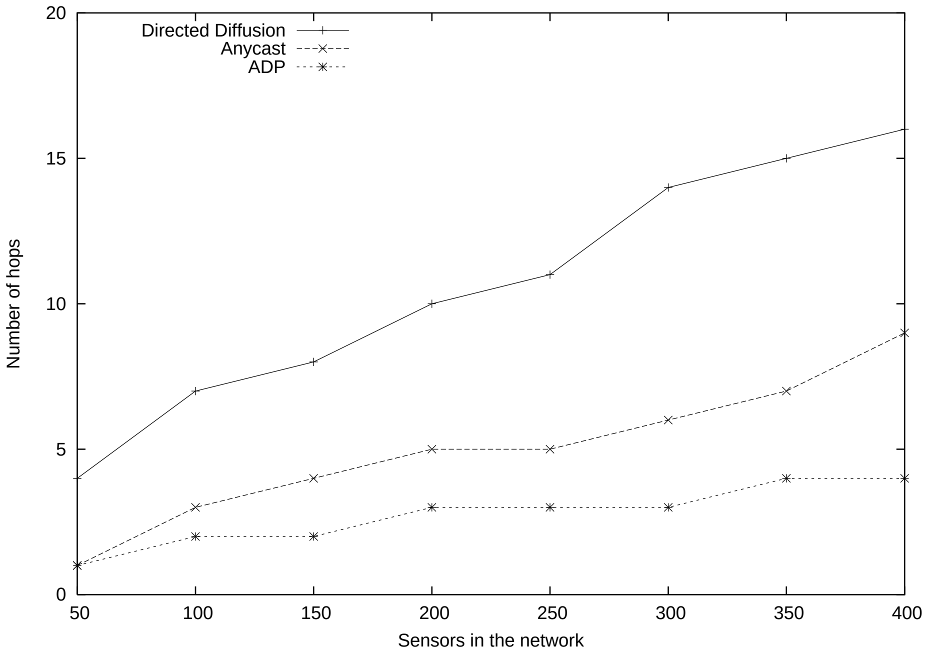

Figure 4.12: Mean number of transmissions per end-to-end path (mean path length)

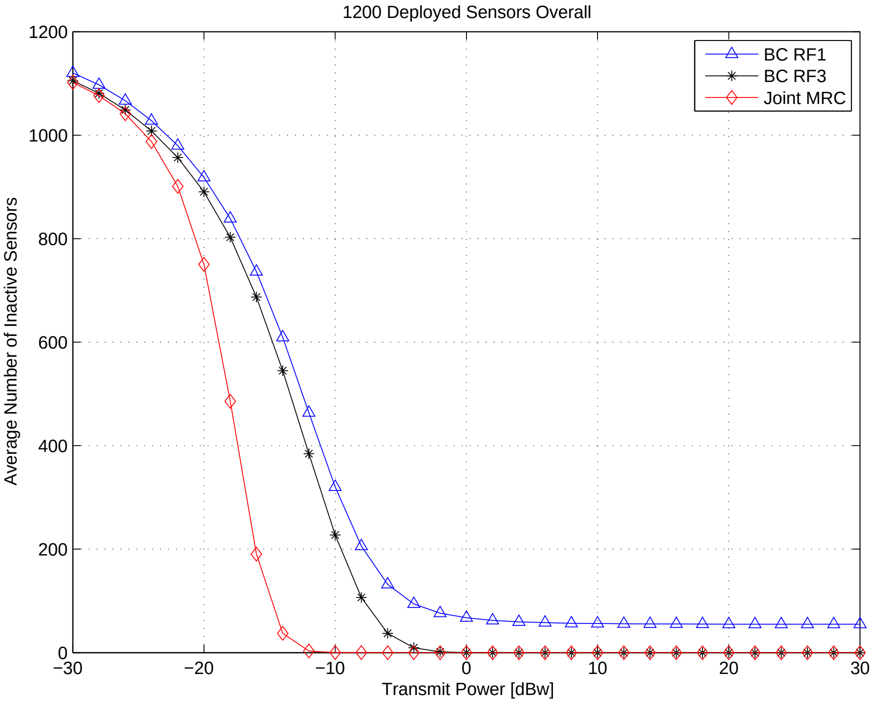

Figure 4.15: Average Number of isolated Sensors vs. Transmit Power.

In Figure 4.16 it is plotted the average number of isolated sensors against the total number of deployed sensors for a different number of deployed sensor nodes, when actuators transmit

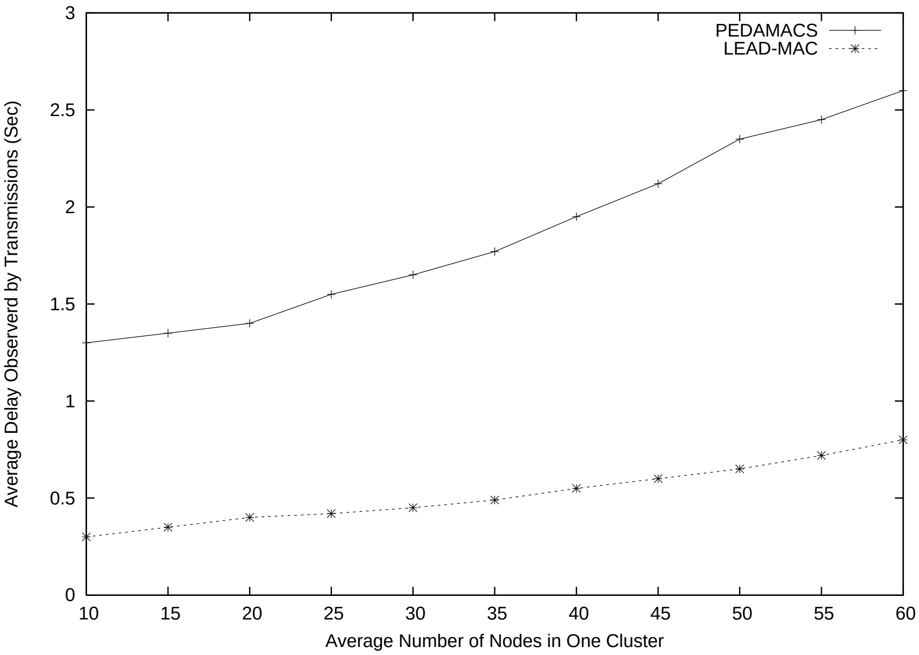

Figure 4.13: Average delay in a cluster → increasing # of nodes

power is -12 dBw. It can be clearly seen that the joint MRC beamforming scheme outperforms the simple Reuse 3 broadcasting, as the average number of isolated sensors is almost 0 for that power level.

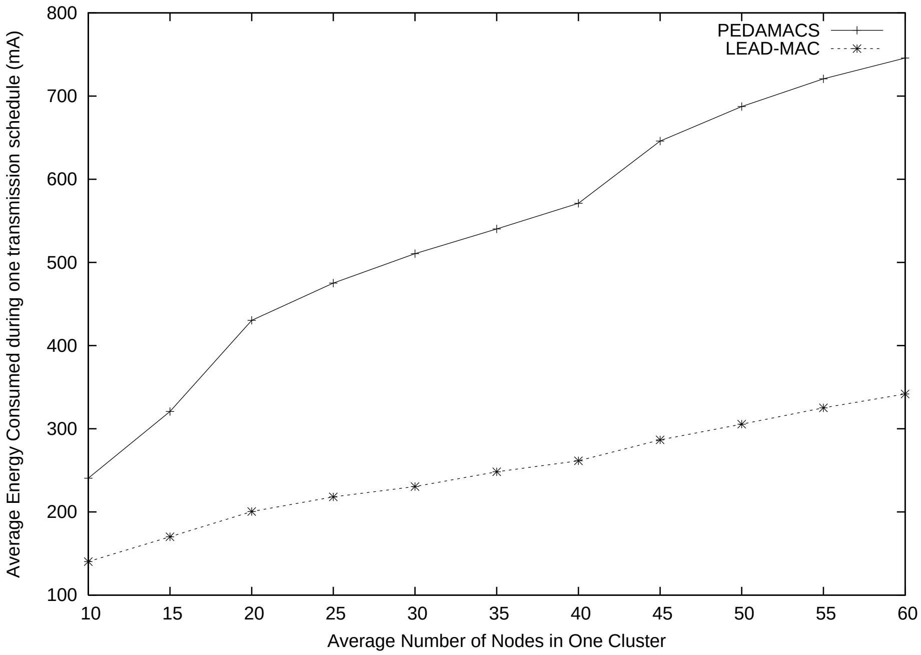

Figure 4.14: Average energy consumption in a cluster → increasing # of nodes

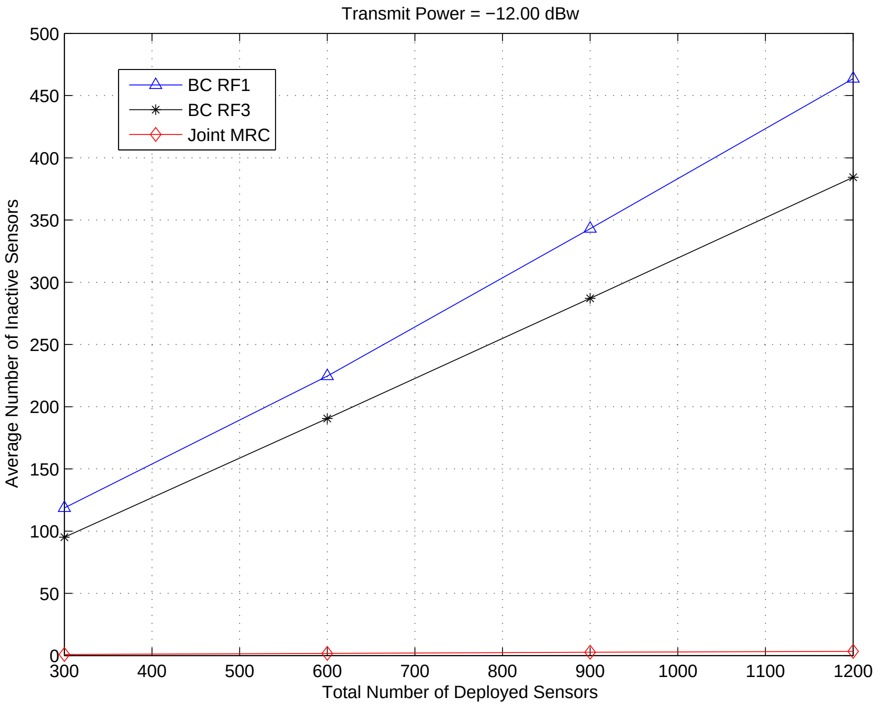

Figure 4.16: Average Number of isolated Sensors Vs. Total Number of Deployed Sensors.

In Figures 4.17, 4.18 and 4.19, the probability of inactivity can be seen in the different areas of the hexagon for the three different transmission schemes considered, when actuators transmit power is -12 dBw. In the cases of RF1 and RF3 schedule broadcasting, the center of the topology experiences a significant probability of inactivity. In a real system implementation, this would result to an important loss of information. On the contrary, Joint beamforming almost eliminates isolated areas in the sensing field at this power level. This turns out to be a very effective actuator transmission scheme that greatly reduces the amount of transmit power needed to ensure very low sensor inactivity. This is because of the beamforming SNR gains and the macro-diversity gains that are provided by the spatially distributed transmitting actuators.

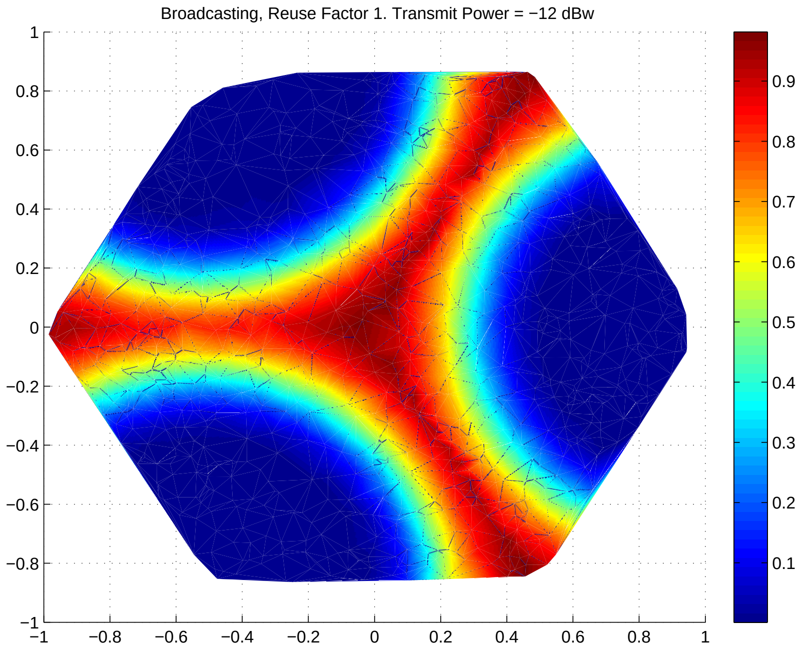

Figure 4.17: Probability of Sensor Inactivity in the areas of the sensing field for the case of Reuse Factor 1 Schedule Broadcast Transmission.

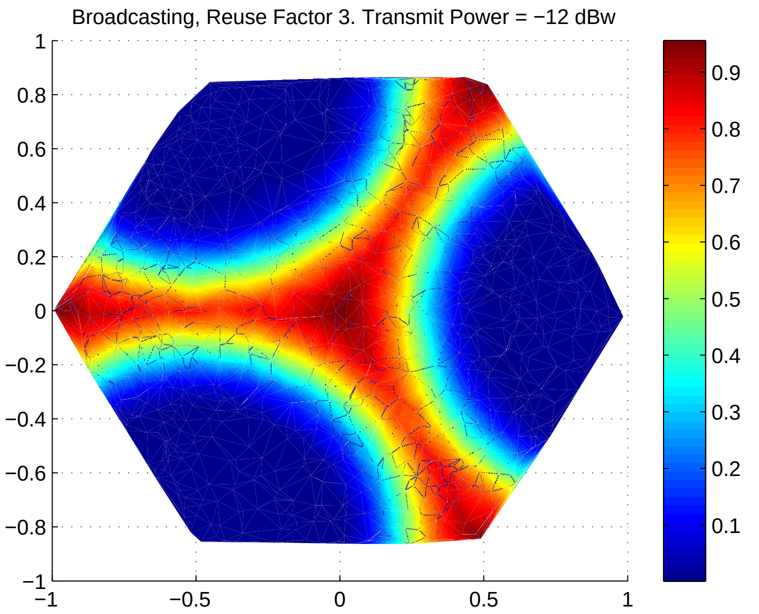

Figure 4.18: Probability of Sensor Inactivity in the areas of the sensing field for the case of Reuse Factor 3 Schedule Broadcast Transmission.

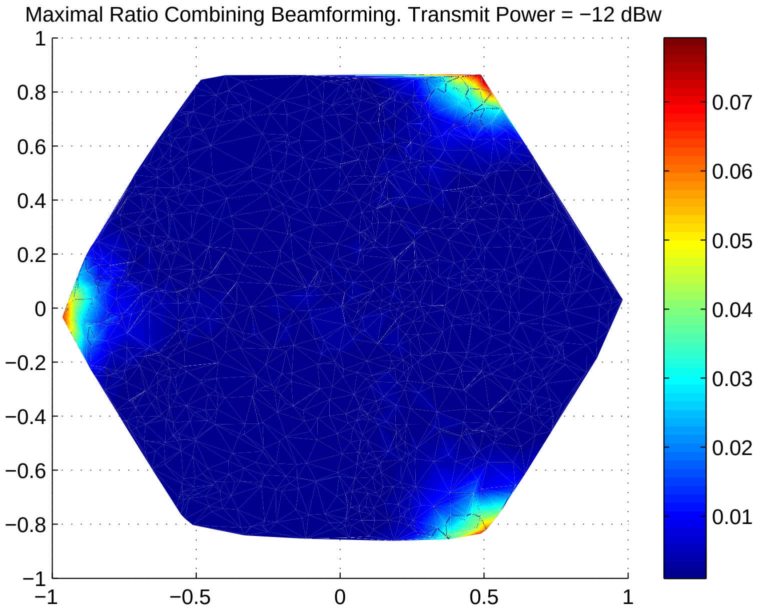

Figure 4.19: Probability of Sensor Inactivity in the areas of the sensing field for the case of joint Maximal Ratio Combining Beamforming.

4.13 Conclusions and Future work

This chapter considers a large scale SANET with multiple actuators as sinks for data generated by the sensors. Since many applications require to have each source node send all its locally generated data to only one actuator for processing, it is necessary to optimally map each sensor to its actuator. Also considering the fact that the end-to-end delays in wireless sensor-actuator networks is a hard constraint, we jointly optimize the actuator selection and optimal flow routing subject to energy and delay constraints with the global aim of maximizing the network lifetime. We proposed and evaluated (using ns-2) our actuator-selection (LEAD-ADP) and routing scheme (LEAD-RP) on top of a TDMA based MAC (LEAD-MAC) protocol. We then use the Lagrangian dual decomposition method to devise a distributed primal-dual algorithm to maximize network-lifetime in the network. The deterministic distributed primal-dual algorithm requires no feedback control and therefore converges *almost* surely to the optimal solution. The results show that the required optimal value of lifetime is achieved for every node in the network by the distributed primal-dual algorithm. We also provide a comparison to the analytical bound. Simulation results show that this approach has near-optimal performance and is practically implementable as compared to earlier *analytical* studies based only on *numerical* evaluations.

This chapter also addresses the problem of isolated regions in the sensing field by letting actuators exchange their CSIT and jointly perform beamforming in order to deliver scheduling information to sensor nodes. The gains of cooperation were shown by simulating the average number of isolated sensors for the case of single actuator transmission and cooperative transmission.

In the near-future, we will consider a real-life SANET application and simulate its *behavior* with the LEAD self-organizing framework to observe its *performance.* We will take into consideration a *dynamic* actuator-assignment scenario to timely transport data in a *mobile* wireless sensor-actuator network.

Table 4.1: Notations

Symbols	Definitions
N	The total number of Sensors in the network
M	The total number of Actuators in the network
B	The total number of BaseStations in the network
e_i	Initial energy of a sensor node i
E_i	Initial energy of an actuator node i
g_i	The locally generated data rate at sensor i
P_{rx}	Power consumption coefficient for receiving data
$c_{i,j}$	Power consumption coefficient for transmitting data from sensor i to sensor j
α, β	Two constants terms in power consumption for transmitting data
$d_{i,j}$	The geographic distance between two nodes i and j
$f_{s_i,s_j}^{s_k,A_l} \left(or\, f_{s_i,A_l}^{s_k,A_l} \right)$	The flow rate from sensor i to sensor j (or actuator l) with source and destination being sensor k and actuator l
$F_{s,s}$ $(or\, F_{s,A})$	The set of flows from one sensor to another (or Actuator node)
F_{s,s_i}	The set of flows coming into sensor i
$F_{s_i,s}$ $(F_{s_i,A})$	The set of flows going out of sensor i to other sensors (or Actuator node)
λ^{s_i,A_l}	If the generated data at sensor i will be transmitted to actuator l, then $\lambda^{s_i,A_l} = 1$; otherwise $\lambda^{s_i,A_l} = 0$
$V_{s_i,s_j}^{s_k,A_l} \left(or\, V_{s_i,A_l}^{s_k,A_l} \right)$	The data volume (in bits) transferred from sensor i to sensor j (or Actuator l) with source and destination being sensor k and Actuator l
$\upsilon_{s,s}$ $(\upsilon_{s,A})$	The set of volume from a sensor to another sensor (or an actuator)
υ_{s,s_i}	The set of incoming volume into sensor i
$\upsilon_{s_i,s}$ $(\upsilon_{s_i,A})$	The set of outgoing volume from a sensor i to another sensor (or an actuator)
μ^{s_i,A_l}	$= \lambda^{s_i,A_l} T$ in MILP-relax
$f_{A_i,A_j}^{A_k,B_l} \left(or\, f_{A_i,B_l}^{A_k,B_l} \right)$	The flow rate from actuator i to actuator j (or BaseStation l) with source and destination being actuator k and BaseStation l
$F_{A,A}$ $(or\, F_{A,B})$	The set of flows from one actuator to another (or BaseStation)
F_{A,A_i}	The set of flows coming into an actuator i
$F_{A_i,A}$ $(F_{A_i,B})$	The set of flows going out of actuator i to other actuators (or BaseStation)
G_i	The locally gathered data at actuator i, $G_i = \sum_i g_i,\ (1 \le i \le N)$
λ^{A_i,B_l}	If the data gathered at actuator i will be transmitted to BaseStation l, then $\lambda^{A_i,B_l} = 1$; otherwise $\lambda^{A_i,B_l} = 0$
$V_{B_i,B_j}^{A_k,B_l} \left(or\, V_{A_i,B_l}^{A_k,B_l} \right)$	The data volume (in bits) transferred from actuator i to actuator j (or BaseStation l) with source and destination being actuator k and BaseStation l
$\upsilon_{A,A}$ $(\upsilon_{A,B})$	The set of volumes from a actuator to another actuator (or a BaseStation)
υ_{A,A_i}	The set of incoming volume into actuator i
$\upsilon_{A_i,A}$ $(\upsilon_{A_i,B})$	The set of outgoing volumes from a actuator i to another actuator (or a BaseStation)
μ^{A_i,B_l}	$= \lambda^{A_i,B_l} T$ in MILP-relax

Table 4.2: Useful states for the sensor node with associated power consumption and delay (time to reach S_4 from any given state)

Operating State	Strong ARM	Memory	ADC	Radio	Power Consumption	Delay (ms)	Notation Used
S_0	Sleep	sleep	Off	Off	50 (μW)	50	E_{node}^{s0}
S_1	Sleep	Sleep	On	Off	5 (mW)	20	E_{node}^{s1}
S_2	Sleep	Sleep	On	Rx	10 (mW)	15	E_{node}^{s2}
S_3	Idle	Sleep	On	Rx	100 (mW)	5	E_{node}^{s3}
S_4	Active	Active	On	Tx, Rx	400 (mW)	NA	E_{node}^{s4}

Table 4.3: The simulation area is such that there are atleast two sensors in each others transmission range

Sensors	Area (m^2)	Actuators
100	500*500	2
150	600*600	3
⋮	⋮	⋮
400	970*970	8

Chapter 5

Cross-Layer Routing in UASNs

UASNs consist of sensors that are deployed to perform collaborative monitoring of tasks over a given volume of water. These networks will find applications in oceanographic data collection, pollution monitoring, offshore exploration, disaster prevention, assisted navigation, tactical surveillance, and mine reconnaissance. The quality of the underwater acoustic link is highly unpredictable, since it mainly depends on *fading* and *multipath*, which are not easily modeled phenomena. This in return severely *degrades* the performance at *higher* layers such as *extremely* long and *variable* propagation delays. In addition, this variation is generally larger in *horizontal* links than in *vertical* ones.

In this chapter, we first analyze a modulation scheme and associated receiver algorithms. This receiver design take advantage of the *TR* and properties of spread spectrum sequences known as Gold sequences. Furthermore, they are much less complex than receivers using *adaptive* equalizers. This technique improves the signal-to-noise ratio (SNR) at the receiver and reduces the bit error rate (BER). We then applied *PC* to the case of network communication. We show that this approach can give almost *zero* BER for a two-hop communication mode compared to the traditional *direct* communication. This link layer information is used at the network layer to optimize routing decisions. We show these improvements by means of analytical analysis and simulations.

5.1 Introduction

Acoustic signaling for wireless digital communications in the sea environment can be a very *attractive* alternative to both *radio* telemetry and *cabled* systems. However, time-varying *multipath* and often harsh ambient *noise* conditions characterize the underwater acoustic channel, often making acoustic communications challenging. The sensors must be organized in an autonomous network that self-configures according to the varying characteristics of the ocean environment. Major challenges in the design of UASNs are:

- The channel is severely *impaired*, mainly due to *multipath*.
- Temporary loss of *connectivity* mainly due to *shadowing*.
- The *propagation delay* is five orders of magnitude higher than in radio frequency *terrestrial* channels and is usually variable [4].
- Extremely low available *bandwidth*.

- Limited *battery* energy at disposal.

In this chapter, we present our analysis of a modulation scheme and associated receiver algorithms. We also present the *quantification* of SNR and BER gains using *PC* in a single transmitter-receiver setting. We then applied *PC* technique to a multi-hop communication systesm. This link layer information is used at the network layer to optimize *routing* decisions. This *cross-layering* improves the network lifetime of battery operated UASNs by reducing the number of *retransmission* attempts.

The organization of this chapter is as follows. Section 5.2 details some of the interesting related work. In Section 5.3, we discuss the basic *building* blocks that *contributed* to the proposed solution. We present the receiver algorithms that take advantage of *TR* and low *cross-correlation* of Gold sequences for single-hop point-to-point communication and its performance analysis in Section 5.4. We apply the idea of *PC* on a linear network to improve some performance metrics in Section 5.5. We also present a distributed *routing* algorithm and its performance *analysis* for a larger network size. In Section 5.6, we conclude the chapter and outline the future directions.

5.2 Related Work

Acoustic underwater communication is a challenging problem [4, 104] for *reliable* high speed communication in the ocean. Underwater communication must deal with the inter-symbol interference (ISI) caused by the time-varying and dispersive multipath shallow water environments. The principle of TR can be used to overcome these challenges.

TR has been investigated and applied widely as time reversal mirrors (TRMs) [105, 106] to solve such problems. Classically, TR is based on *spatial reciprocity* and *time symmetry* of the wave equation. TR is a process where a source at one location transmits sound. This is received at another location, *time reversed*, and retransmitted. The retransmitted sound is then *focused* back at the original source location. Time-reversing acoustic technologies were proven to be effective for acoustic focusing under unknown acoustic environmental conditions. The experiments [105] conducted by the Marine Physical Laboratory and the NATO Undersea Research Center showed that a TRM can produce significant focusing at long distances in a 125 m deep-water channel.

P. Roux *et al.* [111] experimentally demonstrated a way to simplify the study of TR in a fluctuating medium without invoking *reciprocity* in the propagation medium. This non-reciprocity based time-reversal (NR-TR) is built from the forward propagation (one-way propagation) between the TRM and the desired focal point. As a consequence, TR provides a *good* focus even when spatial reciprocity does not *hold* in the medium. For guided wave propagation, a very high degree of *orthogonality* between the signal [107] is *necessary* to allow an *accurate* measure of the whole multipath structure of the transfer function. Instead of TRMs, a point-to-point (without arrays of sources or receivers) acoustic communication *framework* using passive phase conjugation (PPC) is investigated in [112] which use families of sequences called Gold codes to compensate the ISI problem. Gold codes are based on m-sequences that are often used in the underwater acoustic community because of their *auto-correlation* properties [113, 114, 115, 116].

D. Pompili *et al.* [120] proposed algorithm for delay-insensitive and delay-sensitive applications in UASNs. They consider a three-dimensional architecture and evaluate their routing algorithms on top of IEEE 802.11 MAC. They provide a way to calculate an optimal packet size given end-to-end route information. The choice of IEEE 802.11 MAC severely degrades the performance metrics at network layer due to the protocol overhead and is not suitable for UASNs [121, 122].

5.3 The Design Criteria

In this section, we detail the basic building blocks that contributed to the solution provided in this chapter [C-11].

5.3.1 Gold Sequences

One important class of periodic sequences which provides larger sets of sequences with good periodic cross-correlation is the class of Gold sequences. A set of Gold sequences can be constructed from any preferred pair of m-sequences which have a special three-valued cross-correlation function $\left\{-1+2^{[(L+2)/2]}, -1, -1-2^{[(L+2)/2]}\right\}$, where L is the length of a shift register. It can be obtained by taking a m-sequence u and deriving a second m-sequence v by using *decimation*. The decimation factor is an integer and has to fulfill the equation $q = 2^J+1$, where $L/gcd\,(L,J))^1$ is odd.

Now that we have generated a preferred pair of m-sequences of period $N = 2^L - 1$, we can construct a set of Gold sequences. From the two m-sequences, all product sequences are achieved by the exclusive chip by chip *modulo-2 adding* with synchronous clocking:

$$G(u,v) \triangleq \{u, v, u \oplus v(iT_c)\} \qquad 0 \le i < N \tag{5.1}$$

Any 2-register Gold code generator of length L can generate a set of $M = 2^L + 1$ sequences of length $N = 2^L - 1$ Gold sequences which have a three valued cross-correlation

$$\left\{\frac{2^{(M+1)/2}-1}{N}, -\frac{1}{N}, -\frac{2^{(M+1)/2}-1}{N}\right\}$$

for L odd and

$$\left\{\frac{2^{(M+2)/2}-1}{N}, -\frac{1}{N}, -\frac{2^{(M+2)/2}-1}{N}\right\}$$

for $L \equiv 2\text{mod}4$. Gold sequences are *bipolar* sequences with values -1 and 1. Their special property is that any two different Gold sequences from the same family have *very low* cross-correlation values.

5.3.2 The Time reversal (phase conjugation) approach

The TR approach avoids the explicit *recovery* of the channel and its subsequent equalization via signal processing, and its associated algorithm complexity. To review how this focusing is achieved in point-to-point communication, we present here the following two configurations: (1) active phase conjugation (APC) and (2) passive phase conjugation (PPC) [C-11]. The channel impulse response (CIR) function is $h\,(t)$ and its Fourier Transform (FT) is $H\,(\omega)$. Recall that, in the frequency domain, the convolution of $h\,(t)$ and transmit signal $s\,(t)$ is $H\,(\omega)\,S\,(\omega)$. Similarly, the correlation of two signals $s_1\,(t)$ and $s_2\,(t)$ is $S_1^*\,(\omega)\,S_2\,(\omega)$.

[1] $gcd\,(\cdot)$ is a binary algorithm which computes the greatest common divisor of two non-negative integers.

In the two-way transmission (APC) configuration, the transmitter send a waveform $S(\omega)$ which travels through the channel and is recorded on the receiver as $H(\omega)\,S(\omega)$. The received waveform is phase conjugated, $H^*(\omega)\,S^*(\omega)$, retransmitted through the same channel and is again *convolved* with $H(\omega)$, producing $|H|^2\,S^*(\omega)$ at the original point. The term $|H|^2$ is the TR or PC *focusing* operator.

In the one-way transmission (PPC) configuration as shown in Figure 5.1, the transmitter send two signals $S_1(\omega)$ and $S_2(\omega)$ one after another, respectively, through the *same* channel. They are received at the receiver as $H(\omega)\,S_1(\omega)$ and $H(\omega)\;S_2(\omega)$. The cross-correlation of $H(\omega)\,S_1(\omega)$ and $H(\omega)\,S_2(\omega)$ is the product $|H|^2\,S_1^*(\omega)\,S_2(\omega)$, in which we again find the time reversal operator (TRO). The auto-correlation of the CIR $|H|^2$ tends to *reconcentrate* or focus the multipath arrivals at *zero* time lag.

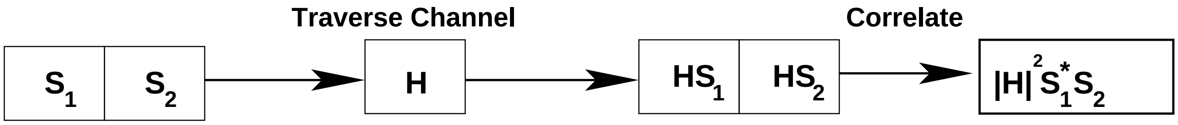

Figure 5.1: Passive Phase Conjugation (PPC)

5.3.3 Underwater Propagation Model

In the underwater acoustic communication, the transmission loss (TL) describes how the acoustic *intensity* decreases as an acoustic *pressure* wave propagates outwards from a sound source. The transmission loss $\mathrm{TL}(d,f)$ [dB] that a narrow-band acoustic signal centered at frequency f [KHz] experiences along a distance d [m] can described by the Urick propagation model [123],

$$\mathrm{TL}(d,f) = x\log d + \alpha(f)\cdot d + A. \tag{5.2}$$

where, the first terms account for geometric spreading, which refers to the spreading of sound energy as a result of expansion of the wave-fronts. It increases with the propagation distance and is independent of frequency. There are two common kinds of geometric spreading: spherical (omni-directional point source, spreading coefficient $x = 20$), which characterizes deep water communications, and cylindrical (horizontal radiation only, spreading coefficient $x = 10$), which characterizes shallow water communication. In this chapter, we will consider the shallow water case. Also, A in (5.2) represents *transmission anomaly* and is measured in [dB].

5.3.3.1 Cylindrical-Spreading Transmission Loss

The cylindrical spreading characterizes shallow water communication and (5.2) provides the following approximation of the TL for cylindrical spread signals as:

$$\mathrm{TL}(d,f) = 10\log d + \alpha(f)\cdot d\times 10^{-3} \tag{5.3}$$

where d is the distance between the source and the receiver (hydrophone), f is the central frequency of the source, α is the medium absorption [dB/m] which can be expressed as following between 4°C and 20°C

$$\alpha(f) = \begin{cases} 0.0601 \times f^{0.8552} & 1 \le f \le 6 \\ 9.7888 \times f^{1.7885} \times 10^{-3} & 7 \le f \le 20 \\ 0.3026 \times f - 3.7933 & 20 \le f \le 35 \\ 0.504 \times f - 11.2 & 35 \le f \le 50 \end{cases} \tag{5.4}$$

Note that A is used usually to apply a spherical spreading model. This is because the transmission anomaly is meant to account for the effects of *refraction* and is therefore not accounted in (5.3). More details can be found in [124] and [125].

5.3.3.2 The passive-sonar equation

We first take a look the sonar parameters and reference locations as shown in Table 5.1. They are added together in forming the sonar equation. Note that, the units of all sonar parameters are in [dB].

Table 5.1: Parameters

Parameter	Symbol	Reference Location
Source Level	SL	1 yd. from source on its acoustic axis
Transmission Loss	TL	1 yd. from source and at target or receiver
Noise Level	NL	at hydrophone location
Receiving Directivity Index	DI	at hydrophone terminals
Detection Threshold	DT	at hydrophone terminals

In the passive case, the target strength becomes irrelevant as only one-way transmission is involved. The passive-sonar equation becomes

$$\mathrm{SL} - \mathrm{TL} = \mathrm{NL} - \mathrm{DI} + \mathrm{DT} \tag{5.5}$$

where SL is Source Level, TL is Transmission Loss, NL is Noise Level, DI is (receiving) Directivity Index, and DT is Detection Threshold.

When the target is just being detected, the SNR equals to DT. An average value for the ambient noise level NL is 70 dB as a representative shallow water case. The directivity index DI is calculated based on the aperture of the antenna.

5.4 Case I: Single-Hop Communication Framework

A receiver algorithm that we will use in this chapter take advantage of time reversal (phase conjugation) and low cross-correlation of the Gold sequences. In PPC configuration, the message must be encoded in the correlation of the two consecutively transmitted waveforms S_1 and S_2. Encoding information in the correlation of two waveforms is not a *typical* signaling scheme and may provide some advantages. Therefore, we choose the PPC configuration for our UASN.

Our modulation cycles through a series of Gold sequences from the same family. With single receiver in acoustic communication using PPC, an ISI problem should be considered. To compensate for this, we rely upon families of Gold sequences to minimize the correlation between the sequences in each family.

5.4.1 Waveform design

Here G_i indicates the ith Gold sequences from a family of 2^L+1 sequences. When transmitted, each G_i is a bipolar sequence modulated by a carrier (BPSK modulation). The transmitted signal of a certain chip k $\left(1 \leq k \leq N = 2^L - 1\right)$ is

$$s_c(t) = g(k)\sqrt{E_c}\sqrt{\frac{2}{T_c}}\cos\left(2\pi f\left(t - kT_c\right) + \phi\right) \tag{5.6}$$

where $g\left(k\right)$'s are real numbers of kth chip, ±1 representations of a Gold sequences, E_c is the energy per chip, f is the carrier frequency and ϕ is the initial phase, and T_c is the chip duration.

Then the expressing of i^{th} Gold sequence in the duration NT_c is

$$G_i\left(t\right) = \sum_{k=0}^{N-1} g\left(k\right)\sqrt{E_i}\sqrt{\frac{2}{T_c}}\cos\left(2\pi f\left(t - kT_c\right) + \phi\right) \tag{5.7}$$

where E_i is the energy of ith Gold sequence.

When $G_i\left(t\right)$ passes through underwater channel, the multipath signal will arrive at receiver with different time delay. In communication system, the received signal is modeled as

$$\begin{aligned} r\left(t\right) &= \sum_p h_p G_i\left(t - \tau_p\right) + N_o\left(t\right) \\ &= \sum_p h_p G_i\left(t\right) \otimes \delta\left(t - \tau_p\right) + N_o(t) \end{aligned} \tag{5.8}$$

where τ_p is the delay of pth path, h_p is considered as pth CIR and N_o is the ambient noise of channel. Because PPC is an approach in the frequency domain, Fourier transform of (5.8) is

$$R\left(\omega\right) = H\left(\omega\right) G_i\left(\omega\right) + N_o\left(\omega\right) \tag{5.9}$$

5.4.2 Pulse position modulation (PPC-PPM)

Pulse position modulation is implemented to take advantage of PPC, we set the first signal transmitted $S_1\left(\omega\right)$ as the reference Gold sequences $\{G_i, 1 \leq i \leq M\}$ with a constant time interval between them as shown in Figure 5.2. If this constant interval can be divided into K resolvable time slots, each pair of G_i's will carry $\log_2 K$ bits of information. The second $\{G_i'\}$ (black) is the repeat of the $\{G_i\}$ (grey) with a varying distance between the reference and second positions. The positions of the second $\{G_i'\}$ is purposely varied to convey the information bits being transmitted.

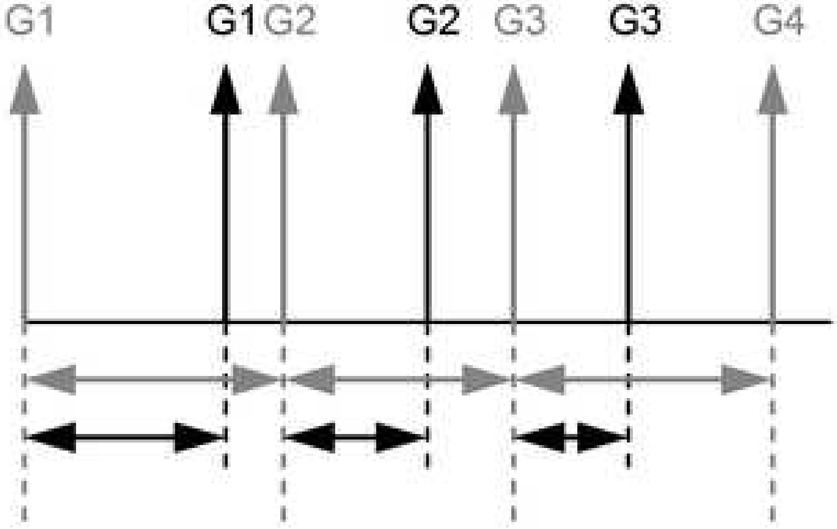

Figure 5.2: Waveform Design for PPC-PPM

At the receiver, the particular order of the Gold sequences being transmitted is known, so the appropriate matched filter tuned to G_i can be applied to each received symbol. The matched filter performs a pulse compression on each G_i and reproduces the multipath arrival structure associated with both instances of G_i. We can illustrate this scheme as following:

$$\{G_i(\omega)\} \longrightarrow H(\omega)\{G_i(\omega)\} \xrightarrow{\text{filter } G_i} H(\omega)|G_i|^2$$
$$\{G_i(\omega - \varphi_i)\} \longrightarrow H(\omega)\{G_i(\omega - \varphi_i)\} \xrightarrow{\text{filter } G_i} H(\omega)|G_i(\varphi_i)|^2$$

When symbols overlap due to multipath, the low cross-correlation property of the Gold sequences ensures that the different matched filters *do not* let through much of the interfering symbols. Then it is possible to decode the information from relative positions of the dominant arrivals only.

The correlation of the two matched filter outputs is

$$\left\langle H(\omega)|G_i|^2, H(\omega)|G_i(\varphi_i)|^2 \right\rangle = |H|^2|G_i|^2|G_i(\varphi_i)|^2 \tag{5.10}$$

and the producing waveform is proportional to $|H|^2$. The auto-correlation function of the multipath impulse response, Γ_h, is defined as following

$$\Gamma_h(0) = \int_0^{NT_c} |h(t)|^2 \, dt = \int_0^{\frac{1}{NT_c}} |H(\omega)|^2 \, d\omega \tag{5.11}$$

where the duration of codewords, $N = 2^L - 1$, is much greater than the multipath channel delay spread.

5.4.3 Calculation of SNR and BER

In this section, we calculate SNR and bit-error rate (BER) that quantifies the robustness of our approach in the presence of external noise.

5.4.3.1 Signal-to-Noise-Ratio (SNR)

The transmitted signals are set to be ±1 pulses controlled by the generated sequence of length N. The total transmitted signal energy per Gold sequence

$$E_t = \frac{1}{T_c}\int_0^{NT_c} s_c^*\left(t\right)s_c\left(t\right)dt \tag{5.12}$$

is N per path because $s_c\left(t\right)$ was normalized to unit power ($E_c = 1$). The total received signal energy E_r over all multi-paths is

$$E_r = \Gamma_h\left(0\right).E_t.m = \Gamma_h\left(0\right).N.m \tag{5.13}$$

where m is the number of Gold sequences used to construct PPM, $1 \leq m \leq M$.

The output of the correlator $y\left(k\right)$ at kth chip received can be written as

$$y\left(k\right) = \Gamma_h\left(0\right)\left|G_i\right|^2\left|G_i\left(\varphi_i\right)\right|^2 + n\left(k\right) \tag{5.14}$$

where $n\left(k\right)$ is the noise output and can be developed from (5.8). Substitute (5.7) into (5.8), we find that beside the desired component $\sqrt{E_c}$, there are still two undesired components which indicate the interferences from multipath and ambient noise by I_{mp} and I_{no}, respectively. Due to the independence of each chip in PN code, the first order statistics of I_{mp} and I_{no} will be zero and their variances can be expressed as

$$\begin{gathered} V_{mp}\left(k\right) = E_c\sum_p h_p^2\cos\left(\phi'\left(p\right)\right) \\ V_{no}\left(k\right) = \tfrac{N_0}{2} \end{gathered}$$

Then, we have the variance of $n\left(k\right)$ as

$$V_n\left(k\right) = V_{mp}\left(k\right) + V_{no}\left(k\right)$$

The received SNR is then expressed as

$$\frac{E_r}{I_o} = \frac{\Gamma_h\left(0\right).N.m}{N.V_n\left(k\right)} = \frac{\Gamma_h\left(0\right).m}{E_c\sum_p h_p^2\cos\left(\phi'\left(p\right)\right) + \frac{N_0}{2}} \tag{5.15}$$

The bit energy SNR is

$$\frac{E_b}{I_o} = \frac{1}{\log_2 K}\cdot\frac{E_r}{I_0} \tag{5.16}$$

where each code word carries $\log_2 K$ bits of information.

5.4.3.2 Bit-Error-Rate (BER)

Using the assumptions of equal *probable* messages, it is possible to write the conditional *error* function rate for the binary case as

$$P\left(a_1, a_2, ..., a_p\right) = \tfrac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\frac{1}{\sqrt{\sum_{i=1}^{p} a_i^2 R_i\left(1-\hat{\lambda}\right)}}\exp\left[-\tfrac{x^2}{2}\right]dx. \tag{5.17}$$

where a_i is the *instantaneous* voltage gain of the i^{th} channel, p is the *multichannel* order, $\hat{\lambda}$ is the real part of the normalized signal *cross-correlation* coefficient and $R_i = \frac{E_i}{N_o}$, where E_i is the energy of each transmitted signal by the i^{th} transmitter.

Then, the classic BER of BPSK in multipath fading channel can be calculated as

$$\text{BER} = f\left(\frac{E_b}{N}\right) \tag{5.18}$$

where $f(\cdot)$ is the bit-error function which is developed from (5.17). More details on (5.17) can be found in [126].

Therefore, the calculation of the BER in our model can be expressed as a function of the ratio between the *energy* of received bit and the *noise* at the receiver. While r [bps] is the considered bit rate, the energy of received bit can be calculated by $P_{i_{max}} \cdot 2N/(r \cdot TL)$, where $P_{i_{max}}$ is the maximum *transmitting* power for node i, and $2N$ is the gain from *pulse compression* (for detail, see (5.14)). We recall (5.5) in Section 5.3.3 for the application of our PPC-PPM construction, SL is the transmitted signal intensity at 1 m from the source according to the following expression:

$$SL = \frac{E_b}{N_o} + 2TL + 70 - DI = 10 \log \frac{I_t}{1\mu Pa} \tag{5.19}$$

where, I_t is in μPa. Solving for I_t yields:

$$I_t = 10^{SL/10} \times 0.67 \times 10^{-18} \tag{5.20}$$

in $Watts/m^2$. Finally, the transmitter power needed to achieve an intensity I_t at a distance of 1 m from the source in the direction of the receiver is expressed as

$$P_t = 2\pi \times 1m \times h \times I_t \tag{5.21}$$

in $Watts$, where h is the water depth in m.

Then, the maximum transmitting power $P_{i_{max}}$ for node i equal to P_t which is calculated from the TL, considered as the maximum *allowable* one-way *transmission loss* in passive sonars. The BER is then finally expressed as

$$BER = f\left(\frac{P_{i_{max}} \cdot 2N}{N_o \cdot r \cdot TL}\right) \tag{5.22}$$

5.4.4 Simulation Results

The PPC-PPM single-hop performance is tested using Matlab-Simulink. The block diagram of system model under consideration is shown in Figure 5.3. A set of Gold sequence will be generated and *pulse positioned* in order to *encode* the bit information. We use the BPSK carrier at 3.5 kHz to modulate the codewords and then send them through the *multipath* fading channel. The channel considered here is a Rayleigh fading channel. At the reception, the signal received is passed through a matched filter tuned to G_i. We then perform the *correlation* of the matched filter outputs to implicitly *auto-correlate* the CIR H by which each of the two G_i receptions have been spread, realizing a filter consisting of the TR (PC) operator $|H|^2$ as discussed in Section 5.3.2.

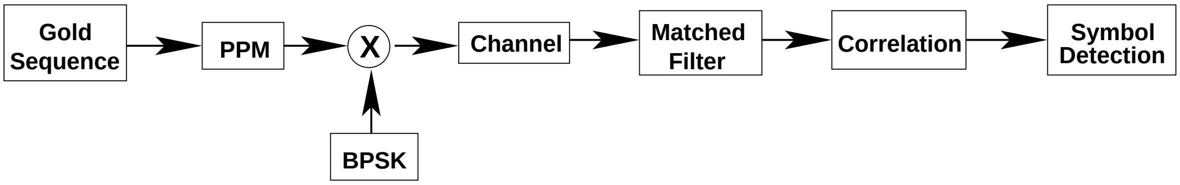

Figure 5.3: Block Diagram of PPC-PPM using Gold sequences

For the simulation, we consider communication between two nodes in shallow water. One node works as transmitter and the other as receiver. Both are centered at 3.5 kHz and have a bandwidth of 500 Hz. The Figures 5.4 and 5.5 shows the classical (theoretical) BER (5.18) and PPC-PPM BER (5.22) for underwater multipath fading channel (for $r = 126$ [bps] and 500 [bps], respectively). The distance between the two nodes, in both cases, is 1 km at depth 50 m. The results from our simulations for PPC-PPM BER gives *better* performance compared to the *theoretical* results. This shows that the PPC-PPM scheme can provide *promising* results for the *low* SNR region. Further, it is also immediately clear that the PPC-PPM BER using a lower value of r (i.e., 126 [bps] in Figure 5.4) gives better results compared to higher values (i.e., 500 [bps] in Figure 5.5). For details on this behavior, see (5.22).

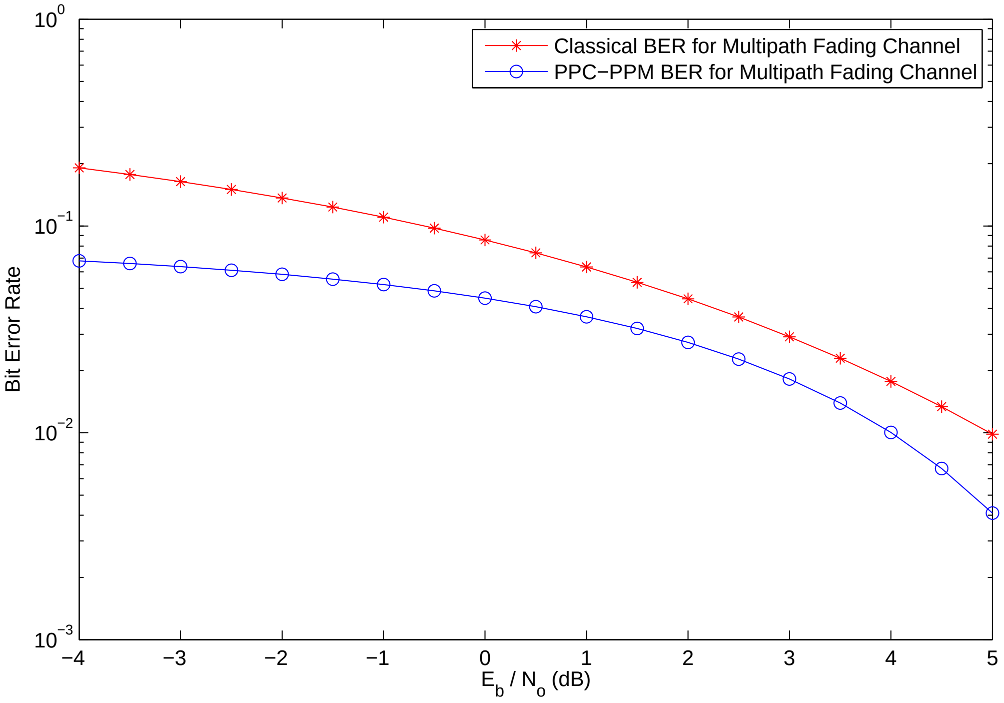

Figure 5.4: Bit-Error-Rate Vs. SNR for 126 [bps]

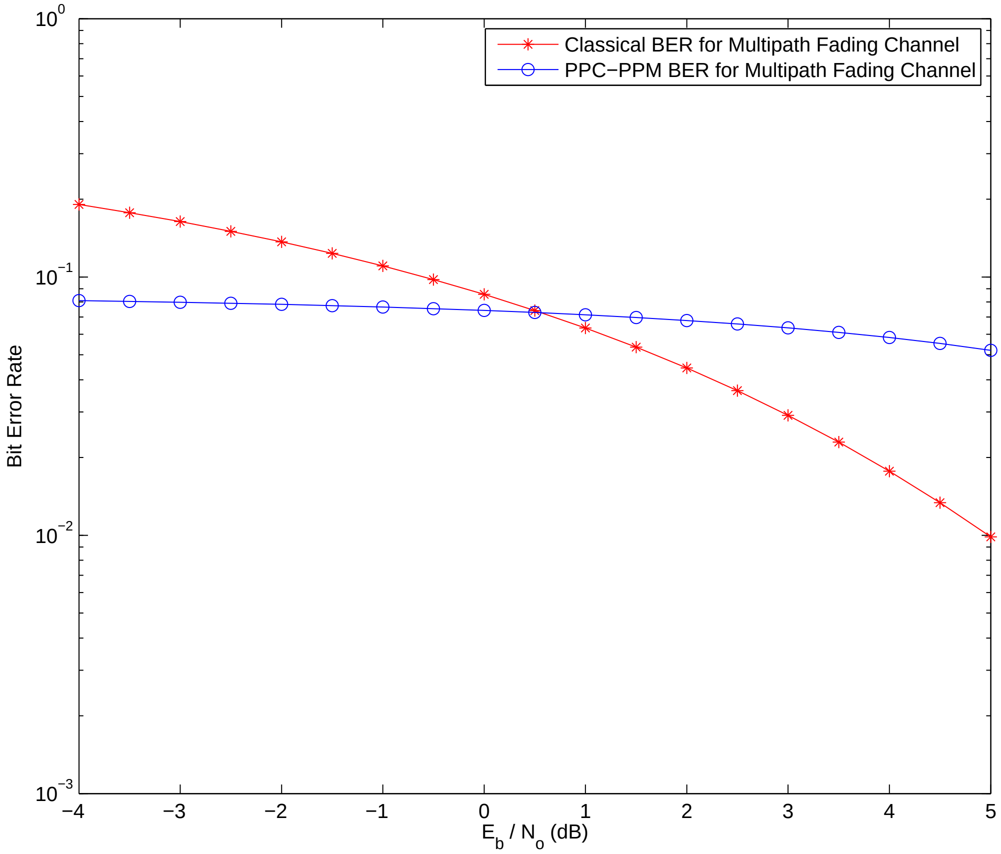

Figure 5.5: Bit-Error-Rate Vs. SNR for 500 [bps]

Figure 5.6 shows the BER w.r.t distance for $r = 126$ [bps], $h = 50$ m, and SNR $= 3$ [dB]. It is evident from the figure that the value of PPC-PPM BER decreases with distance. This behavior is also clear from (5.22), i.e., when the distance is increased, the TL will increase (see (5.3)) but it will decrease the BER.

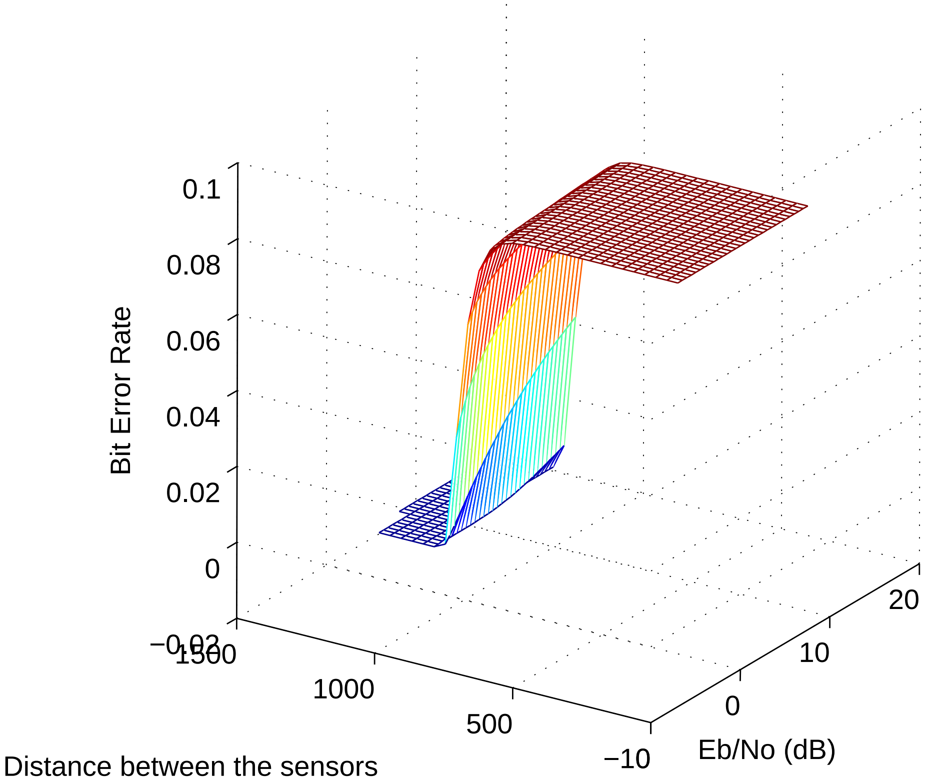

Figure 5.6: Bit-Error-Rate Vs. Distance (m)

Figure 5.7 shows the BER w.r.t physical layer rate (r) at SNR = 3 [dB], $d = 1\ km$ and $h = 50\ m$. The BER increases with the increase in transmission rate. This behavior also confirms the results of Figure 5.5. Therefore, it is ideal to use low transmission rates in order to achieve a desired (application specific) value of BER.

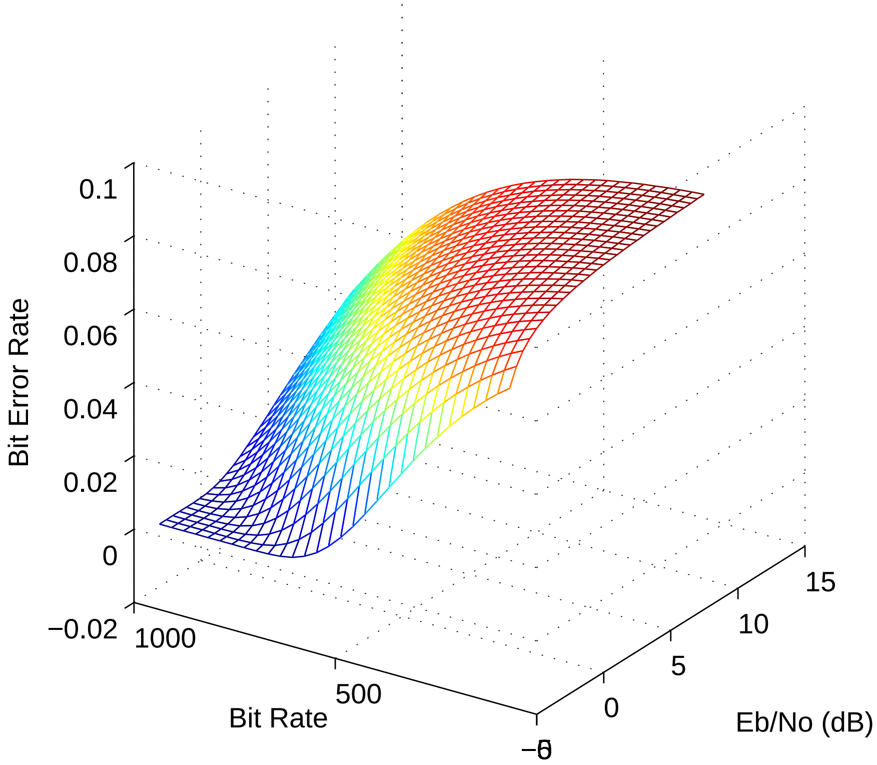

Figure 5.7: Bit-Error-Rate Vs. Physical layer Rate

Figure 5.8 shows the BER w.r.t depth at SNR = 3 [dB], $d = 1$ km, and $r = 126$ [bps]. The BER decreases with the increase in depth. This result is again evident from (5.21), where p_t is the required power to transmit the information at a given h. The results of Figures 5.6, 5.7, and 5.8 will help greatly in optimizing routing decisions when we will deal with communication infrastructure using a network of underwater acoustic sensor nodes.

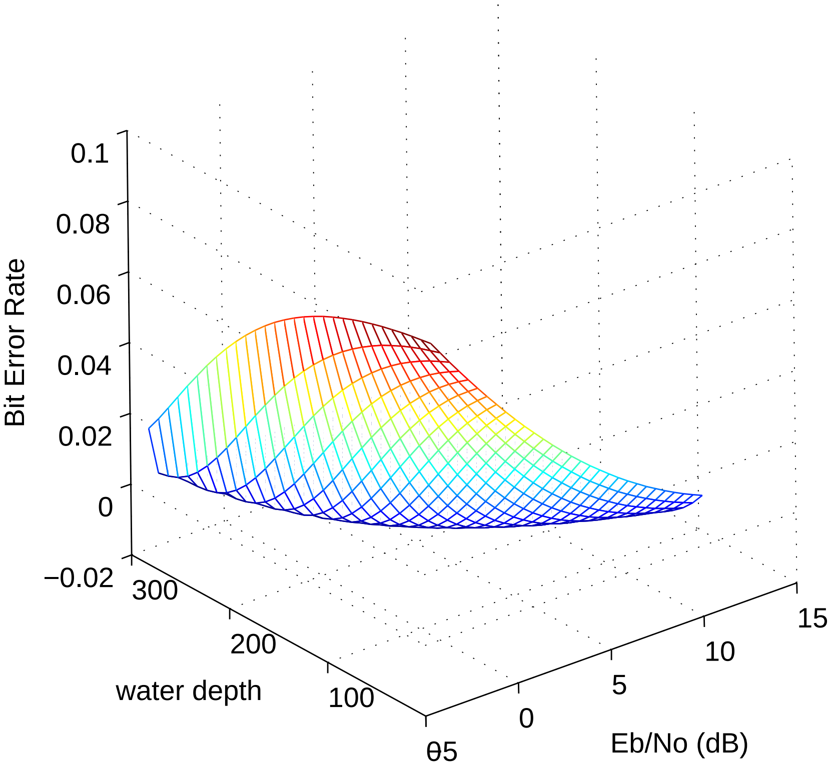

Figure 5.8: Bit-Error-Rate Vs. Depth

5.5 Case II: Multi-Hop Communication Framework

In this section, we consider a multi-hop communication architecture, for three-dimensional UASNs, that will be used in the formulation of our routing algorithms. These networks are used to detect and observe phenomena that cannot be adequately observed by means of ocean bottom sensor nodes, i.e., to perform cooperative sampling of the 3D-ocean environment. In three-dimensional UASNs, underwater sensor nodes float at different depths to observe a given phenomenon. There can be different means to deploy such network. Sensors can coordinate their depths in such a way as to guarantee connectivity, i.e., at least one path from every sensor to the sink always exists, and achieve communication coverage [4].

5.5.1 A Three-Node Linear Network

We first take an example of linear 3-node network. The node 1 sends a signal to node3, which is 2-hop away and is relayed by node 2 which can hear both 1 and 3. We do not discuss the MAC behavior for this example (given in the following section). The routing in this case is simple as the network is linear. Node 2 will receive the packets from node 1 (source) and relays them to node 3 (destination). In the one-way transmission PPC-configuration, the first-transmitter send two signals $S_1(\omega)$ and $S_2(\omega)$ one after another, respectively, through the same channel. They are received at the first-receiver as $H(\omega)S_1(\omega)$ and $H(\omega)S_2(\omega)$. The cross-correlation of $H(\omega)S_1(\omega)$ and $H(\omega)S_2(\omega)$ is the product $|H|^2 S_1^*(\omega)\ S_2(\omega)$ in which we find the TRO. The auto-correlation of the CIR $|H|^2$ tends to re-concentrate or focus the multipath arrivals at zero time lag. After the first transmission, the bit information is taken out of the data because of the phase difference, we *purposely* applied. For the second transmission, we combine a new set of Gold Sequence to this bit information and send this set again through the channel. Then, the first-receiver transmits the signals $S_1^{'}(\omega)$ and $S_2^{'}(\omega)$ one after another, respectively, through the same channel. They are received at the second-receiver as $H(\omega)S_1^{'}(\omega)$ and $H(\omega)S_2^{'}(\omega)$. The cross-correlation of $H(\omega)S_1^{'}(\omega)$ and $H(\omega)\ S_2^{'}(\omega)$ is the product $|H|^2 S_1^{'*}(\omega)S_2^{'}(\omega)$ in which we again find the TRO. Again, the auto-correlation of the CIR $|H|^2$ tends to focus the multipath arrivals. In addition, from the auto-correlation of $S_1^{'*}(\omega)S_2^{'}(\omega)$, we find almost *zero* BER compared to the first-hop transmission as shown in Figure 5.9.

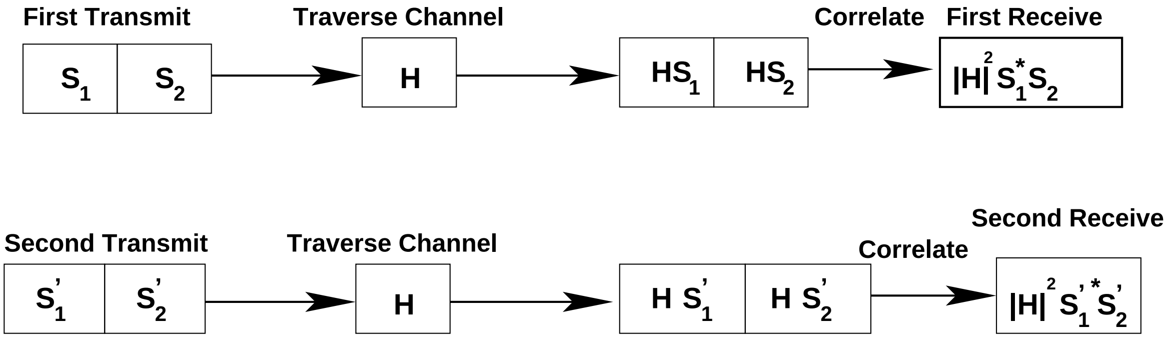

Figure 5.9: Passive Phase Conjugation (PPC) in a 3-node network

Proof: For the first transmission-reception pair as shown in Figure 5.9, if $S_1 = S(\omega)$ and $S_2 = S(\omega)e^{-j\frac{\varphi}{2\pi}}$, where φ is the phase difference. Then

$$Corr\left(HS_1, HS_2\right) = \int_{-\infty}^{+\infty} \left(H(\omega)S(\omega)\right)^* d\omega$$
$$H(\omega)S(\omega)e^{-j\frac{\varphi}{2\pi}}d\omega = |H(\omega)|^2 |S(\omega)|^2 e^{-j\frac{\varphi}{2\pi}}$$

And for the 2nd transmission-reception pair as shown in Figure 5.9, we have

$$Corr\left(HS_1^{'}, HS_2^{'}\right) = \int_{-\infty}^{+\infty} \left(HS_1^* S_2\right)^* H\left(S_1^* S_2\right)(\varphi)\, d\omega$$

$$= \int_{-\infty}^{+\infty} |S(\omega)|^2 e^{j\frac{\varphi}{2\pi}} H^*(\omega)H(\omega)|S(\omega)|^2 e^{-j\frac{\varphi}{\pi}} d\omega$$

$$= |S(\omega)|^4 \left(\int_{-\infty}^{+\infty} H^*(\omega)H(\omega)\,d\omega\right) e^{-j\frac{\varphi}{2\pi}}$$

$$= |H(\omega)|^2 |S(\omega)|^4 e^{-j\frac{\varphi}{2\pi}}$$

The proof is complete.

5.5.2 Network Model

In this chapter, we consider a UASN with N underwater acoustic sensor nodes with a common sink (referred to as *dest* in our modeling).

Channel Model: We assume that the channel is severely impaired due to multipath and fading: a node can decode a transmission successfully iff there is no other interfering transmission, if itself in not transmitting, and if the SNR is higher than a predefined threshold.

Antenna and Frequency: Each sensor node is equipped with one directional antenna with some maximum constraint on power. The transmit antenna has defined finite power levels. The sensor nodes share the same frequency band.

Neighborhood relation model: Given is an $N \times N$ neighborhood relation matrix $\mathcal{N}$ that indicates the node pairs for which direct communication is possible. We will assume that $\mathcal{N}$ is a symmetric matrix, i.e., if sensor i can transmit to sensor j, then j can also transmit to sensor i. For such sensor pairs, the $(i,j)^{th}$ entry of the matrix $\mathcal{N}$ is unity, i.e., $\mathcal{N}_{i,j} = 1$ if sensors i and j can communicate with each other; we will set $\mathcal{N}_{i,j} = 0$ if sensors i and j can not communicate. For any sensor i, we define $\mathcal{N}_i = \{j : \mathcal{N}_{i,j} = 1\}$, which is the set of neighboring sensors of i.

Data Collection Process and Relaying: Each sensor node is assumed to be sensing its environment at a predefined rate. Also, each sensor node wants to use the sensor network to forward its sampled data to a common sink, which is assumed to be a part of the network. Thus, each sensor node also acts as a forwarder of data from other sensor nodes in the network.

Channel Access Mechanism: The acoustic underwater channel has large signal propagation delay. Usual network communication is based on electromagnetic waves traveling at the speed of about $3 \times 10^8 m/s$. The speed of sound is approximately equal to $1.5 \times 10^3 m/s$. The difference in propagation speed can have a great impact on how protocol works. Aloha is a class of MAC protocols that do not try to prevent packet collision. Whereas, Slotted Aloha is a modified form of Aloha which introduces discrete time slots to improve channel utility. Assuming that there is a way to synchronize the nodes so that they could implement Slotted Aloha, we have to look at the propagation delay. Although the nodes sent the packets in predefined slots, there is no guarantee that they will arrive in time slots. Therefore, Slotted Aloha in underwater environment has no different performance than Aloha itself [121, 122] and we use Aloha like MAC for our modeling and simulation purposes.

5.5.3 The Routing Algorithm

In the following, we introduce a distributed routing solution for underwater applications. The algorithm is energy-aware and based only on local information as end-to-end route-information might not be available.

The objective of the distributed solution is to efficiently exploit the channel and to minimize the energy consumption [J-3]. It tries to exploit those links that guarantee a low packet error rate (PER), in order to maximize the probability that the packet is correctly decoded at the receiver. We now cast the distributed routing solution.

Algorithm 5.1 UWRP: UnderWater Routing Protocol

Given : i, $\mathcal{N}_i$, $\mathcal{P}_i$

$\mathcal{N}_i$ is the neighborhood set of i, and $\mathcal{P}_i$ is the positive advance set, composed of nodes closer to sink than node i, i.e., $j \in \mathcal{P}_i$ iff $d_{j,dest} < d_{i,dest}$.

Find : $j^* \in \mathcal{N}_i \cap \mathcal{P}_i$

$f\left(j^*\right) = \min f\left(j\right)$ such that

$f\left(j\right) = \alpha . \frac{PER_{i,j}}{\sum_{j\in\mathcal{N}_i PER_{i,j}}} + \left(1-\alpha\right) . \left(1 - \frac{P_j}{\sum_{j\in\mathcal{N}_i} P_j}\right)$

where $\alpha \in [0, 1]$ is a smoothing parameter applied to the choice of next-hop for routing.

$PER_{i,j} = 1 - \left(1 - BER_{i,j}\right)^s$ is the Packet-Error-Rate on the link (i, j) for a packet size s.

Also, P_j is the remaining energy of sensor j.

Note that the selection of j^* is subject to two objective functions which need only local neighborhood knowledge and can be readily available without any protocol *overhead*. By using $PER_{i,j}$ with some positive value of α, we want to use a link (i, j) with the *minimum* value of PER on that link in order to *minimize* the number of retransmission attempts. And using P_j with its given weight $(1 - \alpha)$, we want to use a link (i, j) with maximum *remaining* energy in order to *prolong* the network-lifetime of battery-operated UASNs. A weighted sum of these two quantities results in the selection of next-hop for routing. Note that this information is available to the nodes *without* any extra signaling overhead. Generally, ad hoc protocols rely on acknowledgments for reliable data transfer. When a node j receives a packet from i, it sends an ACK packet to j to notify successful packet reception. This ACK packet is modified to indicate the values of certain parameters like P_j and $PER_{i,j}$.

5.5.4 Simulation Results

The PPC-PPM multi-hop performance is tested using Matlab-Simulink. The block diagram of physical layer system model under consideration is shown in Figure 5.3. A set of Gold sequence will be generated and pulse positioned in order to encode the bit information. We use the BPSK carrier at 3.5 kHz to modulate the codewords and then send them through the multipath fading channel. We consider $N = 50$ underwater acoustic sensor nodes that are *randomly* distributed in the network with sink located roughly in the *center* of the topology. The transmission range of sensors is such that there are no *isolated* sensors in the network. The sensor nodes learn the network topology during the initial network deployment to form an *aggregation* tree towards the sink. All sensors are the sources of data and generate a packet every $5mins$. The value of α for our simulations is set to $\alpha = (0.65, 0.8]$. Multihop routing is used and the next-hop is decided as the outcome of the distributed routing algorithm presented in Section 5.5.3. All other details of the simulation model can be found in Section 5.5.2.

Figure 5.10 shows the PER, when 1-hop, traditional routing, between link $(i, j))$ versus 2-hop routing is used. We would like to mention here that 2-hop case does not mean state that all the sensor nodes are two hops away from the sink, it merely says that instead of a one-hop routing between nodes i and j, we use an intermediate sensor node k and route data on links (i, k) and (k, j) instead of link(i, j) . This result shows that instead of performing a 1-hop routing on link (i, j), a 2-hop routing (on links (i, k) and (k, j)) results in a dramatic decrease in PER even when i and j are in the transmission range of each other. An interesting factor here is the choice of optimal k for each link (i, j). From the neighborhood information $\mathcal{N}_i$, we choose $k(k \neq i, j)$ in a similar fashion as j^* in Algo. 5.1 . This particular choice of routing also *minimizes* the *average* number of transmission attempts of a packet on a given link. This result is quite evident from Figure 5.11. It shows the average number of transmission attempts for a packet under different values of PER. The network-lifetime of battery operated underwater acoustic sensor nodes can be three times longer using our routing approach compared to any other multi-hop energy-aware routing protocol and is clear from Figure 5.11. Although, this is achieved at the cost of slightly higher network-layer delay . We use $\hat{N}_{i,j}^{tx}$ to represent the average number of retransmit attempts on the link (i, j) and is calculated as follows:

$$\hat{N}_{i,j}^{tx} = \left\lceil \frac{1}{1 - PER_{i,j}} \right\rceil . \tag{5.23}$$

Equation (5.23) assumes *independent* errors among adjacent packets, which holds when the channel *coherence* time is *shorter* than the average *retransmission* delay, i.e., the average time that a sender needs to *retransmit* an unacknowledged packet. Therefore, given the harsh characteristics of the underwater channel, our routing scheme (based on TR approach) performs better than the traditional multi-hop routing approaches in UASNs. The value of TR (PC) is its ability to focus sound back at the original source without any a priori knowledge of the underwater channel. The TR focus enables the underwater sensor network to *minimize* the effect of *multipath* and channel *fading* thus improving the SNR and reducing the BER (PER for multihop case) and ISI.

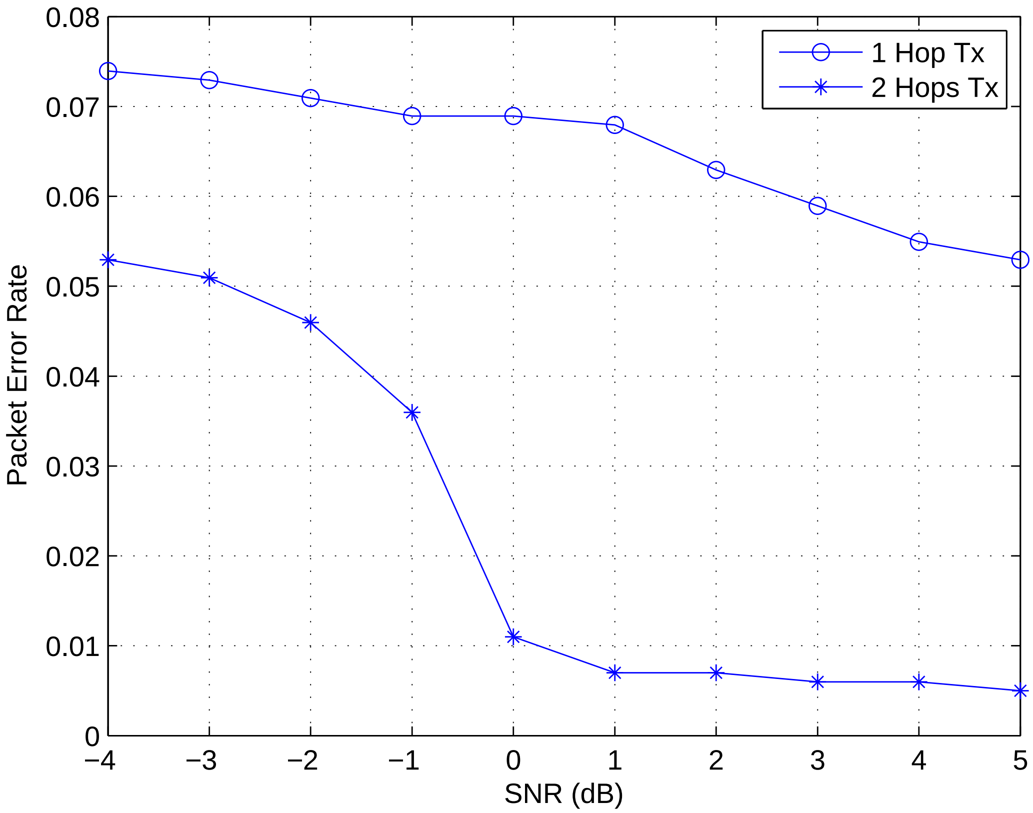

Figure 5.10: Packet-Error-Rate Vs. SNR

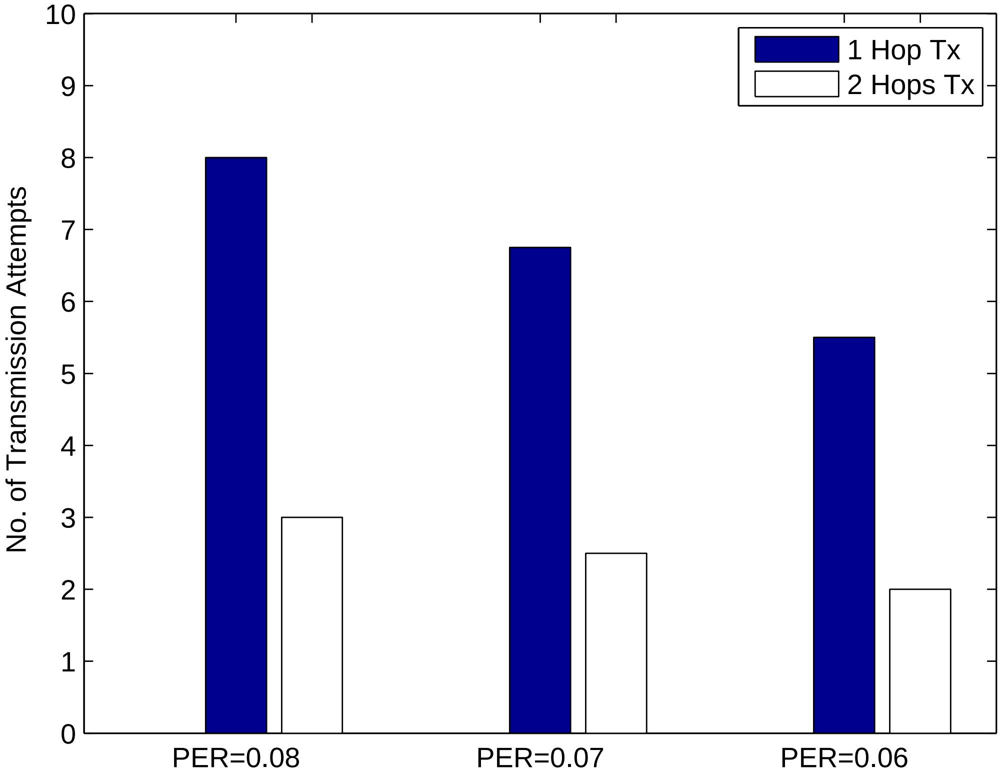

Figure 5.11: Average number of packet transmission attempts

5.6 Conclusions and Future Work

In this chapter, we considered UASNs that are deployed to perform collaborative monitoring of tasks over a given volume of water. We first use a *single* antenna transmitter and receiver for *spatial focusing*. PPC implicitly equalizes the channel by *refocusing* channel spread. We have *quantified* the gain of PPC temporal compression, in a single-hop point-to-point setup, using Gold sequences to implement receivers based on PPM modulation schemes. We have also shown that PPC-PPM can be applied to a network to achieve almost *zero* BER for a two-hop communication mode compared to single-hop communication. This result is of *significant* importance in making routing decisions for UASNs. In particular, it allows a node to select its next hop with the aim of *minimizing* the energy consumption. This cross-layering (PHY-Routing) improves the network lifetime of battery operated UASNs by reducing the number of *retransmission-attempts* between any given pair of nodes. It also minimizes the energy consumption per successful transmission.

In future, we will consider a multihop 3D underwater acoustic sensor network with a real-life application and extend the results of this chapter to optimize the scheduling layer (MAC) and routing layer with the aim of minimizing energy consumption per successful transmission and maximizing the network lifetime.

Chapter 6

Conclusion and Outlook

In this chapter, we first summarize briefly the contributions of this thesis in Section 6.1. In Section 6.2, we discuss extensions of this work that will provide interesting challenges for future research in the field of cross-layer optimizations in wireless sensor and sensor-actuator networks.

6.1 Summary of Contributions

In this thesis, we have considered three different types of sensor networks: 1) WSNs, 2) SANETs, and 3) UASNs. WSNs consist of large number of distributed sensor nodes that organize themselves into a multihop wireless network. Typically, these nodes coordinate to perform a common task. On the other hand, SANETs are composed of possibly a large number of tiny, autonomous sensor devices and actuators equipped with wireless communication capabilities. Distributed systems based on networked sensors and actuators with embedded computation capabilities enable an instrumentation of the physical world at an unprecedented scale and density, thus enabling a new generation of monitoring and control applications. SANETs is an emerging technology that has a wide range of potential applications including environment monitoring, medical systems, robotic exploration, and smart spaces. Such networks consist of large number of distributed sensor and few actuator nodes that organize themselves into a multihop wireless network. SANETs are becoming increasingly important in recent years due to their ability to detect and convey real-time, in-situ information for many civilian and military applications. In contrast to terrestial WSNs, UASNs consist of sensors that are deployed to perform collaborative monitoring of tasks over a given volume of water. These networks will find applications in oceanographic data collection, pollution monitoring, offshore exploration, disaster prevention, assisted navigation, tactical surveillance, and mine reconnaissance. Moreover, UUVs and AUVs equipped with sensors, will enable the exploration of natural undersea resources and gathering of scientific data in collaborative monitoring missions. Underwater acoustic networking is the enabling technology for these applications.

Throughout this work, we have addressed the design, implementation, and the *cross-layer* optimization of routing protocols in WSNs, SANETs, and UASNs. Since the introduction of the WSNs and the associated routing/aggregation protocols, this topics has attracted significant research efforts, with the ultimate hope to set the basic theory that would be able to model the diversity of constraints and requirements to characterize it. Meanwhile, different aspects of the problem are addressed continuously by different approaches, enabling more and more advances in the treatment of the subject.

In dealing with the routing protocols for WSNS, SANETs, and UASNs, we have considered *cross-layering* between: application/MAC, routing/MAC, MAC/PHY layers of the protocol stack for such networks.

For WSNs with random channel access, we proposed a cross-layered (application/MAC) data sampling approach that guarantees a *long term* data sampling rate while *minimizing* the end-to-end delays. Simulation results show that performance of this scheme is better than the traditional layered architecture, where the channel access mechanism is independent of the data sampling process. We also saw that the proposed scheme does not require *tedious* parameter tuning as is the case for the layered architecture.

For general purpose WSNs, we proposed another cross-layered approach and obtained some important insights into various *tradeoffs* that can be achieved by varying certain network parameters. Some of them include: 1) Routing can be crucial in determining the stability properties of the networked sensors. 2) Whether or not the forwarding queues can be stabilized (by appropriate choice of WFQ weights) depends only on *routing* and *channel access rates* 3) We have also seen that the end-to-end throughput is independent of the choice of WFQ weights. We therefore, proposed a distributed learning algorithm to achieve Wardrop equilibrium for the end-to-end delays incurred on different routes from a sensor node to a fusion center (sink). From the simulation results, we have seen a very high delay for a single-queue system (provided the system was stable) compared to two-queues system.

Another objective was to minimize the total delay in the network in *layered* architecture, where the constraints are the arrival-rate and service-rate of a node. Particularly, we have shown that the objective function is strictly convex for the entire network. We then use the Lagrangian dual decomposition method to devise a distributed primal-dual algorithm to minimize the delay in the network. The deterministic distributed primal-dual algorithm requires no feedback control and therefore converges almost surely to the optimal solution. The results show that the required optimal value of *service rate* is achieved for every node in the network by the distributed primal-dual algorithm. It is important to pay equal attention to the observed delay in the network and energy consumption for data transmissions. A fast convergence only means that a small amount of extra energy is consumed to perform local calculations to achieve the desired optimizations. Similarly for the stochastic delay control algorithm, we have shown a probability one convergence and its rate of convergence properties.

We then proposed a *learning* algorithm, applicable to both the open system (layered-architecture) as well as the closed system (cross-layered architecture), to achieve Wardrop equilibrium for the end-to-end delays incurred on different routes from sensor nodes to the fusion center. For the closed system, this algorithm also adapted the *channel access rates* of the sensor nodes.

For wireless sensor-actuator networks with random channel access, we propose that each sensor must transmit its readings toward one actuator only in order to take the burden of relaying, toward different actuators, away from energy-constrained sensors in a straight forward fashion. We then proposed a data sampling approach that guarantees a *long term* data sampling rate while minimizing the end-to-end delays. Simulation results show that performance of this scheme is better than the traditional layered architecture where the channel access mechanism is independent of the data sampling process. We also saw that the proposed scheme does not require tedious parameter tuning as is the case for the layered architecture. We also proposed an algorithm for an optimal actuator selection that provides a good mapping between any sensor and an actuator in the network. The selection algorithm finds a delay-optimal actuator for each sensor in polynomial time. We then proposed a learning algorithm, applicable to both the open system as well as the closed system, to achieve Wardrop equilibrium for the end-to-end delays incurred on different routes from sensor nodes to their attached delay-optimal actuators. For the closed system, this algorithm also adapted the channel access rates of the sensor nodes. We finally propose a distributed actuation control mechanism for SANETs that is responsible for an efficient actuation process.

The actuators can dynamically coordinate and perform power control to maintain a defined level of connectivity subject to throughput constraints. The control overhead for static and mobile actuator scenarios is analyzed using ns-2 simulations. The PC heuristic algorithm is applicable to multihop SANETs to increase throughput, battery life and connectivity.

We then consider a large scale SANET with multiple actuators as sinks for data generated by the sensors. Since many applications require to have each source node send all its locally generated data to only one actuator for processing, it is necessary to optimally map each sensor to its actuator. Also considering the fact that the end-to-end delays in wireless sensor-actuator networks is a hard constraint, we jointly optimize the actuator selection and optimal flow routing subject to energy and delay constraints with the global aim of maximizing the network lifetime. We proposed and evaluated (using a standard network simulator, ns-2) our actuator-selection (LEAD-ADP) and routing scheme (LEAD-RP) on top of a TDMA based MAC (LEAD-MAC) protocol. We then use the Lagrangian dual decomposition method to devise a distributed primal-dual algorithm to maximize network-lifetime in the network. The deterministic distributed primal-dual algorithm requires no feedback control and therefore converges almost surely to the optimal solution. The results show that the required optimal value of lifetime is achieved for every node in the network by the distributed primal-dual algorithm. We also provide a comparison to the analytical bound. Simulation results show that this approach has near-optimal performance and is practically implementable as compared to earlier analytical studies based only on numerical evaluations.

First, this chapter addresses the problem of isolated regions in the sensing field by letting actuators exchange their CSIT and jointly perform beamforming in order to deliver scheduling information to sensor nodes. The gains of cooperation were shown by simulating the average number of isolated sensors for the case of single actuator transmission and cooperative transmission. Since many applications require to have each source node send all its locally generated data to only one actuator for processing and the fact that the end-to-end delays in SANETs is a *hard* constraint, we jointly optimize the actuator selection and optimal flow routing subject to delay-energy constraints. This approach has near-optimal *performance* and is *practically implementable.*

We use a *single* antenna transmitter and receiver for *spatial focusing.* PPC implicitly equalizes the channel by *refocusing* channel spread. We have *quantified* the gain of PPC temporal compression, in a single-hop point-to-point setup, using Gold sequences to implement receivers based on PPM modulation schemes. We have also shown that PPC-PPM can be applied to a network to achieve almost *zero* BER for a two-hop communication mode compared to single-hop communications. This result is of *significant* importance in making routing decisions for UASNs. In particular, it allows a node to select its next hop with the aim of *minimizing* the energy consumption. This cross-layering (PHY-Routing) improves the network lifetime of battery operated UASNs by reducing the number of *retransmission-attempts* between any given pair of nodes. It also minimizes the average-energy consumption per successful transmission.

6.2 Future Directions

- Since the objective of the distributed routing algorithm was only to converge to a Wardrop equilibrium, at this moment it is not able to make a judicious choice among multiple Wardrop equilibria, if they exist. We will work on modifications of the algorithm to make it converge to an *efficient* equilibrium.

- In the near-future, we will present a detailed simulation based study of PC heuristic algorithm in different networking scenarios with some application specific actuation requirements and practical evaluation of distributed multiple-actuator actuation process. We will also work on the development of PC heuristic algorithm to improve some MAC layer performance metrics using a cross-layer approach. As a consequence of a very fast convergence to Wardrop equilibrium, we are also tempted to perform the energy-analysis of the proposed learning and routing scheme in the context of network lifetime.

- We will consider a real-life SANET application and simulate its behavior with the LEAD self-organizing framework to observe its performance. We will take into consideration a dynamic actuator-assignment scenario to timely transport data in a *mobile* wireless sensor-actuator network.

- We will also consider a multihop 3D underwater acoustic sensor network with a real-life application and extend the results of this chapter to optimize the scheduling layer (MAC) and routing layer with the aim of minimizing energy consumption per successful transmission and maximizing the network lifetime.

Chapter 7

Résumé en Francais

Les récents progrès dans les systèmes micro-électro-mécaniques (MEMS) technologie, les communications sans-fil, et de l'électronique numérique ont permis le développement de faible coût, de faible puissance, nœuds de capteurs multifonctionnels qui sont de petite taille et de communiquer non attachés à de courtes distances. Wireless Sensor Networks (RdC) se composent d'un grand nombre de nœuds de capteurs distribués qui s'organisent en un réseau sans fil multi-sauts comme le montre la Figure 7.1. Un réseau de capteurs sans fil est un réseau ad hoc avec un grand nombre de nœuds qui sont des micro-capteurs capables de récolter et de transmettre des données environnementales d'une manière autonome. Chaque nœud a un ou plusieurs capteurs, processeurs embarqués, et les radios de faible puissance, et est normalement fonctionner sur batterie. Typiquement, ces nœuds coordonner pour accomplir une tâche commune. Ces nœuds de capteurs minuscules, qui sont composées de détection, de traitement des données, et communiquer les composants, l'idée d'exploiter les réseaux de capteurs basés sur l'effort de collaboration d'un grand nombre de nœuds. Les réseaux de capteurs représentent une amélioration importante par rapport aux capteurs traditionnels, qui sont déployés dans les deux manières suivantes [1]:

- Les capteurs peuvent être placés loin du phénomène réel, c'est à dire quelque chose de connu par la perception sensorielle. Dans cette approche, grands capteurs qui utilisent des techniques complexes de distinguer les cibles de bruit dans l'environnement sont nécessaires.

- Plusieurs détecteurs qui effectuent uniquement la détection peut être déployé. Les positions de la topologie de capteurs et des communications peuvent être réalisées avec soin. Ils transmettent des séries temporelles du phénomène à senti les nœuds centraux où les calculs sont effectués et les données sont fusionnées. L'entité centrale est montré dans la Figure 7.1. Il peut être placé en fonction de nulle part sur l'application a besoin.

Un réseau de capteurs est composé d'un grand nombre de nœuds de capteurs, qui sont fortement affecté, soit à l'intérieur du phénomène ou très proche de lui. La position de nœuds de capteurs ne doivent pas être conçus ou pré-déterminé. Ils peuvent être aléatoirement dispersés dans une zone géographique, appelée « champ de captage » correspondant au terrain d'intérêt pour le phénomène capté. Cela permet un déploiement aléatoire dans des terrains inaccessibles ou les opérations de secours aux sinistrés. D'un autre côté, cela signifie aussi que les protocoles de réseau de capteurs et d'algorithmes doivent posséder des capacités d'auto-organisation. Une autre caractéristique unique du RdC est l'effort de coopération de nœuds

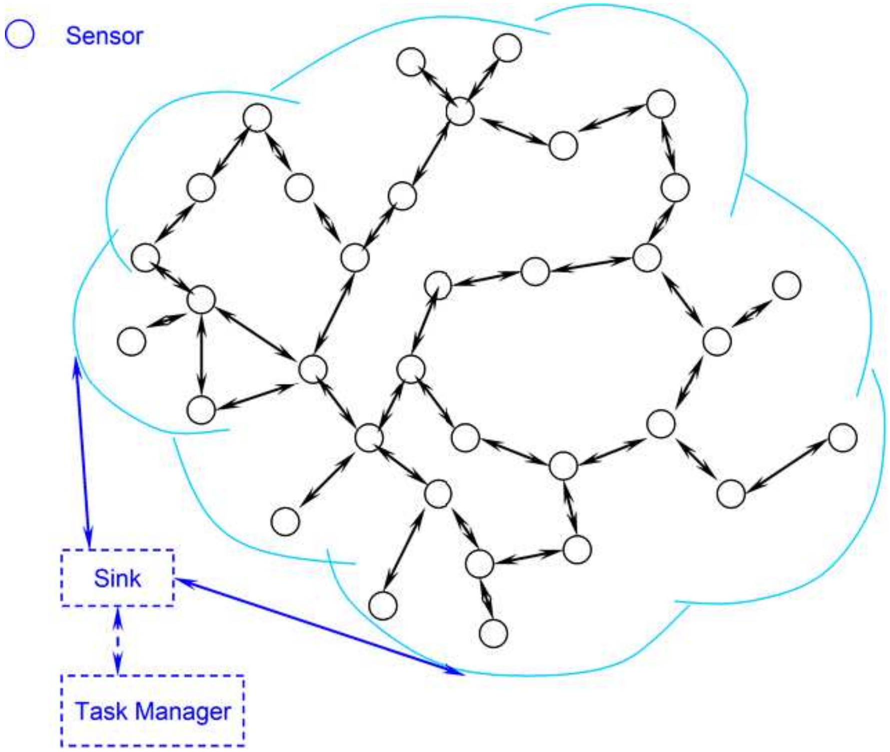

Figure 7.1: Un réseau des capteurs sans-fil

de capteurs. Nœuds de capteurs sont équipés d'un processeur embarqué. Au lieu d'envoyer des données brutes pour les nœuds responsable de la fusion, les nœuds de capteurs utilisent leurs capacités de traitement d'effectuer localement à partir des calculs simples et transmettre des données que le nécessaire et partiellement traitées.

Les caractéristiques décrites ci-dessus d'assurer une gamme étendue d'applications pour les réseaux de capteurs. Quelques-uns des domaines d'application sont la santé, de l'armée, l'environnement, civil, et la sécurité. Par exemple, les données physiologiques sur un patient peut être surveillé à distance par un médecin. Bien que ce soit plus pratique pour le patient, il permet également au médecin de mieux comprendre l'état actuel du patient. Les réseaux de capteurs peuvent également être utilisés pour détecter des agents chimiques étrangers dans l'air et l'eau. Ils peuvent aider à identifier le type, la concentration et la localisation des polluants. En substance, les réseaux de capteurs peuvent fournir à l'utilisateur final, avec intelligence et une meilleure compréhension de l'environnement. Nous imaginons que, dans l'avenir, les réseaux de capteurs sera une partie intégrante de notre vie, plus que l'actuel nos ordinateurs personnels.

Ces réseaux de faible puissance et de perte (LLNs) sont constitués de nombreux dispositifs intégrés avec une puissance limitée, la mémoire et les ressources de traitement. Ils sont reliés entre eux par une série de liens, tels que IEEE 802.15.4, Bluetooth, WiFi basse puissance, câblés ou d'autres de faible puissance PLC (Powerline Communication) liens. LLNs sont la transition vers une bout-en-bout solution IP pour éviter le problème de la non-interopérabilité des réseaux interconnectés par des passerelles de conversion de protocoles et les proxies. Protocoles de routage existants, tels que OSPF, IS-IS, AODV et OLSR ont été évaluées par le ROLL groupe de travail IETF [2] et avoir en leur forme actuelle, été jugées conformes à pas toutes les exigences en matière de routage spécifiques WSN. Le groupe travaille actuellement sur la normalisation des fonctionalités de routage pour les exigences spécifiques posées par LLNs.

Réseaux de capteurs et actionneurs sans-fil (SANETs) sont parmi les domaines de recherche le plus traité dans le domaine des technologies d'information et de la communication (TIC), ces jours, aux Etats-Unis, en Europe et en Asie. SANETs sont composés de, éventuellement, un grand nombre de minuscules dispositifs de capteurs autonomes et actionneurs équipés de capacités de communication sans fil montré dans la Figure 7.2. Un des aspects les plus pertinents de ce domaine de recherche se trouve dans sa multidisciplinarité et le large éventail de compétences qui sont nécessaires à l'approche de leur conception. SANETs sont une technologie émergente qui dispose d'un large éventail d'applications possibles, y compris surveillance de l'environnement, les systèmes médicaux, l'exploration robotique et des espaces intelligentes. SANETs sont de plus en plus importante ces dernières années en raison de leur capacité à détecter et à transmettre en temps-réel, des informations in-situ pour de nombreuses applications civiles et militaires.

Chaque nœud de capteur possède un ou plusieurs capteurs (y compris le multimédia, i.e., la vidéo et l'audio ou de données scalaires, i.e., température, pression, lumière, infrarouge et magnétomètre), les processeurs embarqués, des radios de faible puissance, et est normalement fonctionner sur batterie. Un actionneur est un dispositif qui convertit un signal de commande électrique à une action physique, et constitue le mécanisme par lequel un agent agit sur l'environnement physique. Du point de vue pris en compte dans cette thèse, toutefois, un actionneur, en plus d'être capable d'agir sur l'environnement au moyen d'un ou plusieurs actionneurs, est également une entité de réseau qui effectue des fonctionnalités liées au réseautage, à savoir, recevoir, transmettre, traiter et relais de données. Par exemple, un robot peut interagir avec l'environnement physique, au moyen de plusieurs moteurs et servo-mécanismes (actionneurs). Toutefois, d'un point de vue de la mise en réseau, le robot constitue une entité unique, ce qui est appelé actionneur. Par conséquent, l'actionneur terme englobe les dispositifs hétérogènes comprenant des robots, des véhicules aériens sans pilote (UAV), et actionneurs en réseau telles que l'arrosage d'eau, caméras panoramiques basculantes, des bras robotiques, applications, etc.

Toutefois, en raison de la présence d'actionneurs, SANETs ont quelques différences par les réseaux de capteurs tel que décrit ci-dessous:

- Bien que les nœuds de capteurs sont de petits dispositifs avec une limite de détection, de calcul et capacités de communication sans-fil, les actionneurs sont généralement riches en ressources dispositifs équipés de capacités de traitement meilleur, plus fort pouvoir de transmission et la durée de vie de la batterie.

- En SANETs, il est peut être nécessaire de répondre rapidement. Par conséquent, la question de la communication en temps réel est très important dans SANETs puisque les actions sont effectuées sur l'environnement après la détection a lieu.

- Un déploiement dense n'est pas nécessaire pour l'actionneur nœuds. Ainsi, dans SANETs le nombre d'actionneurs est beaucoup plus faible que le nombre de capteurs.

- Afin d'assurer l'efficacité de détection et d'agir, un mécanisme distribué de coordination locale est nécessaire entre les capteurs et actionneurs. Dans les réseaux de capteurs, l'entité centrale exerce les fonctions de collecte de données et la coordination.

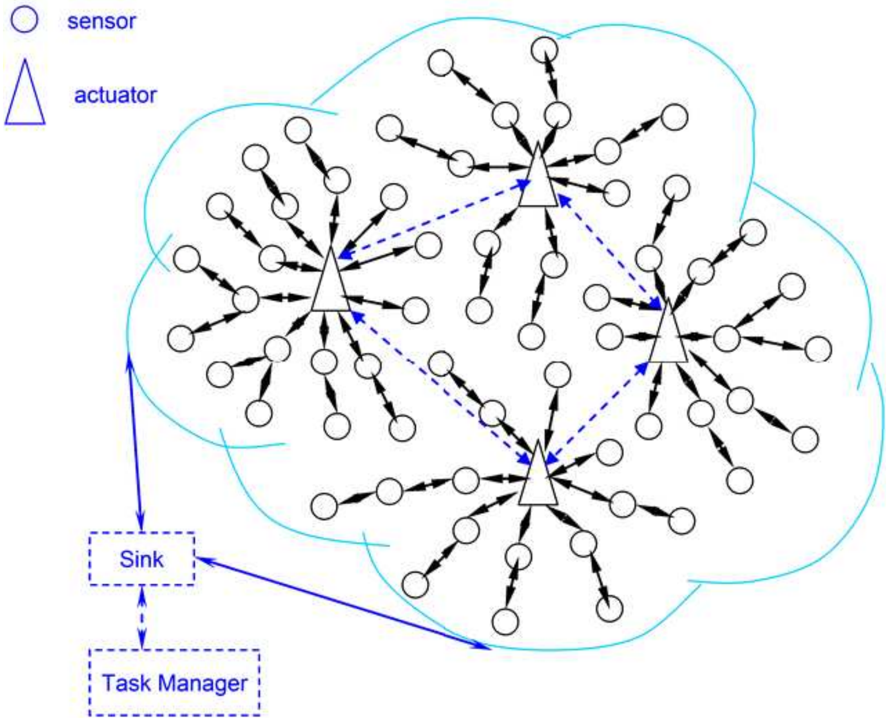

Figure 7.2: Un réseau sans fil capteurs-actionneurs

> Dans SANETs, des phénomènes nouveaux appelé: capteur-actionneur, et actionneur-actionneur peut se produire.

De nombreux protocoles et algorithmes ont été proposés pour les réseaux de capteurs dans les années récentes [3]. Toutefois, puisque les conditions énumérées ci-dessus imposent des contraintes plus strictes, qu'ils ne peut pas être bien adapté aux caractéristiques propres et l'application de SANETs.

Nœuds de capteurs de fond de l'océan sont réputés pour permettre aux applications de collecte de données océanographiques, surveillance de la pollution, l'exploration offshore, de la prévention des catastrophes, avec l'aide de navigation et les applications de surveillance tactique. Unmanned/Autonomous Underwater Vehicles (UUV, AUV), équipé de capteurs sous-marins, sera également trouver une application dans l'exploration des ressources naturelles sous-marines et la collecte de données scientifiques à des missions de surveillance en collaboration. Pour que ces applications viables, il est nécessaire pour permettre des communications sous-marines parmi les appareils sous-marins. Nœuds de capteurs sous-marins et des véhicules doivent posséder des capacités d'auto-configuration, autrement dit, ils doivent être en mesure de coordonner leurs activités par l'échange de configuration, la localisation et des informations de mouvement et à relayer les données surveillées à une station à terre.

Le réseau sans fil acoustique sous-marine est la technologie habilitante pour ces applications. Underwater Acoustic Sensor Networks (UASN) consistent en un nombre variable de capteurs et de véhicules qui sont déployés pour accomplir les tâches de surveillance de collaboration sur une zone donnée. Pour atteindre cet objectif, les capteurs et les véhicules auto-organiser en un réseau autonome qui peut s'adapter aux caractéristiques de l'environnement océanique [5].

Mise en réseau sous-marine est une zone relativement inexplorés bien que les communications sous-marines ont été expérimentées depuis la 2ème Guerre Mondiale, quand, en 1945, un téléphone sous-marin a été développé aux Etats-Unis pour communiquer avec les sous-marins. Communications acoustiques sont la technologie à couche physique dans les réseaux sous-marins. En fait, les ondes radio se propagent à de longues distances dans l'eau de mer conductrice seulement à des fréquences très basse (30-300 Hz), qui nécessitent de grandes antennes et la puissance de transmission élevée. Ondes optiques ne souffrent pas d'une forte atténuation tels peut sont influencées par la diffusion. En outre, la transmission de signaux optiques à une haute précision de pointage du laser faisceaux étroits. Ainsi, les liens dans les réseaux sous-marins sont basés sur les communications acoustiques sans fil.

L'approche traditionnelle pour la marine à fond ou la colonne de surveillance des océans est de déployer des capteurs sous-marins que les données enregistrées au cours de la mission de suivi, et récupérer ensuite les instruments. Inconvénients suivants sont offerts par ce régime:

- Suivi en temps réel n'est pas possible. Ceci est particulièrement critique dans la surveillance ou dans les applications de surveillance environnementale tels que la surveillance sismique. Les données enregistrées ne sont pas accessibles que lorsque les instruments sont récupérées, mais qui arrive plusieurs mois après le début de la mission de surveillance.

- Aucune interaction n'est possible entre les systèmes de contrôle à terre et en instruments de contrôle. Cela entrave toute adaptative mise au point des instruments, et il n'est pas possible de reconfigurer le système après des événements particuliers se produisent.

- Si des défaillances ou erreurs de configuration se produit, elle ne peut pas être possible de les détecter avant les instruments sont récupérés. Ceci peut facilement conduire à la rupture complète d'une mission de surveillance.

- La quantité de données qui peuvent être enregistrées au cours de la mission de suivi par chaque capteur est limitée par la capacité des dispositifs de stockage embarqué (mémoires, disques durs, etc.).

Par conséquent, il est nécessaire de déployer des réseaux sous-marin qui permettra le suivi en temps réel des zones océaniques sélectionnée, la configuration à distance et l'interaction avec les opérateurs sur terre. Ceci peut être obtenu en connectant les instruments sous-marins au moyen de liaisons sans fil basé sur la communication acoustique.

De nombreux chercheurs sont actuellement engagés dans le développement de solutions de réseautage pour les réseaux de capteurs terrestres. Bien qu'il existe de nombreuses récemment mis au point des protocoles de réseau pour les réseaux de capteurs, les caractéristiques uniques de la chaîne de communication acoustique sous-marine, comme la capacité de bande passante limitée et des retards variables, nécessitent des protocoles très efficace et fiable des données de communication. La qualité du lien acoustique sous-marine est très imprévisible, car elle dépend principalement de la décoloration et le multi-trajet, qui ne sont pas des phénomènes facilement modélisés. Ceci, en retour, se dégrade considérablement les performances au niveau des couches supérieures comme les très longs et variables retards de propagation. En outre, cette variation est généralement plus importante dans les liens horizontaux que dans les verticales. Signalisation acoustiques pour les communications numériques sans fil dans l'environnement de mer peuvent être une alternative très attrayante tant pour les systèmes de radio et de télémétrie câblé. Toutefois, variables dans le temps des trajets multiples et souvent rudes conditions de bruit ambiant caractériser le canal acoustique sous-marine, ce qui rend souvent les communications acoustiques difficiles. Des défis majeurs dans la conception de UASNs sont les suivants:

- Le canal est gravement altérée, principalement en raison de trajets multiples.
- La perte temporaire de connectivité principalement à l'ombrage.
- Le retard de propagation est de cinq ordres de grandeur plus élevé que dans les canaux de fréquence radio terrestre et est généralement variable [4].
- Très faible bande passante disponible.
- Garantie limitée de l'énergie de la batterie à disposition.

Puisque les missions de surveillance sous-marins peuvent être extrêmement coûteuses en raison des coûts élevés impliqués dans des appareils sous-marins, il est important que le réseau déployé être très fiables, de manière à éviter l'échec des missions de surveillance en raison de défaillance des dispositifs simples ou multiples. Par exemple, il est crucial d'éviter de concevoir la topologie du réseau avec les points uniques de défaillance susceptible de compromettre le fonctionnement global du réseau. La capacité du réseau est également influencée par la topologie du réseau. Comme la capacité de la chaîne sous-marine est extrêmement limitée, il est très important d'organiser la topologie du réseau ainsi un tel qu'aucun des goulets d'étranglement de communication sont mis en place.

Fonctionnement

Les données captées par les nœuds sont acheminées grâce à un routage multi-saut à un nœud considéré comme un "point de collecte", appelé nœud-puits (ou sink). Ce dernier peut être connecté à l'utilisateur du réseau (via Internet, un satellite ou un autre système). L'usager peut adresser des requêtes aux autres nœuds du réseau, précisant le type de données requises et récolter les données environnementales captées par le biais du nœud puits.

Les progrès conjoints de la microélectronique, microtechnique, des technologies de transmission sans fil et des applications logicielles ont permis de produire à coût raisonnable des micro-capteurs de quelques millimètres cubes de volume, susceptibles de fonctionner en réseaux. Il intègrent :

- une unité de captage chargée de capter des grandeurs physiques (chaleur, humidité, vibrations, rayonnement...) et de les transformer en grandeurs numériques,
- une unité de traitement informatique et de stockage de données et un module de transmission sans fil (wireless).

Ces micro-capteurs sont donc de véritables systèmes embarqués. Le déploiement de plusieurs d'entre eux, en vue de collecter et transmettre des données environnementales vers un ou plusieurs points de collecte, d'une manière autonome, forme un réseau de capteurs sans fil.

Historique

Jusque dans les années 1990, hormis pour quelques balises radio, pour acheminer les données d'un capteur au contrôleur central il fallait un câblage coûteux et encombrant.

De nouveaux réseaux de capteurs sont apparus dans les années 1990, notamment dans les domaines de l'environnement et de l'industrie, permis par les récents progrès du domaine des techniques sans-fil (wireless). Aujourd'hui, grâce aux récents progrès des techniques sans-fil, de nouveaux produits exploitant des réseaux de capteurs sans-fil sont employés pour récupérer ces données environnementales.

Enjeux

Pour le magazine Technology Review du MIT, le réseau de capteurs sans fil est l'une des dix nouvelles technologies qui bouleverseront le monde et notre manière de vivre et de travailler. Il répond à l'émergence ces dernières décennies, de l'offre et d'un besoin accru d'observation et de contrôler des phénomènes physiques et biologiques dans différents domaines :

- industriels, techniques et scientifique (monitoring de la température, la pression, l'hygrométrie, la luminosité...).

- écologie et environnement (surveillance des UV, de la radioactivité, de polluants tels que les HAP, les métaux lourds, ou de l'ozone ou du le NO2 ou encore le CO2 et d'autres gaz à effet de serre santé (suivi des malades, veille éco-épidémiologique et épidémiologique,

- sécurité,

- transports (automatisations diverses, prévention des accidents...),

- l'automatisation des bâtiments domotique,

- etc)

Applications

La diminution de taille et de coût des micro-capteurs, l'élargissement de la gamme des types de capteurs disponibles (thermique, optique, vibrations,...) et l'évolution des support de communication sans fil, ont élargi le champ d'application des réseaux de capteurs. Ils s'insèrent notamment dans d'autres systèmes tels que le contrôle et l'automatisation des chaînes de montage. Ils permettent de collecter et de traiter des informations complexes provenant de l'environnement (météorologie, étude des courants, de l'acidification des océans, de la dispersion de polluants, de propagules, etc.

Certains prospectivistes pensent que les réseaux de capteurs pourraient révolutionner la manière même de comprendre et de construire les systèmes physiques complexes, notamment dans les domaines militaire, environnemental, domestique, sanitaire, de la sécurité, etc.

Applications militaires

Comme dans le cas de plusieurs technologies, le domaine militaire a été un moteur initial pour le développement des réseaux de capteurs. Le déploiement rapide, le coût réduit, l'auto-organisation et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui rendent ce type de réseaux un outil appréciable dans un tel domaine. Un réseau de capteurs déployé sur un secteur stratégique ou difficile d'accès, permet par exemple d'y surveiller tous les mouvements (amis ou ennemis), ou d'analyser le terrain avant d'y envoyer des troupes (détection d'agents chimiques, biologiques ou de radiations). Des tests concluants auraient déjà été réalisés par l'armée américaine dans le désert de Californie. Certains exampeles sont les suivants:

- Asset Monitoring: les commandants peuvent surveiller les emplacements des troupes, des armes et des fournitures pour renforcer le contrôle et la communication.
- BattleField Monitoring: les vibrations et les capteurs magnétiques peuvent localiser et suivre des forces ennemies dans le champ de bataille.
- Urban Warfare: capteurs autorisé le déploiement dans les immeubles peuvent empêcher leur réoccupation et de suivre l'activité des ennemis à l'intérieur.
- Protection: prévention et protection contre les radiations, les armes biologiques et chimiques peuvent être atteints par le déploiement d'un WSN, qui détecte le niveau de rayonnement ou de la présence de produits toxiques.
- Distributed Tactical Surveillance: AUVs et capteurs fixes sous-marins peuvent collaborer pour la surveillance des zones de surveillance, de reconnaissance, de ciblage et de systèmes de détection d'intrusion. Par exemple, dans [7], un réseau de capteur 3D sous-marin est conçu pour un système de surveillance tactique qui est capable de détecter et classifier les sous-marins, véhicules de livraison de petite taille (SDVs) et des plongeurs sur la base des données recueillies par rayonnement mécanique, magnétique microcapteurs et acoustique. Avec des systèmes de sonar à l'égard du radar traditionnel, UASNs peut atteindre une précision plus élevée, et permettre la détection et la classification des cibles à faible signature en combinant également des mesures de différents types de capteurs.
- Mine Reconnaissance: Le fonctionnement simultané des AUV multiples avec capteurs acoustiques et optiques peuvent être utilisées pour effectuer l'évaluation rapide de l'environnement et de détecter les mines comme des objets.

Applications à la sécurité

Les structures d'avions, navires, automobiles, métros, etc pourraient être suivies en temps réel par des réseaux de capteurs, de même que les réseaux de circulation ou de distribution de l'énergie. Les altérations de structure d'un bâtiment, d'une route, d'un quai, d'une voie ferrée, d'un pont ou d'un barrage hydroélectrique (suite à un séisme ou au vieillissement) pourraient être détectées par des capteurs préalablement intégrés dans les murs ou dans le béton, sans alimentation électrique ni connexions filaires. Certains capteurs ne s'activant que périodiquement peuvent fonctionner durant des années, voire des décennies. Un réseau de capteurs de mouvements peut constituer un système d'alarme distribué qui servira à détecter les intrusions sur un large secteur. Déconnecter le système ne serait plus aussi simple, puisqu'il n'existe pas de point critique. La surveillance de routes ou voies ferrées pour prévenir des accidents avec des animaux (roadkill) ou des êtres humains ou entre plusieurs véhicules est une des applications envisagées des réseaux de capteurs.

Selon leurs promoteurs, ces réseaux de capteurs pourraient diminuer certaines failles de systèmes de sécurité et mécanismes de sécurisation, tout en diminuant leur coût. D'autres craignent aussi des dérives sécuritaires ou totalitaires si l'usage de ces réseaux n'est pas assujetti à des garanties éthiques sérieuses.

Applications environnementales

Des thermo-capteurs peuvent être dispersés à partir d'avions, ballons, navires et signaler d'éventuels problèmes environnementaux dans le champ de captage (incendie, pollution, épidémies, aléa météorologique...) permettant d'améliorer la connaissance de l'environnement et l'efficacité des moyens de lutte. Des capteurs pourraient être semés avec les graines par les agriculteurs afin de détecter le stress hydrique des plantes ou le taux de nutriment de l'eau du sol, pour optimiser les apports d'eau et de nutriments ou le drainage et l'irrigation. Sur les sites industriels, les centrales nucléaires ou dans les pétroliers, des capteurs peuvent être déployés en réseau pour détecter des fuites de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.) et alerter les utilisateurs et secours plus rapidement, pour permettre une intervention efficace. Une grande quantité de micro-capteurs pourrait être déployée en forêt ou dans certaines aires protégées pour recueillir des informations sur l'état des habitats naturels et sur les comportements de la faune, de la flore et de la fonge (déplacements, activité, état de santé..). L'université de Pise (Italie) a ainsi réalisé des réseaux de capteurs pour le contrôle de parcs naturels (feux, animaux,..). Des capteurs avalés par les animaux ou placés sous leur peau sont déjà parfois utilisés). Il devient ainsi possible "d'observer la biodivesité", sans déranger, des espèces animales vulnérables au dérangement ou difficiles à étudier dans leur environnement naturel, et de proposer des solutions plus efficaces pour la conservation de la faune.

Les éventuelles conséquences de la dispersion en masse des micro-capteurs dans l'environnement ont soulevé plusieurs inquiétudes. En effet, ceux-ci sont généralement doté d'une micro-batterie contenant des métaux nocifs. Néanmoins, le déploiement d'un million de capteurs de 1 millimètre cube chacun ne représente qu'un volume total d'un litre. Même si tout ce volume était constitué de batteries, cela n'aurait pas des répercussions désastreuses sur l'environnement. Certains exampeles sont les suivants:

- Surveillance de L'habitat: un WSN déployé dans un environnement sous-glaciaire [11, 12] peut recueillir des renseignements sur les calottes glaciaires et les glaciers. Les réseaux de capteurs peuvent également être déployés pour mesurer la population d'oiseaux et autres espèces [13]. Aussi, WSN peut fournir un avertissement d'inondation [14] et surveiller l'érosion côtière [15].

- La Détection des Catastrophes: les incendies de forêt peuvent être détectées et localisées par un densément déployés WSN.

- Les Réseaux D'échantillonnage de L'océan: les réseaux de capteurs et d'AUV, tels que l'Odyssée AUV-classe [16], peut effectuer l'échantillonnage, synoptique coopérative adaptative de l'environnement 3D océan côtier [17]. Des expériences telles que l'expérience sur le terrain de Monterey Bay [18] a démontré les avantages de réunir les nouveaux véhicules sophistiqués de robotique avec les modèles d'océan de pointe pour améliorer la capacité d'observer et de prévoir les caractéristiques de l'environnement océanique.

- Surveillance de L'environnement: UASNs peut effectuer la surveillance de la pollution (chimique, biologique et nucléaire). Par exemple, elle mai être possible de détailler le lisier chimique des antibiotiques, des hormones de type œstrogène et d'insecticides pour contrôler les ruisseaux, les rivières, les lacs et les baies de l'océan (qualité de l'eau une analyse in-situ) [19]. Surveillance des courants océaniques et les vents, les prévisions météo s'était améliorée, la détection des changements climatiques, comprendre et prévoir l'effet des activités humaines sur les écosystèmes marins, la surveillance biologique comme le repérage des poissons ou des micro-organismes, sont d'autres applications possibles. Par exemple, dans [20], la conception et la construction d'un réseau simple détection sous-marine est décrit de détecter des gradients de température extrêmes (thermocline), qui sont considérés comme un terrain fertile pour certains micro-organismes marins.

- Les Explorations Sous-Marine: UASNs peut aider à détecter les gisements sous-marins ou de réservoirs, déterminer des routes pour la pose de câbles sous-marins, et aider à l'exploration des minéraux précieux.

- Prévention des Catastrophes: les réseaux de capteurs qui mesurent l'activité sismique dans les endroits éloignés peuvent fournir des alertes aux tsunamis dans les zones côtières [21], ou d'étudier les effets des tremblements de terre sous-marin (compris sous la mer).

- Détection D'incendie au Forêt: un SANET pourraient être déployés pour détecter un incendie de forêt à ses débuts [22]. Un certain nombre de nœuds doivent être pré-déployée dans une forêt. Chaque noeud peut rassembler les différents types d'informations provenant de capteurs, comme la température, l'humidité, de pression et de position. Toutes les données de détection est envoyé par la communication multi-hop au centre de contrôle via un nombre d'actionneurs (dispositifs de passerelle) répartis à travers la forêt. Les actionneurs sont connectés à des réseaux mobiles (par exemple, Universal Mobile Telecommunications System - UMTS) et sera positionné de manière à réduire le nombre de sauts de la source de détection d'incendie au centre de contrôle. Les actionneurs permettra également de réduire la congestion du réseau dans les déploiements à grande échelle par l'extraction des données depuis le réseau en des points prédéterminés. Il mai également être possible dans ce scénario que certains patrons d'unités mobiles forêts agissent comme actionneur mobile, la collecte de données sur l'environnement sur tout leur parcours à travers la forêt. Dès qu'un événement lié à l'incendie est détecté, comme l'élévation de température brusques, le centre de contrôle sera aussitôt très effrayée. Les opérateurs dans le centre de contrôle peut juger s'il s'agit d'une fausse alarme en utilisant soit les données recueillies à partir d'autres capteurs ou en envoyant une équipe pour vérifier la situation sur place. Ensuite, les pompiers et les hélicoptères peuvent être envoyées pour éteindre l'incendie avant qu'il ne croît à un incendie de forêt majeurs.

Applications médicales et vétérinaire

La surveillance des fonctions vitales d'un organisme vivant pourrait à l'avenir être facilitée par des micro-capteurs avalés ou implantés sous la peau. Des gellules multi-capteurs ou des micro-caméras pouvant être avalées existent déjà, pouvant sans recours à la chirurgie, transmettre des images de l'intérieur d'un corps humain (avec une autonomie de 24 heures). Une récente étude présente des capteurs fonctionnant dans le corps humain, qui pourraient traiter certaines maladies. Un projet est de créer une rétine artificielle composée de 100 micro-capteurs pour corriger la vue. D'autres ambitieuses applications biomédicales sont aussi présentées, tel que : la surveillance de la glycémie, la surveillance des organes vitaux ou la détection précoce de cancers. Des réseaux de capteurs permettraient théoriquement une surveillance permanente des patients et une possibilité de collecter des informations physiologiques de meilleure qualité, facilitant ainsi le diagnostic de quelques maladies.

- Surveillance à domicile: surveillance à domicile pour les malades chroniques et personnes âgées [23] permet de soins de longue durée et peut réduire la durée du séjour hospitalier.

- Surveillance des patients: nœuds de capteurs déployés sur le corps des patients dans les hôpitaux [24], permettent la collecte des données à caractère périodique ou continue comme la température, la pression artérielle, etc.

Applications commerciales

Des nœuds capteurs pourraient améliorer le processus de stockage et de livraison (pour garantir la chaine du froid en particulier). Le réseau ainsi formé, pourra être utilisé pour connaître la position, l'état et la direction d'un paquet ou d'une cargaison. Un client attendant un paquet peut alors avoir un avis de livraison en temps réel et connaître la position du paquet. Des entreprises manufacturières, via des réseaux de capteurs pourraient suivre le procédé de production à partir des matières premières jusqu'au produit final livré. Grâce aux réseaux de capteurs, les entreprises pourraient offrir une meilleure qualité de service tout en réduisant leurs coûts. Les produits en fin de vie pourraient être mieux démontés et recyclés ou réutilisés si les microcapteurs en garantissent le bon état. Dans les immeubles, le système domotique de chauffage et climatisation, d'éclairage ou de distribution d'eau pourrait optimiser son efficience grâce à des micro-capteurs présents dans des tuiles aux plancher en passant par les murs, huisseries et meubles. Les systèmes ne fonctionneraient que là où il faut, quand il faut et à la juste mesure. Utilisée à grande échelle, une telle application permettrait de réduire la demande mondiale en énergie et indirectement les émissions de gaz à effet de serre. Rien qu'aux États-Unis, cette économie est estimée à 55 milliards de dollars par an, avec une diminution de 35 millions de tonnes des émissions de carbone dans l'air. Le monde économique pourrait ainsi diminuer ses impacts environnementaux sur le climat. Certains exampeles sont les suivantes:

- Surveillance: un réseau de capteur peut détecter les incendies dans les immeubles et donner des informations sur sa localisation. Il peut également détecter des intrusions et du suivi de l'activité humaine.

- La Prévention des Catastrophes: nœuds de capteurs déployés dans le cadre de l'eau peut empêcher de catastrophe comme le tsunami du tremblement de terre océaniques ou imminente.

- Smart Metering Solutions: Smart Metering Solutions, fourni par Coronis, basé sur Wavenis [8] la technologie sans fil ont été déployées dans des millions d'installations résidentielles, industrielles et commerciales autour du monde, reliant les consommateurs de gaz, de l'eau et d'électricité de manière efficace avec l'opérateur est de retour l'information et de systèmes de facturation. Wavenis technologie sans fil fournit l'ultra-longue portée et de la consommation d'énergie extrêmement basse qui sont essentiels pour l'efficacité mile dernière, la couverture extérieure en matière de comptage des réseaux qui servent des villes entières, y compris les zones urbaines denses ainsi que tentaculaire des zones suburbaines et commerciale.

- Navigation Assistée: les capteurs peuvent être utilisés pour identifier les dangers sur le fond marin, de localiser les bancs de roches dangereuses dans les eaux peu profondes, des postes d'amarrage, submergé épaves, et d'effectuer le profilage bathymétrie.

- Reprise après Sinistre: après un séisme ou une attaque terroriste, les nœuds de capteur peut détecter des signes de vie à l'intérieur d'un bâtiment endommagé.

- Smart Park: un système de contrôle distribué soutenu par SANET. Elle améliore la mobilité dans la zone urbaine en trouvant des places de parking gratuits pour les conducteurs désireux de se garer [9, 10]. Elle diminue également le risque d'éventuels accidents, la pollution, et d'éliminer la rage au volant.

Plates-formes

Parmi les standards les plus aptes à être exploités dans les réseaux de capteurs sans-fil se retrouvent la double pile protocolaire Bluetooth / Zigbee.

- La Bluetooth, dont Ericsson a initié le projet en 1994, a été standardisé sous la norme IEEE 802.4.15 et a comme but la création et le maintien de réseaux à portée personnelle, PAN (Personal Area Network). Un tel réseau est utilisé pour le transfert de données à bas débit à faible distance entre appareils compatibles. Malheureusement, le grand défaut de cette technique est sa trop grande consommation d'énergie et ne peut donc pas être utilisée par des capteurs qui sont alimentés par une batterie et qui idéalement devraient fonctionner durant plusieurs années.

- Le ZigBee combiné avec IEEE 802.15.4 offre des caractéristiques répondeant encore mieux aux besoins des réseaux de capteurs en termes d'économies d'énergie. ZigBee offre des débits de données moindres, mais il consomme également nettement moins que Bluetooth. Un faible débit de données n'handicape pas pour un réseau de capteurs où les fréquences de transmission sont faibles.

Les constructeurs tendent à employer des « techniques propriétaires » ayant l'avantage d'être spécifiquement optimisées pour une utilisation précise, mais avec l'inconvénient de ne pas être compatibles entre elles.

Matérielles et Logiciels

De nouvelles techniques influenceront l'avenir des réseaux de capteurs. par exemple, UWB (Ultra wideband) est une technique de transmission permettant des consommations extrêmement basses grâce à sa simplicité matérielle. De plus, l'atténuation du signal engendré par des obstacles est moindre qu'avec les systèmes radio à bande étroite conventionnels.

Le domaine des capteurs sans fil semble promis à un grand essor. De nombreux nouveaux produits logiciels sont attendus, y compris dans le domaine de l'open-source avec par exemple TinyOS développé à l'Université de Berkeley ; un système d'exploitation "open source" conçu pour les capteurs embarqués sans-fil qui est déjà utilisé (en 2009) par plus de 500 universités et centres de recherche dans le monde. La réalisation de programmes sur cette plateforme s'effectue exclusivement en NesC (dialecte du C). Cet OS a notamment pour particularité une taille extrêmement réduite en termes de mémoire (quelques kilo-octets).

Architecture d'un micro-capteur

Un « nœud capteur » contient quatre unités de base : l'unité de captage, l'unité de traitement, l'unité de transmission, et l'unité de contrôle d'énergie. Selon le domaine d'application, il peut aussi contenir des modules supplémentaires tels qu'un système de localisation (GPS), ou bien un système générateur d'énergie (cellule solaire). Quelques micro-capteurs, plus volumineux, sont dotés d'un système mobilisateur chargé de les déplacer en cas de nécessité.

L'unité de captage

Le capteur est généralement composée de deux sous-unités : le récepteur (reconnaissant l'analyte) et le transducteur (convertissant le signal du récepteur en signal électrique). Le capteur fournit des signaux analogiques, basés sur le phénomène observé, au convertisseur Analogique/Numérique. Ce dernier transforme ces signaux en un signal numérique compréhensible par l'unité de traitement.

L'unité de traitement

Elle comprend un processeur généralement associé à une petite unité de stockage. Elle fonctionne à l'aide d'un système d'exploitation spécialement conçu pour les micro-capteurs (TinyOS par exemple). Elle exécute les protocoles de communications qui permettent de faire « collaborer » le nœud avec les autres nœuds du réseau. Elle peut aussi analyser les données captées pour alléger la tâche du nœud puits.

L'unité de transmission

Elle effectue toutes les émissions et réceptions des données sur un medium « sans-fil ». Elle peut être de type optique (comme dans les nœuds Smart Dust), ou de type radio-fréquence.

- Les communications de type optique sont robustes vis-à-vis des interférences électriques. Néanmoins, ne pouvant pas établir de liaisons à travers des obstacles, elles présentent l'inconvénient d'exiger une ligne de vue permanente entre les entités communicantes.

- Les unités de transmission de type radio-fréquence comprennent des circuits de modulation, démodulation, filtrage et multiplexage ; ceci implique une augmentation de la complexité et du coût de production du micro-capteur.

Concevoir des unités de transmission de type radio-fréquence avec une faible consommation d'énergie est un défi car pour qu'un nœud ait une portée de communication suffisamment grande, il est nécessaire d'utiliser un signal assez puissant et donc une énergie consommée importante. L'alternative consistant à utiliser de longues antennes n'est pas possible à cause de la taille réduite des micro-capteurs.

L'unité de contrôle d'énergie

Un micro-capteur est muni d'une ressource énergétique (généralement une batterie). Étant donné sa petite taille, cette ressource énergétique est limitée et généralement non-remplaçable. ceci fait souvent de l'énergie la ressource la plus précieuse d'un réseau de capteurs, car elle influe directement sur la durée de vie des micro-capteurs et donc du réseau entier. L'unité de contrôle d'énergie constitue est donc un systèmes essentiel. Elle doit répartir l'énergie disponible aux autres modules, de manière optimale (par exemple en réduisant les dépenses inutiles et en mettant en veille les composants inactifs). Cette unité peut aussi gérer des systèmes de rechargement d'énergie à partir de l'environnement via des cellules photovoltaïque par exemple.

Architectures Réseau de capteurs

Il existe plusieurs topologies pour les réseaux à communication radio. Nous discutons ci-dessous des topologies applicables aux réseaux de capteurs.

La Topologie en étoile

Dans cette topologie une station de base envoie ou reçoit un message via un certaine nombre de nœuds. Ces nœuds peuvent seulement envoyer ou recevoir un message de l'unique station de base, il ne leur est pas permis de s'échanger des messages.

- *Avantage*: simplicité et faible consommation d'énergie des nœuds, moindre latence de communication entre les nœuds et la station de base.
- *Inconvénient*: la station de base est vulnérable, car tout le réseau est géré par un seul nœud.

La topologie « en toile » ou « en grille » (Mesh Network)

Dans ce cas (dit « communication multi-sauts »), tout nœud peut échanger avec n'importe quel autre nœud du réseau (s'il est à portée de transmission). Un nœud voulant transmettre un message à un autre nœud hors de sa portée de transmission, peut utiliser un nœud intermédiaire pour envoyer son message au nœud destinataire. Avantage : Possibilité de passer à l'échelle du réseau, avec redondance et tolérance aux fautes, Inconvénient : Une consommation d'énergie plus importante est induite par la communication multi-sauts. Une latence est crée par le passage des messages des nœuds par plusieurs autres avant d'arriver à la station de base.

Un réseau Mesh est le nom des systèmes en réseau embarqués qui partagent plusieurs caractéristiques telles que:

- Multi-Hop—La possibilité d'envoyer des messages de capteur en capteur jusqu'à une station de base, ceci permettant l'extension du réseau en escalade.
- Self-Configuring—Capacité à créer le réseau sans intervention humaine.
- Self-Healing—Capacité d'ajouter et de supprimer des noeuds du réseau automatiquement sans avoir à reconfigurer le réseau.
- Dynamic Routing—Capacité à déterminer de manière adaptative le chemin vers la base dynamiquement en fonction des conditions du réseau. Ces caractéristiques associées à une gestion de la consommation d'énergie permettent aux réseaux de capteurs une grande autonomie, un déploiement facile et une réactivité face à un problème au sein du réseau.

La topologie hybride

Une topologie hybride entre celle en étoile et en grille fournit des communications réseau robustes et diverses, en assurant la minimisation de la consommation d'énergie dans les réseaux de capteurs. Dans ce type de topologie, les nœuds capteur autonome en énergie ne routent pas les messages, mais il y a d'autres nœuds qui ont la possibilité de faire le routage des messages. En général, ces nœuds disposent d'une source d'énergie externe.

Motivations et Objectifs

Les réseaux de capteurs sont similaires à des réseaux ad hoc en ce sens que les réseaux de capteurs emprunter massivement sur l'auto-organisation et les technologies de routage développé par le groupe ad hoc de la recherche. Toutefois, un objectif majeur pour la conception des réseaux de capteurs est de réduire le coût de chaque nœud. Pour de nombreuses applications, le coût souhaité pour une connexion sans fil périphérique est inférieur à un dollar.

Nous, dans cette thèse, prenons un ensemble de capteurs répartis sur une région pour effectuer un opération. Chacun de ces capteurs a un émetteur-récepteur sans fil qui transmet et reçoit sur une seule fréquence, qui est commun à tous ces capteurs. Au fil du temps, certains de ces capteurs générer/collecter des informations pour être envoyé à un autre capteur. En raison de la capacité de la batterie limitée de ces capteurs, un capteur ne peut pas communiquer directement avec les nœuds qui sont loins. Dans de tels scénarios, l'une des possibilités de transfert d'informations entre deux noeuds qui ne peuvent pas communiquer directement est d'utiliser d'autres nœuds de capteurs dans le réseau. Pour être précis, les capteurs source transmet ses informations à l'un des capteurs qui est à sa portée de transmission. Le capteur intermédiaire utilise ensuite la même procédure afin que l'information a finalement atteint sa destination.

Un ensemble composé de la paire ordonnée de noeuds constituent une liaison qui est utilisé pour faciliter la communication entre les deux paires donné de nœuds (par exemple, un capteur et une station de base). Il s'agit d'un problème standard de routage multi-sauts dans les réseaux de capteurs. Le problème de routage optimal a été largement étudié dans le cadre des réseaux filaires où habituellement plus court chemin (algorithme de routage) est utilisé: chaque maillon du réseau a un poids qui lui est associée et l'objectif de l'algorithme de routage est de trouver un chemin qui permet d'atteindre le poids minimal entre deux noeuds donnée. De toute évidence, le résultat d'un tel algorithme dépend de la fonction de poids associée à chaque lien dans le réseau. Dans les réseaux de capteurs, l'optimalité de l'algorithme de routage est défini à prolonger la vie de réseau (là où la vie est défini comme le temps engendré par le réseau pour certains agrégation des données jusqu'à ce nœud vivant en premier est déconnecté suite à une coupure de l'énergie). Toutefois, une compréhension complète de l'effet de routage sur les performances de réseaux de capteurs et de l'utilisation des ressources (en particulier, la stabilité en couche MAC et, par conséquent, le retard bout-en-bout) n'a pas reçu beaucoup d'attention.

Pour SANETs, nous nous concentrerons sur deux niveaux de coordination les plus restreintes à savoir: capteur-actionneur, et l'actionneur-actionneur. En SANETs, pour capteur-actionneur de coordination, il est nécessaire d'élaborer des protocoles qui sont en mesure de fournir des services en temps réel avec des bornes retard donné, en fonction des contraintes d'application et d'assurer une communication efficace de l'énergie entre les capteurs et actionneurs. Les actionneurs peuvent communiquer les uns avec les autres, en plus de communiquer avec les capteurs. Comme il existe peu de nombre de nœuds d'actionneur et les capacités de puissance de ces nœuds sont plus élevés que les nœuds de capteurs, communication est similaire à la communication en réseau sans fil ad-hoc. Nous considérons également un SANET qui prolonge la durée de vie du réseau en minimisant la consommation d'énergie et, parallèlement, prend soin de retard-sensibilité des données captées.

En dehors de SANETs, nous considérons également UASNs qui sont déployés pour accomplir les tâches de surveillance de collaboration sous-marine. Les capteurs doivent être organisés en un réseau autonome qui s'auto-configurer en fonction des caractéristiques variables de l'environnement océanique. La plupart des déficiences du canal acoustique sous-marine sont adéquatement traitées au niveau physique, par la conception de récepteurs qui sont en mesure de faire face à des taux peu élevé d'erreur, la décoloration, et les interférences inter-symboles (ISI) causée par trajets multiples. Il ya eu des efforts à développer des égaliseurs de canaux et d'adaptation des techniques de traitement spatial de sorte que la modulation de phase cohérente peut être utilisé pour atteindre la haute efficacité spectrale souhaitée. Ces techniques sont exigeants en calcul avec les réglages de paramètres multiples et des exigences qui ne sont pas particulièrement adaptés aux applications où l'autonomie, l'adaptabilité et de longue vie de batterie d'opération sont envisagées. Par conséquent, nous analysons les facteurs influant sur les communications acoustiques dans le but de préciser les défis posés par les chaînes sous-marines pour les réseaux de capteurs sous-marins.

Contribution

Le chapitre 'Introduction' présente de manière générale le domaine des réseaux de capteurs sans-fils et les applications possibles. Il présente aussi rapidement les contributions ainsi que le plan du document de thèse. Une différence est ainsi faite pour les réseaux de capteurs sans-fils (wireless sensor networks, WSN), les réseaux de capteurs-actionneurs (sensor-actuator networks, SANET), et réseaux de capteurs sous-marine (underwater acoustic sensor networks, UASNs).

Après un premier chapitre introductif qui donne en particulier les hypothèses, les motivations et les objectifs du travail réalisé, le deuxième chapitre présente le problème du routage multi-sauts et de l'échantillonnage (un couplage entre la couche applicative, celle qui effectue la capture des données, et la couche d'accès au support) dans les réseaux de capteurs sans fiI. On aborde d'abord les deux possibilités les plus connues, à savoir un échantillonnage indépendant ou dépendant des possibilités de transmission, et on montre que la deuxième solution est celle qui donne les meilleures performances, i.e., garantit l'échantillonnage et minimise les délais de bout-en-bout. On propose ensuite d'associer deux files à chaque capteurs, dont une de retransmission des messages reçus, et on montre que la stabilité de ces dernières est fonction du routage et des taux d'accès au canal. Enfin, on propose un algorithme de routage optimal qui recherche l' équilibre de Wardrop et fournit le meilleur délai dans de tels réseaux.

Nous considérons un WSN dans lequel les nœuds de capteurs sont des sources du trafic qui doit être transféré dans un mode multi-hop à un centre de traitement commun. Nous considérons: les nœuds de capteurs ont un processus d'échantillonnage indépendants (architecture en couches) de l'opération de transmission, montré dans la Figure 7.3. Ce système est comme le réseau de paquets radio (PRN) pour lesquels l'analyse exacte n'est pas disponible. Nous montrons aussi que la condition de stabilité proposé dans la littérature de PRN n'est pas précis. D'abord, une condition de stabilité correcte pour un tel système est fourni. Puis, nous avons proposé un échantillon stratifié d'échantillonnage des données schéma dans lequel, le capteur de noeuds exemple de nouvelles données seulement quand il a une occasion (architecture inter-couches), de la transmission des données montré dans la Figure 7.3. Pour le couplage (appelé closed-system par opposition à l'open system qui est sans couplage), l'idée générale est que la couche d 'accès au support demande à la couche applicative de lancer une capture lorsqu'elle n'a pas de paquets en attente de transmission. Toutes les applicauons ne peuvent pas fonctlonncr selon ce mode. Ensuite, une étude analytique de la stabilité du système est aussi proposée. On peut également remarquer que ce système donne une meilleure performance en termes de retard et est d'ailleurs favorable à l'analyse.

Le mécanisme de couplaga proposé péut effectivement être vu comme un mécanisme inter-couches. Il peut aussi se ramener à un système classique de contrôle du type maitre-esclave, pour lequel les avantagcs et les inconvénients sont connus et une littérature abondante disponible. L'étude de la stabilité du système est intéressante, avec une phase de modélisation et une phase de résolution avec des méthodes d'évaluation de performance.

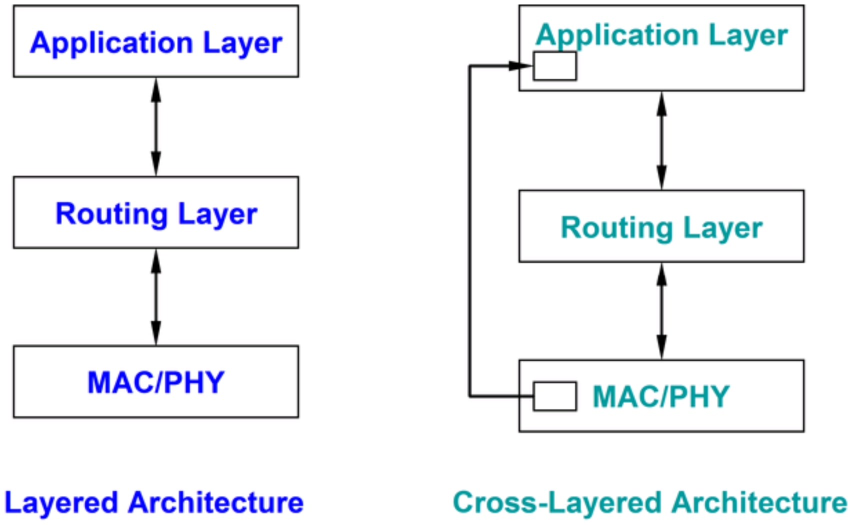

Figure 7.3: Une architecture en couches et inter-couches

De fournir des services intéressants tels que la surveillance d'urgence, réuni en temps réel et les contraintes énergétiques et de la stabilité au contrôle d'accès au support (MAC) couche sont les exigences de base des protocoles de communication dans ces réseaux. Nous proposons également une architecture en couches croisées avec deux files d'émission de la couche MAC, c'est à dire, l'un pour ses propres données générées, et l'autre pour le trafic de transmission montré dans la Figure 7.4. Nous utilisons une discipline probabiliste pour les files d'attente. Notre premier résultat concerne principalement la stabilité des files d'attente à effectuer au niveau des noeuds. Il précise que si les files d'attente de transfert peut être stabilisé, par le choix approprié des files d'attente, ne dépend que de routage et les taux d'accès aux canaux des capteurs. En outre, le poids des WFQs jouer un rôle dans la détermination du compromis entre la puissance allouée pour la communication et le retard de l'acheminement du trafic.

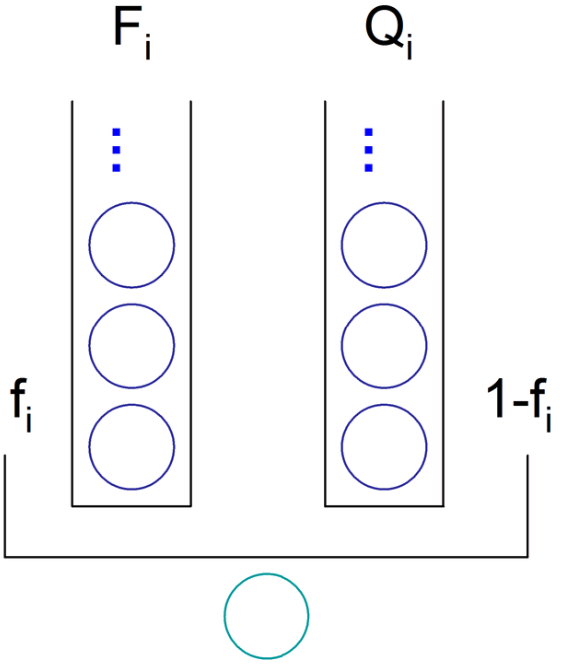

Figure 7.4: Un système avec deux files d'attente au-MAC

En ce qui concerne le routage, on propose un routage minimisant le retard d'acheminemenet en prenant en compte le retard de chaque chemin, dans I'hypothèse où plusieurs chemins sont possibles. Il est supposé que ce retard d'acheminemenet est disponible. De plus, il est connu que des oscillations néfastes peuvent apparaître si le mode des calcul de ces retards ou la sélections des routes optimales est mal adapté. Pour cela, on utilise une méthode qui a été proposée par allieurs pour déterminer les estimations des retards et les probabilités de routage afin de parvenir à un équilibre de type Wardrop. On montre la stabilité par simulâtion, avec des résultats conséquents.

Nous abordons ensuite le problème du routage optimal qui vise à minimiser le retard bout-en-bout. Depuis, nous nous permettre le partage de trafic à des noeuds sources, nous proposons un algorithme qui cherche l'équilibre de Wardrop au lieu d'un seul chemin avec le retard minimum. Wardrop equilibria première apparition dans le contexte des réseaux de transport. Le premier principe de Wardrop dire: Le temps de trajet en toutes les voies utilisées en fait sont égaux et inférieurs à ceux qui seraient subis par un seul véhicule, sur toutes routes non utilisés. Chaque utilisateur non-coopérative cherche à minimiser son coût de transport.

Les flux de circulation qui satisfont à ce principe sont généralement qualifiés de 'l'équilibre utilisateur' (UE) flux, puisque chaque utilisateur de choisir l'itinéraire qui est le meilleur. Plus précisément, un utilisateur-optimisé l'équilibre est atteint lorsque aucun utilisateur peut abaisser son coût de transport par une action unilatérale.

Le régime de routage distribué est conçue pour une large classe de réseaux de capteurs qui converge (au sens de Cesaro) à l'ensemble de Cesaro-Wardrop equilibria. Chaque lien est attribué un poids et l'objectif est d'acheminer à travers les sentiers de poids minimum en utilisant itérative régime de mise à jour. La convergence est établie en utilisant les résultats standard de la littérature connexe et validée par les résultats des simulations TinyOS. Notre algorithme peut s'adapter à l'évolution du trafic sur le réseau et les retards. Ce régime est fondé sur le temps de multiples rapprochement échelle algorithmes stochastiques. L'algorithme est simulé dans TOSSIM et des résultats numérique de ces simulations sont fournis.

Pour les réseaux de capteurs à l'accès au canal aléatoire, nous avons proposé une approche d'échantillonnage des données qui vous garantit un taux d'échantillonnage tout en minimisant le retard bout-en-bout. De simulation et des résultats numériques montrent que les performances de ce système est mieux que l'architecture traditionnelle, où le mécanisme d'accès canal est indépendante d'échantillonnage. Nous avons également vu que le régime proposé ne nécessite pas de paramètre de réglage fastidieux comme c'est le cas pour l'architecture en couches.

Nous avons également obtenu quelques aperçus importants sur des arbitrages différents qui peuvent être obtenus en faisant varier certains paramètres du réseau. Certains d'entre eux comprennent: 1) Le routage peut être crucial dans la détermination des propriétés de stabilité des capteurs en réseau. 2) Que ce soit ou non les files d'attente de transfert peut être stabilisé (par le choix approprié des poids WFQ) ne dépend que de routage et les taux d'accès au canal 3) Nous avons vu également que le retard bout-en-bout est indépendante du choix des poids WFQ.

Nous avons alors proposé un algorithme d'apprentissage, applicable à tous les deux systèmes, d'atteindre l'équilibre de Wardrop pour le retard bout-en-bout. Pour le système fermé, cet algorithme a aussi adapté le taux d'accès au canal des nœuds de capteurs. À partir des résultats de simulation, nous avons vu un délai très élevé pour un système unique file d'attente (à condition que le système était stable) par rapport au système avec deux files d'attente.

Depuis, l'objectif de l'algorithme n'a été que de converger vers un équilibre de Wardrop, à ce moment ce n'est pas en mesure de faire un choix judicieux parmi les équilibres de Wardrop multiples, si elles existent.

Le troisième chapitre étend le précédent en considérant des réseaux de capteurs et d'actionneurs, les actionneurs devant tous recevoir leurs informations en temps minimum. Ce chapitre donne d'abord une architecture cross-layer fermée, en opposition aux architectures ouvertes de l'internet, et montre que cette proposition est plus performante et converge plus vite que la solution classique (hiérarchique) ouverte, ceci à la fois pour la stabilité des files d'attente et pour la minimisation des délais de bout-en-bout. En considérant une architecture de communication à deux niveaux, celui des capteurs aux actionneurs, et celui des actionneurs entre eux, on conduit une étude de conception complète et propose en particulier successivement un nouvel algorithme permettant à chaque capteur de choisir son actionneur de façon optimale et un nouvel algorithme de routage (de tous les flux des capteurs vers leurs actionneurs respectifs), optimal aussi, qui est adaptatif et converge vers un équilibre de Cesaro-Wardrop.

Le chapitre3 présente des contributions pour les réseaux de capteurs-actionneurs (SANETs). On énonce que ces systèmes engendrent des interactions plus variées: capteur-capteur, capteur-actionneur, actionneur-actionneur. Dans le cadre de l'interaction capteur-actionneur, on propose ensuite un mécanisme pour la sélection d'un actionneur pour un ensemble des capteurs donnés. En ce qui conccrne les interactions actionneur-actionneur, on propose une méthode basée sur le contrôle de la puissance d'émission.

Nous considérons un SANET et résoudre le problème du retard minimum pour l'agrégation des données. Nous analysons le moyen retard bout-en-bout dans le réseau. L'objectif est de minimiser le retard total sur le réseau. Nous prouvons que cette fonction objectif est strictement convexe pour l'ensemble du réseau. Nous proposons ensuite un cadre d'optimisation distribuée à atteindre l'objectif requis. L'approche est basée sur l'optimisation convexe et algorithme déterministe distribué sans contrôle rétroactif. Seul le savoir local est utilisé pour mettre à jour les mesures algorithmiques. Plus précisément, nous formulons l'objectif comme une fonction de réduction des retards au niveau du réseau où les contraintes sont la capacité du reception et le taux de service aux couche MAC. En utilisant la méthode lagrangienne, nous déduisons un algorithme primal-dual distribué pour minimiser le retard dans le réseau. Nous développons en outre un retard stochastique primal-dual contrôle algorithme en présence d'un environnement bruyant. Nous présentons également la convergence et la vitesse de convergence associée.

Ce chapitre étudie également un problème de sélection d'actionneur avec le retard optimal pour SANETs. Chaque capteur doit transmettre ses données locales à un seul des actionneurs. Un algorithme en temps polynomial est proposé pour le sélection d'actionneur. Nous proposons enfin un mécanisme distribué pour le contrôle d'actionnement qui couvre toutes les exigences pour un processus de commande efficace.

Pour les réseaux de capteurs-actionneurs sans fil avec accès au canal aléatoire, nous proposons que chaque capteur doit transmettre ses lectures vers un actionneur. L'objectif pour le système ouvert est de minimiser le retard total du réseau. En particulier, nous avons montré que la fonction objectif est strictement convexe pour l'ensemble du réseau. Les résultats montrent que la valeur optimale du taux de service requis est atteint pour chaque noeud du réseau par le primal distribués algorithme dual. Il est important d'accorder une attention égale à la fois le retard observé dans le réseau et la consommation d'énergie pour les transmissions de données. Une convergence rapide signifie que seul un petit supplément d'énergie est consommée pour effectuer des calculs locaux pour atteindre les optimisations souhaitées. Seule l'efficacité énergétique de routage peut pas servir un objectif, pour certaines applications de réseaux de capteurs. De même pour l'algorithme stochastique, nous avons montré une probabilité égale à un convergence et son taux de convergence.

Nous proposons enfin un mécanisme de contrôle distribué de commande destinés à SANETs qui est responsable d'un processus de commande efficace. Les actionneurs peuvent dynamiquement coordonner et effectuer le contrôle de puissance pour maintenir un niveau défini de la connectivité sous réserve de contraintes de débit. Les frais généraux de contrôle des scénarios actionneur statique et mobile est analysée à l'aide des simulations ns-2. L'algorithme PC heuristique est applicable à SANETs multihop pour augmenter le débit, la vie de la batterie et de connectivité.

À l'avenir, nous présenterons une étude basée sur une simulation détaillée de PC algorithme heuristique dans les scénarios de mise en réseau différents avec quelques actionnement. Nous allons également travailler sur le développement de PC algorithme heuristique pour améliorer certains indicateurs de performance couche MAC utilisant une approche multi-couche. Comme conséquence d'une convergence très rapide à l'équilibre de Wardrop, nous sommes également tenté d'effectuer l'analyse des études proposé et du programme d'itinéraire dans le cadre de vie du réseau.

Dans le quatrième chapitre, on présente de nouvelles extensions de nos algorithmes nous permettant d'optimiser à la fois les délais et la consommation d'énergie. On étudie en particulier une architecture du système dans laquelle les capteurs sont organisés en clusters autour des actionneurs, et étend alors les propositions précédentes sur le choix du meilleur actionneur, et celui du routage optimal, et de plus propose un nouveau protocole de niveau MAC qui complémente les deux précédents afin de minimiser encore le couple délai-énergie. L'ensemble $<$ choix des actionneurs, routage optimal et protocole MAC adapté $>$ constitue ainsi une nouvelle architecture, originale, qui est de plus évaluée, et conduit à une solution quasi-optimale pour I'augmentation la durée de vie du système.

Ce chapitre présente l'architectur LEAD (low-energy, adaptive and distributed) pour exploiter des réseaux de capteurs présentant un grand nombre de noeuds. Ce travail comporte en fait plusieurs contributions à plusieurs niveaux: organisation, routage, couche d'accès MAC et PHY. Globalement LEAD regroupe sous une architecture commune des contributions origlnales et complémentaires. On peut citer un protocole de découverte d 'actionneurs, un mécanisme de réveil pour économiser l'énergie (LEAD-Wakeup) et une méthode de transmission.

Nous considérons un trois niveaux SANET et de présenter la conception, la mise en œuvre et l'évaluation des performances d'un protocol (LEAD) cadre d'auto-organisation. Ce cadre assure la coordination, le routage et protocoles de la couche MAC d'organisation en réseau et de gestion. Le cadre est illustré à la Figure 7.5. Nous organisons la SANET en groupes hétérogènes où chaque grappe est géré par un actionneur. Pour maximiser la durée de vie du réseau et d'atteindre retard minimale bout-en-bout, il est essentiel pour une adaptation optimale à chaque nœud de capteurs à un actionneur et de trouver un schéma de routage optimal. Nous fournissons un protocole de découverte du récepteur (ADP) qui découvre un actionneur de destination pour chaque capteur dans le réseau basé sur le résultat d'une fonction de coût. En outre, une fois que la destination des actionneurs sont fixes, nous fournissons une solution de routage optimal dans le but de maximiser la durée de vie du réseau. Nous proposons alors un TDMA MAC protocole en conformité avec l'algorithme de routage. L'actionneur-sélection, le routage optimal, et TDMA MAC régimes de garanties ainsi une durée de vie quasi-optimale. La proposition est validée par des moyens d'analyse et de NS-2 résultats de simulation.

Les contraintes énergétiques et du retard ont un impact significatif sur la conception et le fonctionnement de SANETs. En outre, la prévention de nœuds de capteurs d'être inactif/isolé est très critique. Le problème de l'inactivité du capteur/isolement découle de la pathloss et à la décoloration qui dégrade la qualité des signaux transmis à partir d'actionneurs à des capteurs, en particulier dans les zones de déploiement anisotrope, par exemple, terrains accidentés et montagneux. La transmission des données de capteurs en SANETs repose largement sur l'information concernant le calendrier que chaque nœud de capteurs reçoit de son actionneur associé. Par conséquent, si le signal contenant des informations d'ordonnancement est parvenue à une puissance très faible en raison des déficiences introduites par le canal sans fil, le nœud de capteurs pourraient être incapables de le décoder et, par conséquent il restera inactif/isolé.

Capteurs transmettent leurs lectures pour les actionneurs. Tous les actionneurs coopérer et conjointement transmettre les informations de programmation à des capteurs à l'utilisation de beamforming. Il en résulte une réduction importante du nombre de capteurs inactifs comparant à la transmission unique pour un niveau donné de puissance d'émission. La réduction

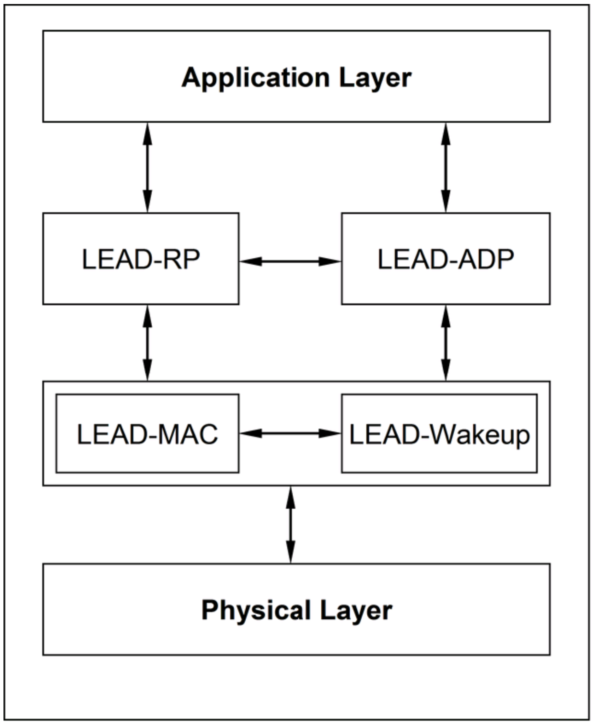

Figure 7.5: The LEAD Framework

est attribuable au gain résultant de tableau et l'exploitation de la macro-diversité qui est fourni par la coopération de l'actionneur. Afin de maximiser la durée de vie du réseau et d'atteindre les retards minimale bout-en-bout, il est essentiel pour une adaptation optimale à chaque nœud de capteurs à un actionneur et de trouver une solution de routage optimal. Une solution distribuée pour la sélection actionneur optimal soumis à des contraintes de retard est également fournie.

Depuis de nombreuses applications nécessitent d'avoir chaque nœud source envoie toutes ses données locales à un seul destination pour le traitement, il est nécessaire de cartographier de façon optimale chaque capteur à son actionneur. Prenant également en considération le fait que le retards bout-en-bout dans les réseaux de capteurs sans fil actionneur est une contrainte dure, nous optimisons conjointement le choix d'actionneur et un débit optimal de routage avec l'objectif global de maximiser la durée de vie du réseau. Nous avons proposé et évalué (à l'aide de ns-2) notre sélection d'actionneur (LEAD-ADP), et le routage (LEAD-RP), et d'un TDMA MAC (LEAD-MAC) protocol. Nous utilisons ensuite la méthode lagrangienne de concevoir un algorithme primal-dual distribués afin de maximiser la durée de vie du réseau. Nous fournissons également un rapport à l'analyse liée. Les résultats des simulations montrent que les performances de cette approche est quasi-optimale et il est pratiquement réalisables par rapport aux précédentes études analytiques basées uniquement sur les évaluations numériques. Dans un avenir proche, nous allons considérer une application réelle SANET et de simuler son comportement avec l'auto-organisation LEAD permettant d'observer ses performances.

Enfin, dans le chapitre cinq, on considère la problématique des réseaux de capteurs sous-marins acoustiques. Ces capteurs sont déployés dans des conditions de transmission très différentes des conditions sans fiI, en effet bien plus contraintes en termes de vitesse et d'erreur.

Ces conditions nous conduisent à considérer maintenant un nouveau niveau physique, et on montre que la technique de conjugaison de phase conduit à un taux d'erreur quasiment nul en utilisant une communication à deux sauts. Cette approche nous permet alors d'utiliser les informations de la couche Ligne afin d'optimiser le routage.

On étudie le cas des capteur sous-marin où le milleu aquatique offre une trés faible bande passante, introduit un très grand taux d'erreurs et des délais accrus. On propose une méthode originale pour exploiter le principe du time reversal et des Gold-sequence dans la modulationd physique du signal. Nous considérons la première analyse d'un schéma de modulation et les algorithmes du récepteur. Cette conception de récepteurs profiter du retournement temporel (TR) et les propriétés du Gold séquences. En outre, ils sont beaucoup moins complexes que les récepteurs utilisant des égaliseurs adaptatifs. Cette technique améliore le rapport signal/bruit (SNR) au niveau du récepteur et réduit le taux d'erreur binaire (BER). On présente des résultats qui montrent que nos propositions permet de réduire le taux d'erreur dans ces environnements très difficiles. L'aspect routage 'inter-couches' est introduit ici de manière indirecte en sélectionnant les liens présentant le plus faible taux d'erreur pour réduire le nombre de retransmissions. Nous avons ensuite appliqué la conjugaison de phase pour la communication réseau. Nous montrons que cette approche peut donner presque zéro BER pour un mode communication en deux sauts par rapport à la communication directe traditionnels. Cette information est utilisée à la couche réseau pour optimiser les décisions de routage. Nous montrons ces améliorations par le biais de l'analyse d'analyse et de simulation.

De plus, dans chaque chapitre, on a soit effectué les simulations nécessaires avec le simulateur NS2, soit réalisé une implémentation des algorithmes en utilisant le système d'exploitation TinyOS. Des résultats numériques sont ainsi donnés pour montrer la qualité et la possibilité d'implémentation de toutes les propositions de cette thèse. Elles montrent clairement les améliorations obtenues avec les propositions faites, et en particulier que la plus évoluée satisfait effectivement les contraintes de garantie et d'énergie demandées.

Dans le chapitre 6, nous présentons un résumé général des travaux réalisés et les conclusions concernant les résultats obtenus lors de cette thèse. Quelques points de vue et des questions ouvertes sont présentées pour la poursuite de ces travaux dans le domaine de la Croix-couche dans les optimisations de capteurs sans fil, capteur-actionneur, et sous l'eau des réseaux de capteurs acoustiques.

Dans un avenir proche, nous allons présenter une étude basée sur une simulation détaillée de PC algorithme heuristique dans les scénarios de mise en réseau différents avec quelques actionnement exigences spécifiques d'application et l'évaluation des pratiques distribués processus en plusieurs actionnement de l'actionneur. Nous allons également travailler sur le développement de PC algorithme heuristique pour améliorer certains indicateurs de performance couche MAC utilisant une approche multi-couche. Comme conséquence d'une convergence très rapide à l'équilibre de Wardrop, nous sommes également tenté d'effectuer l'énergie-analyse des études proposé et du programme d'itinéraire dans le cadre de vie du réseau.

Nous allons considérer une application réelle SANET vie et de simuler son comportement avec l'auto-organisation LEAD cadre permettant d'observer ses performances. Nous prendrons en considération un vérin dynamique scénario de cession de transport de données en temps opportun dans un capteur sans fil de réseau mobile de l'actionneur. Nous allons également envisager une multihop 3D sous l'eau du réseau de capteurs acoustiques avec une application réelle, la vie et étendre les résultats de ce chapitre afin d'optimiser la couche d'ordonnancement (MAC) et la couche de routage dans le but de minimiser la consommation d'énergie par la transmission efficace et en maximisant le réseau durée de vie.

Bibliography

[1] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. IEEE/ACM Transactions on Networking, Vol. 11, pp. 2-16, Feb. 2003.

[2] http://tools.ietf.org/wg/roll/

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci. WSNs: a survey. Article Computer Networks, Vol. 38, No. 4, pp 393-422, March 2002.

[4] Ian F. Akyildiz, D. Pompili, T. Melodia. Underwater acoustic Sensor Networks: research challenges. Ad Hoc Journal, 3(3):257-279, March 2005.

[5] UnderWater Sensor Networks at BWN Laboratory, Georgia Institute of Technology, Available from <http:// www.ece.gatech.edu/research/labs/bwn/UWASN/>.

[6] M. Ilyas and I. Mahgoub. Handbook os WSNs: compact wireless and wired sensing systems, CRC Press LLC, 2005.

[7] E. Cayirci, H. Tezcan, Y. Dogan, V. Coskun. Wireless Sensor Networks for underwater surveillance systems. Ad Hoc Networks, in press; doi:10.1016/j.adhoc.2004.10.008.

[8] http://www.coronis.com

[9] http://www.smartpark.net

[10] SmartPark http://smartpark.epfl.ch

[11] R. B. Alley. Continuity comes first: recent progress in understanding subglacial deformation. in Geological Society Special Publication, 2000, vol. 176, pp. 171–180.

[12] J.K. Hart and J. Rose. Approaches to the study of glacier bed deformation. in Quaternary International, 2001, vol. 86, pp. 45–58.

[13] R. Szewczyk et al.. Lessons from a sensor network expedition. in 1st European Conference on Wireless Sensor Networks (EWSN'04), Berlin, Jan. 2004, pp. 307-322.

[14] http://envisense.org/floodnet.htm

[15] http://envisense.org/secoas.htm

[16] AUV Laboratory at MIT Sea Grant, Available from <http://auvlab.mit.edu/>.

[17] Ocean Engineering at Florida Atlantic University, Available from <http://www.oe.fau.edu/research/ams.html>.

[18] Second field test for the AOSN program, Monterey Bay August 2003, Available from <http://www.mbari.org/ aosn/MontereyBay2003/MontereyBay2003Default.htm>.

[19] X. Yang, K.G. Ong, W.R. Dreschel, K. Zeng, C.S. Mungle, C.A. Grimes. Design of a wireless sensor network for long-term, in-situ monitoring of an aqueous environment. Sensors 2 (2002) 455–472.

[20] B. Zhang, G.S. Sukhatme, A.A. Requicha. Adaptive sampling for marine microorganism monitoring. in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004.

[21] N.N. Soreide, C.E. Woody, S.M. Holt. Overview of ocean based buoys and drifters: Present applications and future needs. in: 16th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, January 2004.

[22] CRUISE website: http://www.ist-cruise.eu

[23] e-SENSE IST project website: http://www.ist-esense.org

[24] C.R. Baker et al. Wireless Sensor Networks for home health care. in 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07), May 2007.

[25] J.-H. Chang and L. Tassiulas. Energy conserving routing in wireless ad hoc networks. In IEEE Infocom Conference Proceedings, Tel Aviv, Israel, Mar. 26-30, 2000, pp. 22-31.

[26] M. Bhardwaj and A. P. Chandrakasan. Bounding the lifetime of WSNs via optimal role assignments. In IEEE Infocom Conference Proceedings, New York, Jun. 23-27, 2002, pp. 1587-1596.

[27] I. F. Akyildiz, I. H. Kasimoglu. Wireless. sensor and actor networks: research challenges. Article Ad Hoc Networks, Vol. 2, No. 4, pp 351-367, October 2004.

[28] J. G. Wardrop. Some theoratical aspects of road traffic research. In Proceedings Inst. Civil Eng., Part 2, 1952, pp. 325-378.

[29] V. S. Borkar and P. R. Kumar. Dynamic Cesaro-Wardrop equilibration in networks. IEEE Transactions on Automatic Control, 48(3):382-296, March 2003.

[30] Razwan Cristescu, Baltasar Beferull-Lozano, and Martin Vetterli, "Networked Slepian-Wolf: Theory, Algorithms, and Scaling Laws," IEEE Transactions on Information Theory, vol. 51, no. 12, pp. 4057-4073, December 2005.

[31] D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources IEEE Transactions on Information Theory, vol. 19, pp. 471-480, July 1973.

[32] A. A. Kherani, R. El Azouzi and E. Altman, "Stability-Throughput Tradeoff and Routing in Multi-hop Wireless Ad-Hoc Networks," Networking 2006 Conference Proceedings, pp. 25-40, Coimbra, Portugal, may 2006.

[33] Jr. R. L. Hamilton and H.-C. Yu, “Optimal routing in multihop packet radio networks,” In Proceedings of IEEE Conference on Computer Communications (INFOCOM), San Francisco, pp. 389-396, June 1990.

[34] LAN-MAN Standards committee of the IEEE Computer Society. 2006. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003).

[35] Kristofer S. J. Pister, Lance Doherty. TSMP: TIME SYNCHRONIZED MESH PROTOCOL. Proceedings of IASTED International Symposium, Distributed Sensor Networks - DSNs 2008, November 16-18, 2008, Orlando, Florida, USA.

[36] Wei Ye, John Heidemann and Deborah Estrin. Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks. IEEE/ACM Transactions on Networking, Vol. 12, No. 3, pp. 493-506, June 2004.

[37] Thomas Watteyne, Abdelmalik Bachir, Mischa Dohler, Dominique Barthel, and Isabelle Augé-Blum. 1-hopMAC: An Energy-Efficient MAC Protocol for Avoiding 1-hop Neighborhood Knowledge. International Workshop on Wireless Ad-hoc and Sensor Networks - IWWAN 2006, June 28-30, 2006, New York, USA.

[38] R. G. Gallagar, “A minimum delay routing algorithm using distributed computation,” IEEE Transactions on Communications, vol. 25, no. 1, pp. 73-85, January 1977.

[39] S. Bandyopadhyay and E. Coyle, "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks," in INFOCOM, vol. 3. IEEE, 30 March-3 April 2003, pp. 1713–1723.

[40] S. Muruganathan, D. Ma, R. Bhasin, and A. Fapojuwo, "A Centralized Energy-Efficient Routing Protocol for Wireless Sensor Networks," IEEE Radio Communications, vol. 43, no. 3, pp. 8–13, March 2005.

[41] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An application specific protocol architecture for wireless microsensor networks. IEEE Trans. Wireless Comm., 1(4), 660-670, Oct. 2002.

[42] S. Lindsey and C Raghavendra. Pegasis: Power-efficient gathering in sensor information systems. in International Conference on Communication Protocols, 2001, pp. 149-155.

[43] O. Younis, M. Krunz, and S. Ramasubramanian, "Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges," IEEE Network, vol. 20, no. 3, pp. 20–25, May/June 2006.

[44] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks," IEEE Transactions on Mobile Computing, vol. 3, no. 4, pp. 366–379, October-December 2004.

[45] A. Boukerche, R. W. Pazzi, and R. Araujo, "Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments," Journal of Parallel and Distributed Computing, vol. 66, no. 4, pp. 586–599, April 2006.

[46] M. Dohler, D. Barthel, F. Maraninchi, L. Mounier, S. Aubert, C. Dugas, A. Buhrig, F. Paugnat, M. Renaudin, A. Duda, M. Heusse, and F. Valois, "The ARESA Project: Facilitating Research, Development and Commercialization of WSNs," in 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), San Diego, CA, USA, June 2007, pp. 590–599.

[47] T. Watteyne, I. Auge-Blum, M. Dohler, and D. Barthel, "Geographic Forwarding in Wireless Sensor Networks with Loose Position-Awareness," in 18th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC). Athens, Greece: IEEE, September 3-7 2007.

[48] T. Watteyne, I. Auge-Blum, M. Dohler, S. Ubda, and D. Barthel, "Centroid Virtual Coordinates - A Novel Near-Shortest Path Routing Paradigm," International Journal of Computer and Telecommunications Networking, Elsevier, Special Issue on Autonomic and Self-Organising Systems, to appear., 2009.

[49] V. S. Borkar. Stochastic approximation with two time scales. System Control Letters, 29:291294, 1996.

[50] V. S. Borkar and S. P. Meyn. The O.D.E method for convergence of stochastic approximation and reinforcement learning. SIAM Journal of Control and Optimization, Vol 38, pp. 447-469, 2000.

[51] V. S. Borkar and D. Manjunath. Charge based control of diffserv-like queues. Automatica, Vol. 40, No. 12, December 2004.

[52] http://www.tinyos.net/

[53] The Network Simulator - ns (version 2.27). http://www.isi.edu/nsnam/ns/.

[54] http://cvs.cens.ucla.edu/emstar/

[55] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for WSNs. In ACM Mobicom Conference Proceedings, pp. 56-67, Boston, MA, USA, 2000.

[56] R. Buche and H. J. Kushner. Rate of convergence for constrained stochastic approximation algorithms. *SIAM Journal on Control and Optimization*, vol. 40, pp. 1011-1041, 2001.

[57] H. Kushner and G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer 2003.

[58] I. F. Akyildiz and I. H. Kasimoglu. Sensor and actor networks: research challenges. Ad Hoc networks (Elsevier), Vol. 2 No. 4, pp. 351-367, October, 2004.

[59] Gkelias, A. Dohler, M. Friderikos, V. Aghvami, A.H. Wireless multi-hop CSMA/CA with cross-optimised PHY/MAC. In Proceedings of Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. 29 Nov. - 3 Dec. 2004, pp. 39 - 43.

[60] R. Ramanathan and R Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In Proceedings of IEEE Infocom Conference, Tel Aviv, Israel, Mar. 26-30, 2000.

[61] J. Czyzyk, M. Mesnier, and J. Mor. The NEOS server. IEEE Journal on Computational Science and Engineering, vol. 5, no. 3, pp. 68–75, July-September 1998.

[62] T. X. Brown, H. N. Gabow, and Q. Zhang. Maximum flow-life curve for a wireless ad hoc network. In ACM MobiHoc Conference Proceedings, Long Beach, CA, Oct. 4-5, 2001, pp. 128-136.

[63] Y. T. Hou, Y. Shi, and H. D. Sherali. Optimal base station selection for anycast routing in wireless sensor networks. IEEE Trans. on Veh. Tech., 55(3), 813-821, May 2006.

[64] H. Zhang and H. Dai. Cochannel Interference Mitigation and Cooperative Processing in Downlink Multicell Multiuser MIMO Networks. In Eurasip Journal on Wireless Comm. and Net., 2004:2, pp. 222-235, 2004.

[65] W. Hu, N. Bulusu, and S. Jha. A Communication Paradigm for Hybrid SANs. International Journal of Wireless Information Networks, Vol. 12, No. 1, pp 47-59, January 2005.

[66] T. Melodia, D. Popili, V. C. Gungor and I. F. Akyildiz. A distributed coordination framework for WSANs. in Proc. of MobiHoc, May 2005.

[67] E. Cayirci, T. Coplu and O. Emirogl. 2005. Power aware many to many routing in wireless sensor and actuator networks. In Proceedings of the Ewsn Conference (Berlin, Germany, Jan. 31-2 Feb.). 236-245.

[68] J. N. Al-Karaki and A. E. Kamal. 2004. Routing techniques in wireless sensor networks: a survey. Article wireless communications. 11. No. 6. December: 6-28.

[69] A. Manjeshwar and D.P. Agarwal. Teen: a routing protocol for enhanced efficiency in wireless sensor networks. In 1st International Workshop on Parallel and Distributed Computing, Issues in Wireless Networks and Mobile Computing, Apr. 2001.

[70] Dust Networks. Technical Overview of Time Synchronized Mesh Protocol, White Paper, http://www.dustnetworks.com/. 2008.

[71] J. Li and G. Lazarou. A bit-map-assisted energy-efficient MAC scheme for wireless sensor networks. In proceedings of IEEE IPSN, 2004.

[72] M. Ali, T. Suleman and Z. Uzmi. MMAC: A Mobility-Adaptive, Collision-Free MAC Protocol for Wireless Sensor Networks. In Proceedings of IEEE IPCCC, 2005.

[73] H. Cao, K. Parker, and A. Arora. O-MAC: A Receiver Centric Power Management Protocol. In Proceedings of IEEE ICNP, Santa Barbara, CA, November 2006.

[74] A. Barroso U. Roedig and C. Sreenan. f-MAC: A Deterministic Media Access Control Protocol Without Time Synchronization. In Proceedings of EWSN, 2006.

[75] J. Son J. Pak and K. Han. A MAC Protocol Using Separate Wakeup Slots for Sensor Network. In Proceedings of ICCSA, pages 1159–68, Glascow, UK, May 2006.

[76] F. N-Abdesselam H. Wang, X. Zhang and A. Khokhar. DPS-MAC: An Asynchronous MAC Protocol for Wireless Sensor Networks. In Proceedings of HIPC, Goa, India, December 2007.

[77] V. Jain, R. Biswas and D. Agrawal. Energy Efficient and Reliable Medium Access for Wireless Sensor Networks. In proceedings of IEEE WoWMoM, Helsinki, Finland, June 2007.

[78] M. Pereira M. Macedo P. Pinto L. Bernardo, R. Oliveira. A Wireless Sensor MAC Protocol for Bursty Data Traffic. In Proceedings of IEEE PIMRC, Athens, Greece, September 2007.

[79] S. Liu, K-W. Fan, and P. Sinha. CMAC: An Energy Efficient MAC Layer Protocol Using Convergent Packet Forwarding for Wireless Sensor Networks. In Proceedings of IEEE SECON, San Diego, CA, June 2007.

[80] G. Halkes and K. Langendoen. Crankshaft: An Energy-Efficient MAC-Protocol For Dense Wireless Sensor Networks. In Proceedings of EWSN, 2007.

[81] AODV: RFC 3561 http://www.ietf.org/rfc/rfc3561.txt

[82] OLSR: RFC 3626, www.ietf.org/rfc/rfc3626.txt

[83] V. P. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, N. Shroff. A Minimum Cost Heterogeneous Sensor Network with a Lifetime Constraint. IEEE Transactions on Mobile Computing, Vol. 4, No. 1, pp. 4-15, January 2005.

[84] G. Xing, C. Lu, Y. Zhang, Q. Huang, and R. Pless. Minimum Power Configuration in WSNs. in Proc. of MobiHoc, May 2005.

[85] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Initializing Newly Deployed Ad Hoc and WSNs. in Proc. of MobiHoc, May 2004.

[86] J. L. Bredin, E. D. Demaine, M. T. Hajiaghayi, and D. Rus. Deploying WSNs with Guaranteed Capacity and Fault Tolerance. in Proc. of MobiHoc, May 2005.

[87] Xiang Ji, and Hongyuan Zha. Sensor Positioning in Ad-hoc WSNs Using Multidimensional Scaling. in Proc. of IEEE Infocom, March 2004.

[88] Yi Shang, and Wheeler Ruml. Improved MDS-Based Localization. in Proc. of IEEE Infocom March 2004.

[89] G. Sarma, and R. Mazumdar. Hybrid WSNs: A Small World. in proc. of MobiHoc, May 2005.

[90] A. Cerpa, J. L. Wong, M. Potkonjak, and D. Estrin. Temporal Properties of Low Power Wireless Links. in Proc. of MobiHoc, May 2005.

[91] I. F. Akyildiz, M. C. Vuran and O. B. Akan. On Exploiting Spatial and Temporal Correlation in WSNs. in Proc. of IWWAN, May 2004.

[92] H. Gupta, V. Navda, S. R. Das, and V. Chowdhary. Efficient Gathering of Correlated Data in WSNs. in Proc. of MobiHoc, May 2005.

[93] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley. Mobility Improves Coverage of WSNs. in Proc. of MobiHoc, May 2005.

[94] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. Network Flows: Theory, Algorithms, and Applications. Englewood Cliffs, New Jersey: Prentice Hall, February 1993.

[95] LAN-MAN Standards committee of the IEEE Computer Society. 1997. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification. IEEE Std 802.11-1997.

[96] T. V. Dam, and K. Langendoen, "An adaptive energy efficient MAC protocol for WSNs", in Proceedings of ACM Sensys Conference, 2003.

[97] J. Polastre, J. Hill and D. Culler, Versatile low power media access for wireless sensor networks, in Proceedings of ACM Sensys Conference, 2004.

[98] I Rhee, A. Warrier, M. Aia and J. Min, Z-MAC: a hybrid MAC for wireless sensor networks, in Proceedings of ACM Sensys Conference, 2005.

[99] W. Ye, J. Heidemann, and D. Estrin. An Energy-Efficient MAC Protocol for Wireless sensor networks. In IEEE Infocom Conference Proceedings, Vol. 3, pp. 1567-1576, Newyork, USA, 2002.

[100] W. Ye, J. Heidemann, and D. Estrin. Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks. IEEE/ACM Trans. on Netw., Vol. 12, No. 3, pp. 493-506, June 2004.

[101] S. Coleri and P. Varaiya, PEDAMACS: Power efficient and delay aware medium access protocol for WSNs. Master Thesis, University of California, Berkeley, CA 94720, USA, 2003.

[102] M. Younis, M. Youssef and K. Arisha, Energy-aware management for cluster-based WSNs, Computer Networks, Vol. 43, No. 5, pp. 649-668, 2003.

[103] S. Coleri and P. Varaiya, PEDAMACS: Power efficient and delay aware medium access protocol for WSNs, IEEE Transactions on Mobile Computing, Vol. 5, No. 7, pp. 920-930, 2006.

[104] E. M. Sozer, M. Stojanovic, and J. G. Proakis. Underwater Acoustic Networks. IEEE Jounral of Oceanic Engineering, Vol. 25, No. 1, pp. 72-83, 2000.

[105] G. F. Edelmann, H. C. Song, S. Kim, W. S. Hodgkiss, W. A. Kuperman, and T. Akal. Underwater Acoustic Communications Using Time Reversal. IEEE Journal of Oceanic Engineering, Vol. 30, No. 4, pp. 852-864, 2005.

[106] M. Fink. Time reversed acoustics. American Institute of Physics, Vol. 20, pp. 3-15, 2001.

[107] T. Folegot. Adaptive instant record signals applied to detection with time reversal operator decomposition. Journal of Acoustical Society of America, Vol. 117, No. 6, pp. 3757-3765, 2005.

[108] H. Song, P. Roux, T. Akal, G. Edelmann, W. Higley, W.S. Hodgkiss, W.A. Kuperman, K. Raghukumar, and M. Stevenson. Time Reversal Ocean Acoustic Experiments At 3.5 kHz: Applications To Active Sonar And Undersea Communications. American Institute of Physics, 2004.

[109] G. F. Edelmann, T. Akal, W. S. Hodgkiss, S. Kim, W. A. Kuperman, H. C. Song, and P. Guerrini. Underwater acoustic communication using time-reversal self-equalization. SACLANT undersea Research Centre, Italy, North Atlantic Treaty Organization, no. SR-341, 2001.

[110] G. F. Edelmann. An overview of time-reversal acoustic communications. 2005. http://www.tica05.org/papers/G_edelman.pdf.

[111] P. Roux, W. A. Kuperman, W. S. Hodgkiss, H. C. Song, and T. Akal. A nonreciprocal implementation of time reversal in the ocean. Acoustical Society of America, Vol. 116, No. 2, pp. 1009-1015, 2004.

[112] P. Hursky, M. B. Porter, J. A. Rice, V. K. McDonald. Passive phase-conjugate signaling using pulse-position modulation. In Proceedings of OCEANS'01, Vol. 4, 2001.

[113] H. M. Kwon and T. G. Birdsall. Digital Waveform Acoustic Codings For Ocean Telemetry. IEEE Journal of Oceanic Engineering, Vol. 16, No. 1, pp. 56-65, 1991.

[114] D. V. Sarwate and M. B. Pursley. Crosscorrelation Properties of Pseudorandom and Related Sequences. In Proceedings of the IEEE, Vol. 68, No. 5, pp. 593-619, 1980.

[115] Anatol Z. TIRKEL. Cross-Correlation of M-Sequences - Some Unusual Coincidences. In Proceedings of ISSTA, 1996.

[116] S-H. Chang, C-H. Weng, J-Y. Chen. Application of Quasi-Orthogonal Sequence in Underwater Acoustic DSSS Communication System. In Proceedings of IEEE, pp. 145-150, 2004.

[117] Ir. J. Meel, Spread Spectum - Introduction, De Nayer Instituut, 1999.

[118] Ir. J. Meel, Spread Spectrum - Applications, De Nayer Instituut, 1999.

[119] J-Y. Chen, S-H Chang. Application of BP Neural Network Based PN Code Acquisition System in Underwater DSSS Acoustic Communication. In Proceedings of IEEE, pp. 627-632, 2002.

[120] D. Pompili, T. Melodia, and I. F. Akyildiz. Routing algorithms for delay-insensitive and delay-sensitive applications in underwater sensor networks. In Proceedings of MobiCom'06, 2006.

[121] L. F. M. Vieira, J. Kong, U. Lee, and M. Gerla. Analysis of Aloha protocols for underwater acoustic sensor networks. Poster at WUWNet'06 Workshop, 2006.

[122] A. Syed, W. Ye, B. Krishnamachari, and J. Heidemann. Understanding Spatio-Temporal uncertainity in medium access with Aloha protocols. In Proceedings of WUWNets'07, 2007.

[123] Robert J. Urick. Principles of Underwater Sound for Engineers. McGraw-Hill Book Company, 1967.

[124] F. H. Fisher and V. P. Simmon. Sound absorption in sea water. Journal of Acoustical Society of America, Vol. 62, No. 3, pp. 558-564, 1977.

[125] R. Jurdak, C. V. Lopes, and P. Baldi. Battery Lifetime Estimation and Optimization for Underwater Sensor Networks. IEEE Sensor Network Operations, 2004.

[126] William C. Lindsey. Error Probabilities for Rician Fading Multichannel Reception of Binary and N-ary Signals. IEEE Transactions on Information Theory, pp. 339-350, October 1964.

[127] Q. Fang, J. Gao, and L. J.Guibas. Locating and Bypassing Routing Holes in WSNs. in proc. of IEEE Infocom March 2004.

[128] J-H. Chang, and L. Tassiulas. Maximum Lifetime Routing in WSNs. IEEE/ACM Transactions on Networking, Vol. 12, No. 4, August 2004.

[129] V. Paruchuri, A. Durresi, and L. Barolli. Energy Aware Routing Protocol for Heterogeneous WSNs. in Proc. of DEXA, August 2005.

[130] E. Yoneki, J. Bacon. A survey of WSN technologies: research trends and middleware's role. Technical Report, No. 646, September 2005.

[131] V. Paruchuri and A. Durresi. 2005. Delay-energy aware routing protocol for sensor and actor networks. In Proceedings of the Icpads Conference (Fukuoka, Japan, Jul. 20-22). 292-198.

[132] J. M. Kahn, R.H. Katz, and K.S.J. Pister, Next century challenges: mobile networking for smart dust. in Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks (Mobicom'99), Washington, USA, 1999, pp. 271-278.

[133] D. Culler, D. Estrin,and M. Srivastava. Overview of WSNs. Aug. 2004, Vol. 37.

[134] C. Chong and S. P. Kumar. WSNs: evolution, oppertunities and challenges. Aug. 2003, Vol. 91, pp. 1247-1256.

[135] I. F. Akyildiz, Y. Sankarasubramaniam, and E. Cayirci, A survey on WSNs. IEEE Communications Magazine, Aug. 2002, Vol. 40, pp. 102-114.

[136] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler. An analysis of a large scale habitant monitoring application. in The Second ACM Conference on Embedded Networked Sensor Systems, Nov. 2004.

[137] K. Martinez, P. Padhy, A. Riddoch, H. L. R. Ong, and J. K. Hart. Glacial environment monitroing using WSNs. in REAL-WSN'05, Sweden, Nov. 2005.

[138] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in WSNs. in Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks (Mobicom'99), Washington, USA,1999, pp. 263-270.

[139] Ivan Stojmenovic. Handbook of WSNs: algorithms and architectures, John Wiley & Sons, 2005.

[140] A. Manjeshwar and D.P. Agarwal. Apteen: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. in Proceedings of the 16th International Parallel and Distributed Processing Symposium (IPDPS'02), 2002, pp. 195–202.

[141] L. Sankaranarayanan, G. Kramer, and N.B. Mandayam. Hierarchical WSNs: capacity bounds and cooperative strategies using the multiple-access relay channel model. in First Annual IEEE Commu- nications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON 2004), Oct. 2004, pp. 191–199.

[142] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, 2004.

[143] F. Baccelli and S Foss. On the saturation rule for the stability of queues. In Jour. of Appl. Prob., 32, 494-507, 1995.

[144] M. Stojanovic. Underwater Wireless Communications: Current Achievements and Research Challenges. http://www.mit.edu/~millitsa/resources/pdfs/newsletter-oes.pdf.

[145] D. P. Bertsekas. *Nonlinear Programming.* Belmont, MA: Athena Scientific, 1995.

List of Publications

Conference Papers

[C-1] Muhammad Farukh Munir and Fethi Filali. A Novel Self Organizing Framework for SANETs. EW, 12th European Wireless Conference, 2-6 April 2006, Athens, Greece.

[C-2] Muhammad Farukh Munir, Arzad Alam Kherani, and Fethi Filali. Achieving Cesaro-Wardrop Equilibrium in Wireless Sensor Networks. IEEE Winter School on Coding and Information Theory, 12-16 March 2007, La Colle sur Loup, France.

[C-3] Muhammad Farukh Munir and Fethi Filali. Analyzing The Performance Of A Self Organizing Framework for Wireless Sensor-Actuator Networks. ACM/SIGSIM CNS 2007, 10th Communications and Networking Simulation Conference, 25-29 March 2007, Norfolk, USA.

[C-4] Muhammad Farukh Munir, Arzad Alam Kherani, and Fethi Filali. On Stability and Sampling Schemes For Wireless Sensor Networks. IEEE/IFIP WiOpt, 5th Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 16-20 April 2007, Limassol, Cyprus.

[C-5] Muhammad Farukh Munir and Fethi Filali. Low-Energy, Adaptive, and Distributed MAC protocol for Wireless Sensor-Actuator Networks. IEEE PIMRC, 18th IEEE Annual International Symposium on Personal Indoor and Mobile Radio Communications Conference Proceedings, September 3-7, 2007, Athens, Greece.

[C-6] Muhammad Farukh Munir and Fethi Filali. Maximizing Network-Lifetime in Large Scale Heterogeneous Wireless Sensor-Actuator Networks: A Near-Optimal Solution, IEEE/ACM MSWIM-PEWASUN, 4th ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, October 22-26, 2007, Chania, Greece.

[C-7] Muhammad Farukh Munir, Arzad Alam Kherani, and Fethi Filali. A Distributed Algorithm to Achieve Cesaro-Wardrop Equilibrium in Wireless Sensor Networks. IEEE-CCNC 2008, 5th IEEE Consumer Communications & Networking Conference, January 10-12, 2008, Las Vegas, USA.

[C-8] Muhammad Farukh Munir, Arzad Alam Kherani, and Fethi Filali. Stability and Delay Analysis of Multi-Hop Single-Sink Wireless Sensor Networks. IEEE PERCOM-PerSeNS 2008, 4th IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing, March 17-21, 2008, Hong Kong.

[C-9] Muhammad Farukh Munir, Agisilaos Papadogiannis, and Fethi Filali. Cooperative Multi-Hop Wireless Sensor-Actuator Networks: Exploiting Actuator-Cooperation and Optimizations. IEEE WCNC 2008, IEEE Wireless Communications and Networking Conference, March 31- April 3, 2008, Las Vegas, USA.

[C-10] Muhammad Farukh Munir and Fethi Filali. Increasing Connectivity in Wireless Sensor-Actuator Networks Using Dynamic Actuator Cooperation. IEEE VTC 2008, IEEE Vehicular Technology Conference, May 9-14, 2008, Singapore.

[C-11] Muhammad Farukh Munir, Hong Xu, and Fethi Filali. Underwater Acoustic Sensor Networking Using Passive Phase Conjugation. IEEE ICC-2008, IEEE International Conference on Communications, May 19-24, 2008, Beijing, China.

[C-12] Muhammad Farukh Munir, Arzad Alam Kherani, and Fethi Filali. Multi-Hop Single-Sink Wireless Sensor Networks: Distributed Convex Optimizations. IEEE/IFIP WiOpt, 7th Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 23-25 June 2009, Seoul, South Korea.

Journal Papers

[J-1] Muhammad Farukh Munir, Arzad Alam Kherani, and Fethi Filali. Achieving Long-Term Stability in Delay-Constrained Wireless Sensor-Actuator Networks. Submitted to an International Journal for publication.

[J-2] Muhammad Farukh Munir and Fethi Filali. LEAD: A Low-Energy Adaptive and Distributed Self-Organizing Framework for Sensor-Actuator Networks. Submitted to an International Journal for publication.

[J-3] Muhammad Farukh Munir, Hong Xu, and Fethi Filali. An Efficient Communication Framework for Underwater Acoustic Sensor Networks Using Passive Phase Conjugation. Submitted to an International Journal for Publication.

Research Reports

[RR-1] Muhammad Farukh Munir and Fethi Filali. An energy aware actuator discovery protocol for SANETs. Rapport de recherche RR-06-158.

[RR-2] Muhammad Farukh Munir and Fethi Filali. Performance analysis of the actuator discovery protocol for static and mobile sensor and actuator networks. Rapport de recherche RR-06-159.

[RR-3] Muhammad Farukh Munir and Fethi Filali. A low-energy adaptive and distributed MAC protocol for wireless sensor-actuator networks. Rapport de recherche RR-06-161.

[RR-4] Muhammad Farukh Munir, Arzad Alam Kherani, and Fethi Filali. Achieving Cesaro-Wardrop equilibrium in wireless sensor networks. Rapport de recherche RR-07-191.

List of Abbreviations

It's not what you say; it's what they hear.

ACK	Acknowledgement
ADP	Actuator Discovery Protocol
AF	Action First
AODV	Ad hoc On Demand Distance Vector
APC	Active Phase Conjugation
API	Application Programing Interface
AUVs	Autonomous Underwater Vehicles
AWGN	Additive White Gaussian Noise
BCDCP	Base-Station Controlled Dynamic Clustering Protocol
BER	Bit Error Rate
BFS	Breadth-First Scheduling
BMA	Bit-Map Assisted
BPSK	Binary Phase Shift Keying
BS	BaseStation
CCI	Co-Channel Interference
CDMA	Code Division Multiple Access
CMAC	Convergent MAC
CPEQ	Clustered PEQ
CSIT	Channel State Information at the transmitter
CSMA	Carrier Sense Multiple Access
DD	Directed Diffusion

DF	Decision First
DPS	Dual Preamble Sampling
FMAC	Framelet MAC
GS	Gold Sequences
HEED	Hybrid Energy-Efficient Distributed Clustering
IP	Internet Protocol
ISI	Inter-Symbol Interference
LEACH	Low-Energy Adaptive Clustering Hierarchy
LEAD	Low-Energy Adaptive and Distributed
LAN	Local Area Network
LFS	Length-First Scheduling
LLNs	Low power and Lossy networks
LP	Linear Program
LPL	Low-Power Listening
MAC	Medium Access Control
MEMS	Micro-Electro-Mechanical Systems
MINLP	Mixed Integer Non-Linear Program
MILP	Mixed Integer Linear Program
MRC	Maximal Ratio Combining
NACK	Negative Acknowledgement
NLP	Non-Linear Program
NP	Non-deterministic Polynomial time
ODE	Ordinary Differential Equations
OLSR	Optimized Link State Routing
PC	Phase Conjugation
PEGASIS	Power-Efficient Gathering in Sensor Information Systems
PEQ	Periodic Event-driven Query-based protocol
PER	Packet Error Rate
PLC	Powerline Communication

PPC	Passive Phase Conjugation
PPM	Pulse position modulation
QoS	Quality of Service
SANETs	Sensor-Actuator Networks
SDVv	Small Delivery Vehicles
SINR	Signal-to-Interference-Noise Ratio
SNR	Singal-to-Noise Ratio
SPT	Shortest Path Tree
TDMA	Time Division Multiple Access
TOSSIM	TinyOS Simulator
TR	Time Reversal
TRM	Time Reversal Mirror
TRO	Time Reversal Operator
TSMP	Time Synchronized Mesh Protocol
UASNs	Underwater Acoustic Sensor Networks
UE	User Equilibrium
UMTS	Universal Mobile Telecommunications System
UUVs	Unmanned Underwater Vehicles
WFQ	Weighted Fair Queueing
WSNS	Wireless Sensor Networks

Abstract

In the first part of the thesis, we consider a WSN. We consider the following data sampling scheme: the sensor nodes have a sampling process independent of the transmission scheme. A correct stability condition for such a system is provided. We propose a cross-layered sampling scheme in which the sensor nodes sample new data only when it has an opportunity to transmit. We then address the problem of optimal routing that minimize the end-to-end delays. We propose an algorithm that seeks the Wardrop equilibrium.

In the second part, we consider a two-tier SANET. We first address the minimum-delay problem for data aggregation and investigate an optimal actuator-selection problem for SANETs. A polynomial time algorithm is proposed for optimal actuator selection. We also propose a distributed mechanism for actuation control which covers all the requirements for an effective actuation process. We then address the minimum-energy consumption problem for data aggregation. We present the design, implementation, and performance evaluation of a novel low-energy adaptive and distributed (LEAD) self-organization framework. This framework provides coordination, routing, and MAC layer solution for network organization and management.

In the last part, we focus on UASNs. We analyze a modulation scheme and associated receiver algorithms. This receiver design take advantage of the time reversal (phase conjugation) and properties of spread spectrum sequences known as Gold sequences. We show that this approach can give almost zero BER for a 2-hop communication compared to single hop. This link layer information is then used at the network layer to formalize routing decisions.

Keywords: WSNs, SANETs, UASNs, Cross-Layering, Optimizations, Analysis, Simulations.

Résumé

Dans la première partie de la thèse, nous considérons un WSN. Nous considérons les données suivantes, des plans d'échantillonnage: le capteur ont un processus d'échantillonnage indépendants de la transmission. Une bonne condition pour la stabilité d'un tel système est fourni. Nous proposons un processus d'échantillonnage 'inter-couche' dans lequel les nœuds échantillon de nouvelles données seulement quand il a la possibilité de transmettre. Nous avons ensuite aborder le problème de routage optimal permettant de minimiser le retard bout-a-bout. Nous proposons un algorithme qui cherche l'équilibre de Wardrop.

Dans la deuxième partie, nous considérons un SANET. Un algorithme en temps polynomial est proposé pour la sélection optimale d'actionneur. Nous proposons également un mécanisme distribué pour l'actionnement de contrôle qui couvre toutes les exigences d'un véritable processus de déclenchement. Nous présentons la conception, la mise en œuvre et évaluation de la performance d'un roman à faible consommation d'énergie distribuée et adaptative (LEAD) d'auto-organisation cadre. Ce cadre prévoit la coordination, de routage, et la couche MAC solution réseau pour l'organisation et la gestion.

Dans la dernière partie, nous nous concentrons sur UASNs. Nous analysons un schéma de modulation et d'algorithmes de récepteur. Cette conception de récepteurs tirer profit du temps d'inversion (conjugaison de phase) et les propriétés des séquences d'étalement de spectre, connu sous le nom de Gold séquences. Nous montrons les ameliorations de cette approche. Cette couche liaison de données sont ensuite utilisées à la couche réseau d'officialiser les décisions de routage.

Mots-clés: WSNs, SANETs, UASNs, Inter-couche, Optimizations, Analyse, Simulations.