Wireless Sensor Networks
Using Network Coding
For Structural Health Monitoring

By

Jelena Skulic

Department of Electrical and Electronics Engineering

Imperial College of Science, Technology and Medicine

London, United Kingdom, SW7 2BT

October 2014

PhD Supervisor: Kin K. Leung

I herewith certify that all material in this dissertation which is not my own work has been properly acknowledged.

Abstract

Wireless Sensor Networks (WSNs) have been deployed for the purpose of structural health monitoring (SHM) of civil engineering structures, e.g. bridges. SHM applications can potentially produce a high volume of sensing data, which consumes much transmission power and thus decreases the lifetime of the battery-run networks. We employ the network coding technique to improve the network efficiency and prolong its lifetime. By increasing the transmission power, we change the node connectivity and control the number of nodes that can overhear transmitted messages so as to hopefully realize the capacity gain by use of network coding.

In Chapter 1, we present the background, to enable the reader to understand the need for SHM, advantages and drawbacks of WSNs and potential the application of network coding techniques has. In Chapter 2 we provide a review of related research explaining how it relates to our work, and why it is not fully applicable in our case.

In Chapter 3, we propose to control transmission power as a means to adjust the number of nodes that can overhear a message transmission by a neighbouring node. However, too much of the overhearing by high power transmission consumes aggressively limited battery energy. We investigate the interplay between transmission power and network coding operations in Chapter 4. We show that our solution reduces the overall volume of data transfer, thus leading to significant energy savings and prolonged network lifetime. We present the mathematical analysis of our proposed algorithm. By simulation, we also study the trade-offs between overhearing and power consumption for the network coding scheme.

In Chapter 5, we propose a methodology for the optimal placement of sensor nodes in linear network topologies (e.g., along the length of a bridge), that aims to minimise the link connectivity problems and maximise the lifetime of the network. Both simple packet relay and network coding are considered for the routing of the collected data packets towards two sink nodes positioned at both ends of the bridge. Our

mathematical analysis, verified by simulation results, shows that the proposed methodology can lead to significant energy saving and prolong the lifetime of the underlying wireless sensor network.

Chapter 6 is dedicated to the delay analysis. We analytically calculate the gains in terms of packet delay obtained by the use of network coding in linear multi-hop wireless sensor network topologies. Moreover, we calculate the exact packet delay (from the packet generation time to the time it is delivered to the sink nodes) as a function of the location of the source sensor node within the linear network. The derived packet delay distribution formulas have been verified by simulations and can provide a benchmark for the delay performance of linear sensor networks.

In the Chapter 7, we propose an adaptive version of network coding based algorithm. In the case of packet loss, nodes do not necessary retransmit messages as they are able to internally decide how to cope with the situation. The goal of this algorithm is to reduce the power consumption, and decrease delays whenever it can. This algorithm achieves the delay similar to that of three-hop direct-connectivity version of the deterministic algorithm, and consumes power almost like one-hop direct-connectivity version of deterministic algorithm. In very poor channel conditions, this protocol outperforms the deterministic algorithm both in terms of delay and power consumption.

In Chapter 8, we explain the direction of our future work. Particularly, we are interested in the application of combined TDMA/FDMA technique to our algorithm.

Acknowledgements

First of all, I would like to thank my supervisor, Professor Kin K. Leung for taking me as a PhD student and giving me an opportunity to study at Imperial College. I appreciate his guidance and support not only for my research but also my life. After three years of his supervision, his impressive leadership becomes a milestone in my life. My adviser treats his students as his own children, and I hope to do the same in a due day. I will always be grateful for the opportunity and support he gave me during my PhD studies.

My parents Milka and Milorad, and my brother Kosta have given me much support and love. I am grateful that they always believed in me, and sacrificed on so many occasions to help me pursue my dreams.

Dr. Athanasios Gkelias has been a great support throughout my work. He spent a lot of his time brainstorming ideas and discussing my work. We co-authored a number of papers, and my PhD would not be as nice experience if I did not have his unselfish support at all times. I am very grateful to have him as a friend, colleague and co-adviser.

My friends and colleagues from Communications and Signal Processing Group J. Murray-Bruce, G. V. Rossi, A. Scholefield, L. Ma, G. Tychogiorgos, A. Akindoyin, Z. Sheng, C. Y. Leow, L. Noto, S. Kanna, and J. Pearson, have helped me in the past three years.

I would like to thank to my SmartEN team for their support and constructive criticism: Professors J. C. Belfiore. T. Onoufriou, K. Berberidis; Dr. M. Kyriakides, Dr. A. Zhao; fellow researchers E. Karseras, R. Soman, O. Oguz, B. Han, A. Mejri, L. Ljano Trueba, S. Hong, K. Jalsen, I. Ferrer Alcover.

Finally, I would like to thank Dr. Mischa Dohler and Dr. John Polak, the examiners at the PhD viva as they carefully read this thesis and provided some great directions both regarding improvements and future work.

Contents

List of Figures

Publications used in this thesis

- J. Skulic, K.K. Leung, “Application of Network Coding in Wireless Sensor Networks for Bridge Monitoring,” PIMRC 2012, Australia, 2012.
- J. Skulic, K. K. Leung, “Monitoring of bridge using a wireless sensor network based on network coding”, IABMAS 2012, Italy, 2012.
- J. Skulic, A. Gkelias, and K. K. Leung: "Node Placement in Linear Wireless Sensor Networks ", EUSIPCO 2013, Morocco, 2013.
- J. Skulic, A. Gkelias, and K. K. Leung: "Delay Analysis of Network Coding in Linear Wireless Sensor Network", ICWiSe 2013, Malaysia, 2013.
- J. Skulic, A. Gkelias, and K.K. Leung, "Adaptive Network Coding Based Cross layer Protocol for Wireless Sensor Networks", to be published.

List of Tables

Glossary

MAC	Medium Access Control
WSN	Wireless Sensor Network
SHM	Structural Health Monitoring
NC	Network Coding
SINR	Signal to Interference Noise Ratio

1. INTRODUCTION

1.1 Methodology

This thesis comprises of several chapters that tackle different challenges of WSNs used for SHM. In the first chapter we discuss the idea of network coding application in the WSNs used to increase the reliability of network, and decrease the power consumption while increasing the throughput. In this thesis, we will focus on structural health monitoring of liner structure e.g. a bridge. We will consider that sensor nodes are deployed along one or both sides of bridge, creating a layout that resembles single or double-parallel-lines. All nodes sense a number of data (temperature, humidity, strain, vibrations) make packets of sensed data and forward them to the sink nodes located at the ends of bride. Sink nodes communicate those data to civil engineers via Internet. All nodes generate packets, forward their own packets as well as packets originating at other nodes towards sink nodes.

In the first technical part, we will present the routing algorithm for linear WSNs - for nodes deployed at one or both sides of bridge. We will show how nodes communicate with each other using network coding. We will allow nodes to use the overhearing capability to decrease the number of transmissions necessary to relay all messages to sink nodes. We discuss algorithm from aspect of total number of transmissions and receptions in the network that are the main consumers of the limited battery supply in these networks.

Following this chapter, we assume more realistic channel in which some of these packets can be lost. Those packets need to be retransmitted which can have significant influence on total power consumption. We will discuss different versions of multi-hop communication where we change the transmission power level to adjust the connectivity among adjacent nodes. This increases the overhearing capability and decreases the number of necessary transmissions, but also exposes more packets to collisions.

Following this we will try to discover what would ideal positions of sensor nodes be in order to decrease the maintenance costs. As sensor nodes have different loads in terms of messages they need to transmit their distance can be used to even out the speed at which their power supply is spent. Thus, technicians will not have to go to the field often to replace batteries which drives costs of monitoring down.

In the following technical chapter we will analyse the time delays in this network. Often, delay is not the most important parameter, so we will look for techniques that will decrease the total delay but not at the expense of power consumption. We will look for lower bound in terms of delays so as to set the - expectations and potential applications of our algorithm straight.

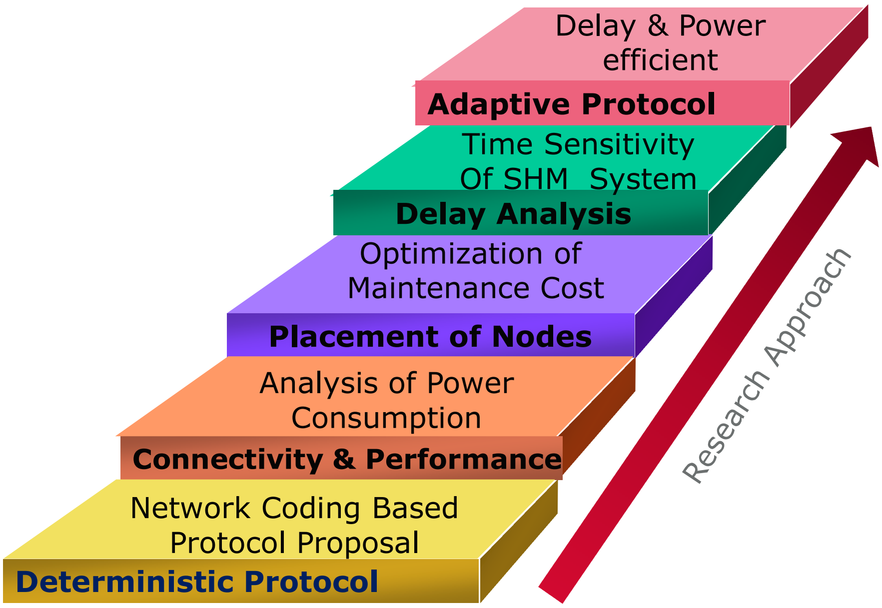


Fig. 1.1 Methodology used in this thesis.

Lastly, we will look for an adaptive network coding based algorithm for WSNs that is robust enough and does not crashes when unexpected events happen. We will try to design this algorithm having primarily power consumption on mind, but also delays in the network. The goal overall is to look for adaptive, self-healing solution. While we were writing this thesis, we decided to test each solution both mathematically and using extensive simulations. This methodology is presented in the Figure 1.1.

1.2 Motivation for Structural Health Monitoring

The aging infrastructure is a big concern for the modern society. In Europe, it is common to find several centuries old infrastructure that is still heavily used. There is a risk associated with the deterioration of infrastructure state including severe economic implications and a loss of lives due to its malfunction. A failure of e.g. bridge may result in the disruption of traffic for a period of time, but it can also compromise the emergency response. Obviously, the financial loss is not limited to the value of the structure.

Structural Health Monitoring (SHM) aims to estimate the state of the structure, and detect the structural changes that have potential to change the performance of the structure. Various types of infrastructure are monitored nowadays such are bridges, buildings, dams, pipelines, aircrafts, ships, off-shore oil rigs etc. Together, they make a pillars of industrial and economic prosperity. Civil engineers consider a number of standardized methodologies in the construction phase to ensure that a structure can be safely used. However, the design cannot anticipate all scenarios of misuse (e.g. overloading) and environmental conditions that may eventually cause the structural deterioration.

Only in the past 10 years a number of structures collapsed due to the deterioration, some of which are: Myllysilta bridge (Finland, 2010), Feltham bridge (UK, 2009), Malahide Viaduct (Ireland, 2009), bridge over river Po (Italy, 2009), E45 Bridge (Denmark, 2006), bridge in Almuñécar (Spain, 2005). Globally, in more populated areas some infrastructure failures took away a number of human lives, mostly in US, China, Canada, Indonesia and India. Also, a number of bridges collapsed due to the severe environmental conditions. For instance, the severe flooding resulted in the collapse of CPR Bonnybrook Bridge (Canada, 2013), five bridges in Cumbria (UK, 2009), Somerton Bridge (Australia, 2008). The earthquakes were fatal for bridges in Kobe (Japan, 1995), Chi-Chi (Taiwan, 1999), Loma Prieta (US, 1989). SHM monitoring can be a good indicator if a structure can be safely altered or not. A collapse of a department store in Seoul in 1995 killed over 500 of people, whereas a collapse of a supermarket ceiling in Latvia in 2013 killed 50 persons. All these tragic events show the vulnerability of structures to the deterioration due to the aging and misuse, and damage during natural catastrophes.

Structural Health Monitoring (SHM) systems evaluate the reliability of civil-engineering infrastructures (e.g., bridges and high-rise buildings) and track their conditions in the real-time. The purpose of SHM is to detect and localize damage, to evaluate the severity of damage, and to predict the remaining lifetime of the infrastructure. Monitoring systems make the infrastructure safer and extend its lifespan. There are two different approaches to SHM: direct and indirect damage detection. Direct damage detection is a form of visual inspection by civil engineers or x-ray inspection, ultra-sonic inspection etc. Direct damage inspection will not be a focus of this thesis. Indirect methods are automated, and they use the combination of the algorithms for damage detection within structural monitoring systems. SHM systems need to detect the time scale of the change in infrastructure (how fast is the condition changing), but also the severity of the change. Some of the SHM systems measure the reaction of infrastructure to events (event-triggered monitoring). Those events can be artificially induced vibrations for testing, or natural catastrophes. More common are systems that perform continuous health monitoring of ambient vibrations, strain, wind, temperature. These systems can perform short term or long term monitoring, depending on specific conditions. Lately, systems that measure the reaction of infrastructure to a strong motion are very popular. Particularly, there is a strong interest in systems that measure reactions of bridges to the fast moving, heavy trains as they cause a number of fatal bridge malfunctions.

SHM systems became an integral part of newly built infrastructures within highly active seismic areas. Those systems are responsible for collection of sensor’s measurements and the storage of measured data - which is the part of monitoring controlled by electrical engineers. The collected data are shipped to civil engineers that need to make a sensible decision concerning the infrastructure.

SHM systems can be classified as global and local damage detection systems. Local damage detection systems are screening a component or a subcomponent of a structure. Global damage detection systems measure global characteristic of a structure, e.g. vibrations (natural frequencies, mode shapes). Local detection systems are used for structural damage e.g. cracks. Usually, a qualified professional would go and inspect the structure personally. Ideally, he would know the damage region, which would enable him to prioritize his work (as it is often not feasible to inspect a complete structure). This kind of monitoring is expensive, slow, unreliable, inefficient and susceptible to human error. Global based damage detection systems use numerical methods to perform the global analysis of characteristics such are vibrations, strain. This type of monitoring emerged only after automated SHM systems became available.

1.3 Wireless vs. Wired Networks Study

Traditional SHM systems were based on piezoelectric patches that are used as actuators and sensors, and are wired to data acquisition boards in PC. Wired data acquisition systems were used to collect data from a number of locations in a monitored structure. For instance, the vibrations induced by ambient sources such as moving vehicles, winds, seismic movements, or artificial test would be monitored. While wire (usually coaxial cable) is a very reliable communication link and almost unlimited resources in terms of capacity, there are several reasons why wired systems proved to be unfeasible solutions.

From financial perspective, the cost of wired system is prohibitively high. The wired SHM system mounted to Tsing Ma bridge in Hong Kong with 350 sensing channels cost around £5.5 million [1]. SHM systems that are monitoring buildings are a little bit cheaper as they use fewer sensors, but some reported systems cost £3500 per sensing channel [2]. Therefore, as the size of wired SHM system grows its cost increases at a much higher rate. The cost imposes the limitation on the size of SHM systems, and as a result, wired SHM systems are rarely high span. Additionally, the implementation and maintenance of these systems is very high. Also, as wires have to run all over the structure, wired systems cause the disruption of normal function of a structure. Commonly used coaxial cables are very sensitive to bending and their performance tends to degrade severely if they are not well integrated with the structure. The problems with vandalism are also reported, and the replacement of these kinds of systems is too expensive.

Wired systems fail easily in the case of a natural disaster, and in those situations they serve no purpose as civil engineers are not able to recover the data. Wired systems also require a stable power source, which is particularly inconvenient for the monitoring of bridges. Wired SHM could integrate only a small number of sensor nodes. When we measure a global characteristics of a structure these systems scale poorly in the case of the localisation of damage, making the monitoring very challenging.

Compared to the wired SHM systems, Wireless Sensor Network (WSN) based monitoring systems provide a number of advantages. The most important one is the significantly lower price of wireless sensor nodes. They cost a tenth of the price of a wired sensing channel. Because of this, WSN based monitoring systems are good candidates for monitoring of large structures, as the increase in price with the network span is linear. Also, they provide an opportunity for dense monitoring (a higher spatial density of sensors). WSNs based monitoring systems can be installed in a couple of hours compared to days or even weeks needed for the installation of the wired monitoring systems. The maintenance of these systems is also simple and inexpensive as all it takes is to replace a node that failed. The disruption of the operation of structure when WSN is deployed is minimal. Therefore, a WSN data acquisition offers ease of deployment and maintenance, scalability, flexibility of monitoring and low cost of monitoring, non-intrusive sensor nodes that integrate well with the structure, and as such these systems are a natural choice for SHM.

The low cost of sensor nodes enables dense monitoring. Often, hundreds of sensors can be installed on a structure making this system a better option for damage screening by monitoring behaviour of critical components, enabling these systems to detect local damage with the high accuracy.

1.4 Requirements for SHM applications

SHM system has several design objectives and requirements that will directly influence the setup of a sensor network used for the monitoring:

- Civil engineers are usually interested in vibration measurements. Vibrations tend to change in a fraction of a second. In order to capture the change, sampling frequency needs to be very high, resulting in the high data rates.
- Large structures and applications that require dense data sensing consequently produce high traffic load.
- Monitoring systems need to be able to report with the minimum delays the measured data to the base station. Thus, one of the indicators of high quality SHM system is the low end-to-end delay.
- A target sampling rate can be as high as 1 [kHz] (though it is usually 200 [Hz]). Combined with 16-bit minimum required digitalization accuracy, there is a need for very low jitter (time uncertainty of the sampled intervals).
- In order to obtain the meaningful vibrations measurements for correlation analysis, time synchronization in sampling throughout the bridge is important. This problem is hard to solve as independent clocks in nodes usually have a drift.
- The most important requirement of SHM is the reliable transfer of data from sensing units to base station(s). Vibrations data are too valuable as they can indicate the state of the structure, to be lost to a communication error.

- Peak amplitudes of detected signals can be high; they can also be as low as 500 [μG]. SHM system must be able to communicate all extremes to base station(s). Thus reporting a linear combination of measurements is not an option.

Civil engineers assess the state of a structure based on summation of the set of modes. Modes are sinusoids with frequencies that correspond to the resonant frequencies of the structure. Fundamental modes of most structures are below 10 [Hz]. Most of the early research in SHM area assumed the sampling frequency of 50 [Hz].

However, as the response to vibrations decays fairly quickly, the sampling frequency of 50 [Hz] does not provide enough of samples for robust reconstruction of spectrum (it provides 25-40 samples of the useful structural response data). This is very prominent in structures such are buildings and bridges that are not designed to sustain prolonged vibrations. It is implied that Niquist criterion does not provide the lower bound for sampling rates on real structures. As the background noise level tends to be high in structures, oversampling is generally accepted as a strategy to improve signal to noise ratio, by reducing the relative noise energy.

For high sampling rates, it is important to control time uncertainty (jitter) as a guaranty of the synchronization of nodes within the network. Both sources of jitter need to be capped. Time-jitter occurs because the software cannot keep up with the aggressive sampling, while special jitter exists due to the internal clocks that are not fully tuned. Usually, the goal is to cap the total jitter to 5% of the sampling interval. Wired SHM systems, have only time-jitter as all nodes share the clock, while spatial jitter is under control. Jitter is usually taken care of by adding an additional microcontroller.

In SHM applications, it is required that no data is lost in communication system, as events of interests do not happen often, and they cannot be duplicated. Thus, the aim is the lossless communication. As we are interested primarily in the global structure monitoring, availability of high span networks is important.

The low cost of wireless sensor nodes allows for many nodes to be deployed, providing denser coverage and multiple data paths to cope with the failures.

1.5 Limitations of Wireless Sensor Networks used for SHM

SHM applications operate with high sampling rates. In the case of global monitoring, sensors need to monitor physically large surfaces. For both global and local monitoring, dense coverage of infrastructure systems is always desirable. This results in high data rates. Data rates were not a problem for wired monitoring systems, but it is a challenging to cope with the high data rates for WSNs based systems. Essentially, nor is the wireless medium convenient for the reliable communication, nor is the internal limited battery power convenient for the high data rates.

High data rates consume much power resources for the transmissions/receptions. Power consumed for transmissions/receptions is much higher than power consumed for simple computations. Due to the high data rates, WSNs are prone to failures of the individual nodes as their batteries get depleted. It is essential that sensing units of all nodes participate in the monitoring at all times. The failures of individual nodes would degrade the reliability of the system. Even if the power resources were unlimited, there would still be a problem of limited bandwidth that cannot comply with the need for high data rates. This issue can be controlled at the expense of the end-to end delays, as the requirement for low delays is not as strict as the requirement for the high reliability of data delivery.

Another component of a SHM system's value for civil engineers is the ability to cap and control the jitter. Unfortunately, it is hard to cap the time-jitter as the data rates are high and nodes are fairly simple and small, so adding a microcontroller is not always an option. WSNs are particularly a poor candidate for a low spatial jitter as internal clocks of nodes are always slightly unsynchronized, and overhead in the form of synchronization messages would unnecessary drain out the battery power. All these issues need to be solved as the upside of WSN-based SHM systems is significant. The main goal of SHM systems remains the high quality of collected and delivered data. Based on those data, the structure quality is assessed and further decisions concerning the infrastructure's safety are made by civil engineers.

1.6 Challenges: low latency, power constraint, and high reliability

Due to the high sampling frequency, the traffic load of raw data is very high. It is recommended to decrease the volume of data prior to their transfer to the data sink node(s). However, data processing should not change the fidelity of data because that would directly degrade the sensitivity of a SHM system. The performance of a SHM system depends on the quality of data received at the sink and the packet loss rate. Thus, the goal is a reliable and lossless communication over potentially a large span network. It is essential to decrease data overhead as much as possible, and to carefully use scarce network resources. Data aggregation techniques are not a good solution for SHM.

Data processing techniques are frequently based on averaging, minimizing or maximising of the value of all received messages. It is important to deliver the accurate information in form of measurements to civil engineers so they can make the best decision. Typical aggregation technique results in the loss of information. A good way to preserve original sensed data, and yet to decrease the volume of data is to deploy network coding. The idea of network coding is to allow the coding of data at intermediate nodes and forwarding of coded packets, each of which is generated by combining and encoding more than one packet received from possibly multiple sources. The simplest encoding scheme is the XOR operation applied to each corresponding bits of two packets of the same size. In traditional routing schemes, nodes forward data packets from each neighboring node towards the sink(s) separately.

1.7 Wireless Sensor Node

A wireless sensor node is a device of small dimensions that consists of a microcontroller, a radio transceiver, an antenna, a power source, and a number of sensors. A node functions thanks to a lightweight operating system that is able to perform a sensor querying, a simple data manipulation, remote access and wireless communication.

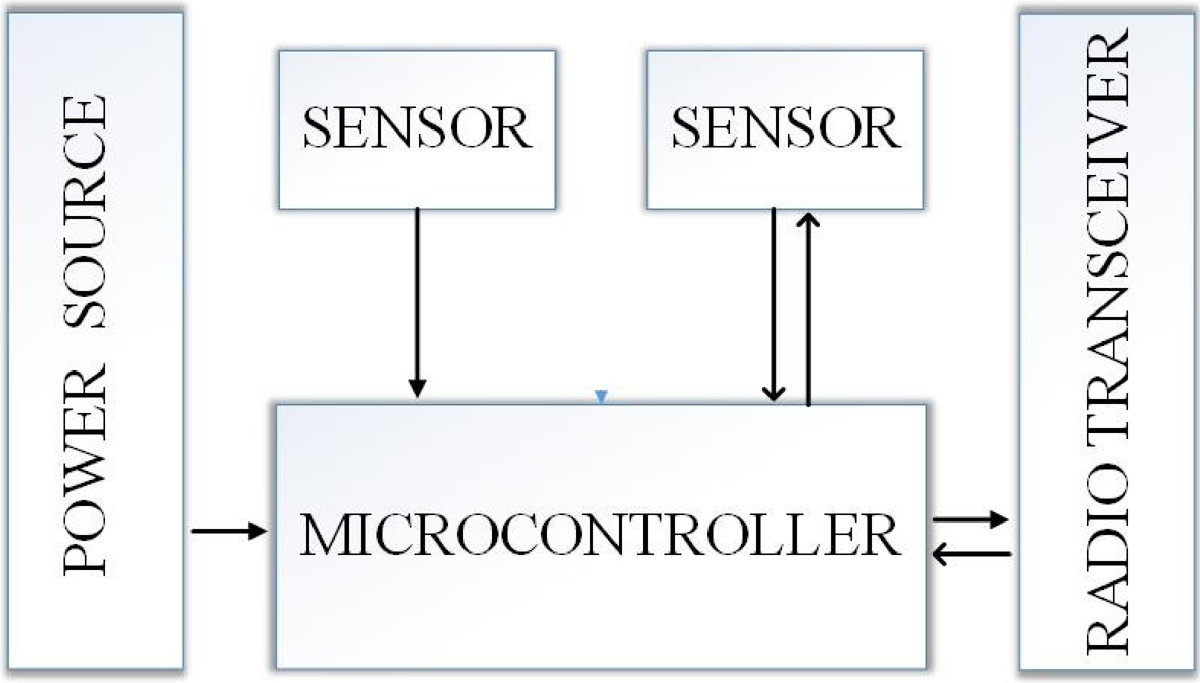


Fig. 1.2 Functional diagram of a wireless sensor node's hardware.

Each sensor node comes equipped with a microcontroller that is the central controlling device that communicates with sensors and radio unit, and uses power source for its operations. The transceiver is a unit responsible for transmission and reception of data packets. Sensors are used for data collection and their type determines the purpose of a node. Typical hardware of a node is presented in a form of a functional diagram in Figure 1.2. It is essential to optimise each unit in these subsystems.

A controller unit is supervising the operation of wireless sensor node. There are different types of microcontrollers on the market, some of which are designed to accommodate the high speed, others the power efficiency. For the purpose of SHM microcontrollers will be used to achieve the high power efficiency (microcontroller for low power).

1.8 Linear Network Coding

In July 2000, Rudolf Ashwede and his team introduced a new class of problems that they named *network information flow* [29]. They considered a point to point communication network in which a number of information sources were able to multicast to the set of destinations. They presented a technique that improves a network's throughput, efficiency and scalability. Instead of simply forwarding the packets of information they receive, the nodes combine a number of packets and transmit the coded packet. This technique is used to maximise the information flow in a network.

Ashwede et al. proved that linear network coding can achieve the upper bound in multicast problems regardless of the number of source nodes. However, linear coding cannot achieve an optimal information flow for multi-sink network, and therefore finding an optimal coding solution for general network problems is an open problem.

The original network coding paper used a butterfly network to present the idea how network coding outperforms simple forwarding, Figure 1.3. At the top of the picture are two source nodes that have information (A) and (B). At the bottom of the Figure 1.3, are two sink nodes. Each sink node is interested in both information (A) and (B). Each link is able to carry only one value at the time. To use a classical routing as an example, any link would be able to carry either (A) or (B) at the time but not both of them. If we decide to send message (A) through the centre, the left sink node would receive twice the message (A). The same problem would face the right sink node if we send (B) through the centre. Therefore, conventional routing is insufficient as no scheme would result in the simultaneous delivery of both (A) and (B) to the sink nodes.

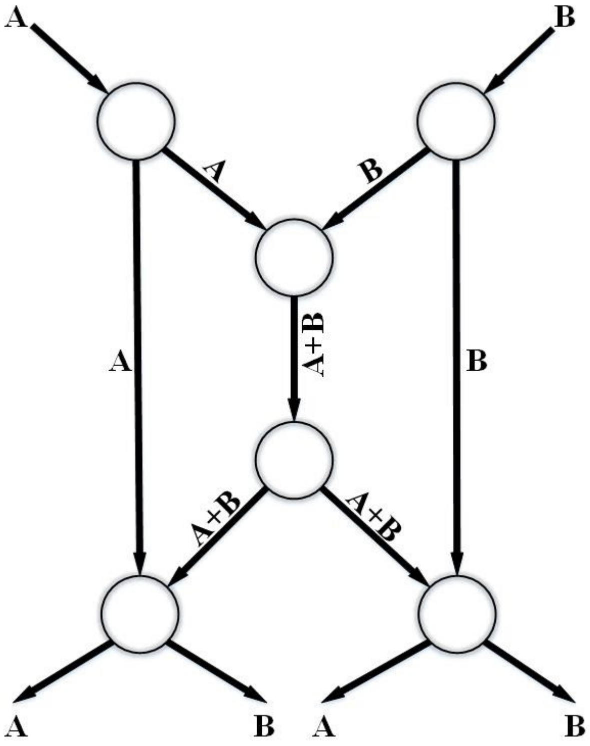


Fig. 1.3. Butterfly network. Linear network coding.

Ashwede et al. suggested that central rely nodes send a linear combination of (A) and (B), or more precisely to use a simple XOR operation to code a packet (A+B). As the left sink node receives messages (A) and (A+B), it will be able to compute the message (B) by applying the same XOR operation to the two received packets, (A+(A+B)) = (B). Using the same idea, the right node will extract the message (A) from received messages (B) and (A+B).

We will use this simple idea as a base for a forwarding scheme that supposed to prolong the network lifetime, but still to deliver the original messages to the sink nodes.

1.8. Challenges of real-life Applications of SHM WSNs based systems

WSNs are distributed systems whose nodes execute software that is made specifically for them. Sensor nodes need to perform various tasks to make WSN functional. They need to schedule and execute tasks determined by software that is running them. They are in charge of data acquisition, signal processing, temporarily storage of acquired data in their buffers, monitoring and control of their own resources (power, memory etc.), self-configuration, reception and transmission of data, scheduling of communication and many other tasks. A node can be programmed to support different protocols, which enables node to perform very specific set of tasks. Sensor nodes are usually programmed using NesC or TinyOS – two most popular open source projects [76][77].

Sensor are battery powered, which significantly limits their abilities. Battery is typically preserved by deployment of low-power hardware but also using sensor units that consume less than 80 [mW]. Data communication is the task that consumes the most of available energy supply. This is why most of WSNs are running as multi-hop systems. Data acquisition is an important part of system reliability. In sensor networks it is typically done using 8 or12 [bits] Analog/Digital Converters, limiting the resolution of data. Data quantization effect can be a problem with low amplitude data, but the performance of 8 and 24 bits AD convertors is not significantly different for data acquisition of high amplitude data.

Duty cycle is often used to limit the energy consumption of nodes. A duty cycle represents a fraction of time when hardware is switched on. The lower the duty cycle, the less energy is spent. Frequent turning on and off of hardware will also result in the increase of energy consumption.

WSNs were deployed many times in the field for the short-term monitoring, but very few times for the longer period (several days). The bridge over Kerasjokk river in Sweden is one of very few systems that were monitored for the long term. Engineers used only 8 nodes and sampling frequency100 [Hz], and the record size of 30 [s]. Complete time history used 6 [kB] or 60% of the microcontroller's capacity, duty cycle was 10%. Nodes used TDMA scheduling to avoid collisions which introduced significant delays. Sensor nodes were collecting data over the period of 30 [days]. It took only 4 [h] to deploy complete WSN. The battery could last for 240 days. [78]

1.9 Related Technologies

Ultra Wide Band (UWB) signal is a signal with either a spectral occupancy in excess of 500 [MHz] or a fractional bandwidth of more than 20%.WSN applications often use UWB based technology called Impulse Radio-UWB (IR-UWB) that relies on ultra-short waveforms that can be free of sine-wave carriers and do not require IF processing because they can operate at baseband. The IR-UWB technique is chosen to be the PHY layer of the IEEE 802.15.4a Task Group for WPAN Low Rate Alternative PHY layer [80][81].

Bluetooth wireless technology is a short-range robust, low power, and low cost communication system intended to replace the cables in WPANs. The IEEE Project 802.15.1 standardized a WPAN based on the Bluetooth v1.1 Specifications. The PHY layer of Bluetooth also operates in the unlicensed ISM band at 2.4 [GHz] using 79 frequency channels spacing of 1 MHz in the ISM band. Bluetooth relies on frequency hopping strategy in order to avoid the interference and fading. Nodes in a Bluetooth networks form piconets that are managed by a master node and have a maximum number of seven active slave nodes. Bluetooth technology uses full duplex transmission deploying a time-division duplex scheme [79].

The ultra-low-power Bluetooth technology is started as a part of European FP6 project MIMOSA, using a name *Wibree*. It is the simplified version of Bluetooth, and hence better fit for WSNs. It uses the same physical layer as regular Bluetooth devices. Wibree is designed to efficiently transmitting very small quantities of data at very low latencies to other devices, making it up to 15 times more efficient compare to the Bluetooth technology as it uses optimises connectable and discoverable modes, the number and size of each individual packet transmitted during connection. Two Bluetooth devices can communicate only if they use the same frequency or channel at the same time. Thus, two devices need to search different channels to find each other. Bluetooth uses 32 channels, while Wibree uses only three channels for advertising. Searching three channels consumes less time and power than searching 32 of them. Also Wibree slave device does not listen to its master if it has nothing to transmit, unlike classical Bluetooth devices.

Z-Wave is a technology developed for low-power remote control applications. Unfortunately, it is not compatible with the 802.15.4 technology. It operates in sub 1 [GHz] band, that is subject to less interference than 2.4 [GHz] based technologies as there is no interference from 802.11 and 802.15.1 standardized devices. The downside is that European regulations limit its operations to 1% or less of duty cycle. There can be maximum 232 nodes in the network because of addressing limitations. Z-Wave operates with 9.6 [kbps] and 40 [kbps] and maximum transmission range of 100 [m].

ANT is a technology designed to run using low cost, low power microcontrollers and transceivers operating in the 2.4 [GHz] ISM band. The ANT WSN protocol achieves an ultra-low power consumption, low latency, extended battery life and lower implementation costs. ANT has an option of trade-off between data rates and power consumption. However it supports the broadcast, burst and acknowledged transactions up to a net data rate of 20 [kbps], which is not quite the target of our application. ANT is a scalable protocol as it can support ad hoc interconnection of large number of nodes. Bluetooth in average consumes ten times more power than ANT protocol. Compared to IEEE 802.15.4 ANT achieves larger data rates (nearly 1 [Mbps]). However, ANT lacks interoperability as it is a proprietary protocol [86].

1.10 Standards

WSNs were first used in military purposes during the Cold War. US military developed a Sound Surveillance System (SOSUS) that was used to detect and track Soviet submarines. The first WSN deployed submerged acoustic sensors,that were distributed along Pacific and Atlantic oceans. This system was extremely expensive at the time. Interestingly, it is still in use today and it follows the activity of wildlife and underwater volcanos, Figure 1.4.

The next milestone in the history of WSNs was the launch of the United States Advanced Research Projects Agency (DARPA) and its Distributed Sensor Networks program. Soon after, this project involved the academia and scientific community around the globe. Government and Universities started applying WSNs in the industries other than military, Figure 1.5.

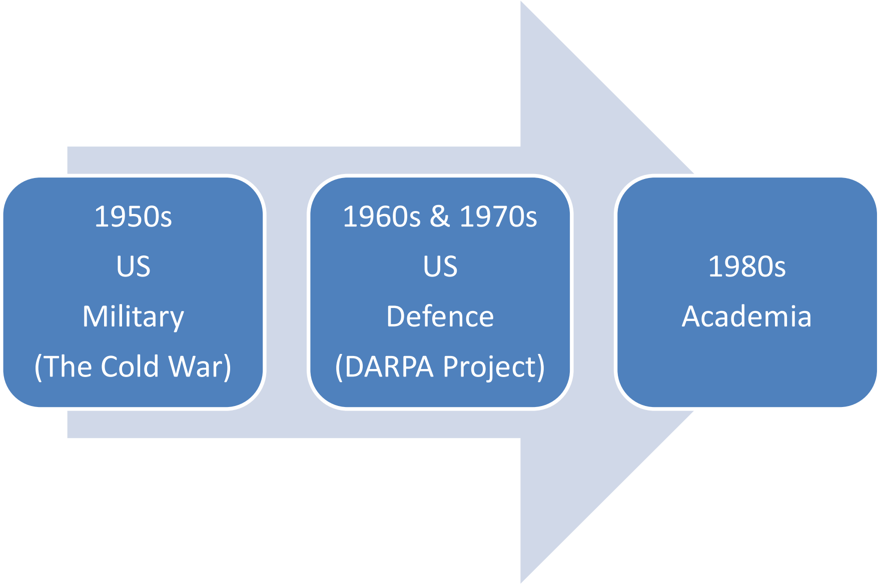


Fig 1.4. Timeline. How WSNs developed.

Academia and industry started a number of open initiatives, as shown in the Figure 1.5.

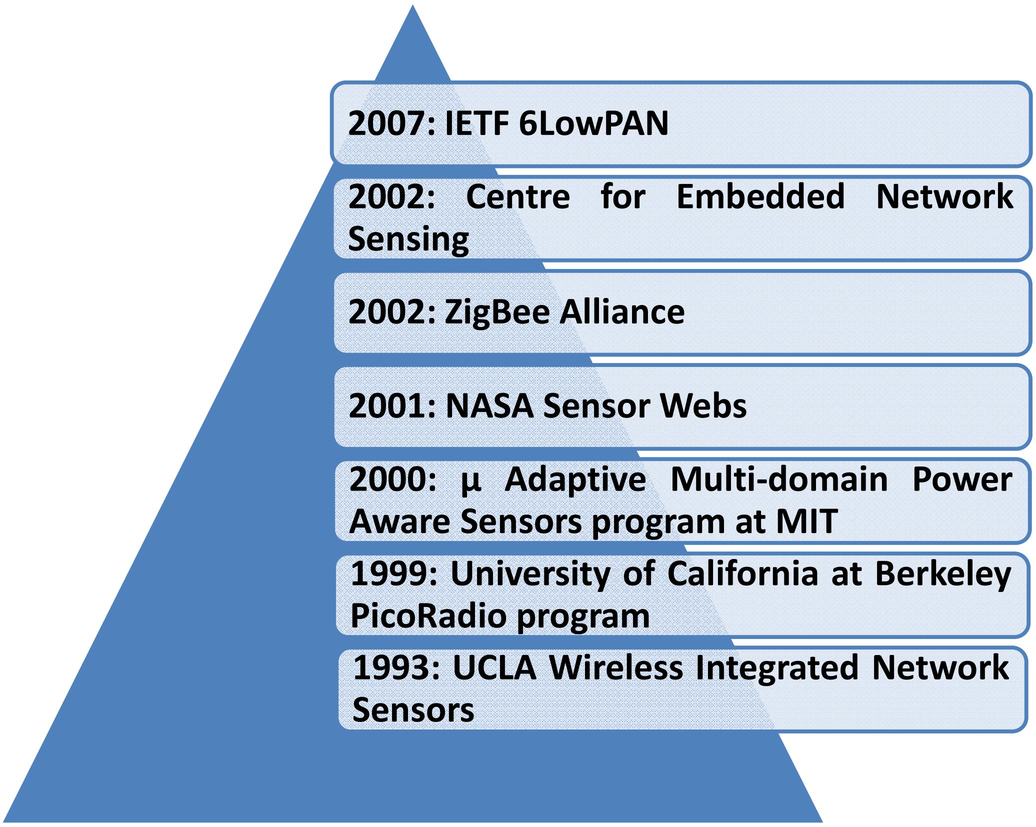


Fig. 1.5. History of Wireless Sensor Networks Academic and industrial Initiatives.

All of these initiatives had for their goal the reduction of maintenance and deployment costs of system as well as energy cost. The four most prominent brunches of research were: sensors research, semiconductor development, networking protocols and energy storage/generation technology. Sensor technology research was very popular, and there are several widely used technologies in the most recent past. CMOS sensors are predominantly used to track ambient conditions like: temperature, humidity, wind, presence of chemicals etc. LED sensors are usually deployed to track ambient light and proximity of something. MEMS sensors can serve as: gyroscopes, magnetometers, pressure-meters, acoustic sensors etc. All of these sensors can be programmed to perform additional functions.

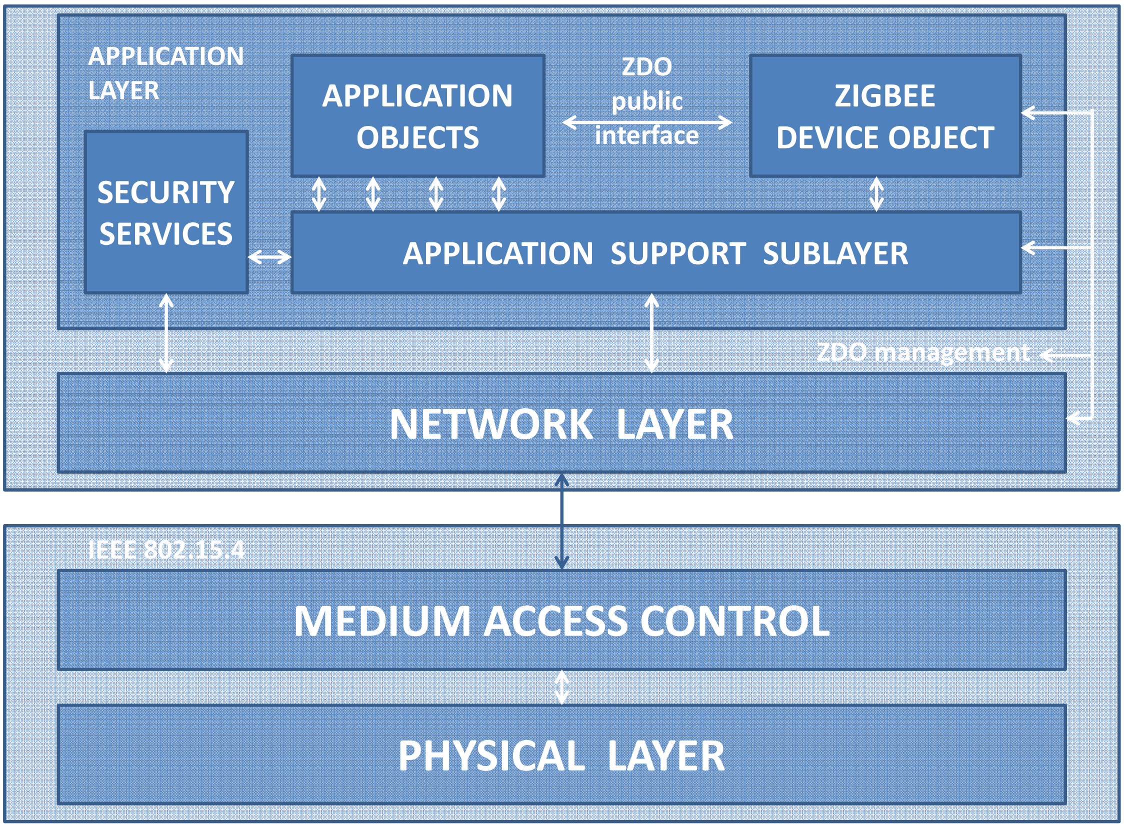


Fig 1.6. Connection between ZigBee standard and 802.15.4 standard.

WSNs are a part of WPAN (Wireless Personal Area Networks). IEEE 802.15.4 specification is intended to give a framework to wireless short-range topology of low complexity, low cost, low power consumption, low data rates applications supported by inexpensive fixed or mobile sensor nodes. This standard covers two lowest layers of the ISO/OSI protocol stack, PHYsical (PHY) and Medium Access Layer(MAC). Upper layers of stack most often use ZigBee protocols or in the past few years 6LowPAN.

Depending on geographical area 802.15.4 physical layer operates in three different bands, using total 27 half-duplex channels. Band A (868 [MHz]) uses a single channel and maximum data rate of 20 [kbps]. RF sensitivity is minimum -92 [dBm] and maximum transmission range is 1 [km]. Band B (915 [MHz]) has ten available channels with rate 40 [kbps], minimum receiver sensitivity of -92 [dBm] and maximum transmission range of 1 [km]. Band C (2.4 [GHz]) ISM band (Industrial, Scientific and Medical band).

Sixteen channels with available data rate 250 [kbps], minimum RF sensitivity of -85 [dBm] and maximum transmission range of 220 [m]. Because of the energy efficiency pat of 802.15.4 standard, networks should aim for low rate and low duty cycle (both transmitter and receiver should be over 99% of time in the idle mode) [82].

IEEE 802.15.4 uses a protocol based on the CSMA/CA algorithm with mandatory listening to the channel prior to transmission to reduce the probability of collisions with other ongoing transmissions. IEEE 802.15.4 specifies two operational modes: the *beacon-enabled* and the *non beacon-enabled* subject to deployed channel access mechanisms. Non beacon-enabled mode uses an unslotted CSMA/CA protocol to access the channel. Beacon-enabled mode, accesses the channel using a superframe, starting with a packet (beacon) transmitted by WPAN coordinator [83][84].

802.15.4 standard is tailored for the short-range communication system meant to allow the applications to have relaxed throughput and latency requirements. ZigBee protocol layer works on top of 802.15.4 standard that is in charge of PHY and MAC layers of protocol stack, Figure 1.6. ZigBee adds: network layer specifications, application layer specifications, ZigBee device objects and manufacturer-defined application objects for customization. While in the rest of the thesis, we will try to remain technology agnostic in testing phase we complied with ZigBee standard.

ZigBee devices are commonly used in industry, as they work well with IEEE 802.15.4 (2003 version) standard applied at lower layers. Basically ZigBee represents a specification for a set of higher layers' communication protocols using small, low-power devices to relay data from sensor node to sensor node, often using ad-hoc communication. ZigBee is low cost standard suits the best applications that require low data rate (150 [kbps]) and long battery life. ZigBee specification is best fit for cheaper and simpler networks, than those using Wi-Fi, UWB and Bluetooth. Available ZigBee chips have flash memory in a range between 60 and 256 [kB]. ZigBee operates in unlicensed spectrum. Operating frequency in Europe is 868 [MHZ], while in US and Australia it is 915 [MHz], and most often worldwide 2.4 [GHz], offering data rates in range between 20 and 250 [kbps]. Spread spectrum techniques are almost always deployed to reduce the interference levels in shared bands. Each network using ZigBee has one coordinator node that controls parameters and serves for the maintenance [85].

Apart of ZigBee, there is another standard for higher layers of OSI/ISO protocol stack, 6LowPAN. It is realised by Internet Engineering Task Force (IETF) in order to make 802.15.4 compatible with IPv6. 6LowPAN devices can communicate with other IP-enabled devices, whereas Zigbee node needs an 802.15.4/IP gateway to interact with an IP network. Thus the standard should be chosen by the target application. 6LowPAN standard is more and more used in SHM as civil engineers tend to prefer that sink nodes directly communicate data to their offices via Internet. If there is no need to interface with IP devices or the packet size is small, it is better to implement ZigBee as 6LowPAN performs fragmentation. We assumed ZigBee standard in parts of this thesis because of the unnecessary fragmentation of packets that 6LowPAN would perform.

1.11 Summary

In this chapter, we explained the motivation for Structural Health Monitoring (SHM). We explained why the underlying technology shifted from wired to wireless systems. We explained the advantages and the challenges of Wireless Sensor Networks (WSNs) based systems. We provided the background on sensor nodes and network coding, and set the goals of this thesis.

2. RELATED WORK

2.1 Applications of Wireless Sensor Networks for Structural Health Monitoring

Traditional Structural Health Monitoring (SHM) systems use wired networks. From financial perspective, the cost of wired system is prohibitively high. The wired SHM system mounted to Tsing Ma bridge in Hong Kong with 350 sensing channels costs around £5.5 million [1]. SHM systems that are monitoring buildings are a little bit cheaper as they use fewer sensors, but some reported systems cost £3500 per sensing channel [2]. As the size of wired SHM system grows its cost increases at a much higher rate. The cost imposes the limitation on the size of SHM systems, and as a result, wired SHM systems are rarely high span. Wired networks were usually combined with the GPS [3]. These systems were extremely expensive, and yet not particularly robust or scalable.

The usage of wired SHM systems is not practical as the cabling and connecting of sensors to a central data acquisition unit takes time and is very challenging, especially when it comes to the objects large in size such are e.g. bridges. To simplify this process WSNs based SHM systems are introduced [4]. However they acquire pre-processing of sensed data in order to comply with severely limited available bandwidth and power supply.

Therefore, a newer approach involved application of WSNs that solved many issues of their wired counterparts [5]. Lot of work on WSNs based SHM systems had for their goal to surpass the traditional wired systems [6, 7]. Pakzad et al. designed a Wiseden system and tested it on a multi-hop wireless sensor network to monitor the Golden Gate Bridge [8]. The WSN implemented on the Golden Gate Bridge generated 20 MB of data (1600 seconds of data, sampling at 50 Hz at 64 sensor nodes). It took longer than nine hours to finish the relying of the data to a central location.

Compared to the wired systems, their wireless counterparts are inexpensive and easy to mount. Some of recent field deployments include WSNs based SHM of Golden Gate Bridge, California [6], Wright Bridge, New York [7], Geumdang Bridge, Korea [8], the Alamosa Canyon Bridge New Mexico [9], and Gi-Lu Bridge, Taiwan [10]. Some WSN based SHM systems are tested on buildings [11].

WSN are not the replacement for wired systems, as they face issues with the power supply, bandwidth, synchronization and data loss. Frequent replacement of batteries would quickly offset the financial gain obtained by the deployment of a cheaper wireless network. There is a number of parallel and distributed algorithms utilizing data processing within the wireless sensor nodes, allowing them to broadcast a relatively small amount of processed data compared to a massive amount of raw data [9, 11, 12]. Distributed data processing is possible because sensor nodes have computational unit. WSNs have limited data rates as they use crowded unlicensed industrial, scientific, and medical (ISM) bandwidths. Thus, in terms of rates WSNs based SHM systems are inferior to their wired counterparts.

In general, Wireless Sensor Networks (WSNs) are used in various monitoring applications such are: structure monitoring [5], habitat monitoring [13], vehicle tracking [14] and many other applications [15].

WSNs are used for SHM of the infrastructure. They are meant to detect the changes in the performance of structure without interfering with its operation. They can monitor the disaster response (to an explosion, flood, earthquake), or they can perform continuous monitoring (vibrations, strain, temperature, wind) [5, 12, 16, 17].

Paek et al. extended the Wisden system. They now considered higher sampling frequency, precisely 200 [Hz]. They improved the data compression scheme, which prevented the unintended filtering. They replaced Mica2 nodes with the more powerful MicaZ nodes which incorporated the ZigBee technology.

Zigbee offers a theoretical bandwidth over twelve times larger than the bandwidth of Mica2 generation radios, resulting in the decrease of transmission delay and data loss [18].

However, due to the high sampling rates, we need to use some form of data compression. Data fusion for WSNs has been considered either in the form of aggregation in [19] or in the form of network coding in [20]. In the both cases information is collected by sensing unit, that is part of a sensor node, and packet is created from it. It is further forwarded towards the fusion centre following the reverse tree layout. Lin et al. derive link loss probabilities based on end-to-end path observation [19]. This approach is useful as there is no overhead caused by collection of the network maintenance messages. They show that the application of a network coding technique changes the fundamental connection between the link loss probabilities and the end-to-end path observation, as network coding offers more reliable communication even without using control messages.

A synchronization of sensor nodes can often be the issue in WSNs based SHM systems. WSNs are traditionally used as low data rate asynchronous systems for distributed monitoring. Some real-time and high data rate applications, such is SHM, face severe channel sharing problems. Goldoni et al. developed a protocol for real-time synchronous monitoring, that is applicable in both single-hop and multi-hop WSNs. PRISM defines a maximum number of sensor nodes, achievable data-rate and the requirements for the nodes synchronization, in order to make possible different types of data acquisition. The performance of this protocol was experimentally tested and results suggest significantly improved efficiency [21].

Network coding is widely used in the applications other than SHM. It is used for multimedia streaming and file distribution in the wireless networks [22]. In optical networks, network coding technique is used as a data protection against network failures [23]. It can also be used for the secure data transmission against the network attacks [24]. And the most important application of it is the improved usage of the network resources such are power consumption and data storage, and increasing data rates in wireless networks [25, 26, 27].

There are three main types of routing protocols: data-centric, hierarchical and location-based. Data-centric protocols receive a query and forward the requested data. Hierarchical protocols tend to form clusters. Cluster heads are in charge of the data compression and reduction of the overhead. They can be managed in the distributed manner. Location based protocols use a location of data to forward the information into required areas. We will be particularly interested in the distributed algorithms. One of those is presented in [28]. Yilmaz et al. propose a shortest hop algorithm that performs the load balancing in WSNs.

2.2 Network Coding

The network coding is a useful technique for gaining network efficiency. In order to maximise the throughput of multicast networks, Ahlswede et al. [29] introduces network coding for the first time. They show how to decrease the traffic load by encoding the received messages at intermediate nodes before forwarding them. They introduced a new class of problems that they named network information flow. They considered a point-to-point communication network in which a number of information sources were able to multicast to the set of destinations. They presented a technique that improves a network's throughput, efficiency and scalability. Instead of simply forwarding the packets of information they receive, the nodes combine a number of packets and transmit the coded packet. This technique is used to maximise the information flow in a network.

Ahlswede et al. show that it is suboptimal to restrain the function of nodes in the network to the routing solely. They show that the multicast capacity (a maximum rate at which a transmitter can send messages to a set of receivers) as symbol size approaches to infinity, is equal to the weakest path separating a source and any receiver. They show that multicast capacity is unachievable by simple routing, but can be achieved by means of network coding.

Li et al. additionally prove that the linear coding using infinite symbol size is sufficient to achieve the multicast capacity [30]. Koetter et al. prove previous results mathematically and provide a condition in a form of an expression to check the validity of a given linear multicast code [31].

Nagajothy et al. [32] investigate the benefits of network coding for enhancing the lifetime of nodes by decreasing the number of transmissions for grid and circular configurations. They study the interplay between the energy and the traffic rate, number and density of nodes.

Hong et al. propose a distributed algorithm for data gathering based on network coding that maximises the network lifetime. However, they neglect the energy spent for data reception. They claim that only energy consumed by transceiver modules contributes to the depletion of the initial energy in nodes [33].

Glatz et al. study network coding with the low computational power, using the butterfly structure. Their method prolongs the network lifetime and loosens resource constraints [34]. In the other paper they design an independent and autonomous network coding layer that is overhead-free. The scheme can be applied to the existing applications on top of the current routing mechanisms without changing the application or networking protocol. It is designed to work without centralized control and to conserve up to 29.3 % of the messages that need to be sent. This can be used combined with our proposed scheme [35].

Wang et al. propose an algorithm based on network coding for the 2-hop information exchange in WSN. Their algorithm is tested on both grid and random topology networks, and it achieves better performance compared to the simple forwarding [36].

Sikora et al. consider a WSN with one source node, one sink node, and multiple relay nodes placed equidistantly between them. They optimised the number of hops so as to achieve the lowest total transmission power, under the constraints of total bandwidth and the end-to-end data rate. They perform the optimisation with an additional constraint of the end-to-end delay [37].

Existing work considered a different topology with only one sink node. To the best of our knowledge, existing research has not considered the joint use of power control and network coding in the WSN used for SHM. Specifically, using bridge monitoring as an example, this paper proposes and investigates the performance trade-offs of power consumption (and thus network lifetime) and the network coding gain due to increased overhearing by high transmission power. Existing research work cannot be applied directly to a bridge monitoring because of the very specific, linear topology of sensor network on the bridge.

There are many approaches for the efficient power management in wireless networks at different stack layers. There are two main ways to manage the power supply: the improvement of the power efficiency of the system; and the prevention of the system's deconstruction due to the unfair power consumption. Zhang et al. list approaches for the optimisation at each layer of the WSN network stack. However, they do not provide any formal analysis of their findings. They explain how the application layer methods like data-fusion and compression, and the network layer methods like routing and distributed storage need to be optimised to avoid the obstruction of one another [38]. At the physical layer, power consumption can be reduced using the directional antennae. At the data-link layer, power consumption can be decreased by avoiding the unnecessary retransmissions; or collisions in channel access by allocating contiguous slots for transmission and reception. At the network layer, in order to reduce power consumption we consider the battery life in the route selection process, and we also reduce the frequency of control messages. At the transport layer, we preserve power by avoiding repeated retransmissions, and by handling packet loss in a localized manner and using power efficient error control methods [38, 39, 40].

2.3 Wireless Channel

Shah-Mansouri et al. investigated the passive loss tomography problem of coded packet in wireless sensor networks. They estimate path loss rates from source and intermediate nodes by inspecting the content of the coded packet at the sink. They suggest an algorithm for the estimation of link loss rates. They show that in coded packet wireless sensor networks, the proposed algorithm outperforms the Bayesian inference algorithm. They offer a precise solution for the loss rate of links in WSNs based on network coding [41].

Lin et al. study the loss due to the inference problem in sensor networks with network coding. They show how the fundamental connection between the path and link loss is changed when network coding is applied. They propose inference algorithms based on Bayesian principles to detect the set of lossy links in WSNs. Through simulation, they show that proposed algorithm achieves high detection and low false-positive rates [19].

Buta et al. performed extensive experiments and developed a method of distance estimation between sensor nodes using the received radio-frequency (RF) power level. They measure the received RF power level at various distances, and afterwards they use the recorded dBm values as an input in the distance determination formula. They notice the influence of humidity and temperature at the path loss exponent [42].

Due to the fading and the interference, data transmission via wireless links is prone to errors. Various propagation models are available. A reader can get more information concerning ways to simulate a realistic wireless channel in the book Modelling the Wireless Propagation Channel: A Simulation Approach with MATLAB [43], but that will not be the topic of this thesis. There are a number of models for a propagation channel under the effect of combined Rayleigh fading, shadowing and path loss. We opted to use the ideas provided in the work of Zorzi et. al [44]. Numerous models for the propagation channel exist and for analytical discussion we will use a model described in [45].

The multipath fading phenomenon has a powerful effect on the quality of wireless channel. One of the important models used to describe multipath fading phenomenon is proposed in [46]. This effect is particularly prominent if there is a lot of reflection, refraction and diffraction of radio waves by different objects. Thus, the transmitted signal reaches the designated receiver via more than one path causing a phenomenon called multi-path fading [46, 47]

In our analysis we consider the interference limited wireless channels (i.e., the interference power is much higher than the noise level, therefore the noise can be neglected) and we use the Shannon capacity formula to define the wireless link data rate [48].

Very common problem in wireless communications is the hidden terminal problem. When a number of nodes are outside of each other's communication range, it is possible that they send data at the same time to the same receiver, causing data loss and triggering packet retransmission. These packets therefore collide at the receiver. Data collisions do not always result in the data loss. If the power level at the receiver of one signal is stronger than all others by a certain margin, the receiver will successfully decode it. This phenomenon is called the capture effect [49].

Following papers are dedicated to the design of the bandwidth efficient network, and they are relevant as they consider a network with nodes deployed equidistantly in a line. One of the most interesting papers is certainly [37]. Sikora et. al observe a certain distance in the network covered by N-hop route (N nodes). For small values of N, the spatial frequency reuse is not recommended because of the potential damage due to the interference, so a simple time-division multiple access (TDMA) scheduling scheme is optimal. The division of distance into N hops results in an increase of the SINR at each node by N^{γ}, where γ is the path loss exponent. However, the end-to-end data rate is reduced by a factor of N. Thus, in order to achieve the same throughput a N-hop scheme should transmit at N times higher data rate at each node (hop) [37].

As the decrease of throughput is linear and the gain from the increased SINR is proportional to $\ln(N)$, it is possible to compute the optimum N for a given end-to-end data rate. Sikora et al. present the mathematical analysis of Shannon capacity in linear networks with N nodes. This kind of system is subject to the overhead due to the delays and additional processing at nodes. In their other paper [50] they come up with a formula (approximation) for the number of nodes that consume the lowest power. They consider the capacity to be a special case of either fixed-rate relying or rate adaptive relying. Those two cases are identical when nodes deployed in the network are equidistant. Authors assumed that point-to-point links are frequency-flat fading channels, while system has a power constraint.

2.4 Routing and Optimisation of the Network

As WSNs based SHM systems are becoming more popular, new data communication concepts are proposed [36]. Inherited issues of WSNs such as power consumption of wireless nodes and consistent interaction mechanisms are explored in [51, 52, 53, 54].

The transmission power of nodes can vary when battery level is low or when there is a physical damage of a node. Blom et al. argue that the actual transmitted power levels at a sensor node is very important for the design of the efficient power control algorithms - that are independent of the operational condition of the wireless sensor device [55].

Yuan et al. investigate the effect of the optimisation of sensor nodes' transmission power on network lifetime. They jointly optimise transmission power, routing, and source quantisation in a wireless sensor network. They show how the optimisation of the overall network can be separated into two problems: source coding at the application layer and power control at the physical layer [56].

Liu et al. describe the design and deployment of WSNs for SHM. They design the architecture based on application requirements such as energy efficiency, routing and high-frequency sampling. They also derive the analytical model to determine energy consumed for the transmission and the reception of packets. Liu et al. perform a cluster-based modal analysis. They divide the network into clusters and identify the vibration characteristics in each cluster that are afterwards assembled together. They optimise the cluster size to minimise total energy consumption. Also, they evaluate by experiments and simulations, the effectiveness and efficiency of the proposed cluster-based modal analysis [57].

Ferentinos et al. proposed an optimisation of application specific WSN that has connectivity and energy consumption constraints, based on genetic algorithms. They provide a fitness function, which is used to assess the network performance such are the status of sensor nodes, the choice of cluster heads. However, they refer to a single-hop routing scheme that is hierarchical. An interesting criterion to use as an objective function is the network lifetime – the time it takes until the network gets partitioned in a way that makes the collection of data from a part of a network impossible [58]. Some papers used multiple objective parameters [59].

To the best of our knowledge, the existing research has not considered the joint use of power control and network coding in WSN used for SHM. We emphasize that the requirements for SHM applications are very different than the rest of WSN applications. Even though these papers are valuable for the area as they explore the interplay between network coding and power consumption, linear multi-sink networks are not discussed yet. As this thesis is focused on a very specific topology directions suggested in the related research are sub-optimal or at times even not applicable. We will take into account specific conditions in the network that are unique to our application. We will try to jointly assess the interplay between network coding and power control, in light of strict reliability requirements.

Last few years there is an increasing interest in planning of WSNs. Researchers considered various objectives such are fault tolerance, coverage area, network connectivity, network lifetime, data fidelity, number of nodes, load balancing and energy efficiency. Zou et al. optimise the sensor nodes' placement with the objective to increase the sensing coverage [60]. Their work measures the probability that an event would happen without being detected.

Fu et al. focused on determining the optimal nodes positions that would reliably diagnose the health of monitored infrastructure, while at the same time they tried to minimise the power consumption for data collection and relying. They observed the energy-balanced routing tree and optimal grid separation in order to diminish the energy consumption [61].

Wang et al. analytically derive the formula for network lifetime optimisation focusing on small-span planar networks. They also proposed a suboptimal iterative approach for large-span multi-hop planar networks' lifetime optimisation [62].

Sagduyu et al. developed a number of networks cross-layer algorithms for the multicast in wireless sensor networks. However, the potential of the broadcasting remains unexploited. They show how the network coding based technique increases the achievable throughput rates for single-source multicasting in wired networks. They conclude that a joint design of medium access control under omnidirectional transmissions, half-duplex operation and interference effects is the best way to proceed [63].

In order to maximise the operational lifetime of sensor nodes which have limited battery supply, sensors need to transmit data at a limited power and hence form a multi-hop network [64]. Based on the transmission power level in this thesis we consider three different network connectivity cases: one-hop, two-hop and three-hop direct-connectivity. Let us further assume omnidirectional transmissions, as previously explained.

2.5 Scheduling and Optimisation of the Network

Diaz and Leung propose a randomized scheduling algorithm for data aggregation in WSNs, RandSched. This distributed TDMA-based algorithm consists of two phases: a testing phase (decides if a set of nodes can be scheduled concurrently collisions-free) and a finalization phase (sink node creates a finalization packet that contains the information about the aggregation process) [65].

Amdouni et al. focuses on a traffic-aware time slot assignment that minimises the schedule length for three and linear topologies. A smaller schedule length results in reduced network delays and changes the energy consumption. They formulize the problem as a linear program and provide their results on the optimal number of slots. They suggest a delay-optimised algorithm with two heuristics that reduces energy consumption and storage capacity. However, their approach cannot be combined with Network Coding (NC) and they used only one sink node, which is not useful for SHM of bridges especially very long bridges [66].

A level based scheduling that minimises the schedule length is proposed in [51]. Firstly, a linear network is built from the initial network. Each node in the same corresponds to a level in the original network. Afterwards, the schedule of transmissions in the original network can be deducted by colouring the original network. In this paper, each node has only one packet to transmit, and only nodes that hear each other interfere.

Gandham et al. compute the theoretical lower bound of the number of slots required in linear, multi-line and tree network topologies. Their solution is close to the bounds for linear and multi-line topologies [67].

Thus, some researchers already worked on a problem of delay in linear wireless networks. However, they used a system model with only one sink, which is simpler and does not satisfy needs of application that is the topic of this thesis. We use two sink nodes, one at the each end of a linear structure.

2.6 Adaptive Network Coding Algorithms

Yu et al. jointly optimise source coding (data quantisation), routing and power control strategy at each sensor, in order to obtain an efficient solution for optimisation of the overall network [68].

Katti et al. worked on opportunistic wireless network coding. They suggest the opportunistic scheduling in the multiple unicast case in order to increase the throughput. Their method performs the optimal scheduling using information about the state of neighbouring nodes. The method makes use of the overhearing capability based on which a node makes an optimally decodable network codes that can always be decoded by the neighbouring nodes resulting in the improved throughput [26].

One of the first network coded based protocols for WSNs was AdapCode. It achieves the high reliability due to its adaptive network coding scheme that uses information it has about the link quality. AdapCode is not suitable for the health care applications as it does not support the multicast [27].

Shwe et al. point out that AdapCode might be missing out on potential network coding opportunities as it includes only some of the neighbouring nodes in the network. They suggest an improved version of AdapCode that uses a power efficient protocol to discover all neighbours of a node. They are trying to develop a more efficient protocol as more opportunities for network coding result in the reduction of power consumption [69].

3. NETWORK CODING BASED ROUTING SCHEME

3.1 Chapter Abstract

Wireless Sensor Networks (WSNs) have been deployed in order to perform the Structural Health Monitoring (SHM) of linear structures. Wireless sensors are deployed on bridges to take (sense) a variety of measurements. SHM applications have a potential to produce a high volume of sensing data because of the nature of certain metrics of interest. This results in the aggressive power consumption and consequently, in the decrease of the lifetime of battery-run networks. The key idea is to make use of network coding to reduce the number of transmissions needed for forwarding of sensing data to the sink nodes. Thus, the network coding technique can be used to improve the network efficiency and extend its lifetime. By increasing transmission power, we change the direct-connectivity of nodes and control the number of nodes that overhear transmitted messages in order to achieve the capacity gain by use of network coding. Specifically, we observe a bridge with the fixed length. Sensor nodes are deployed at a uniform distance along one or both sides of the bridge. In this chapter we present the way in which the proposed algorithm functions for different layouts and connectivity. We present the mathematical analysis of our proposed algorithm, and provide the closed form expressions for the total number of transmissions in the network. By simulation, we verify our mathematical analysis. This solution is by no means limited to the bridge monitoring and it can be used for the monitoring of different type of linear infrastructure. The results presented in this chapter are published at PIMRC'12 [70].

3.2 Related Work

Nagajothy et al. [32] investigate the benefits of network coding for enhancing the lifetime of nodes by decreasing the number of transmissions for grid and circular configurations. They study the interplay between energy and traffic rate, number and density of nodes. In [33], Hong et al. propose a distributed algorithm for data gathering based on network coding that maximises the network lifetime. However, they neglect the energy spent for data reception. They claim that only the energy consumed by transceiver modules contributes to the depletion of the initial energy in nodes.

Glatz et al. [34] study network coding with the low computational power, using the butterfly structure. Their method prolongs the network lifetime and loosens resource constraints. In their other paper [35] they design an independent and autonomous network coding layer that is overhead-free. The scheme can be applied to the existing applications on top of the current routing mechanisms without the change of the application or the networking layer protocols. Layer is designed to work without centralized control and it conserves up to the 29.3 % of the messages that need to be sent. This approach can be used together with our proposed scheme.

Wang et al. [36] propose the algorithm based on network coding for the 2-hop information exchange in WSNs. Their algorithm is tested on both grid and random topology networks, and it achieves better performance compared to the simple forwarding.

Sikora et al. [10] consider a WSN with one source node, one sink node, and multiple relay nodes placed equidistantly between them. They optimised the number of hops in order to achieve the lowest total transmission power consumption, under the constraints of total bandwidth and the end-to-end data rate. They perform the same optimisation with an additional constraint of the end-to-end delay.

Existing work considered a number of topologies with only one sink node. To the best of our knowledge, the existing research has not considered the joint use of power control and network coding in the WSNs used for SHM. Specifically, using bridge monitoring as an example, this paper proposes and investigates the performance trade-offs of power consumption (and thus network lifetime) and network coding gain due to increased overhearing by high transmission power. Existing research work cannot be applied directly to the bridge monitoring because of the very specific, linear topology of sensor nodes deployed on the bridge.

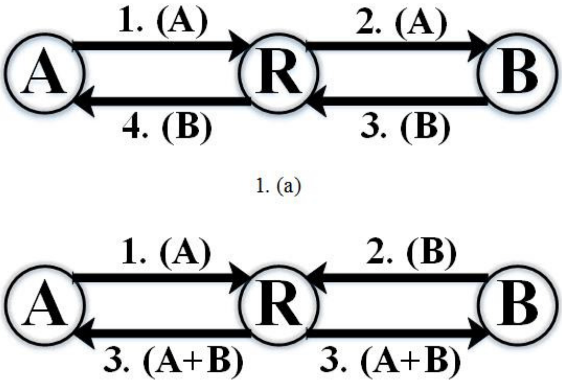
3.3 Motivation

Recent developments in Structural Health Monitoring (SHM) led to the deployment of wireless sensor networks (WSNs) instead of wired sensing systems. It is cheaper, easier and quicker to use WSN than wired sensor network. Wireless sensor nodes are not intrusive and they integrate well with the most of the structures. WSNs are scalable to a large number of nodes compared to the wired sensor networks which are restricted in size. Scalability allows dense coverage of the infrastructure systems. However, the nodes' power supply is often internal and limited. In addition to the limited power supply, WSNs impose a variety of limitations for system design due to the limited bandwidth and high packet loss rate.

The main objective of SHM systems is to provide the high quality data, based on which the structure quality is assessed and further decisions concerning the infrastructure safety are made by civil engineers. Data sampling should be performed simultaneously at all nodes. Additionally, the sampling rate should be high as the measured vibrations in structure change quickly. Due to the high sampling frequency and dense sensing, the traffic load of raw data is very high. Thus, it is recommended to decrease the data volume before data are transferred to sink nodes. However, applied data processing techniques should not harm the fidelity of data as that would consequently degrade the sensitivity of SHM systems. The performance of a SHM system depends on the quality of data received at the sink(s) and the packet loss rate. Thus, the goal of SHM systems is reliable and lossless communication over potentially a large span network.

It is essential to decrease the data overhead as much as possible and carefully use scarce network resources. Data aggregation techniques are not the right choice for SHM. They are frequently based on linear combination of the values of all received messages (e.g. averaging, minimizing or maximising). In monitoring infrastructure, it is important to deliver the accurate information to civil engineers so they can make the best decision. A typical aggregation technique results in the information loss.

A good way to preserve the original sensed data and yet decrease the volume of data is to deploy the network coding. The idea of network coding is to allow the coding of data at intermediate nodes and forward coded packets, each of which is generated by combining and encoding more than one packet received from possibly multiple sources. The simplest encoding scheme is the XOR operation applied to each corresponding bits of two packets of the same size. In traditional routing schemes, nodes forward data packets from each neighbouring node towards the sink separately. Let us observe the communication between nodes A and B, via relay node R (the top part of Figure 3.1).



1. (a)

1. (b)

Fig. 3.1. Nodes A and B are exchanging messages (A) and (B) via relay node R. a) Conventional forwarding requires 4 transmissions and 4 transmission slots. b) Network coding based routing requires 3 transmissions and 3 transmission slots.

Firstly, node A sends its message (A) to the relay node R, which forwards the message (A) to the node B. In the third time slot, the message from node B, labelled as (B), is sent to the relay node R, and then it is forwarded to the node A in the fourth time slot. The total number of transmissions needed for this communication is four. However, if both nodes A and B send their messages to the relay node R, in respectively time instances 1 and 2, node R can XOR the received messages and broadcast the coded packet (A+B) in the third time slot (the lower part of Figure 3.1). Once nodes receive that message, they will be able to XOR it with their own message, and to decode the information sent from the other node. For instance, the node A XORs the received message (A+B) with its own message (A), and extracts the message (B). Node B will extract the message (A) from XOR combination of the messages (B) and (A+B). Thus, this approach requires 3 transmissions for the communication at the expense of a small increase in the computation complexity.

The network coding is a useful technique to improve the network efficiency. In order to maximise the throughput of multicast networks, Ahlswede et al. [29] introduces network coding for the first time. They show how to decrease the traffic load at nodes by encoding received messages in the intermediate nodes prior to their forwarding.

Most of the previously applied solutions use different forms of data aggregation. However, data aggregation is not necessary a good idea for SHM applications. Data aggregation indeed reduces data volume at the expense of degradation of data quality. In the case of SHM, it is very important to collect and deliver non-distorted data. Thus, any form of data aggregation that uses maximisation, minimization or averaging is not desirable as the proper operation of SHM systems directly depends on the quality of collected and received data. In order to ensure the desirable operation and outcome of the SHM systems, we apply network coding to reduce the data volume. At a small expense of added computation, we achieve a significant reduction in data volume, while transferring non-distorted data to the sink node.

Due to the specific linear layout of sensor nodes at the bridge, the SHM of bridges should be studied on its own. In this chapter, we present the algorithm that is designed specifically for SHM of bridges and linear structures based on network coding. We show that proposed algorithm achieves a significant improvement in terms of the number of transmitted messages and energy consumption. We show that the increase in transmission power can change not only the network topology, but also the number of necessary transmissions. We provide the mathematical analysis of proposed algorithm for different connectivity and layouts of sensor nodes. At the end, using extensive simulations we test the performance of our proposed algorithm. To the best of our knowledge, existing research has not yet considered the joint use of power control and network coding in the WSN used for SHM. Specifically, using bridge monitoring as an example, this paper proposes and investigates the performance trade-offs of power consumption (and thus network lifetime) and the network coding gain due to increased overhearing by high transmission power. Existing research work cannot be applied directly to the bridge monitoring because of the very specific, linear topology of sensor nodes on the bridge.

3.4 Scenario

3.4.1 System Model

It is expected that sensor nodes will be deployed on either both sides of bridge (common for wide bridges) or on only one side of bridge (common for narrow bridges). Thus, nodes will appear in a single-line or in a double-parallel-lines.

For now, we will assume that nodes are equidistant. Also, the sink node is likely to be installed at the end of the bridge, as that is a convenient place to transfer collected data over Internet. This is because it is easier to install a node with the stable power supply at the end of the bridge, and still not to disturb the normal function of the bridge.

Since the most important feature of the SHM systems is reliability, we consider that sink nodes are placed at both ends of a bridge. That way we enhance the reliability of the overall network. In the case of the link loss or strong interference at one of sink nodes, the other sink node will still receive a copy of the missing data. The cost of this is expected to be small compared to a bridge failure.

This chapter will introduce the mechanism of data routing, and for simplicity we will assume a perfect MAC layer. No losses will happen due to the interference and packets collisions.

3.4.2 Routing scheme with Network Coding

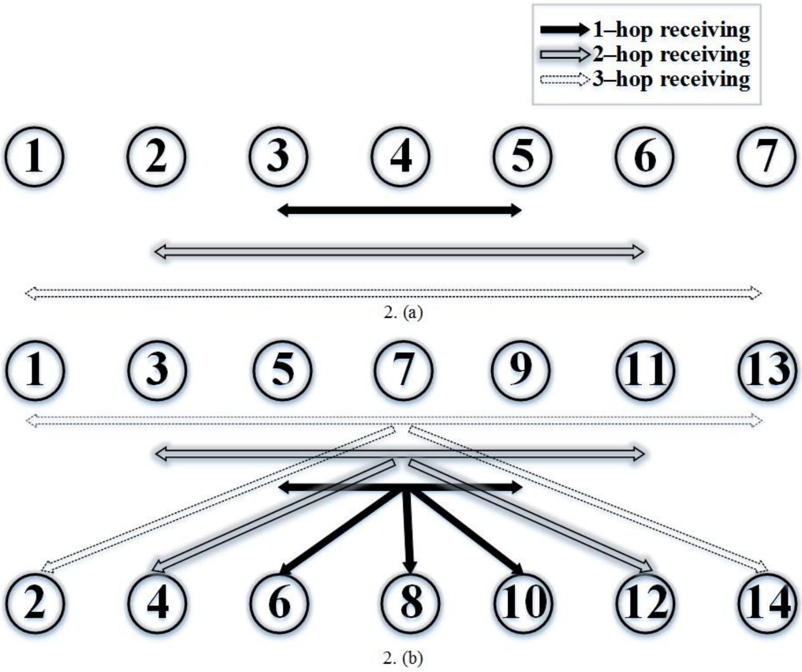


Fig. 3.2 Connectivity of nodes. a) Single-line layout of nodes. b) Double-parallel-lines layout of nodes.

We consider two different layouts of sensor nodes at the bridge: a single-line layout and a double-parallel-lines layout as shown in Figure 3.2. As shown in the upper part of Figure 3.2, each node can communicate with its immediate (one-hop) neighbours on both its right and left sides. For example, if a node with the ID 4 transmits, nodes with the IDs 3 and 5 decode its message (in Figure 3.2a). In the double-parallel-lines scenario, in addition to the immediate neighbouring nodes on the same line, each node additionally communicates with the neighbouring nodes from the opposite line. Thus, when node with ID 7 transmits, nodes with IDs 5, 6, 8, 9, and 10 can decode the message. In both cases, sink nodes are positioned at the ends of each line.

We put sink nodes on both ends of each line to provide the redundancy for the SHM in the case of wireless link failures. In addition, three sub-cases are discussed. We say that a node can communicate directly with its immediate neighbours if only one-hop neighbouring nodes can be reached directly. For example, when node with ID 4 transmits and nodes with IDs 3 and 5 receive its message, we will talk about one-hop direct-connectivity, as shown in the Figure 3.2a. Correspondingly, when node with the ID 7 transmits, nodes with the IDs 5, 6, 8, 9 and 10 decode its message in Figure 3.2b. In the case of one-hop direct-connectivity, nodes transmit with the minimum power that is just high enough to maintain the network connectivity. We say that nodes communicate directly with the two-hop neighbours if each node in addition to the nodes one-hop away, can directly reach its neighbouring nodes located two-hops away. For example, when node with the ID 4 transmits, nodes with the IDs 2, 3, 5 and 6 decode its message in the single-line scenario in Figure 3.2a. Similarly, when node with the ID 7 transmits, nodes with the IDs 3, 4, 5, 6, 8, 9, 10, 11 and 12 can decode the message in the double-parallel-lines case in Figure 3.2b. If nodes further increase the transmission power, they can achieve three-hop direct-connectivity. For instance, when node with the ID 4 transmits, nodes with the IDs 1, 2, 3, 5, 6 and 7 can decode the message in the single-line scenario shown in Figure 3.2a. Likewise, when node with the ID 7 transmits, nodes with the IDs 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13 and 14 decode the message in the double-parallel-lines case in Figure 3.2b. We here propose a network coding algorithm for bridge monitoring.

3.4.3 Single-line layout of sensor nodes

We will now discuss the networks whose nodes are deployed in a single-line. This is often the case when narrow bridges are monitored.

3.4.3.1 One-hop direct-connectivity subcase

To present the proposed routing algorithm with network coding, we consider the single-line topology (Figure 3.2a). We assume that each node has transmission power adjusted to communicate only with its immediate (one-hop) neighbours. At this point for simplicity we also assume that perfect transmission schedule is in use, so that transmitted packets do not collide and messages can be properly received by their intended receiving nodes. While we present the scheme we assume that no losses happen due to the poor conditions in the wireless channel either. We use an example with a six-node network as shown in Figure 3.3 to illustrate our routing scheme with network coding.

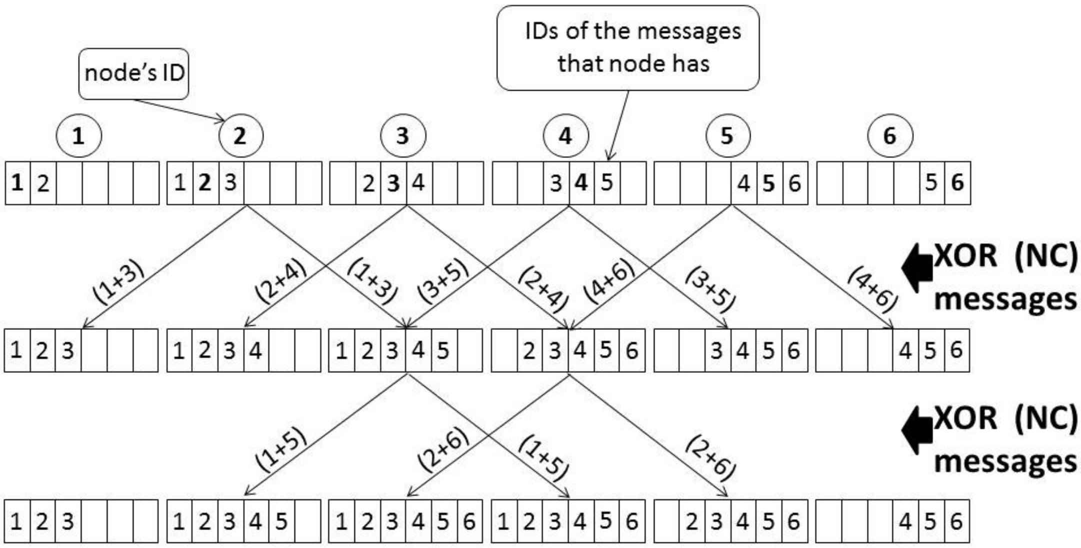


Fig. 3.3 Network coding based algorithm for the single-line layout. Each node can reach only its immediate neighbours.

Initially, each node performs sensing and creates one packet of sensing data. That packet needs to be transferred to the sink nodes at the both ends of a bridge. In this step each node broadcasts its original message. Thus, at this point each node will be aware of its own message and messages from its immediate neighbours. As shown at the top row of Figure 3.3, each node (with index in boldface) is associated with six rectangles. Each rectangle belonging to a node (say *k*) shows the index of the source node associated with the message that has been received by node *k* after each round of message exchanges. For example, after the first round of message exchanges, node 3 has received messages from itself (node 3) and its immediate neighbouring nodes 2 and 4, as indicated in the top row of rectangles in Figure 3.3. In the next step, each node XORs the two received messages just from its two immediate neighbours and broadcasts the coded packet. For example, node 4 XORs messages from nodes 3 and 5, and broadcasts the coded message to nodes 3 and 5. Since node 3 knows its own message, performing an XOR operation on coded packet received from node 4 and the original message of node 3 will enable node 3 to decode the original message from node 5. Similarly, node 3 receives and decodes the original message from node 1 as well. As a result, the original messages that each node holds after two rounds of message exchanges are shown in the middle row of rectangles in Figure 3.3. For example, node 3 now holds the original messages of nodes 1, 2, 3, 4 and 5. In essence, after two rounds of message exchanges using network coding, each node possesses the original messages from two-hop neighbouring nodes. Therefore, each node excluding the sink nodes, behaves like a relay node for its immediate neighbours, as shown in the Figure 3.1a. Each node XORs a broadcasted message received from its two neighbours with a message it already has, in order to decode new messages. This process continues until the last round of broadcasting, when no node receives more than one coded (XORed) message. The rest of messages can simply be forwarded to the sinks via the nearby nodes. The number of transmissions can be reduced further, if we increase the transmission power.

3.4.3.2 Two-hop direct-connectivity subcase

Contrast to the case of one-hop direct-connectivity, after the initial phase only nodes that receive messages from two two-hop neighbours behave like relay nodes in the first round of network coding. In the first round, they XOR the messages received from the two-hop neighbours. After that, the algorithm continues in the same manner as it functioned for the one-hop connectivity case.

If the transmission power of a node increases, so that a message from each node can directly reach its second-hop neighbours we will talk about two-hop receiving (two-hop connectivity). In the case of two-hop receiving the number of total transmissions decreases further due to the overhearing capability. Initially, all nodes perform the sensing, based on which they create their own packets. These packets should be forwarded to sink nodes at the both ends of the linear structure (e.g. bridge). Thus, nodes will broadcast their own messages (collected from their own sensing units) in the first stage of the algorithm. After this stage, all nodes will be aware of their own messages, and messages originating from their first-hop and the second-hop neighbours. In the Figure 3.4, this algorithm is presented for the case of a linear network that consists of nine sensor nodes that operate as relay and sensing units. As shown at the top row of the Figure 3.4 each node (with index in boldface) is associated with nine rectangles. Each rectangle belonging to a node (say *k*) shows the index of the source node associated with the message that has been received by node *k* after each round of message exchanges. For example, after the first round of message exchanges, node 5 has received messages from itself (node 5) and its two physically nearest neighbouring nodes from each side, namely from nodes 3, 4, 6 and 7 as indicated in the top row of rectangles in Figure 3.4. This concludes the first phase of the algorithm.

In the next phase, each node that received two messages, one from the second-hop neighbour from each side, XORs those two messages and broadcasts the coded packet. For example, node 5 XORs messages from nodes 3 and 7, and broadcasts the coded message to the nodes 3, 4, 6 and 7. Since the nodes 3 and 4 know the message (3), performing an XOR operation on the coded packet from the node 5 and the original message of nodes 3 will enable nodes 3 and 4 to decode the original message from the node 7. Similarly, nodes 6 and 7 receive and decode the original message from node 3 as well. As a result, the original messages that each node holds after two rounds of messages exchange are shown in the middle row of rectangles in Figure 3.4. For example, node 3 now holds the original messages of nodes 1, 2, 3, 4, 5, 6 and 7. In essence, after two rounds of message exchanges using network coding, each node possesses the original messages from one-hop, two-hop, three-hop and four-hop neighbouring nodes. As shown above, each node excluding the sink nodes, behaves like a relay node for its two-hop neighbours, Figure 3.1a. Each node XORs message received with a corresponding message that it already has, as a mean to decode new messages. This process continues until the last round of broadcasting when no node receives more than one coded (XORed) message from its two-hop neighbours. This concludes the second round of the algorithm. The rest of messages can simply be forwarded to sinks via the nearby nodes.

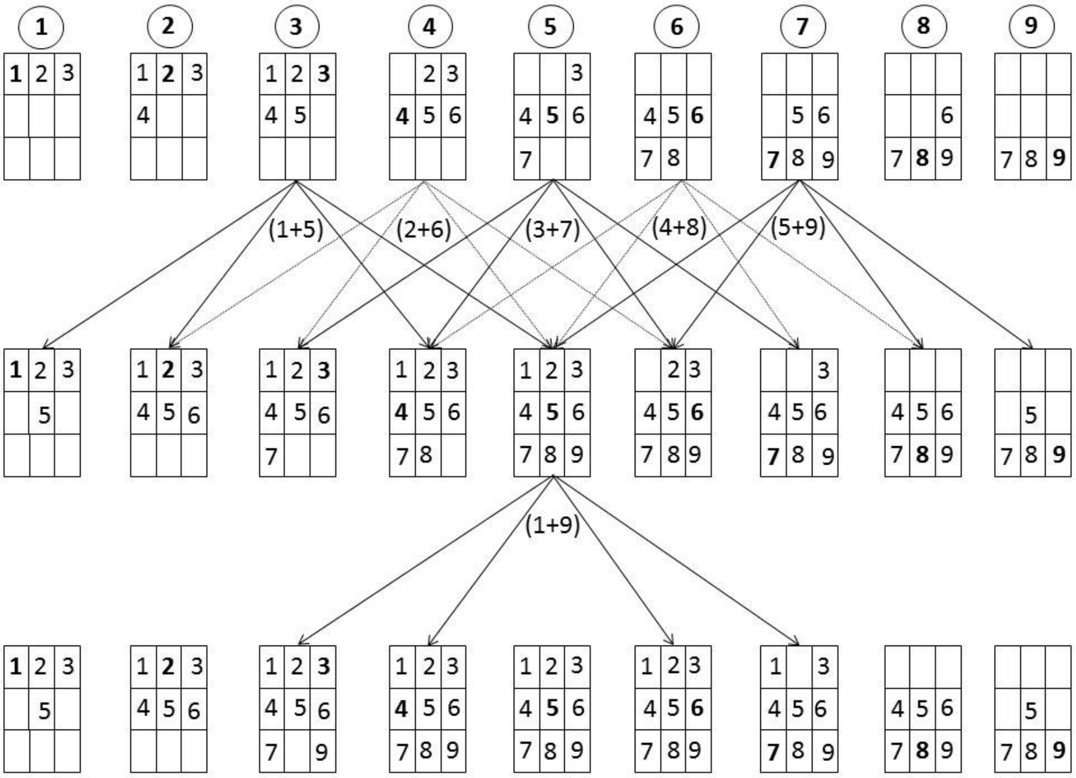


Fig. 3.4. Network coding based algorithm for single-line layout. Each node can reach directly its second-hop neighbours.

3.4.3.3 Three-hop direct-connectivity subcase

The number of transmissions can be reduced further with the increase of the transmission power. Three-hop receiving subcase operates in a similar manner. There will be even more of overhearing. First and third phase of the algorithm will be the same as in the case of one-hop and two-hop direct-connectivity schemes, and all messages will be overheard by one-hop, two-hop and three-hop distant neighbours. The only change is that in the second phase, nodes qualify for transmission only if they have received two different messages from their three-hop neighbours in the previous stage, Figure 3.5.

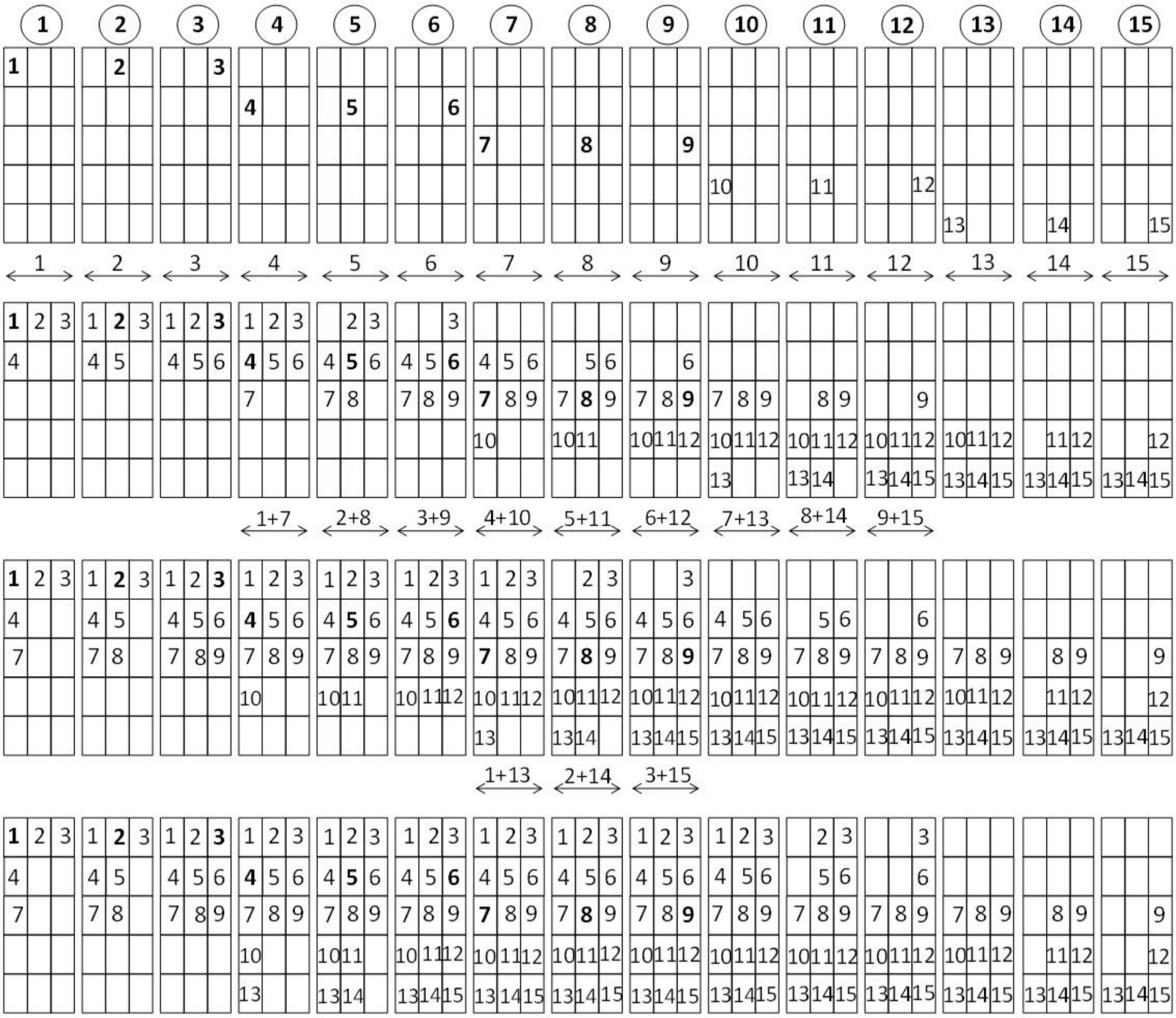


Fig. 3.5. Network coding based algorithm for single-line layout. Each node can reach directly its third-hop neighbours.

For instance, in the network coding phase (starts at the second row of rectangles) a node with the ID 4, received a message from nodes 1 and 7 and it will encode them and broadcast the coded message. In the next instance (the third row of rectangles), a node with the ID 7 did not receive two different messages from its three-hop neighbours so it will not participate in the network coding any more. We may notice that the number of necessary transmissions decreases significantly due to the overhearing.

3.4.4 Double-parallel-lines layout of sensor nodes

We will now discuss the networks whose nodes are deployed at both sides of a bridge, resembling double-parallel-lines topology. This is often a case when wide bridges are monitored.

3.4.4.1 One-hop direct-connectivity subcase

Sometimes, civil engineers place sensor nodes along both sides of bridge, creating a topology that resembles double-parallel-lines. This is the case for very wide bridges where sensors placed at one side of bridge do not cover the area at the opposite side of bridge. Thus, these sensor nodes are not able to collect high quality data concerning the events at the other side of bridge. Therefore, we will now discuss how our algorithm operates in the case of double-parallel-lines layout of nodes.

We present the routing algorithm that uses network coding for double-parallel-lines topology as shown in the Figure 3.2b. We assume that each node has transmission power adjusted to communicate only to its immediate one-hop neighbours, including the immediate neighbours placed on the opposite line. For simplicity we keep the assumption of perfect MAC layer and lossless channel. We present the algorithm using a ten-node network as an example, Figure 3.6. Initially, each node performs sensing and creates a packet of sensing data that needs to be transferred to sink nodes at the both sides of bridge (total 4 sink nodes, one at each end of each line). The next step is that each node broadcasts its original message. Thus, at this point each node will be aware of its own message and the messages from its immediate neighbours. As shown at the top row of rectangles in Figure 3.6, each node (with index in boldface) is associated with ten rectangles. Each rectangle belonging to a node (say *k*) shows the index of the source node associated with the message that has been received by node *k* after each round of message exchanges. For example, after the first round of message exchanges, node 5 received messages from itself (node 5) and its immediate neighbouring nodes 3, 4, 6, 7 and 8, as indicated in top rows of rectangles in Figure 3.6. This concludes the broadcasting phase of the algorithm.

In the next phase, each node XORs two received messages (that represent the information obtained in the previous instance of algorithm) just from its two immediate neighbours in the same line, and broadcasts the coded packet. For example, node 5 XORs messages from nodes 3 and 7, and broadcasts the coded message to the nodes 3, 4, 6, 7 and 8. Since nodes 3 and 4 know the message (3), performing an XOR operation on the coded packet from node 7 and the original message of node 3 will enable nodes 3 and 4 to decode the original message from node 7. Similarly, nodes 6 and 7 receive and decode the original message from node 3. As a result, the original messages that each node has after two rounds of message exchanges are shown in the two middle rows of rectangles in Figure 3.6. For example, node 3 now holds the original messages of nodes 1, 2, 3, 4, 5 and 6. In essence, after two rounds of message exchanges using network coding, each node possesses the original messages from one- and two-hop neighbouring nodes. As it is already shown, each node excluding the sink nodes, behaves like a relay node for its immediate neighbours from the same line, as in Figure 3.1b. Each node XORs messages received from its two neighbours with a message it already has, as a mean to decode new messages. This process continues until the last round when no node receives more than one coded (XORed) message from nodes in the same line. This concludes the network coding phase. The rest of the messages can simply be forwarded to sinks via nearby nodes. The number of transmissions can be reduced further, if we increase the transmission power, and consequently the level of direct-connectivity. Contrary to the case of one-hop direct-connectivity, after the initial phase, only nodes that receive messages from the two two-hop neighbours in the same line behave like relay nodes in the phase of network coding. Sensor nodes XOR messages received from the two-hop neighbours with the corresponding message that they already have. Nodes on the other line are able to decode coded messages too due to the overhearing. Upon network coding phase, the rest of messages can simply be forwarded to sinks via nearby nodes.

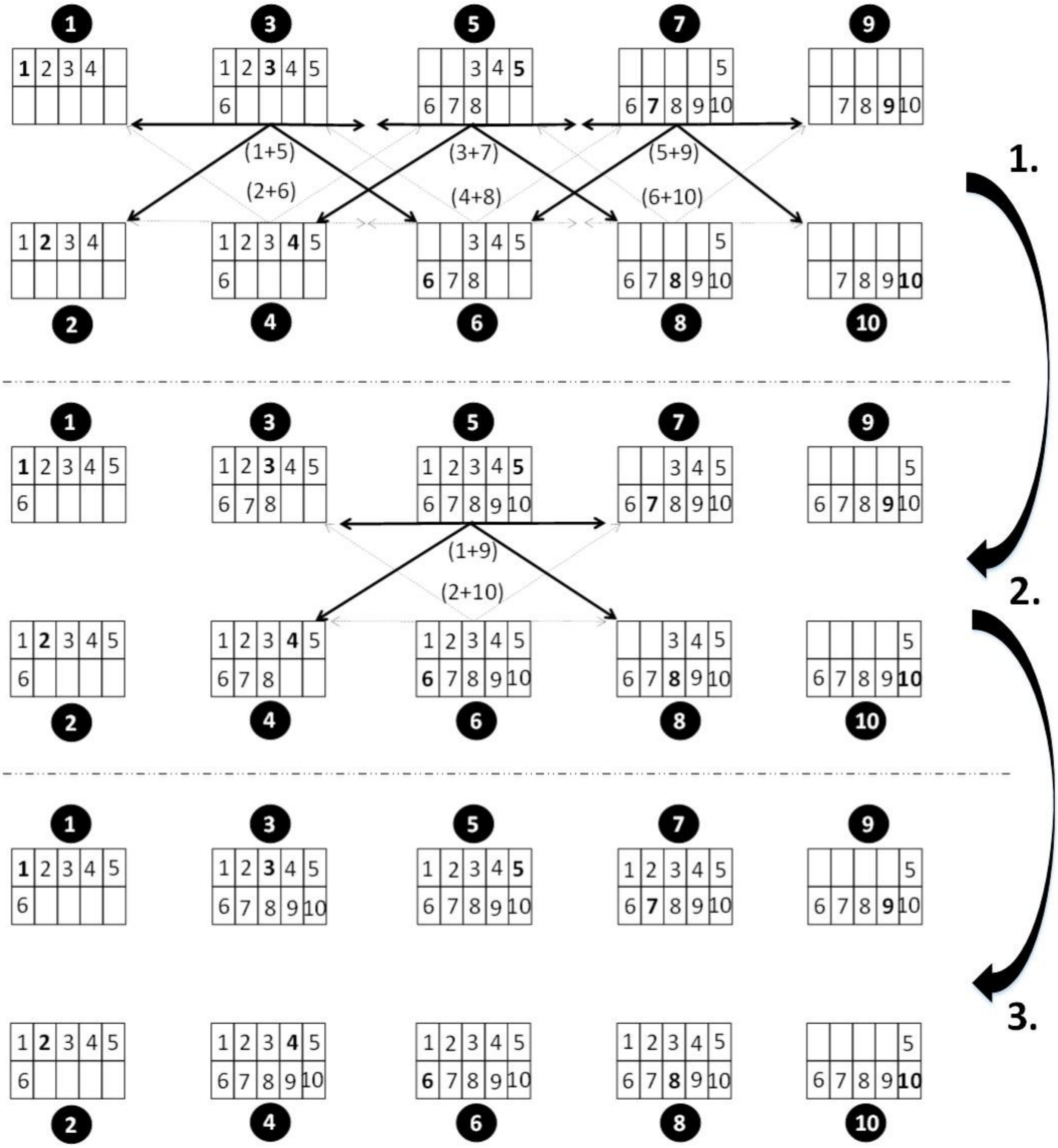


Fig. 3.6. Network coding based algorithm for double-parallel-lines layout. Each node can directly reach only its immediate neighbours.

3.5 Mathematical Analysis

We presented the network coding based message forwarding. Now, we will explore performance of the proposed message forwarding scheme. We applied network coding in order to decrease the total number of transmissions needed to relay all messages to sink nodes, but yet to preserve the original content of packets. The number of total receptions per node will not be the topic of this chapter as each node can extract only one new message at the time from any received packet. We will assume that node sleeps if it does not expect a message. Thus, all nodes will receive the same number of packets. Therefore, in the ideal conditions when no losses happen, the total number of receptions will be the same for the regular data forwarding and for the network coding based routing. As the energy consumed for these simple computations is much lower than the energy consumed by transmitting/receiving unit, we will assess the performance of the algorithm looking only into the number of required transmissions.

3.5.1 Single-line layout of sensor nodes

The total number of transmissions required until all generated packets reach sink nodes consists of three parts: broadcasting of the original nodes' messages, transmission of the network coded messages and conventional forwarding of simple packets coming from the other nodes. The greater the proportion of messages is coded, the better the network's performance. Not only that network coding reduces the energy consumption, it also improves the capacity. Let *N* be the number of nodes in the network deployed in a single line in the further text.

3.5.1.1 One-hop direct-connectivity subcase

Consider that nodes can directly communicate only with their next-hop neighbours. Following our algorithm, we notice that rules imposed by the scheme determine how many transmissions each node will perform in which phase.

We present the idea using a 7-nodes network arranged in a single-line, Figure 3.7. Nodes can communicate only with their immediate neighbours. As previously explained, in the first phase nodes perform sensing, they create a packet out of sensed data and broadcast their own message. We refer to this phase as the broadcasting phase.

In the next phase, each node that received two different messages from its immediate neighbours participates in the network coding phase. This phase consists of several separated instances (α), which is represented in the figure as the number of rows in the network coding phase. In the first instance of this phase, each node that received two different messages from its immediate neighbours in broadcasting phase transmits an XORed packet that contains those two messages. In the next instance (the next row of rectangles), all nodes that received two different messages in the previous instance XOR those messages. This will continue until no node has received two different messages in the previous instance. Once that happens, the network coding phase will be finished.

In the last phase, all remaining messages will be forwarded to sink nodes (nodes with the IDs 1 and 7). Forwarding phase consists of a number of instances (β) that is represented as a number of rows in the figure. To understand this phase we will observe separately nodes with the IDs lower than the middle node (left hand half), and nodes with the IDs higher than the middle node (right hand half). Let us explain the forwarding phase using the right hand half of the nodes. To calculate the total number of transmissions, we simply multiply the total number of transmissions performed by nodes from the right hand side by two, as the situation is symmetrical. Each sensor node that has a message that is still unknown to their first-hop neighbour from the right side will transmit in this phase. A node can have a number of messages at the beginning of the instance (one row) that is required by its right hand neighbour. In the fifth row (the first instance of the forwarding phase), node 5 has two of those messages (messages with the IDs 1 and 2 are required by node 6). Therefore, node 5 will perform two transmissions in the first instance of the forwarding phase. Node 6 will also perform two transmissions as sending messages with ID 3 and 4 is required by node 7.

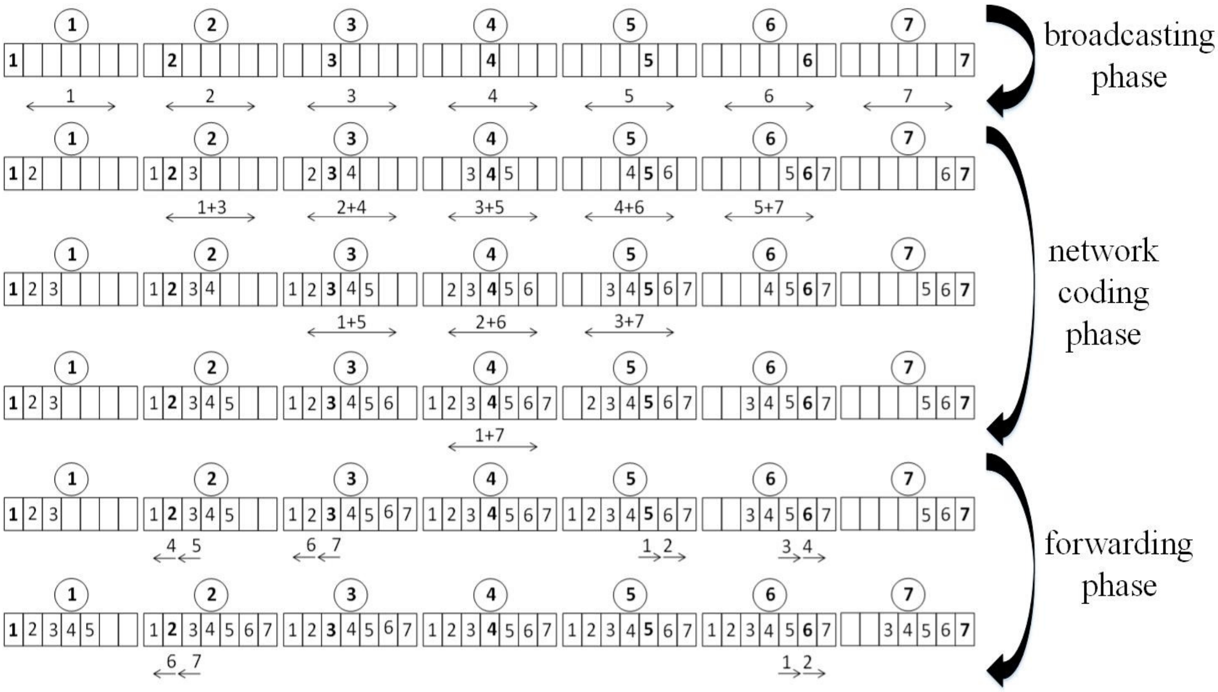


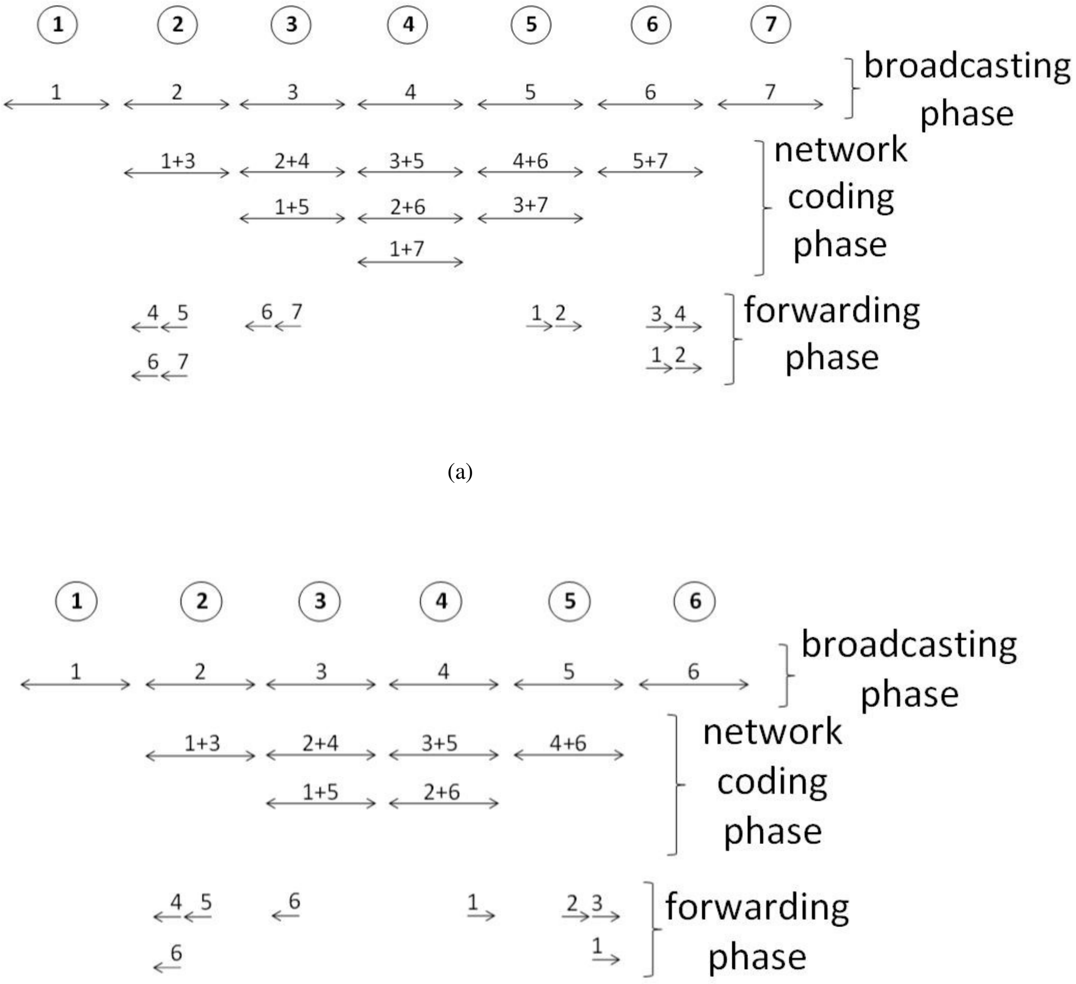
Fig. 3.7. Single-line layout, one-hop direct-connectivity. The three phases of the proposed algorithm.

If we think of the previous figure as of a simple flow of messages, we will be able to spot patterns and use them to provide a closed form expression for the total number of transmissions. Let *N* be the total number of nodes deployed in a line. In the Figure 3.7, *N* is equal to 7.

- In the **broadcasting phase**, each node in the network will transmit one packet. Thus, there are total *N* transmissions in this phase.
- In the **network coding phase**, we first need to estimate the number transmissions per instance. Each participating node (in this phase) transmits one message per instance. In the first instance, there are *N-2* participating nodes. In the each following instance, the number of participating nodes will decrease by two. If i denotes the i-th instance, there will be $N - 2i$ of nodes that

transmit a message each in the instance i. Now, we need to estimate the number of instances in this phase. This is where we need to take into account if *N* is the odd or even number. An instance consists of at least one transmission. If *N* is the odd number, we start with $N-2$ transmissions in the first instance, where $mod(N-2,2)=1$. In the last instance we have only one transmission. Therefore there will be $\alpha_{odd}=\left\lceil\frac{N-2}{2}\right\rceil$ of those instances. Figure 3.8b shows the message flow extracted from the Figure 3.3 for the case when the number of nodes in the network is set to the $N=6$. We see that for the even number of nodes (Figure 3.8a) the last instance consists of two transmissions. Therefore in this case $\alpha_{even}=\frac{N-2}{2}$. This means that in the general case, for any value of *N*, the number of instances is equal to $\alpha=\left\lceil\frac{N-2}{2}\right\rceil$. Therefore, the total number of messages transmitted in this phase will be: $\sum_{i=1}^{\alpha}(N-2i)$.

- In the **forwarding phase**, we also need to estimate the number of instances and the number of transmissions per instance. If we observe the Figure 3.8, we may notice that there are $N-3$ messages missing at the sink node (node with the ID 7), at the end of the network coding phase. Let us observe only the nodes with the IDs higher than the ID of the middle node. The situation is symmetrical for the nodes with the lower ID than the middle node. If a network has the odd number of nodes, in each instance a sink node can receive two new messages. Therefore, the number of instances in this phase for odd values of *N* will be $\beta_{odd}=\frac{N-3}{2}$. If we observe the Figure 3.3, we realise that the sink node is expecting $N-3$ messages in this phase. In each but the last instance, exactly two new messages are relayed to the sink node. In the last instance, only one message is relied to the sink node. Therefore, the total number of instances in the forwarding phase for even values of *N* is: $\beta_{even}=\left\lceil\frac{N-3}{2}\right\rceil$. Thus, in the general case for arbitrary value of *N*, the number of instances in the forwarding phase is $\beta_{\frac{even}{odd}}=\left\lceil\frac{N-3}{2}\right\rceil$. In the instance i, $N-2i-3$ messages are transmitted, regardless the value of i. Therefore the total number of transmissions for nodes with the IDs higher than the middle node is $\sum_{i=0}^{\beta-1}(N-2i-3)$, where $\beta = \left\lceil \frac{N-3}{2} \right\rceil$. Thus, the total number of messages transmitted in the forwarding phase is $2\sum_{i=0}^{\beta-1}(N-2i-3)$.



(a)

(b)

Fig. 3.8. Single-line layout, one-hop receiving. a) Flow of messages in the each phase for the odd number of nodes in the network. b) Flow of messages in the each phase for the even number of nodes in the network.

The total number of transmissions required to transfer data from all nodes to the sinks is:

$$N_{tr} = N + N_{tr} = N + \sum_{i=1}^{\alpha}(N-2i) + 2\sum_{i=0}^{\beta-1}(N-2i-3), \tag{3.1}$$

where $\alpha = \left\lceil \frac{N-2}{2} \right\rceil$ and $\beta = \left\lceil \frac{N-3}{2} \right\rceil$.

However, if the network coding is not applied, for a transmission of every broadcasted XORed message in the previous formula, two transmissions are required. Therefore, the only change compared to the case when network coding is applied, is that the number of transmissions in the network coding phase doubles. Thus, when the network coding is not used, the total number of transmissions increases exactly by the number of transmissions of coded messages. So, without network coding, the total number of transmissions is given by:

$$N_{tr} = N + 2\sum_{i=1}^{\alpha}(N-2i) + 2\sum_{i=0}^{\beta}(N-3-2i). \tag{3.2}$$

3.5.1.2 Two-hop direct-connectivity subcase

If transmission power is increased to the point when a second-hop neighbour can decode a message, the number of necessary transmissions will decrease. This happens because more nodes overhear messages and they do not need any longer separate transmissions. However, in this case, a scheduling algorithm must be carefully designed to avoid an excessive interference in light of the increased transmission power. The increase in the transmission power reduces the number of transmissions, but it also consumes more energy. The trade-off between these two factors should be carefully studied. Nevertheless, if network coding is used, and two-hop connectivity is achieved according to our algorithm, the total number of transmissions can be obtained following steps similar to the one-hop direct-connectivity case. There are four distinct cases to which we should pay attention depending on the characteristics of the total number of nodes deployed in the network (*N*): $mod(N, 4) = \{1, 2, 3, 0\}$ that we consider in order to obtain the closed form solution for the total number of transmissions in a the linear network with the two-hop direct-connectivity. All cases follow one of these four patterns.

- In the **broadcasting phase,** each node broadcasts a packet that it formed from data gathered by its own sensors. The total number of transmissions in this phase, given lossless communication will be equal to the total number of nodes in the network, *N*.

- In the **network coding phase,** there will be a number of transmission cycles, each of which comprises of the different number of transmissions. In each of these cycles a node can transmit no more than one packet. A cycle is represented with a row of arrows in the Figure 3.9. There are four distinctive patterns based on the value of $mod(N,4) = \{0,1,2,3\}$. The last cycle in this phase accordingly consists of 4, 1, 2 or 3 transmissions. Examples of these four patterns are shown respectively in the Figures 3.9, 12a, 12b and 12c. We need to compute the number of transmission cycles and the number of transmissions per transmission cycle. Regardless of the value of *N*, in the first cycle of this phase total $N-4$ nodes transmit their message. The last cycle makes difference and it comprises of 1, 2, 3 or 4 transmissions. Therefore, the number of cycles in this phase is $\alpha = \left\lceil \frac{N-4}{4} \right\rceil$. In the cycle i, there are $N-4i$ transmissions. Thus, the total number of transmissions in this phase is given by: $\sum_{i=1}^{\alpha}(N-4i)$.

- In the **forwarding phase,** we only observe transmissions by nodes with the IDs higher or equal to the middle node in the network. The situation is symmetrical for the remaining nodes. We may notice that a network that consists of *N* sensor nodes, where $mod(N,4) = \{1,2,3\}$ has respectively 1, 2 and 3 transmissions respectively, in the last transmission cycle, while Figure 3.4 suggests that $N-4$ packets need to be transferred to the sink node in this phase. In the each cycle but the last one, sink nodes receive four new packets. The total number of cycles in this phase is given by: $\beta = \beta_{1,2,3} = \left\lceil \frac{N-4}{4} \right\rceil$, and in i-th cycle there are $N-4i$ transmissions. Only networks that comprise of *N* nodes, when $mod(N,4) = \{0\}$ are the exception to this rule, as they have one not four transmissions in the last cycle. In this case the number of transmissions is equal to the number of transmissions in the previous three cases increased by one transmission that happens in the last cycle. The total number of transmissions in the data forwarding phase will be: $2\sum_{i=1}^{\beta-1}(N-4i)+\begin{cases}0. & for\ mod(N,4)=\{1,2,3\}\\ 2, & for\ mod(N,4)=\{0\}\end{cases}$.

The total number of transmissions required to transfer all messages to the sink nodes in a network with a single line layout and two-hop direct-connectivity is:

$$\text{for } N=4k\ ,\ N_{tr}=N+\sum_{i=1}^{\alpha}(N-4i)+2\sum_{i=0}^{\beta-1}(N-4i-4)+2; \tag{3.3}$$

$$\text{otherwise, } N_{tr}=N+\sum_{i=1}^{\alpha}(N-4i)+2\sum_{i=0}^{\beta-1}(N-4i-4), \tag{3.4}$$

Similar to the previous case, the total number of transmissions without network coding can be easily derived, by doubling the number of transmissions in the network coded phase:

$$\text{for } N=4k\ ,\ N_{tr}=N+2\sum_{i=1}^{\alpha}(N-4i)+2\sum_{i=0}^{\beta-1}(N-4i-4)+2; \tag{3.5}$$

$$\text{otherwise, } N_{tr}=N+2\sum_{i=1}^{\alpha}(N-4i)+2\sum_{i=0}^{\beta-1}(N-4i-4) \tag{3.6}$$

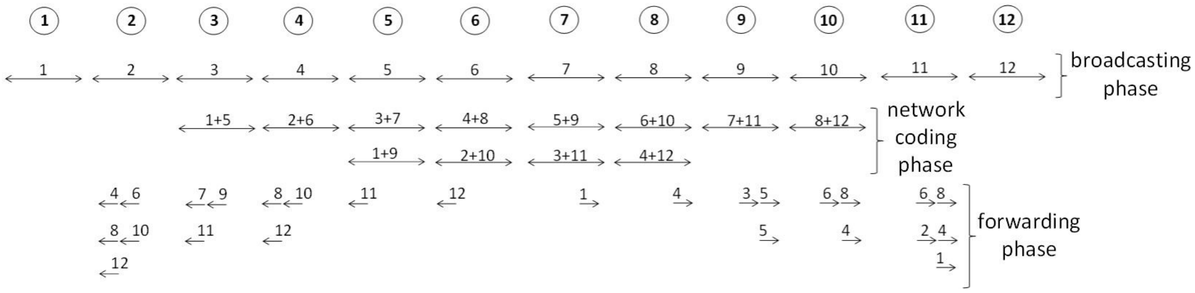
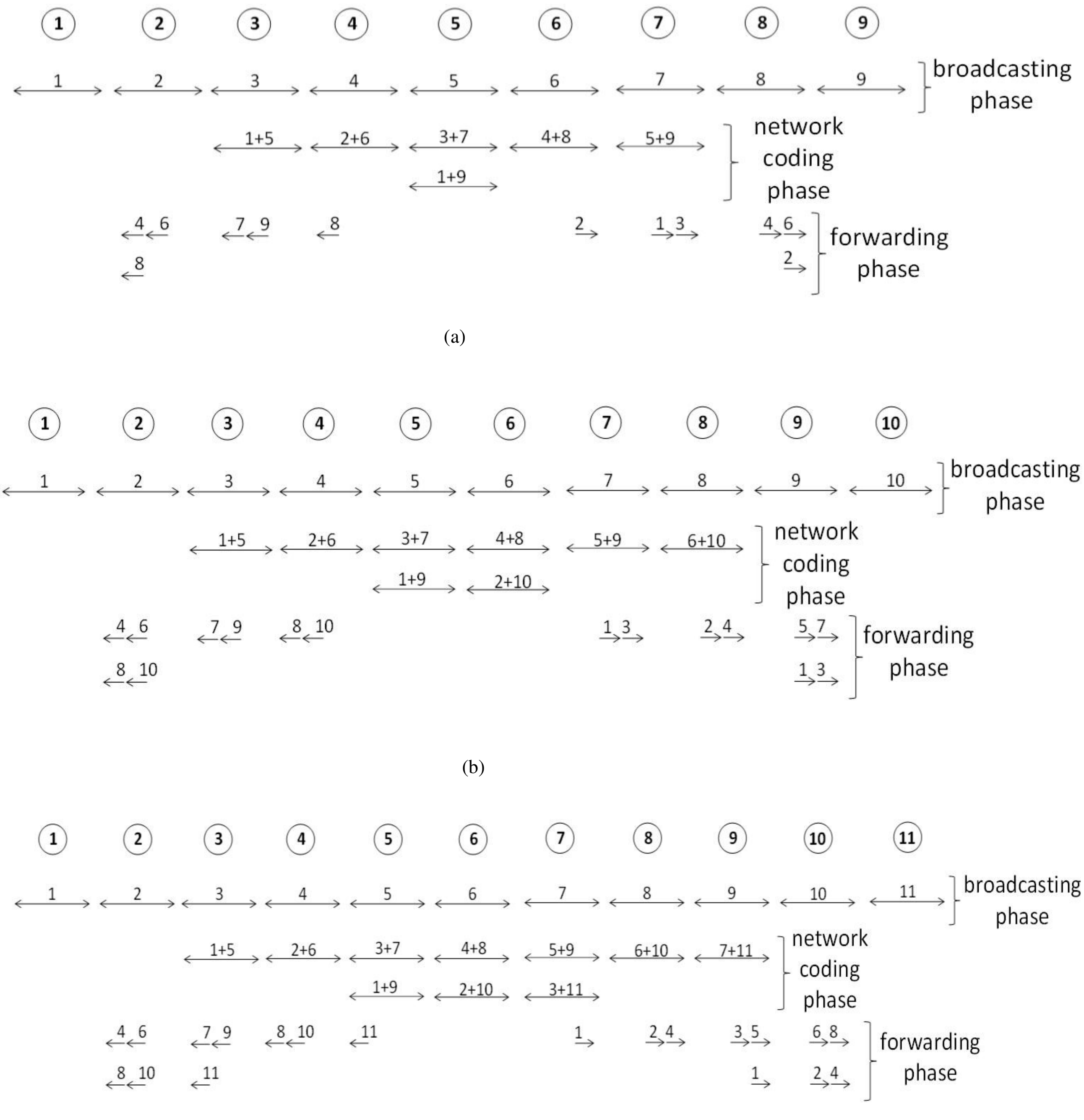


Fig. 3.9. Single-line layout, two-hop direct-connectivity. Flow of messages for *mod(N,4)=0*.



(a)

(b)

(c)

Fig. 3.10. Single-line layout, two-hop direct-connectivity. a) Flow of messages for *mod(N,4)=1*. b) Flow of messages for *mod(N,4)=2*. c) Flow of messages for *mod(N,4)=3*.

3.5.1.3 *Three-hop direct-connectivity subcase*

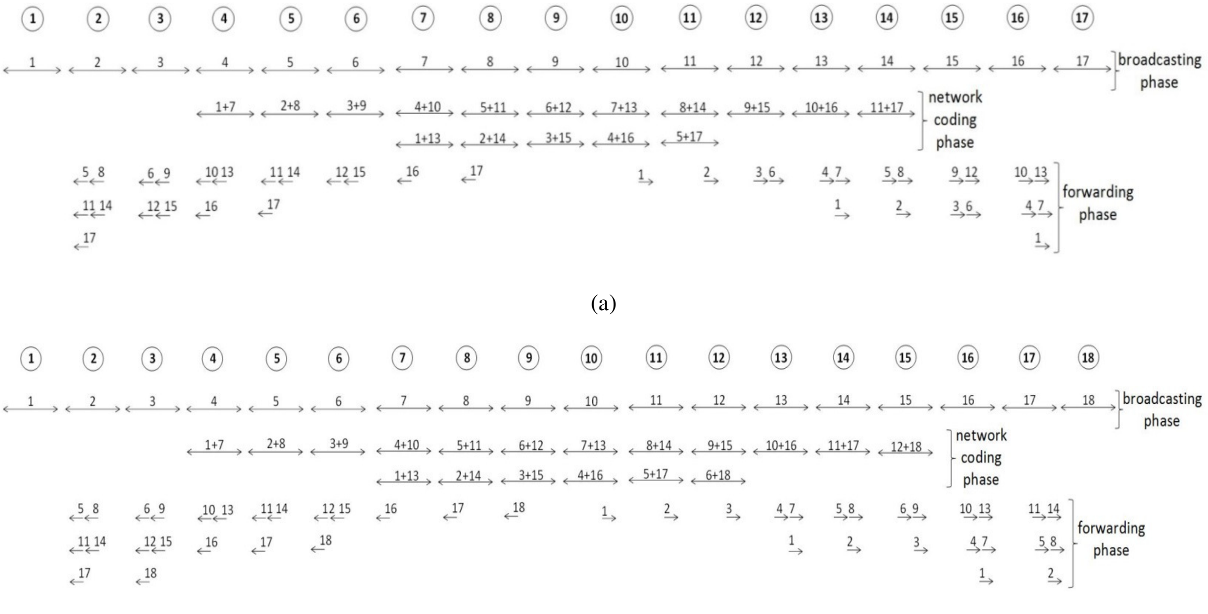
If we increase the transmission power to the point where the direct three-hop connectivity is achieved, the number of necessary transmissions will decrease further, due to the overhearing. The scheduling algorithm must be carefully designed to avoid excessive interference in light of the increased transmission power. The increase in transmission power reduces the number of transmissions, but it also consumes more energy as packets should be successfully decoded at the larger distance. Nevertheless, if the network coding is used, and the three-hop direct-connectivity is achieved according to our algorithm, the total number of transmissions can be obtained following steps similar to the one-hop and two-hop direct-connectivity cases. There are six distinctive cases to which we should pay attention depending on the value of number of nodes deployed in the network N , $mod(N,6) = \{1,2,3,4,5,0\}$. All cases follow one of the six patterns, Figure 3.11 and Figure 3.12.

- In the **broadcasting phase,** each node broadcasts a packet created from data collected by its own sensors. The total number of transmissions in this phase, given the lossless communication is equal to the number of nodes in the network, *N*.

- In the **network coding phase,** we need to estimate the number of cycles, and the number of nodes per cycle. In each cycle, a node will transmit no more than one packet. A cycle is represented with a row of arrows in Figures 3.11 and 3.12, e.g. the second row of arrows is one cycle. There will be six distinctive patterns based on the value of $mod(N,6) = \{1,2,3,4,5,0\}$. The last cycle in this phase will accordingly have 1, 2, 3, 4, 5 or 6 transmissions. Examples of those four cases are shown respectively in the Figures 3.11 and 3.12. We see that in the first cycle of this phase total N-6 nodes transmit their message, whereas in the last cycle 1, 2, 3, 4, 5 or 6 transmissions are performed. Therefore, the number of cycles in this phase is given by $\alpha = \left\lceil \frac{N-6}{6} \right\rceil$.

In the cycle i, there will be $N-6i$ transmissions. Given the lossless communication, the total number of the transmissions in this phase is: $\sum_{i=1}^{\alpha}(N-6i)$.

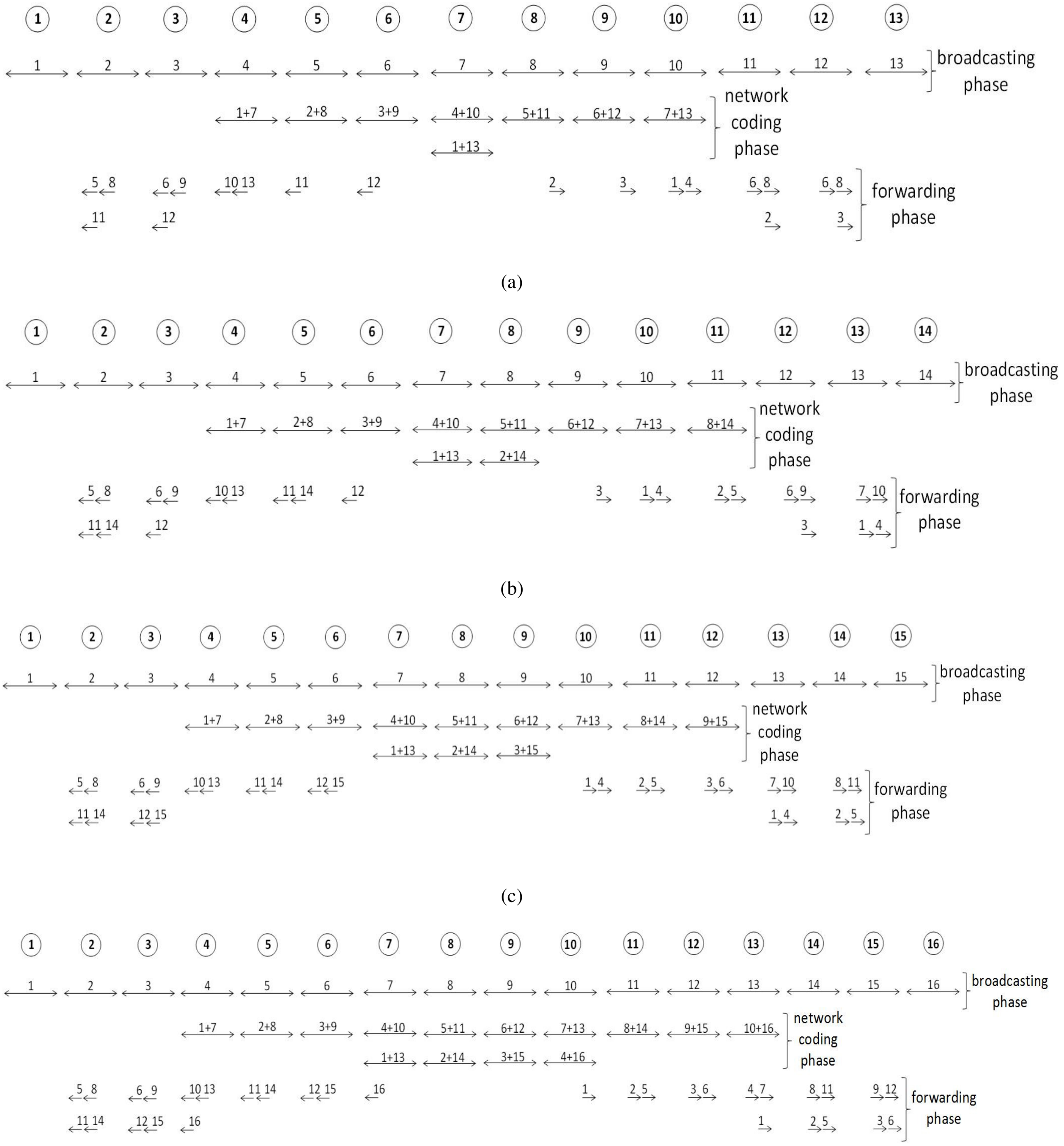
- In the **forwarding phase,** we will observe only transmissions from the nodes with the ID higher or equal to the middle node. As the situation with the remaining nodes is symmetrical, we obtain the total number of transmissions by multiplying the number of transmissions by two. We may also notice that a network with the total number of nodes such that $mod(N,6)=\{1,2,3,4\}$ has respectively 2, 3, 4 and 5 transmissions in the last cycle of transmissions. There are $N-5$ packets that need to be handed to the each sink node in this phase. In each previous transmission cycle a sink node receives six new packets. The total number of cycles in this phase is given by $\beta=\left\lceil\frac{N-5}{6}\right\rceil$, and in i-th cycle there are $N-6i-5$ transmissions. Networks that comprise of N nodes, where $mod(N,6)=\{5,0\}$ are the exception to the rule, as they have one transmission surplus in the last cycle. Therefore, the total number of transmissions in this phase will be: $2\sum_{i=0}^{\beta-1}(N-6i-5)+\begin{cases}0. & for\ mod(N,6)=\{1,2,3,4\}\\ 2, & for\ mod(N,6)=\{5,0\}\end{cases}$.



(a)

(b)

Fig. 3.11. Single-line layout, two-hop direct-connectivity. Flow of messages for a) *mod(N,6)=5*, b) *mod(N,6)=0*.



(a)

(b)

(c)

(d)

Fig. 3.12. Single-line layout, two-hop direct-connectivity. Flow of messages for a) *mod(N,6)=1*, b) *mod(N,6)=2*, c) *mod(N,6)=3*, d) *mod(N,6)=4*.

Therefore, in the case of network coding, for $mod(N,6)=\{1,2,3,4\}$:

$$N_{tr}=N+\sum_{i=1}^{\alpha}(N-6i)+2\sum_{i=0}^{\beta-1}(N-6i-5), \tag{3.7}$$

without network coding total number of transmissions is:

$$N_{tr}=N+2\sum_{i=1}^{\alpha}(N-6i)+2\sum_{i=0}^{\beta-1}(N-6i-5). \tag{3.8}$$

Whereas if $mod(N,6)=\{5,0\}$, for the case when network coding is deployed:

$$N_{tr}=N+\sum_{i=1}^{\alpha}(N-6i)+2\sum_{i=0}^{\beta-1}(N-6i-5)+2. \tag{3.9}$$

While, for the case without network coding:

$$N_{tr}=N+2\sum_{i=1}^{\alpha}(N-6i)+2\sum_{i=0}^{\beta-1}(N-6i-5)+2\ , \tag{3.10}$$

where $\alpha=\left\lceil\frac{N-6}{6}\right\rceil$ and $\beta=\left\lceil\frac{N-5}{6}\right\rceil$.

3.5.2 Double-parallel-lines layout of sensor nodes

The total number of transmissions required until all generated packets reach sink nodes consists of three parts: the broadcasting of the original nodes' messages, the transmission of network coded messages and the conventional forwarding of simple packets coming from the other nodes. The greater the proportion of messages is coded, the better the network's performance.

Not only that network coding reduces energy consumption, it also improves the capacity. Let $\frac{N}{2}$ be the number of nodes in the network deployed in a single line. Total number of nodes in the network is *N*. This is to achieve the fairness so that sink nodes expect the same number of messages as in the case of single-line layout.

3.5.2.1 One-hop direct-connectivity subcase

Nodes can overhear more transmissions from other nodes in the double-parallel-lines layout. However, since each node can extract only one new packet from each XORed (coded) packet, coding gain will be limited. Thus, in order to be useful, each packet received by a node should consist of a number of already known packets (that is greater or equal to 0) and exactly one new packet. If received XORed packet contains two or more unknown packets to a node, that node will not be able to decode any new message. Ideally, all nodes that are receiving XORed message can decode a new packet. The coding is optimal when all nodes that overhear encoded message extract a new packet for themselves. If at least one node cannot decode any new packet from a received packet, coding is sub-optimal.

The total number of transmissions required until the information from all nodes reaches the sink nodes consists of: the broadcasting of nodes' original messages, the transmissions of the network coded messages and the conventionally forwarding of other nodes' original messages. Sometimes, civil engineers deploy nodes at the both sides of bridge, most often when they monitor very wide bridges. Let *N* be the total number of nodes deployed in the network.

For the network with one-hop direct-connectivity, there are only two distinctive patterns that need to be observed, and they are: $mod\left(\frac{N}{2},2\right)=\{0,1\}$. We show the example of those schemes in a form of message flow, for N=10 and N=12 in the Figure 3.12.

- In the **broadcasting phase**, each node broadcasts a packet formed from data collected by its own sensors. The total number of transmissions in this phase, given the assumed lossless communication, is equal to the total number of nodes deployed in the network, *N*.
- In the **network coding phase**, there is a number of cycles. In the each cycle a node transmits maximum one packet, but different number of nodes participates in each cycle. In the Figure

3.13, a cycle is represented with a row of arrows, e.g. the second row of arrows makes one cycle. There are two distinctive schemes, one for the number of nodes in the network that satisfies $mod\left(\frac{N}{2},2\right) = \{1\}$, and the other one when $mod\left(\frac{N}{2},2\right) = \{0\}$. They have 1 and 2 transmissions in the last cycle of this phase (in each line), respectively. Whereas, in the first cycle $\frac{N}{2} - 2$ nodes from each line participate. Therefore, the number of cycles in this phase is $\alpha = \left\lceil\frac{N-4}{4}\right\rceil$. In the cycle i, there are $N - 4i$ transmissions. Hence, the total number of the transmissions in this phase is: $\sum_{i=1}^{\alpha}(N - 4i)$.

- In the **forwarding phase,** we take into account only transmissions from nodes with the IDs higher or equal to the middle nodes from both lines. The situation with the remaining nodes is symmetrical, so we multiply the obtained number of transmissions by two, to get the total number of transmissions in this phase. Sink nodes are waiting to receive $N - 6$ messages in this phase, while in each cycle sink nodes at one side receive maximum four new messages. Thus, the total number of cycles in this phase is $\beta = \left\lceil\frac{N-6}{4}\right\rceil$. In i-th cycle there are $N - 4i - 6$ transmissions. The total number of transmissions in this phase is: $2\sum_{i=0}^{\beta-1}(N - 4i - 6)$.

The total number of transmissions in the case when a message can be received only by its immediate neighbours is:

$$N_{tr} = N + \sum_{i=1}^{\alpha}(N-4i) + 2\sum_{i=0}^{\beta-1}(N-4i-6)\,. \tag{3.11}$$

If the network coding is not applied, the total number of transmissions is:

$$N_{tr} = N + 2\sum_{i=1}^{\alpha}(N-4i) + 2\sum_{i=0}^{\beta-1}(N-4i-6)\,, \tag{3.12}$$

where $\alpha = \left\lceil\frac{N-4}{4}\right\rceil$ and $\beta = \left\lceil\frac{N-6}{4}\right\rceil$. This is because the only difference that instead of any message from the network coding phase, two separate messages should be transmitted.

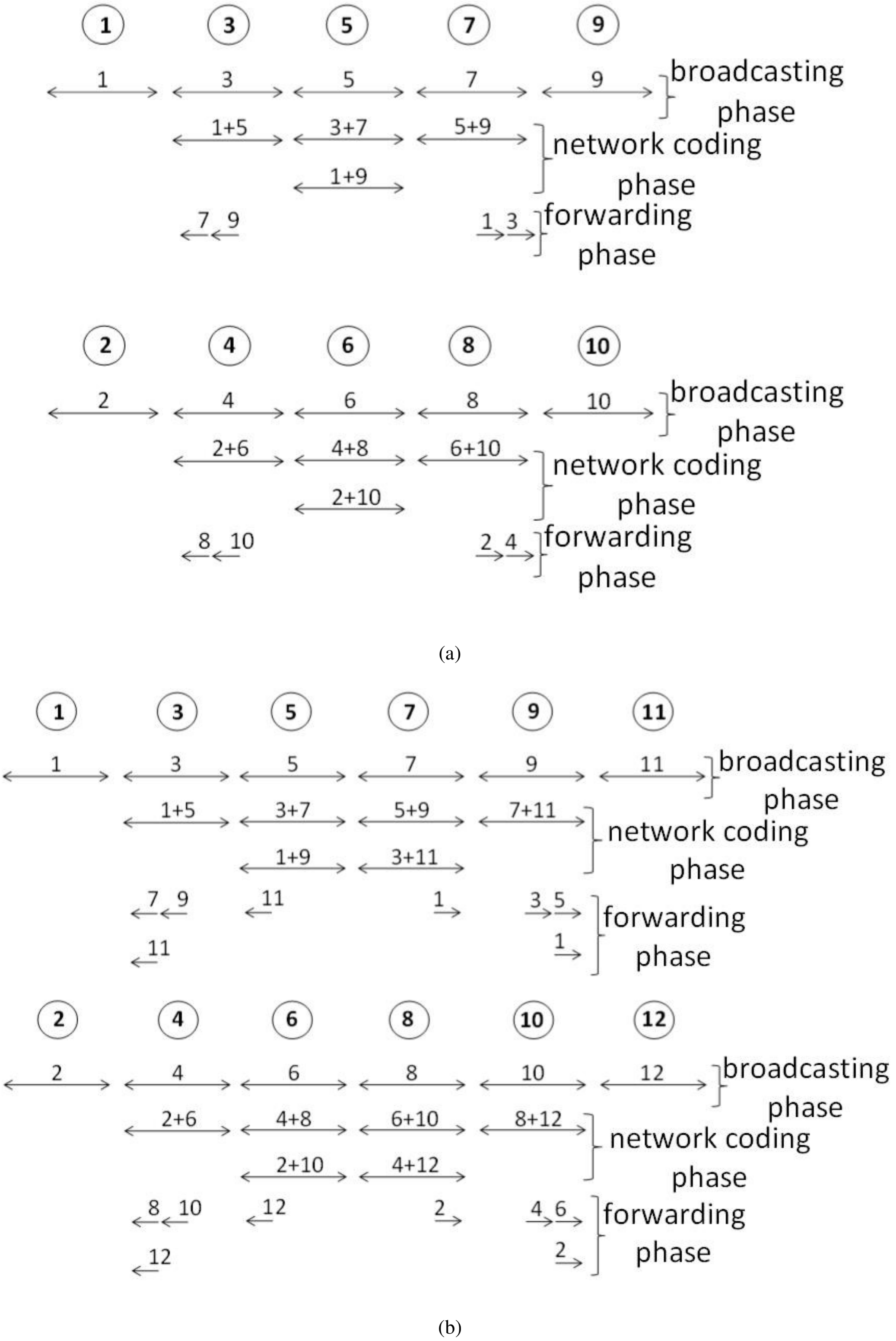


Fig. 3.13. Double-parallel-lines layout, one-hop direct-connectivity. a) Flow of messages for the odd values of *N*. b) Flow of messages for the even values of *N*.

3.5.2.2 *Two-hop direct-connectivity subcase*

If nodes form the double-parallel-lines topology, and we increase the transmission power so that a node can directly communicate with its neighbours that are two-hops away, the number of necessary transmissions further decreases as the overhearing is more prominent. Closed form expressions are derived using the exact same rules as for two-hop connectivity cases in the single-line scenario, combined with the layout discussed in the previous example.

For the network with two-hop direct-connectivity, there are four distinct cases when it comes to the total number of sensor nodes, depending on the value of $mod\left(\frac{N}{2},4\right)$ that can be $\in \{1,2,3,0\}$.

- In the **broadcasting phase,** each node broadcasts a packet created by its own sensors. The total number of transmissions in this phase, given the assumed lossless communication is equal to the total number of sensor nodes deployed in the network *N*.

- The **network coding phase** consists of a number of cycles. In the each cycle a node transmits maximum one packet. There is an analogy between these two layouts given the two-hop direct-connectivity. In the first cycle of this phase $\frac{N}{2}-4$ nodes from each line participate. Every next cycle has four less participating nodes from each line. Therefore, the number of cycles in this phase is $\alpha=\left\lceil\frac{N-8}{8}\right\rceil$. In the cycle i, there are $N-8i$ transmissions. Hence, the total number of transmissions in this phase is: $\sum_{i=1}^{\alpha}(N-8i)$.

- In the **forwarding phase,** we observe only transmissions from the nodes with the IDs higher or equal from the middle nodes in each line. The situation with the remaining nodes is symmetrical, so we multiply the computed number of transmissions by two, to obtain the total number of transmissions. Sink nodes are waiting to receive $N-8$ messages in this phase, while in the each cycle a sink node receives maximum eight new messages. Therefore, the total number of cycles in this phase is $\beta = \left\lceil \frac{N-8}{8} \right\rceil$. In i-th cycle there are $N - 8i - 8$ transmissions. The total number of transmissions in this phase is: $2\sum_{i=0}^{\beta-1}(N - 8i - 8)$.

Thus, if NC is applied the total number of transmissions is:

$$N_{tr} = N + \sum_{i=1}^{\alpha}(N-8i) + 2\sum_{i=0}^{\beta-1_1}(N-8i-8)\,. \tag{3.13}$$

If network coding is not applied, the total number of transmissions will be:

$$N_{tr} = N + 2\sum_{i=1}^{\alpha}(N-8i) + 2\sum_{i=0}^{\beta-1}(N-8i-8). \tag{3.14}$$

The only change between the case with and without network coding is that instead of each coded message a node has to transmit two different messages.

3.5.2.3 Three-hop direct-connectivity subcase

If nodes further increase the transmission power and as a result the messages they transmit can be decoded by their third-hop neighbours, the number of necessary transmissions decreases further. For the network with the three-hop direct-connectivity, there are six distinctive patterns depending on the value of $mod(\frac{N}{2}, 6)$ that can take values $\{1, 2, 3, 4, 5, 0\}$.

- In the **broadcasting phase,** each node broadcasts a packet that containing information from its own sensors. The total number of transmissions in this phase, given the assumed lossless communication is equal to the total number of sensor nodes deployed in the network, *N*.

- In the **network coding phase,** we need to estimate the number of sensing cycles, and the number of transmissions per cycle. In the each cycle a node transmits no more than one packet. In the first transmission cycle, $\frac{N}{2} - 6$ nodes from each line participate. In the each following cycle the number of participating nodes from the each line decreases by six. Therefore, the number of cycles in this phase is $\alpha = \left\lceil \frac{N-12}{12} \right\rceil$. In the cycle i, there are $N - 12i$ transmissions. Therefore, the total number of transmissions in this phase is: $\sum_{i=1}^{\alpha}(N - 12i)$.

- In the **forwarding phase** we observe the transmissions from nodes with the IDs higher or equal to the middle nodes from both lines. The situation with the remaining nodes is symmetrical, so we multiply the obtained number of transmissions by two, to compute the total number of transmissions in the network. Sink nodes are waiting to receive $N - 10$ messages in this phase, while in the each cycle a sink node receives a maximum of twelve new messages. The total number of cycles in this phase is $\beta = \left\lceil \frac{N-10}{12} \right\rceil$. In the i-th cycle there are $N - 12i - 10$ transmissions. Thus, the total number of transmissions in this phase is $2\sum_{i=0}^{\beta-1}(N - 12i - 10)$.

For $mod\left(\frac{N}{2}, 6\right) = \{0\}$, the total number of transmissions is:

$$N_{tr} = N + \sum_{i=1}^{\alpha}(N - 12i) + 2\sum_{i=0}^{\beta-1}(N - 12i - 10) + 4. \tag{3.15}$$

Otherwise,

$$N_{tr} = N + \sum_{i=1}^{\alpha}(N - 12i) + 2\sum_{i=0}^{\beta-1}(N - 12i - 10). \tag{3.16}$$

When network coding is not applied, and $mod\left(\frac{N}{2}, 6\right) = \{0\}$, number reaches:

$$N_{tr} = N + 2\sum_{i=1}^{\alpha}(N - 12i) + 2\sum_{i=0}^{\beta-1}(N - 12i - 10) + 4. \tag{3.17}$$

Otherwise,

$$N_{tr} = N + 2\sum_{i=1}^{\alpha}(N - 12i) + 2\sum_{i=0}^{\beta-1}(N - 12i - 10). \tag{3.18}$$

We notice that the total number of transmissions in the case with traditional forwarding and the case with network coding differs only by the number of network coded messages.

3.5.3 General Case

At this point, we can derive the set of equations that apply to both layouts and all connectivity levels:

- In the **broadcasting phase** each node broadcasts a packet formed by its own sensors. Therefore, the total number of transmissions, given the lossless communication is equal to the total number of nodes deployed in a given network, regardless of its layout, *N*.

- In the **network coding phase** in order to compute the number of transmissions we need to estimate the number of cycles and the number of transmissions per cycle. In the each cycle a node transmits no more than one packet. The number of cycles in this phase is $\alpha = \left\lceil \frac{N-2*c*l}{2*c*l} \right\rceil$, where *c* stands for connectivity level. The value of the parameter is: $c = 1$ for the one-hop direct-connectivity, $c = 2$ for the two-hop direct-connectivity, and $c = 3$ for the three-hop direct-connectivity. The next parameter of interest is *l* that is determined by the layout. The value of this parameter is $l = 1$ for the single-line layout and $l = 2$ for the double-parallel-lines layout. In the cycle i, there are $N - 2 * c * l * i$ transmissions. The total number of transmissions in this phase is $\sum_{i=1}^{\alpha}(N - 2 * c * l * i)$.

- In the **forwarding phase** we first observe transmissions from nodes with the IDs higher or equal to the middle nodes in the (both) line(s). The situation with the remaining nodes is symmetrical, so we multiply the number of transmissions from one side by two, to obtain the total number of transmissions in the network. A sink node is waiting for the delivery of $N - \mathrm{l} * (\mathrm{c} + 2)$ messages in this phase, while in the each cycle a sink node receives a maximum of $2 * c * l$ new messages. Therefore, the total number of cycles in this phase is given by $\beta = \left\lceil \frac{N-l(c+2)}{2cl} \right\rceil$. In i-th cycle there are $N - 2 * c * l * i - l * (c + 2)$ transmissions. Thus, the total number of transmissions in this phase is: $2\sum_{i=0}^{\beta-1}(N-2*c*l*i-l*(c+2))$. Parameters l and c take the same values as explained in the previous case.

Therefore, the total number of transmissions needed for the delivery of all messages to sink nodes, when network coding is applied is:

$$M = N + \sum_{i=0}^{\alpha}(N-2*c*i) + 2\sum_{i=1}^{\beta-1}\big(N-2*c*l*i-l*(c+2)\big). \tag{3.19}$$

If network coding is not used, the number of transmissions is computed by doubling the number of network coded messages:

$$M = N + 2\sum_{i=0}^{\alpha}(N-2*c*i) + 2\sum_{i=1}^{\beta-1}\big(N-2*c*l*i-l*(c+2)\big). \tag{3.20}$$

There is a correction factor for the following cases:

$$M' = M + 2*l \quad \text{for } c = \{1,2\}\ \Lambda\ mod\left(\frac{N}{l}, 2c\right) = 0. \tag{3.21}$$

3.6 Simulation testing

Application of network coding always decreases the total number of transmissions, regardless of the level of direct-connectivity between nodes. We observe the single-line layout, Figure 3.14a. The higher the level of connectivity, the smaller the total number of transmissions is. The network coding technique is an effective replacement for data aggregation that actually preserves the fidelity of information. We may see that the three-hop direct-connectivity level reduces the necessary number of transmissions by two thirds compared to the one-hop direct-connectivity level. This happens due to the overhearing.

We use MATLAB to run simulations that will test how many messages each node has to transmit in order to relay messages created at each node deployed at the bridge to all sink nodes that are situated at the ends of the bridge. We consider that wireless environment is lossless at this point. Thus, using Monte

Carlo simulations does not make sense as numbers had to be exact match to derived mathematical expressions. We started by generating a message at each node and we looked into the number of transmissions each node had to perform given an idealistic, lossless environment. At the beginning each node had exactly one message to relay to sink nodes. We assumed that nodes do not create new message until their message reached all sink nodes.

Whenever a node had a chance to send an XOR combination of two messages (previously received from neighbours from its different sides) – it used the network coding. All nodes broadcasted their own message (from their own sensing units). A node cannot use its own message for XORed packet, as no one else has that message at the beginning and no one else would be able to extract any information when both messages within XORed packet were unknown. Thus, we assume that each node has a buffer where it stores messages. At this point we assume all transmitted messages will be received, thus we do not consider the usage of ACK/NAC control messages but they will be used in the following chapters.

Figure 3.14b shows the total number of transmissions in the case of double-parallel-lines layout. X-axis shows the number of nodes deployed in a line. We emphasize that we discuss the case when we use the same number of nodes for both layouts, for the fixed length of the monitored bridge. This means that nodes will be placed at twice the distance in the case of double-parallel-lines.

Figure 3.13b plots the situation at the one line of nodes in a network with the double-parallel-lines layout. The total number of transmissions can be obtained by doubling the values at y-axe as nodes are equally distributed at both lines in this network. We see that the total number of transmissions is very similar in these two cases, as total number of nodes in these two networks is the same.

While the network benefits from the deployment of network coding, we cannot assess the extent of it based on Figure 3.11. Therefore, we define a new parameter, network coding as:

.

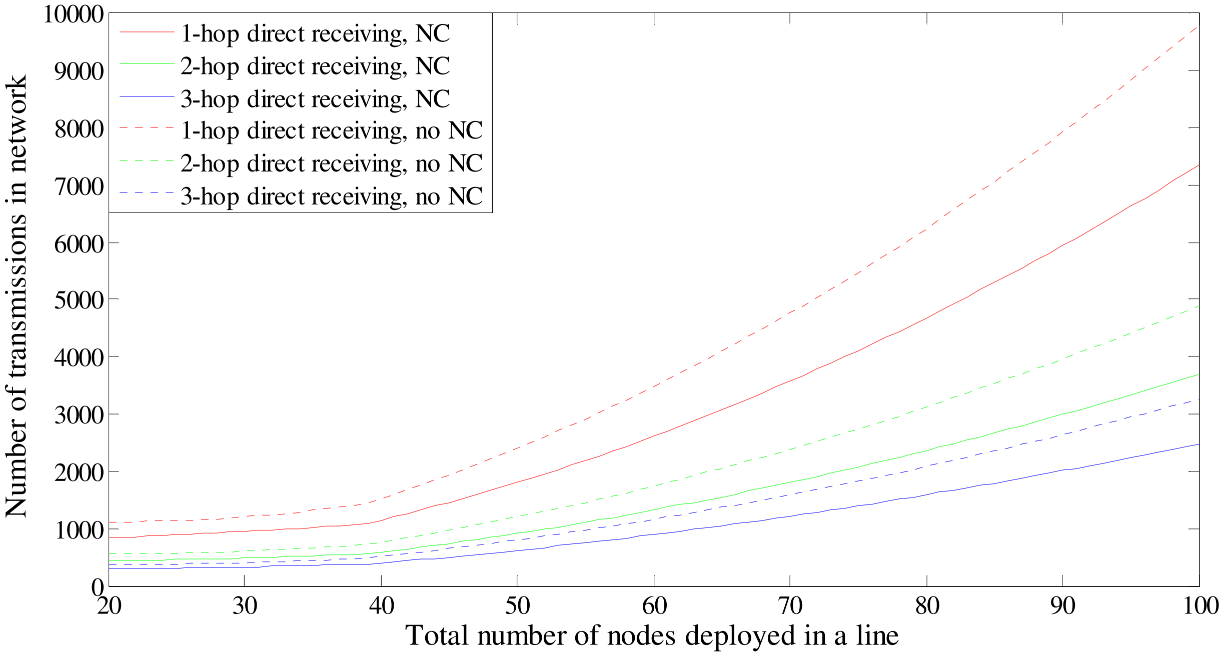
$$coding\_gain = 100 * (\frac{no.\_of\_transmissions\_with\_conventional\_routing}{no.\_of\_transmissions\_with\_network\_coding} - 1) \tag{3.22}$$

Network coding gain shows the relative improvement in terms of the total number of necessary transmissions for the case of network coding based network and traditional forwarding based network. We compare the performance of the network coding based algorithm, for the single-line and double-parallel-lines layout and different direct-connectivity levels. Thus, using this definition of coding gain as a metric, we evaluate the performance of proposed algorithm for different network topologies and connectivity.

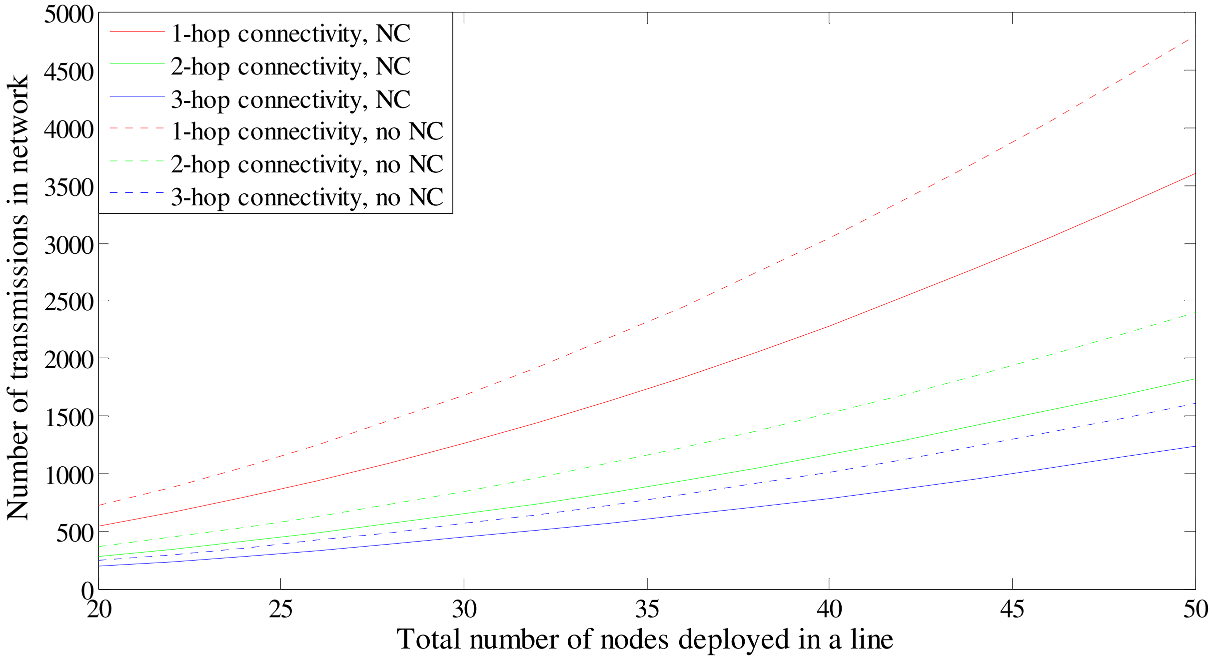
We see that network coding gain is independent of the nodes' layout, and it only depends on a given direct-connectivity level. This is because, regardless of the layout, the total number of transmissions will be the same for the network with the fixed number of nodes, as that is the only information network coding gain uses. We may also see that for the small-span networks, one-hop direct-connectivity algorithm performs the best. This happens as the higher order connectivity levels use the overhearing to their advantage.

At the same time, the higher proportion of messages is relied via simple broadcasting/packet forwarding. This effect is more noticeable for the small span networks as they offer extremely limited network coding opportunities.

For the large span networks the value of network coding gain will converge to 33.5%, 32.5% and 31.5% for one-hop, two-hops and three-hops direct-connectivity, respectfully, Figure 3.15. This happens because at some point the network will achieve an equilibrium between the percentage of coded messages and simple messages that are just forwarded.



(a)



(b)

Figure 3.14. The total number of transmissions needed for the delivery of all messages to the sink nodes. a) for single-line layout. b) for double-parallel-lines layout.

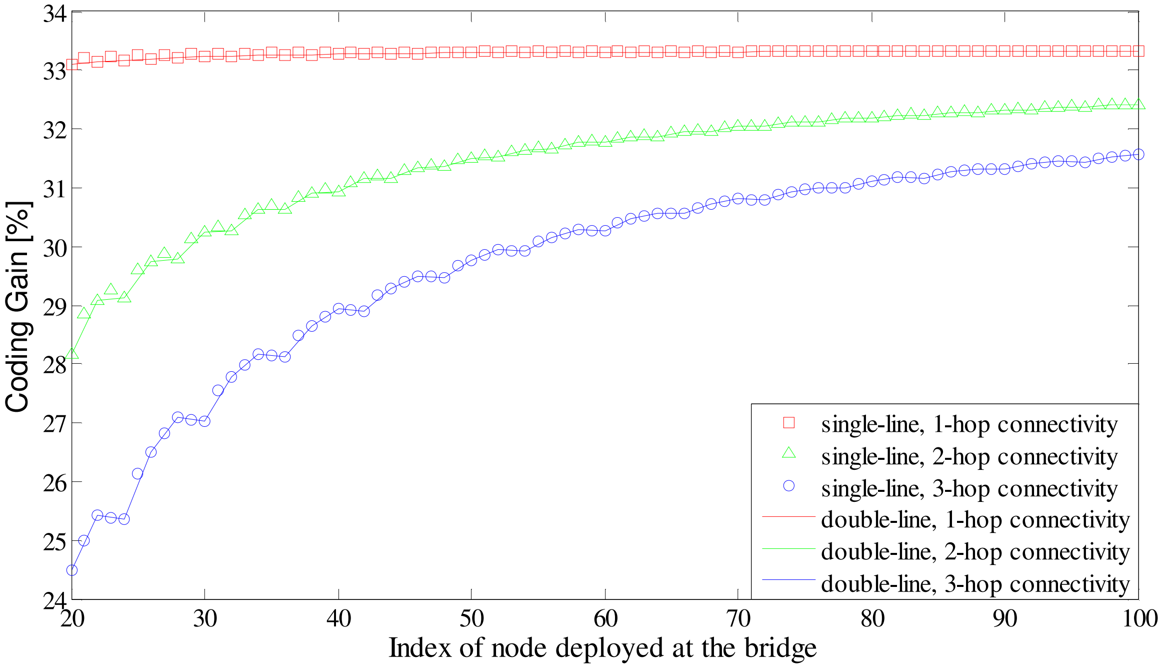


Fig. 3.15 Network coding gain.

3.7 Summary

In this chapter we explained how our proposed algorithm operates for different layouts and connectivity levels. We derived the closed form expressions for the total number of transmissions for both layouts of nodes and all connectivity levels. We used power control to change the connectivity of nodes, and consequently influenced the total number of transmissions in the network. As the underlying assumption we used the perfect Medium Access Layer (MAC) and the lossless communication channel. We are fully aware that these assumptions are idealistic, thus provided results in the practise present the upper bound rather than exact results.

A given number of sensors are placed at one or both sides of the bridge uniformly. Depending on the transmission power, the sensor nodes have one-hop, two-hops or three-hops direct-connectivity, as described earlier. To achieve the fair comparison, we define the coding gain as the ratio of the total number of transmissions with network coding to that for the conventional routing method.

Using this definition of coding gain, we evaluate the performance of proposed algorithm for different network topologies and connectivity. Figure 3.14 shows that the best performance of the network coding algorithm is achieved when nodes can reach only their immediate (one-hop) neighbours. Note that for a given degree of connectivity, results for the single and double-parallel-lines layouts are very close to each other for network that is using the same number of nodes, as shown in Figure 3.12. In the case of one-hop connectivity, each node receives the original messages (data) generated at its second order neighbours via the coded messages received from its immediate neighbours. In contrast, for the case of two-hop direct-connectivity, a node receives the original messages directly from its second order neighbours as data is transmitted at a higher power to maintain the two-hop direct-connectivity. As a result, a higher proportion of messages is conveyed via network coding for networks with one-hop connectivity, compared to the case of two-hop connectivity where more original messages are received directly from second order neighbours. Therefore, coding gain will be higher for one-hop direct-connectivity than for two-hop direct-connectivity in spite of the decrease (due to the overhearing capability) in the total number of transmissions. The same comment applies to the three-hop connectivity. Our results in the Figure 3.15 reveal that our algorithm provides up to 33.5% reduction in the number of transmissions. The figure also suggests that we can achieve the maximum coding gain even for a small number of nodes, thus confirming that our proposed algorithm performs well for relatively small-span sensor networks. However, the proposed algorithm yields better performance gain for networks that consist of a moderate and large number of sensor nodes. These results are published at PIMRC'12 [70].

4. CONNECTIVITY AND PERFORMANCE STUDY

4.1 Chapter Abstract

In the previous chapter, we presented the network coding based algorithm for structural health monitoring of linear structures. We observed two different topologies, namely the single-line and the double-parallel-lines topology. Nodes in any line are equidistant. We use power control to exploit the overhearing capability. To assess the performance of our proposed algorithm we focus on total number of transmissions in the network. In the third chapter, we observed the idealistic case where no losses happen. Thus, assuming perfect MAC protocol and the lossless channel we focus on total number of transmissions in the network. We chose number of transmissions as a metric because our protocol does not change the number of receptions, and power consumed by a transceiver unit is much higher than the power consumed by a microcontroller or a sensing unit. We show that our protocol decreases the number of transmissions by approximately a third, for both layouts and all connectivity levels.

A frequently overlooked disadvantage of network coding in WSNs is the volatility of gains due to the instable link quality. After a message is lost or an overhearing opportunity is missed due to the insufficient received signal power or excessive interference, decoding cannot be performed at the intermediate nodes which significantly degrades the performance. For bridge monitoring, sensor nodes are likely to form a linear topology, for instance when nodes are deployed along one or both sides of a bridge. To ensure a proper data transfer and decoding by use of network coding, we propose here to control the transmission power as a mean to adjust the number of nodes that can overhear a message transmission by a neighbouring node. Network coding gain relies on such message overhearing. On the other hand, too much of overhearing by high transmission power consumes too much of limited battery supply. By simulation, we study the trade-off between the overhearing and the power consumption for the network coding based scheme. Specifically, we consider a bridge with fixed length and sensor nodes that are deployed at a uniform distance along one or both sides of the bridge. Each radio link is characterized by: exponential path loss, shadowing and Rayleigh fading. Our numerical results reveal that appropriate choices of transmission power (thus, the degree of communication connectivity) achieve the optimal extent of overhearing for network coding gain while minimizing the overall power consumption for the WSNs. The results presented in this chapter are published at IABMAS'12 [71].

4.2 Related Work

Yuan et al. [53] investigate the effect of the optimisation of sensor nodes' transmission power on network lifetime. They jointly optimise transmission power, routing, and source quantisation in a wireless sensor network. They show how optimisation of the overall network can be separated into two problems: source coding at the application layer and power control at the physical layer.

Shah-Mansouri et al.[41] investigated the passive loss tomography problem in coded packet wireless sensor networks. They estimate path loss rates from both source and intermediate nodes by inspecting the content of the coded packet at the sink. They suggest an algorithm for the estimation of link loss rates. They show that in the coded packet wireless sensor networks, the proposed algorithm outperforms the Bayesian inference algorithm.

Lin et al. study the loss inference problem in sensor networks with network coding [19]. They show how the fundamental connection between the path and link loss changes when network coding is applied. They propose inference algorithms based on Bayesian principles to detect the set of lossy links in WSNs. Through simulation, they show that proposed algorithm achieves high detection rates and low false-positive rates.

Buta et al. performed the extensive experiments and developed a method of distance estimation between sensor nodes using the received radio-frequency (RF) power level. They measure the received RF power level at various distances and afterwards they use recorded [dBm] values in the distance determination formula. They discuss the influence of humidity and temperature on the path loss exponent [42]

Liu et al. describe the design and deployment of WSNs for SHM. They design the architecture based on application requirements such as energy efficiency, routing and high-frequency sampling. They also derive an analytical model to determine the energy consumed for transmissions and receptions of packets [57]. Liu et al. perform a cluster-based modal analysis. They divide the network into clusters and identify vibration characteristics in each cluster that they afterwards assemble together. They optimise the cluster size to minimise the total energy consumption. They also evaluate the effectiveness and efficiency of the proposed cluster-based modal analysis using experiments and simulations.

To the best of our knowledge, the existing research has not yet considered the joint use of power control and network coding in the WSNs used for SHM. We emphasize that the requirements for SHM applications are very different than the rest of WSNs applications. Even though these papers are relevant for our area of research, as they discuss the interplay between network coding and power consumption, they do not focus on linear multi-sink networks. As we use very specific topology for monitoring suggested directions are sub-optimal or at times even not applicable in our case. We will take into account the conditions in the network that are very specific to our application. We will try to jointly assess the interplay between the network coding and power control, in light of the strict reliability requirements.

Specifically, using a bridge monitoring as an example, this chapter proposes and investigates the performance trade-offs of power consumption (and thus network lifetime) and the network coding gain due to increased overhearing by high transmission power.

4.3 Motivation

Due to the fading and interference data transmission via wireless links is prone to errors. Various propagation models are available. A reader can get more information concerning the ways to simulate a realistic wireless channel in the book *Modelling the Wireless Propagation Channel: A Simulation Approach with MATLAB* [43], but that will not be the topic of this thesis.

There is a number of models for a propagation channel under the effect of combined Rayleigh fading, shadowing and path loss. We chose to use the ideas provided in work of Zorzi et. al [44].

For the purpose of analytical discussion we use the values for path loss coefficient suggested by Rappaport in [45]. The power level at a receiving sensor node, P_R is given by:

$$P_R = R^2 e^{\varepsilon} K \frac{P_T}{r^{\gamma}}, \tag{4.1}$$

where P_T is the transmission power level, γ is the path loss exponent, r is the distance between transmitting and receiving sensor node, Factor $K\frac{P_T}{r^{\gamma}}$ describes the influence of path loss on transmitted power level and is the only deterministic part of the equation. Attenuation due to the shadowing is described by the factor e^{ε}, where ε is Gaussian random variable with the zero mean and variance σ^2. Factor R^2 accounts for Rayleigh fading. The random variable R makes the envelope of received signal to be Rayleigh distributed while making the distribution of its power exponential random variable. Random variables R and ε are independent from user to user, and they are identically distributed. Consequently, the Signal to Interference/Noise Ratio (*SINR*), is given by:

$$SINR_i = \frac{P_{R,i}}{P_N + \sum_{j=1, j\neq i}^{k} P_{R,j}}, \tag{4.2}$$

where $P_{R,i}$ stands for the received power level of the observed signal coming from a node with the ID i, which is the signal that we are trying to receive. P_N is the power level of the noise at the receiver, and $P_{R,j}$ stands for the power level of the interfering signals (signals from nodes other than i that reach the receiving node simultaneously), k is the number of interferers using the same time slot. Thus, the *SINR* level at the receiver when desired signal is coming from the node with the ID *i is:*

$$SINR_i = \frac{R_i^2 e^{\varepsilon_i}}{W + \sum_{j=1, j \neq i}^{k} R_j^2 e^{\varepsilon_j} (\frac{r_i}{r_j})^{\gamma}} \ , \tag{4.3}$$

where factor W is given by:

$$W = \frac{P_N}{K P_{T,i}} r_i^{\gamma} . \tag{4.4}$$

In the formulas (4.3) and (4.4), r_i is a distance of a node with ID i from a receiving node, ε_i is the Gaussian random variable that influences signal coming from the node with ID *i*.

Rappaport [45] provides more details concerning the wireless signal propagation, based on which we set the path-loss exponent to 3 in the analytical model. As vehicles are crossing the bridge, the signal transmitted by nodes is likely to undergo multiple paths and shadowing. Moreover, nodes are usually mounted in a way that immediately exposes them to the shadowing effect. Following steps provided in [43] and [44] we simulate the wireless channel to test the performance of our algorithm in non-ideal channel conditions.

Using simulations, we obtain the probability of successful decoding given the various transmission power levels and distances between the nodes. We define the probability of successful decoding of a packet as:

$$P_i = P(SINR_i > SINR_{threshold}) \ , \tag{4.5}$$

where $SINR_{threshold}$ is the sensitivity level of a sensor node. That value of $SINR_{threshold}$ is determined by the type of sensor nodes that we choose to deploy in our network. We use computer simulation to study the performance of improved connectivity and power consumption. We assume combined path loss, shadowing and Rayleigh fading for each radio link between any given pair of consecutive sensor nodes. Since radio link changes with the time, we adjust the transmission power at each node such that its signal can reach the immediate neighbours with a target SINR above a fixed threshold with a given probability. The goal of this chapter is to analyse the interplay between the network coding, power control and power consumption.

4.4 Scenario

4.4.1 System Model

In the previous chapter, we assumed that our routing algorithm operates in the ideal conditions. The wireless link is lossless and the perfect MAC layer removed the effect of the interference. We assessed its performance based on total number of transmissions and the network coding gain. Now, we want to explore how does the decrease in the total number of transmissions translates to the total power consumption per node and the overall power consumption of a network. We consider that the most of power is consumed by transceiver on receptions and transmissions, while the powers consumed by sensing and computation units is neglected. We abandon the assumption of the ideal channel conditions. We assume that the channel is under combined influence of Rayleigh fading, shadowing and path loss. Consequently, a packet can be lost due to the poor channel conditions. As we are still exclusively assessing the performance on routing level, we assume that the perfect MAC is still in function and no losses due to the interference happen. We still observe a sensor network with a layout that resembles either a single-line or double-parallel-lines topology. We assume that sink nodes are deployed at the ends of the bridge of a fixed length, and that all sinks need to receive all messages. This is one of the ways we are boosting the reliability - the most important requirement of SHM application. The sensor nodes are equidistantly placed along one side of a bridge (single-line layout) or both sides of a bridge (double-parallel-lines layout).

We simulate the wireless channel and asses the reliability of successful decoding of received messages for varying distances among adjacent nodes in a line and different transmission levels. We assume that the ZigBee Pro Module 2PM3570 nodes are deployed in the network. They have very high sensitivity level of -103dBm. Thus, every signal whose power is above this level at the receiver will be successfully decoded. In order to obtain the probabilities of a successful reception, we use the Monte Carlo simulations. Monte Carlo simulations converged after 100.000 samples in this case.

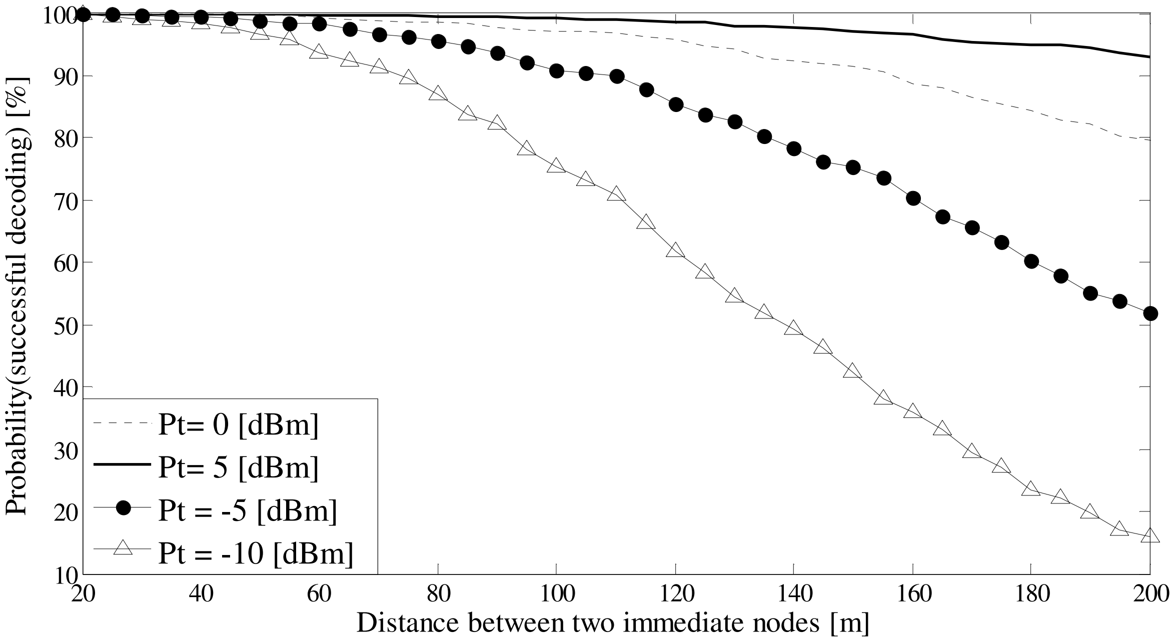
We obtained the probability of successful decoding in this channel empirically, using the following formula:

$$P_{succ} = \frac{number\ of\ successfully\ decoded\ messages}{total\ number\ of\ transmitted\ messages} \tag{4.6}$$

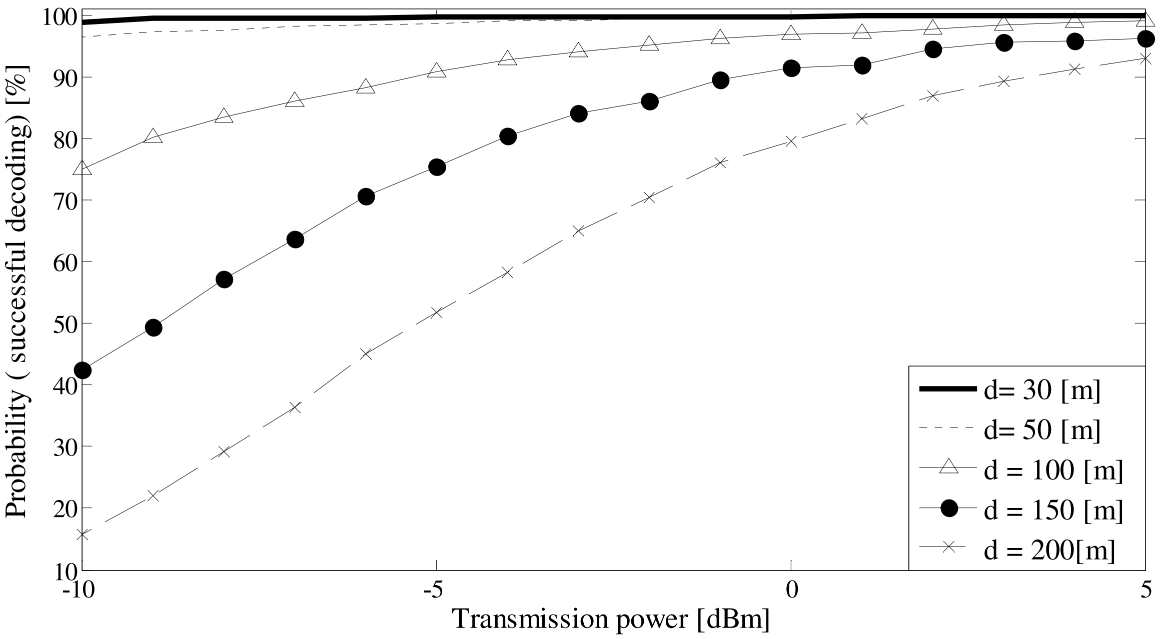
The goal is to achieve 99.99% probability of successful decoding, at the furthest designated receiving node.

4.5 Simulation Testing

We consider a transmission to be successful only if the power level of a signal at the receiver is above the receiver's sensitivity of -103 [dBm] (as we assume ZigBee Pro Module 2PM3570 sensor nodes). We assume 250 [kbps] data rate, with the maximum packet (message) length of 128 [kB], as given in the 802.15.4 standard. As explained in Liu et al. [57], to receive k bits of data a node consumes $k * \alpha$ [nJ], where $\alpha = 50$ [nJ/bit].



(a)



(b)

Fig. 4.1. Probability of successful reception depending on: a) transmission power b) distance between adjacent nodes.

As we moved away from the assumption that once a message is transmitted, it cannot be lost - we need to take into account acknowledgements (ACK/NACK) messages so that nodes know if their transmission was successful or not. We opted to use NACK messages as we have more successful attempts than errors. Thus, choosing NACK over ACK decreases the control messages overhead. These messages are only 1 [B] long, and do not need a whole transmission slot. We use signal processing techniques to model the wireless environment and at all times measure the signal level at some distance. We assume that all nodes use the same transmission power. Signal level of each node will be strongly dependant on its distance from the originating node (via path loss). Signal level also changes due to the shadowing and fading, which are mostly random components. To ensure that results represent the reality in channel we run the simulation for 100.000 times (Monte Carlo simulation) and we plot the average values. Thus, values shown in the Figure 4.1 represent the average state of the channel. We chose to implement simulations in MATLAB as it is based on matrices. We represented each node as a matrix whose rows are made of various values (e.g. row 1: set of distances from originating node, row 2: power levels at those distances, row 3: number of slots until message originating from our node got delivered at corresponding distance etc.). This structure was easy to follow. MATLAB is also equipped with number of built in functions that were useful to statistically describe shadowing and path loss.

Due to the fading and interference, data transmission via a wireless link is prone to errors. We consider a radio link characterized by an exponential path loss, shadowing and fast fading. Since many vehicles are passing over a bridge, we consider the presence of fading and shadowing on top of the signal path loss. From the simulation results, we can read the probability of a successful reception for various transmission power levels and distances between nodes, Figure 4.1.

Results suggest that the higher transmission power leads to the higher probability of successful reception, given the absence of the interference. For example, for the transmission power of 5 [dBm], the probability of successful transmission is higher than 95% even if nodes are placed 200 [m] from each other, as shown in the Figure 4.1b.

We are interested to assess how much power is consumed per node, until a message from each node is relayed to sink nodes for different layouts and connectivity levels. We first assess the distribution of transmissions and receptions load per node in the network. Using this information and an average consumption per transmission/reception for a given connectivity and layout we assess the power consumption distribution in the network if a node uses our proposed algorithm. We choose corresponding transmission power level that provides 99.99% probability of successful decoding, which leaves us with the probability of error of $P_e = 10^{-4}$, Figure 4.1a.

We will try to achieve as fair comparison as possible between the double-parallel-lines layout and the single-line layout. In the first case, we use N nodes for a network with the single-line layout, and $2N$ nodes for a network with the double-parallel-lines layout. Thus, in this case distances between adjacent nodes in a line are fixed and equal for both layouts of nodes. This is fair in a sense that for both layouts nodes relay messages to the same distance, and the density of sensing is the same. However, sink nodes in the network organised in the double-parallel-lines layout are expecting twice as many packets as sink nodes in the single-line layout.

Additionally, a double-parallel-lines topology is almost twice as expensive as a single-line topology, due to the additional sensor nodes required.

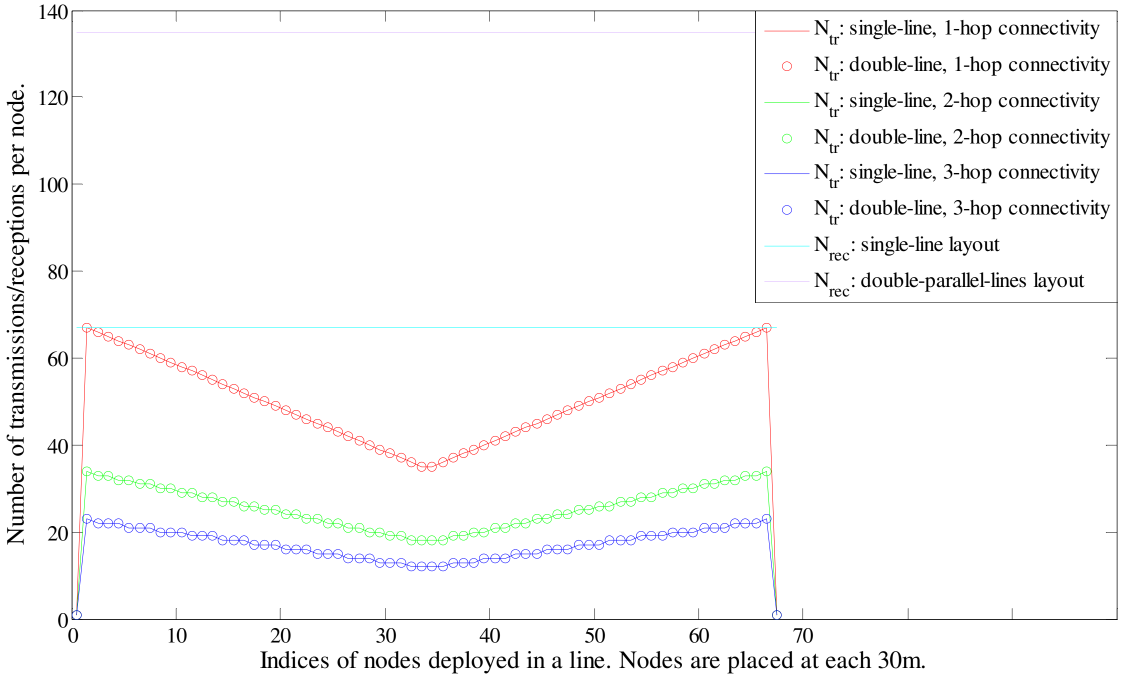
We also discuss the second case, when we use the same number of nodes N for both network layouts. Then, sink nodes will expect the same number of packet regardless of the layout. However, only $\frac{N}{2}$ nodes are available at each line to cover the same bridge length in the double-parallel-lines layout. As a result, the distance between adjacent nodes will be approximately twice as large as the distance between adjacent nodes in a corresponding single-line network.

4.5.1 Case 1: Single-line network with N nodes, double-parallel-lines network with 2N nodes

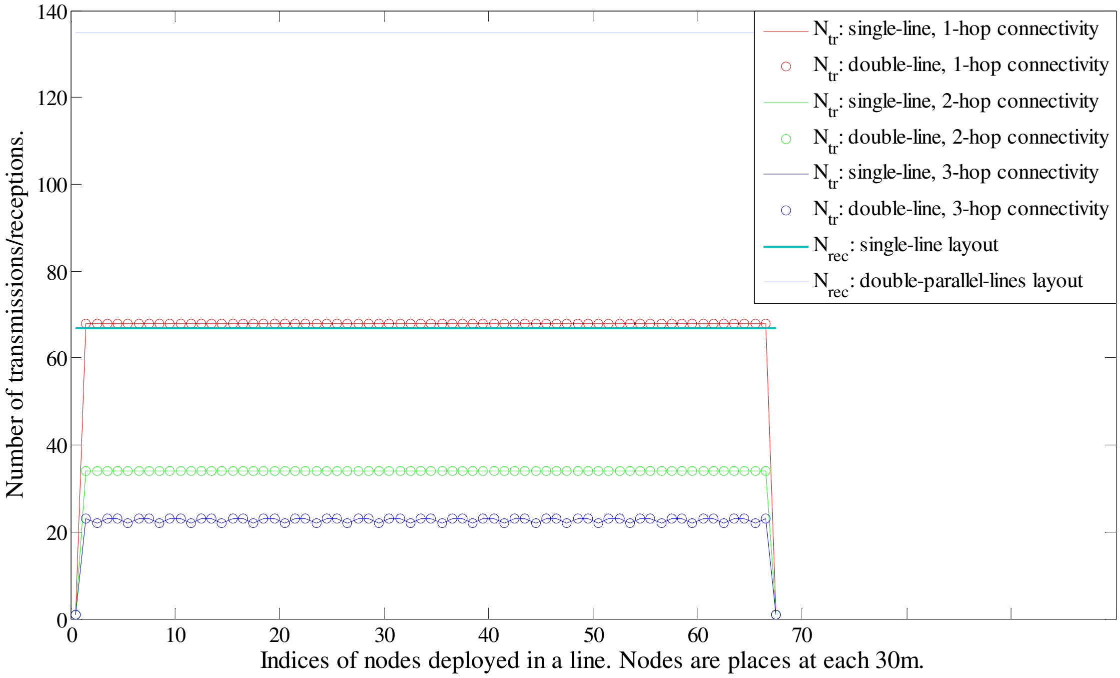
All nodes that belong to the same line are equidistant. The distance between adjacent nodes is the same for both layouts. Thus, in this case a network with the double-parallel-lines layout has twice as many packets to relay as a corresponding network with the single-line layout. If we closely observe the process shown in Figure 3.6, we may notice that one line is in charge of relaying the messages that originated from nodes with the odd IDs, and the other line is in charge for relaying of messages from nodes with the even IDs. Thus, as shown in Figures 4.2 and 4.4, both layouts for a given connectivity level impose the same load in terms of the number of transmissions per node. The difference is in the number of receptions, as we can see from Figure 4.2 and 4.4.

Individual nodes organised in a double-parallel-lines layout receive twice as many packets as nodes organised in a single-line layout. When we plot the number of transmissions in the network with the single-line and the double-parallel-lines layout in the same figure, we observe one line (e.g. all nodes with the odd IDs), as situation is symmetrical for both lines in the double-parallel-lines topology.

Figure 4.2 shows the network that uses 68 nodes for the single-line layout, and 136 nodes for the double-parallel-lines layout. Figure 4.4 shows the network with 40 nodes for the single-line layout, and 80 nodes for the double-parallel-lines layout. Network coding always decreases the total number of transmissions, particularly for nodes in the middle that are physically more challenging to replace. Network coding however does not impact the total number of receptions, as a node can extract a maximum one new packet from any received packet. Thus, in order to extract 68 packets, the minimum number of messages a network has to receive is 68 with or without network coding.



(a)



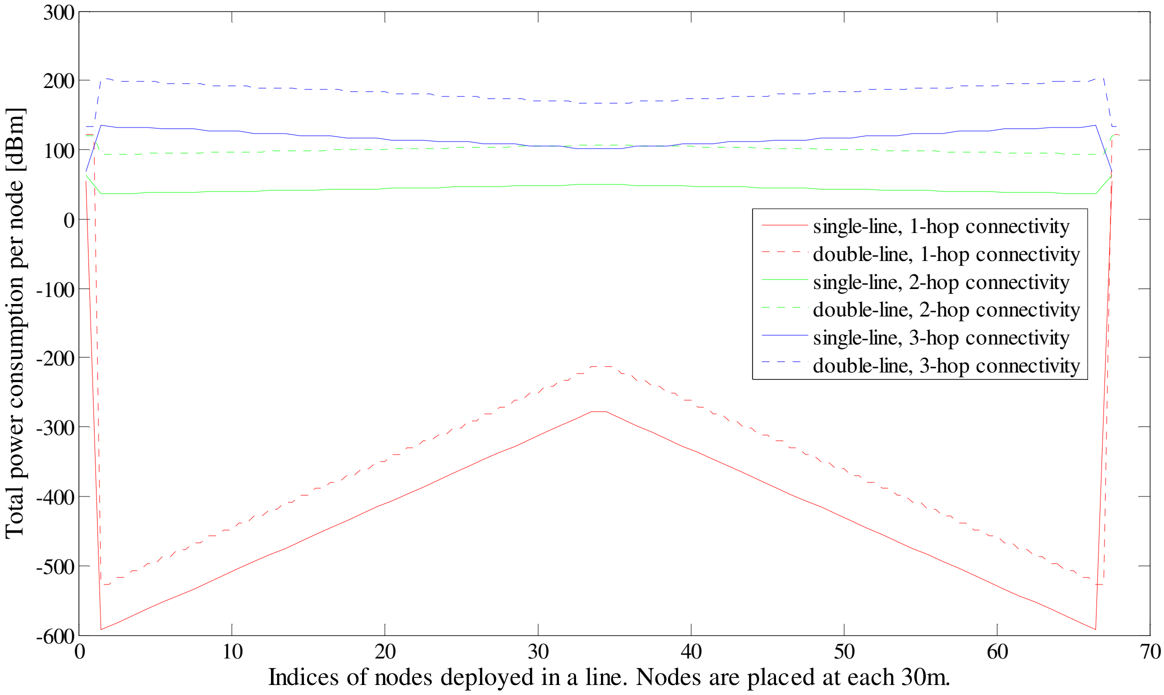
(b)

Fig 4.2. The total number of transmissions and receptions per node when a) network coding b) traditional data forwarding, is used.

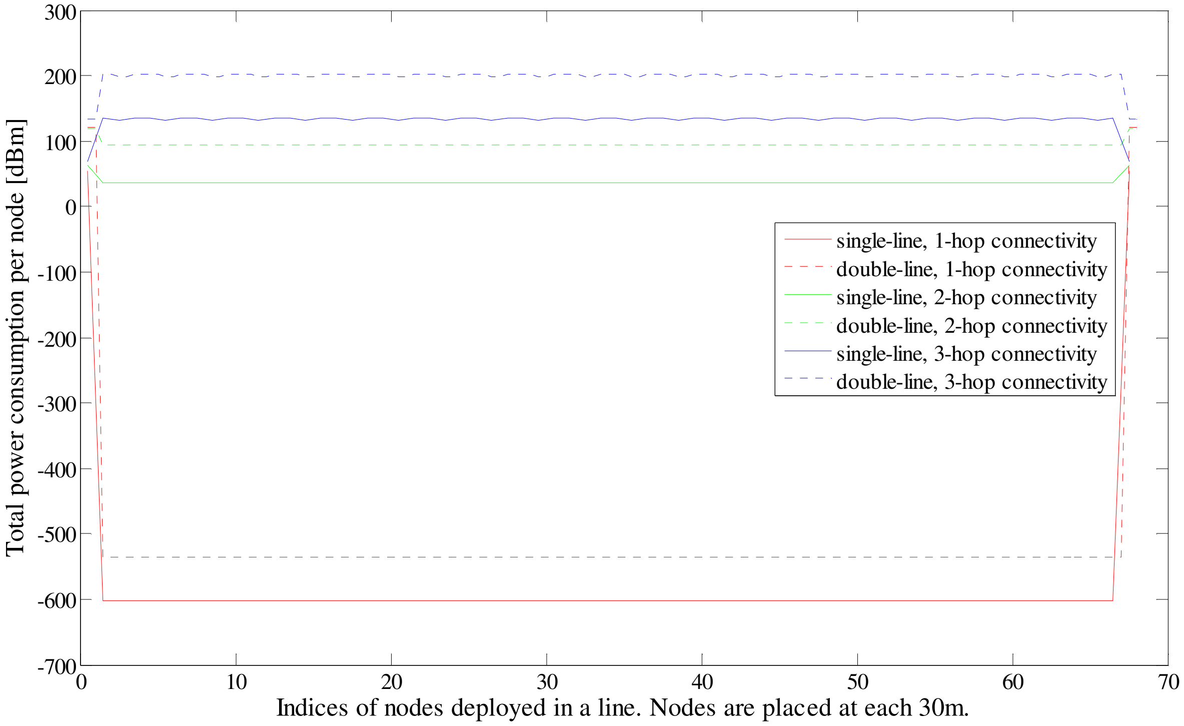
Consequently, when network coding is used, nodes consume less power, as shown in the Figure 4.3. However, even though the higher degree of connectivity results in fewer transmissions, given the higher transmission powers it turns out that simple one-hop direct-connectivity network coding consumes the least power. Figure 4.3 shows that network coding always results in a decrease of power consumption. Thus, the layout in this case is not the key factor for the power consumption. In this case the direct-connectivity level is the most important factor. Effectively, double-parallel-lines layout consumes more power by a margin that is result of an increased number of packet receptions per node. We see that the usage of network coding and one-hop connectivity level performs the best in terms of power consumption per node.

We change the number of nodes deployed in a network from 68 to 40 to see how results scale and to show they are independent of a number of nodes deployed in the network. Figures 4.2 and 4.4 show that these effects are not in any way related to the specific choice of number of nodes. However, due to the fewer number of total messages that need to be relied to sink nodes the average consumption per node is lower. However, one must carefully draw conclusions from this comparison, as it is possible to identify the point when situation reverts as the distance between adjacent nodes may outweigh the number of messages.

We now discuss the total power consumption of network (power needed to relay information from all nodes to the sink nodes) Figure 4.3. It is clear that network consumes significantly less power for lower levels of direct-connectivity among nodes. It is also shown that single-line layout outperforms double-parallel-lines layout, particularly due to the redundant receptions. The smaller distance between equidistant nodes resulted in the increased power consumption as twice as many messages were transferred through the network. Even though double-parallel-lines layout turned out to be inferior to the single-line layout, we cannot completely discard this layout as at times it is the best solution (for very wide bridges) where nodes deployed at one of its sides cannot detect the events from the opposite side of a bridge.

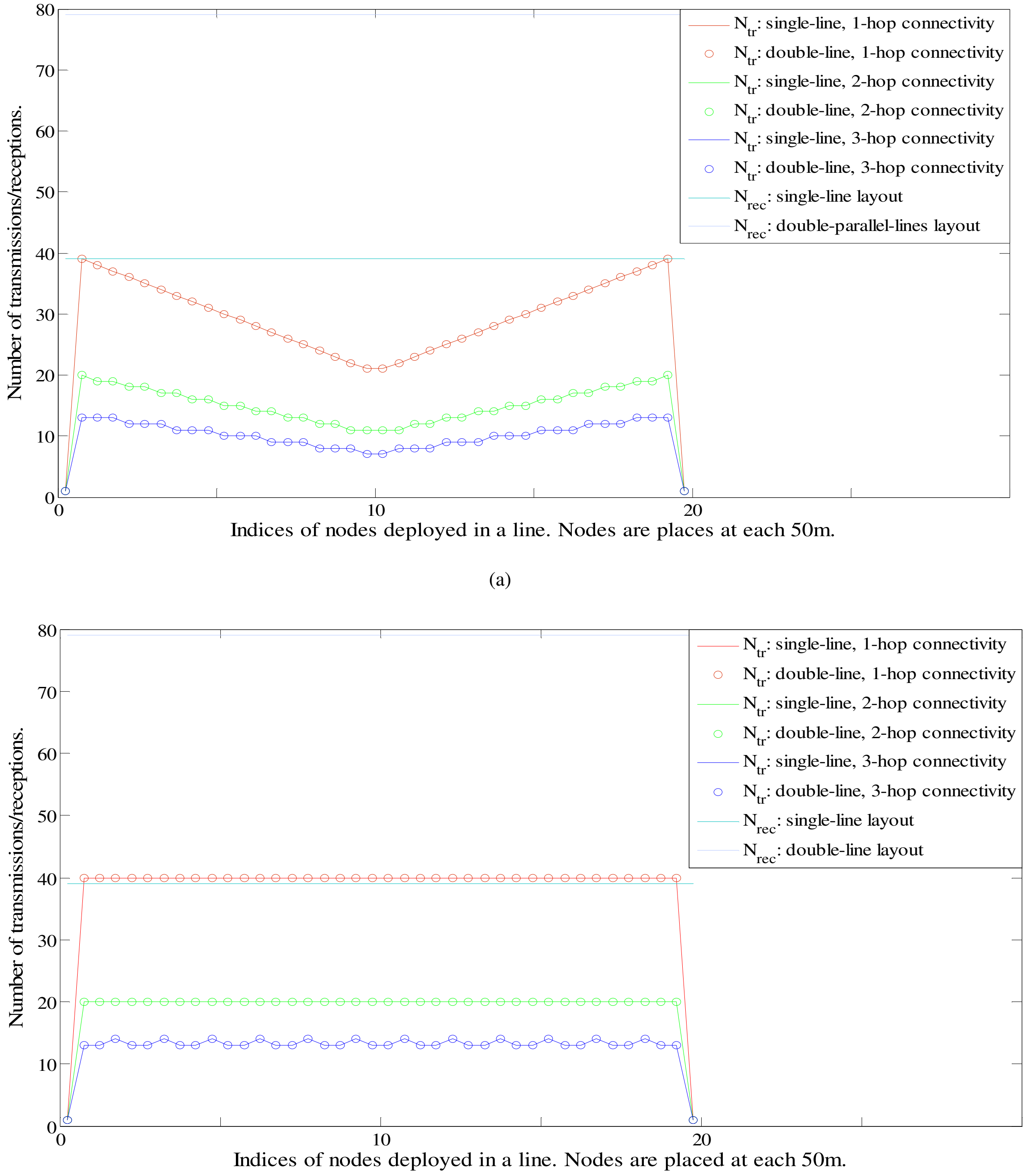


(a)



(b)

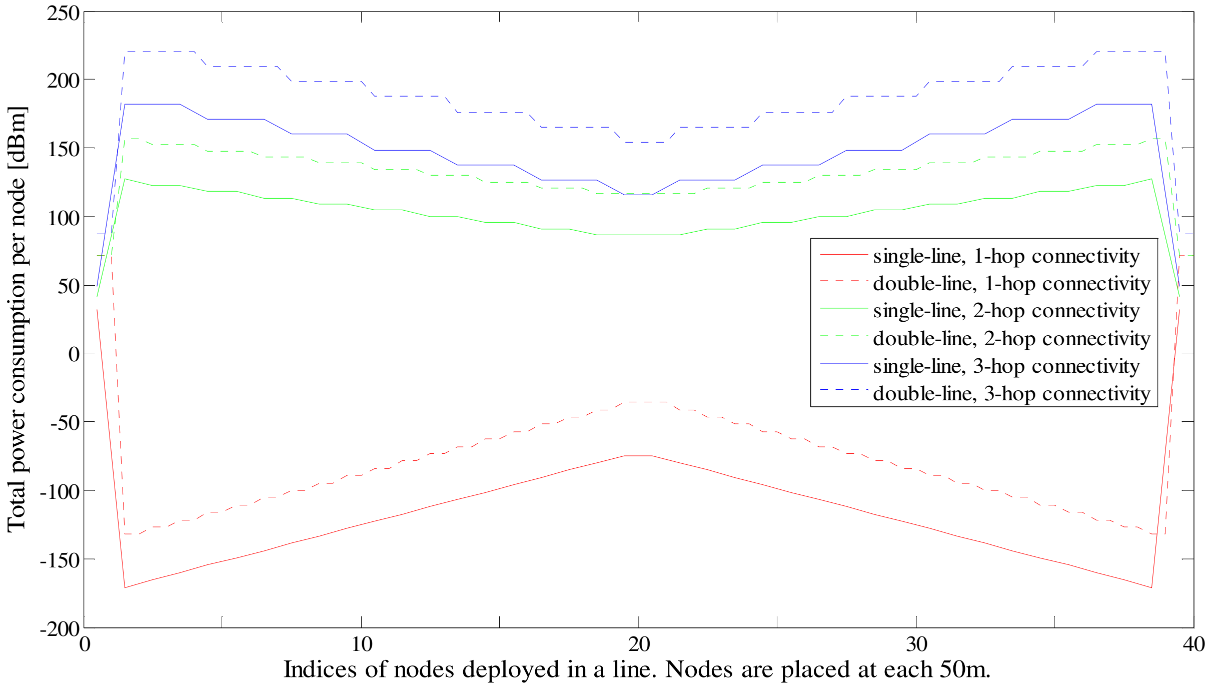
Fig. 4.3. Power consumption per node when a) network coding b) traditional data forwarding, is used.



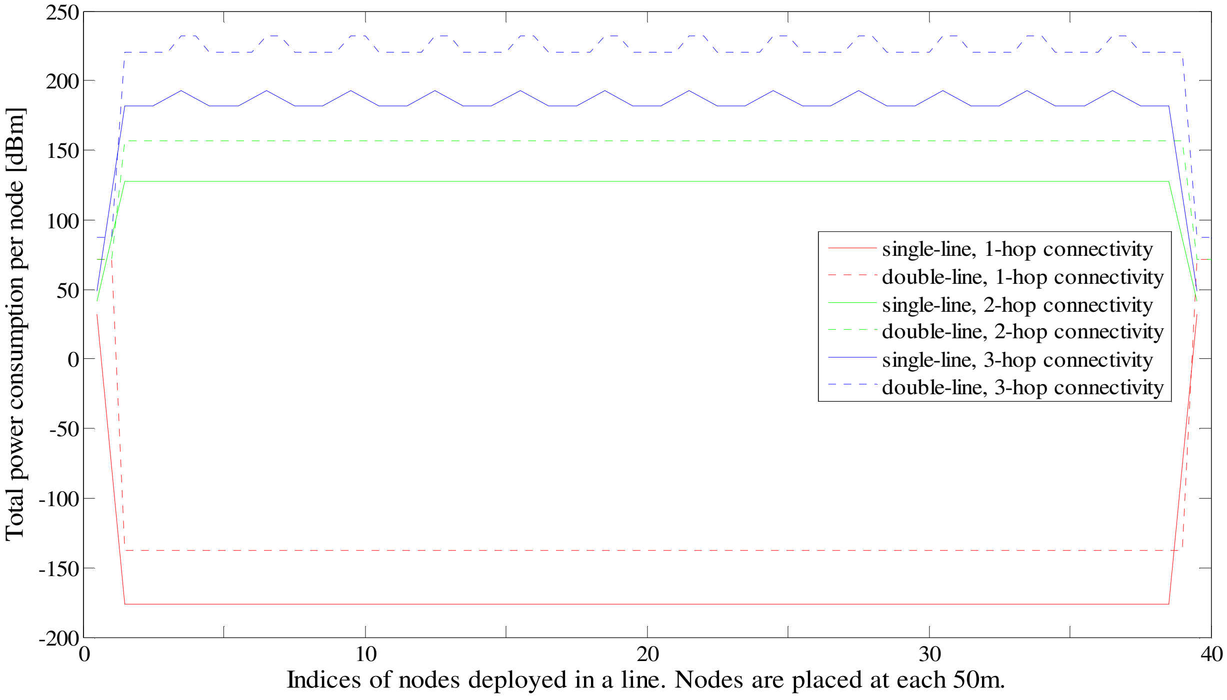
(a)

(b)

Fig. 4.4 The total number of transmissions and receptions per node when a) network coding b) traditional data forwarding, is used.



(a)



(b)

Fig 4.5. Power consumption per node when a) network coding b) traditional data forwarding, is used.

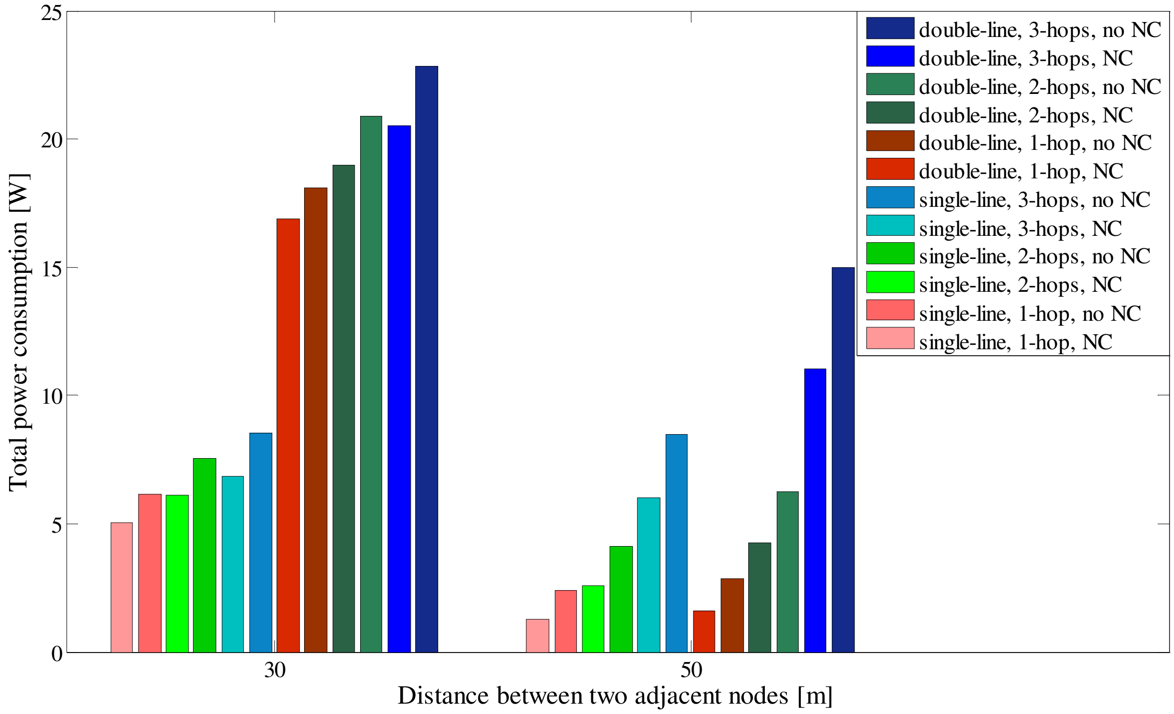


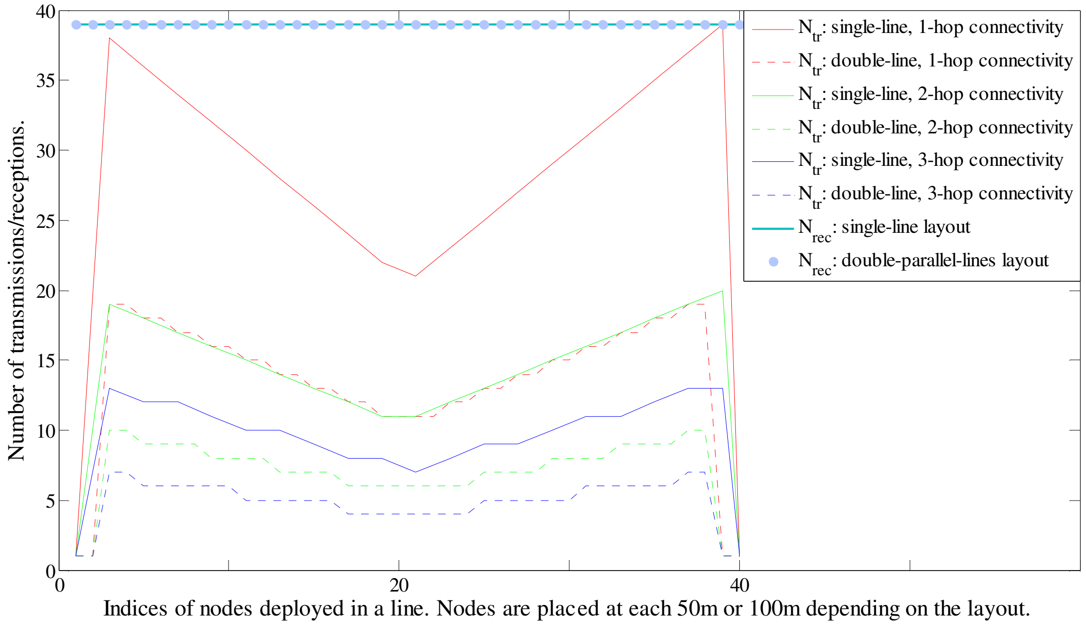
Fig. 4.6. Total power consumption in the network using N nodes for the single-line topology and 2N nodes for the double-parallel-lines topology.

We simulated the performance of a WSN deployed at the 2 [km] long bridge under the constraint that minimum 99.99% of receptions at the furthest designated node(s) are successful. We considered that each side of a given bridge has 40 nodes deployed, and the distance between two adjacent nodes is 50 [m]. For the case of double-parallel-lines topology, power consumption of nodes deployed on opposite sides of bridge is symmetrical. Even though energy consumed per node in the case of one-hop connectivity is lower, the energy consumption in the case of two-hop direct-connectivity is better balanced amongst nodes. When nodes are placed 50 [m] from each other, a total of 40 nodes are deployed in the single-line scenario and 80 nodes are used in the double-parallel-lines scenario. For one-hop connectivity, nodes need to transmit at -5.39 [dBm] to secure 99.99% rate of success at a distance of 50 [m], whereas for the two-hop direct-connectivity, nodes should transmit at 4.58 [dBm] to achieve 99.99% success rate at a 100 [m] distance. Finally to achieve three-hop direct-connectivity with the same probability of successful reception, nodes should transmit at 11.13 [dBm]. We observe how the power consumption is balanced amongst nodes in a network. Since nodes in the middle of a bridge relay almost all messages by means of network coding, they have the lowest power consumption. On the other hand, nodes towards the ends of a bridge send almost all of their messages using regular message forwarding and hence they consume more power compared to the nodes at the middle of the bridge. The lower degree of connectivity, the longer lifetime of the network is.

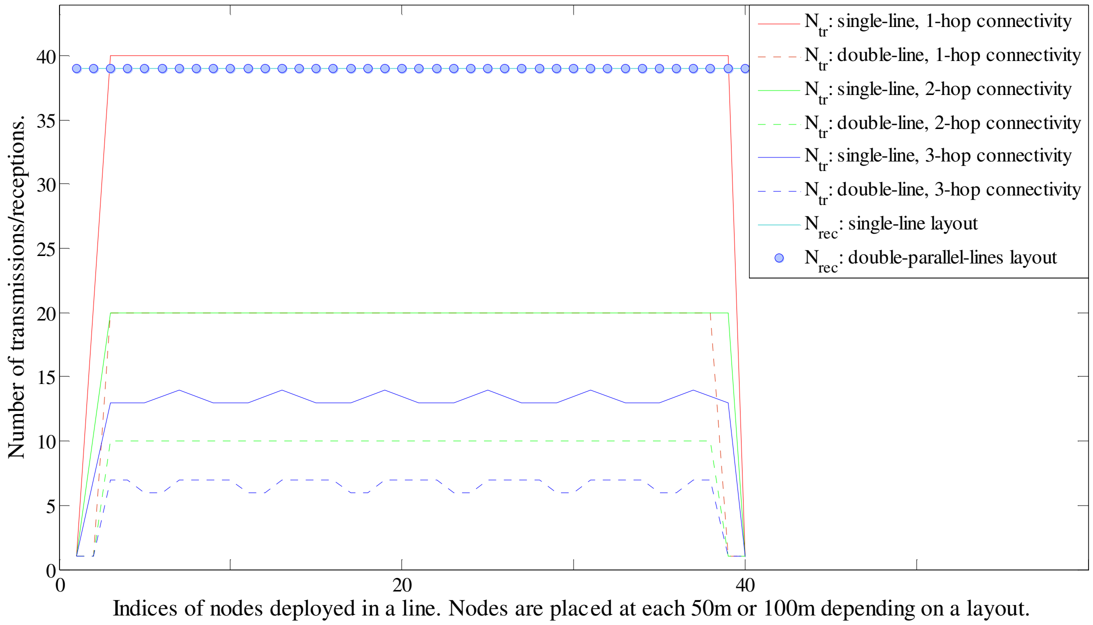
Figure 4.6 shows the total power consumption by all nodes in the network for various combinations of layouts and connectivity when the distance between two adjacent nodes is 30 [m] and 50 [m], respectively. For a given distance, the number of nodes is twice higher in the case of double-parallel-lines layout than it is in the case of single-line layout. It is shown that the single-line layout with the one-hop direct-connectivity consumes the least amount of energy in total (by all nodes). Clearly, for any combination of layout and connectivity the figure reveals that the use of network coding always reduces the power consumption. Compared to the traditional forwarding, our algorithm achieves up to 20% reduction in power consumption. We also see that for a fixed adjacent nodes' distance, the higher degree of connectivity requires the higher power consumption to ensure proper packet decoding. Combining all the effects, the single-line layout with one-hop direct-connectivity among all other combinations performs the best in terms of the power consumption and as a result achieves the longest network lifetime.

4.5.2 Case 2: Single-line network uses N nodes, same as a double-parallel-lines network

Now we observe the situation when regardless of the nodes' layout, the same number of nodes is deployed in both topologies. This is fair in a sense that sink nodes wait for the same number of packets regardless the network layout. However, lines in the double-parallel-lines topology deploy $\frac{N}{2}$ nodes each, meaning that the nodes have to use the higher transmission powers compared to the nodes in a network with the single-line layout. If we closely observe the process in Figure 3.5, we see that nodes from one line relay messages with the odd IDs, and the nodes in the opposite line relay messages with the even IDs.



(a)

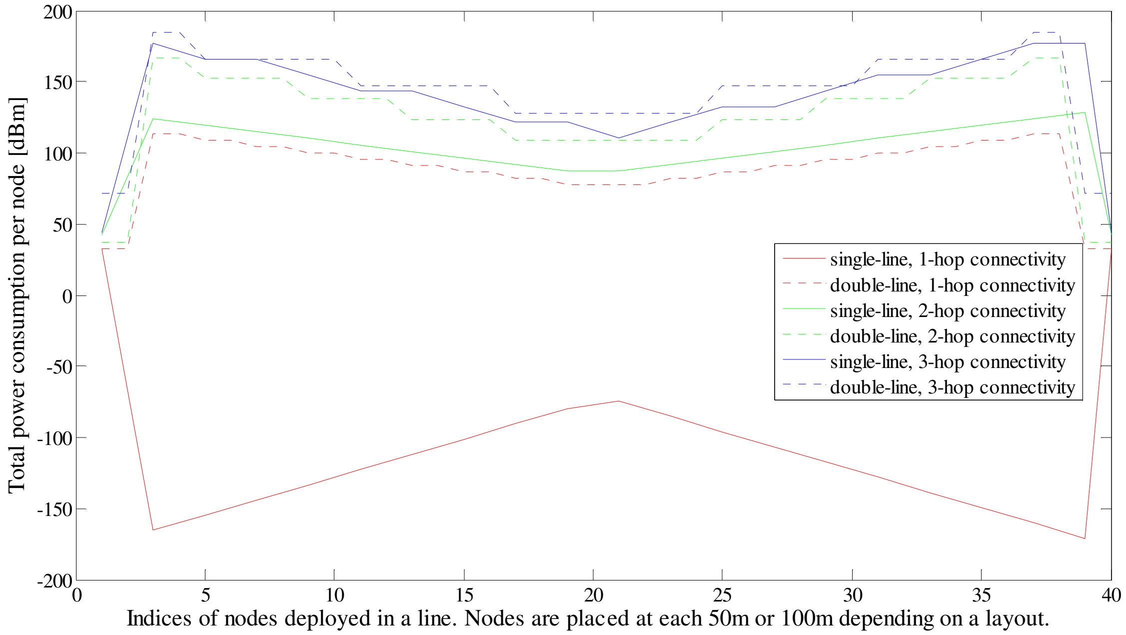


(b)

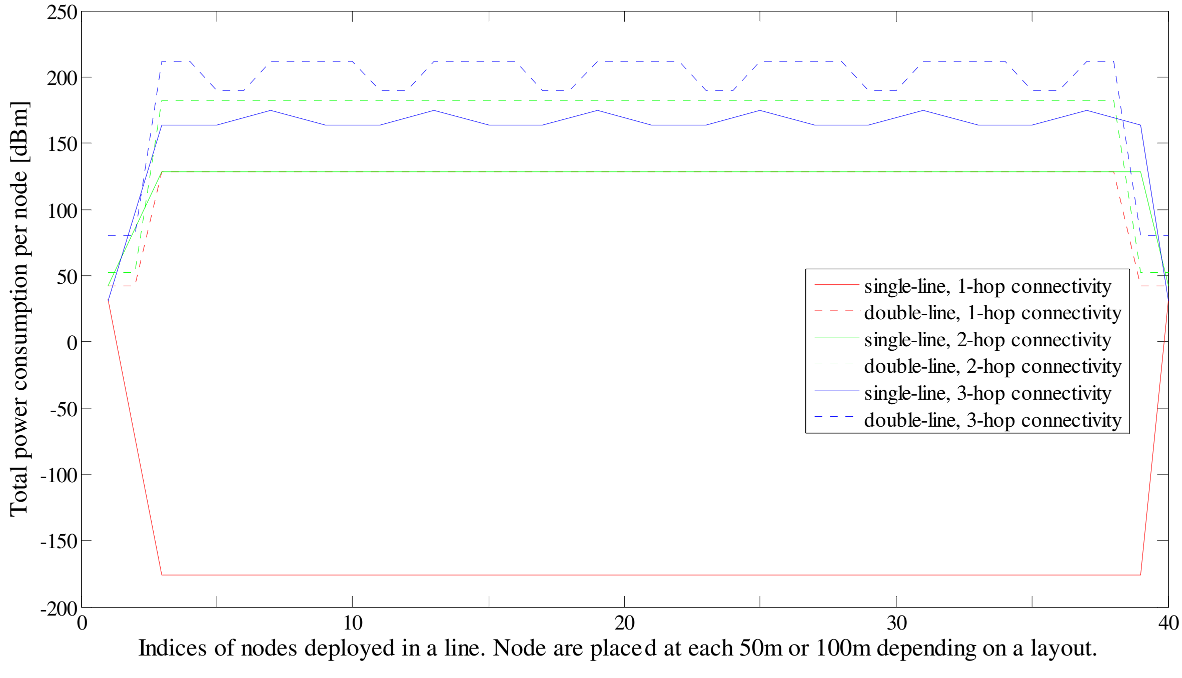
Fig. 4.7. The total number of transmissions and receptions per node when a) network coding b) traditional data forwarding, is used.

Thus, nodes deployed in a network with double-parallel-lines layout transmit 50% less messages compared to the nodes in the single-line layout, but the messages need to be successfully decoded at twice the distance – thus, consume significantly more power. The layout does not affect the number of receptions by each node, as that is solely determined by the number of nodes in a network. Thus, we explore how fewer transmissions, consuming more power each influence the total power consumption distribution. From the data quality point of view, a network with the double-parallel-lines layout is not performing as dense sensing as the same network in the Case1, with twice as many nodes covering the same distance. This reflects poorly on the quality of collected data.

We simulated the performance of a WSN deployed at the 2 [km] long bridge under the constraint that min 99.99% of decoding at the furthest designated node(s) should be successful. We considered that there are 68 nodes deployed at the bridge, resulting in the distance between two adjacent nodes of 30 [m]. For the case of the double-parallel-lines topology, the power consumption for nodes deployed in both lines is symmetrical. We see that even though the energy consumed per node in the case of one-hop connectivity is lower, the energy consumption in the case of two-hop direct-connectivity is better balanced amongst nodes. As the same number of nodes is deployed in the double-parallel-lines scenario, the nodes in a line are placed 60.6 [m] from each other. For one-hop connectivity and single-line layout, nodes need to transmit at -9.81 [dBm] to secure 99.99% rate of success at a distance of 30 [m], whereas for the two-hop direct-connectivity, nodes should transmit at -0.81 [dBm] to achieve 99.99% success rate at the 60 [m] distance. Finally to achieve the three-hop direct-connectivity at the same probability of success, nodes should transmit at 3.08 [dBm]. Now we observe a double-parallel-lines layout. For one-hop connectivity, nodes need to transmit at -0.84 [dBm] to secure 99.99% rate of success at a distance of 60.6 [m], whereas for the two-hop direct-connectivity, nodes should transmit at 9.02 [dBm] to achieve 99.99% success rate at the 181.8 [m] distance. Finally to achieve the three-hop direct-connectivity with the same probability of success, nodes should transmit at 12.1 [dBm].



(a)

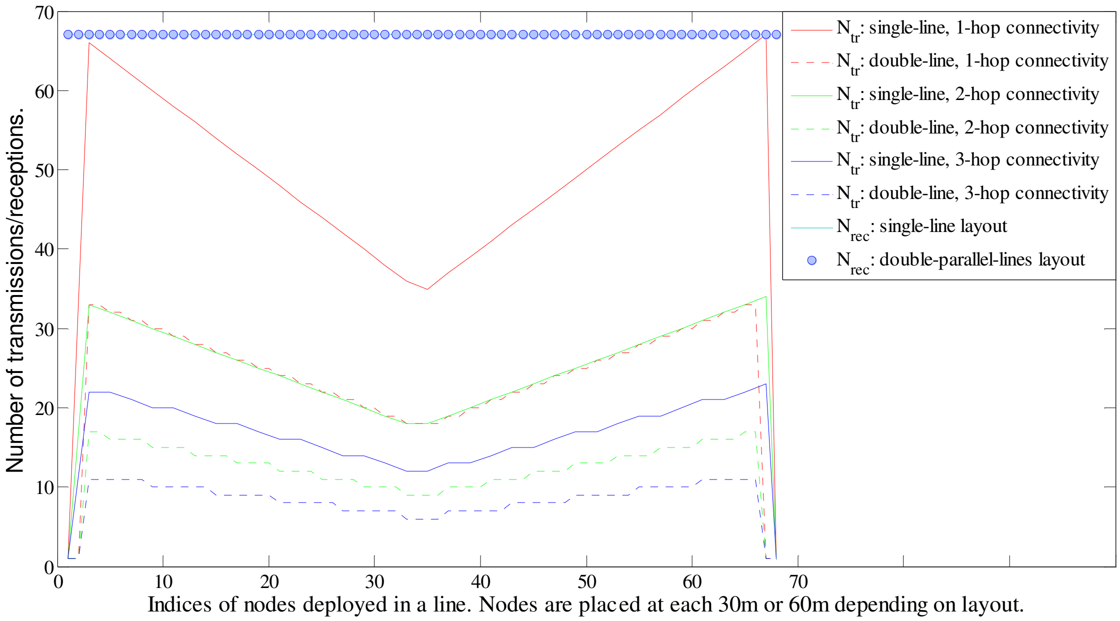


(b)

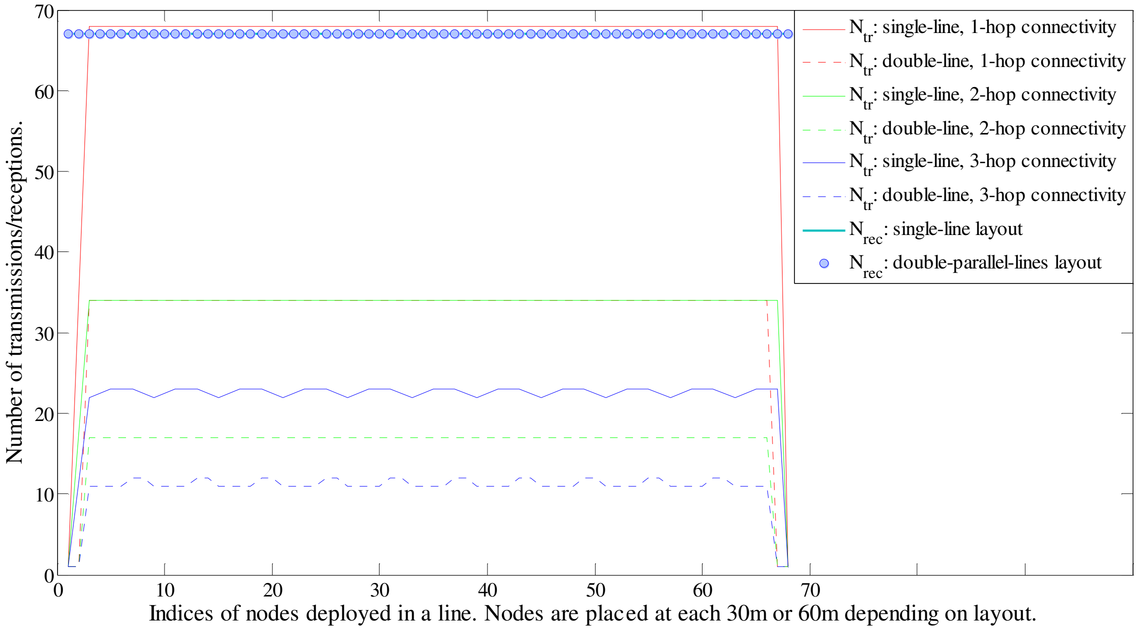
Fig 4.8. Power consumption per node when a) network coding b) traditional data forwarding, is used.

The Figure 4.9 shows that the application of the network coding technique results in the decrease of total number of data transmissions, while it has no effect on total number of receptions. We may notice that there is almost the overlap of curves describing the single-line two-hop direct-connectivity case and the double-parallel-lines one-hop direct-connectivity case. This is because in both cases the same number of neighbours receives a new piece of information, for each transmission. We see that for a given connectivity a network with double-parallel-lines topology performs fewer transmissions. In spite of the fewer transmissions, the double-parallel-lines topology is still inferior to the single-line topology as the transmission powers in the second case are much higher in order to achieve the same probability of successful reception at twice the distance. That is why one-hop connectivity case does not result in similar power consumption characteristics for both layouts. It shows that power consumption is more sensitive to changes in distance among adjacent nodes than the number of messages.

We simulated the performance of a WSN for the 2 [km] long bridge under the constraint that min 99.99% of receptions should be successful. We considered that 40 nodes are deployed at a bridge, for the distance between two adjacent nodes of 50 [m]. For the case of the double-parallel-lines topology, power consumption for nodes in both lines is symmetrical. Even though the energy consumed per node in the case of one-hop connectivity is lower, the energy consumption in the case of two-hop direct-connectivity is better balanced among nodes. As the same number of nodes is deployed in the double-parallel-lines scenario, the nodes in a line are placed 105.26 [m] from each other. For one-hop connectivity and single-line layout, nodes need to transmit at -5.39 [dBm] to secure 99.99% rate of success at a distance of 50 [m], whereas for the two-hop direct-connectivity, nodes should transmit at 4.58 [dBm] to achieve 99.99% success rate at the 100 [m] distance. Finally to achieve three-hop connectivity with the same probability of success, nodes should transmit at 11.13 [dBm]. Now we observe the double-parallel-lines layout. For one-hop connectivity, nodes need to transmit at 4.52 [dBm] to secure 99.99% rate of



(a)



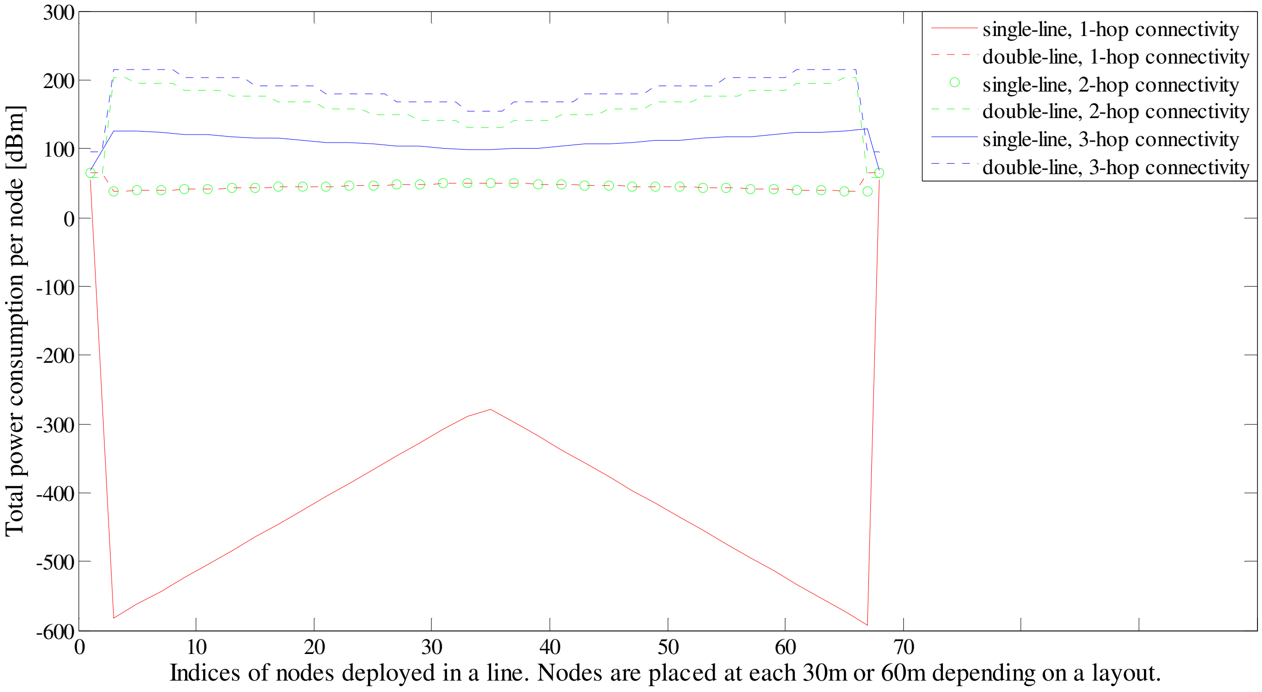
(b)

Fig 4.9. The total number of transmissions and receptions per node when a) network coding b) traditional data forwarding is used.

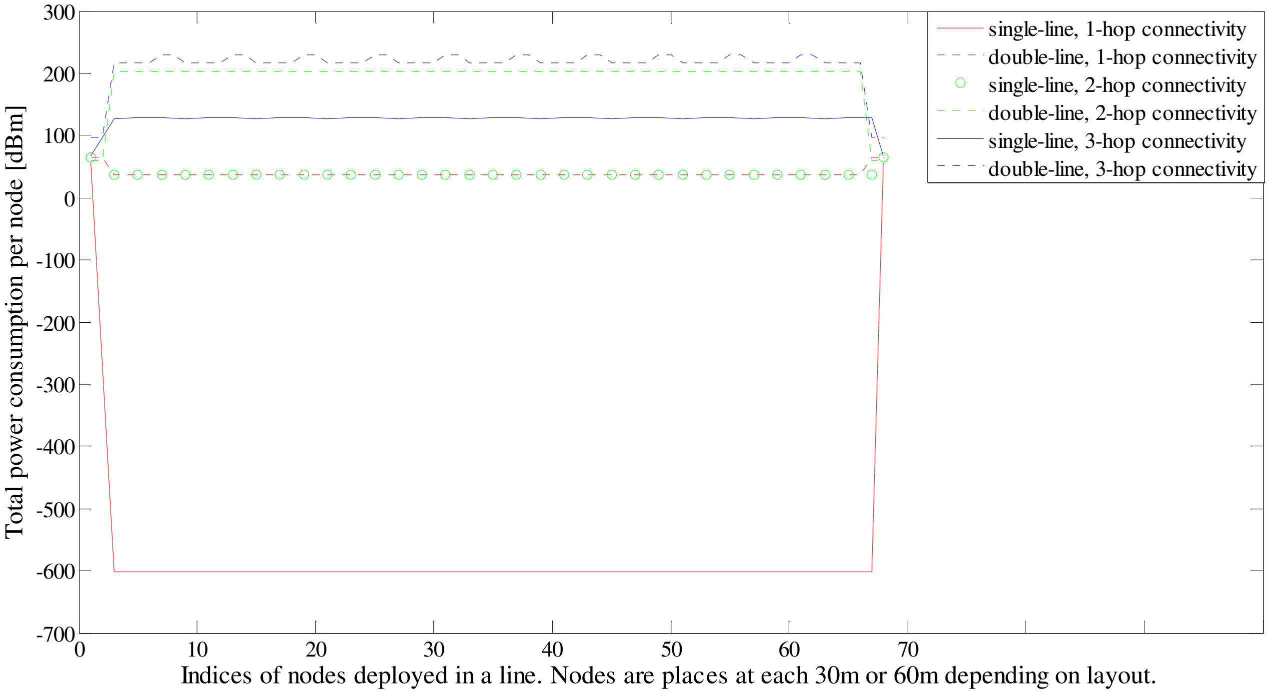
success at the distance of 105.26 [m], whereas for the two-hop direct-connectivity, nodes should transmit at 14.4 [dBm] to achieve 99.99% success rate at a 210.52 [m] distance. Finally to achieve the three-hop direct-connectivity with the same probability of success, nodes should transmit at 18.82 [dBm]. We repeat simulations to see how results scale for the case of a network with different number of nodes. We almost double the number of nodes, and we have around 20% increase in the power consumption. This happens as nodes can decrease their transmission powers to keep the same direct-connectivity level with the same probability. It is shown that for a given connectivity, a network with single-line layout of nodes always performs better than a network with the same number of nodes with the double-parallel-lines topology. It is also shown that for a given layout, the higher degree of connectivity, the higher power consumption is. Single-line layout combined with one-hop direct-connectivity performs the best in terms of power consumption.

Figure 4.11 shows the total power consumption by all nodes in the network for various combinations of layouts and connectivity, where the distance between two adjacent nodes is 30 [m] and 50 [m], respectively. We assume that a network always has the same number of sensor nodes regardless of the layout. It is shown that the single-line layout with the one-hop direct-connectivity consumes the least amount of power in total. For any combination of layout and connectivity, the Figure 4.11 shows that the use of network coding always reduces the power consumption.

Compared to the traditional data forwarding, our algorithm achieves up to 20% reduction in the power consumption. We can also see from figure that for a fixed adjacent nodes' distance, the higher degree of connectivity requires higher power consumption to secure the successful message reception. Combining all the effects, the single-line layout with the one-hop direct-connectivity performs the best in terms of power consumption and as a result achieves the longest network lifetime. In spite of the fact that this case requires the highest number of transmissions.



(a)



(b)

Fig. 4.10. Power consumption per node when a) network coding b) traditional data forwarding is used.

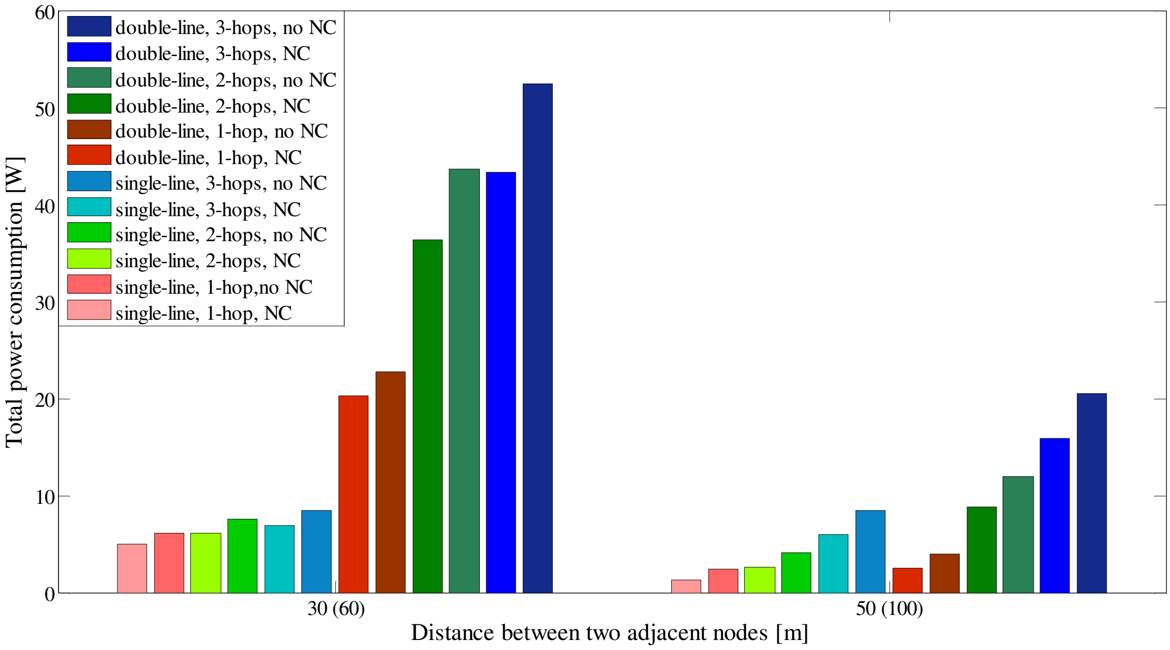


Fig. 4.11. Total power consumption for WSN with network coding and with traditional data forwarding. Distance between adjacent nodes is 30 [m] and 50 [m] for single-line, and 60 [m] and 100 [m] for double-parallel-lines layout.

4.6 Summary

So far, we proposed and analysed the performance of a new algorithm that employs the network coding technique for SHM of bridges and linear structures. Wireless sensors are deployed on a bridge to take (sense) a variety of measurements. The key idea is to use the network coding technique to reduce the number of transmissions needed to forward all sensing data to sink nodes. We analysed the performance gain of the network coding technique. We adjusted the transmission powers to cope with the propagation environment characterized by combined fading, shadowing and path loss to assure a high probability of successful receptions. We evaluate the power consumption by all nodes in the network. It has been found that the network coding method always reduces the power consumption for a given node layout and degree of connectivity. However, special attention has to be paid to the layout and connectivity if the objective is to minimise the overall power consumption (or to maximise the network lifetime) when the network coding technique is applied. For the numerical examples considered, the scenario with the single-line layout, one-hop connectivity and network coding technique applied, consumes the least amount of energy. It is so because networking coding gain depends on the node layout and connectivity in some complicated ways.

In this chapter we evaluated the performance of to the best of our knowledge, the only algorithm that deploys network coding for structural health monitoring of bridges. We used the power control technique to cope with the volatile propagation environment effected by fading, shadowing and path loss; to assure the high probability of successful receptions.

We evaluate the power consumption of each node, and the overall power consumption. We compared the result with the case when evaluated algorithm is not applied, which we refer to as traditional data forwarding. We conclude that network coding should be carefully applied. Transmitting with higher powers creates the overhearing opportunities and increases the network coding gain. However, in order to reach directly the second-hop neighbours a node has to increase significantly the transmission power which consumes too much of power. Thus, we show that single-hop connectivity is a better option in terms of power consumption. For a given layout, a network with higher level of connectivity consumes more power than the same network with the lower level of connectivity. The results presented in this chapter are published at IABMAS'12 [71].

5. NODE PLACEMENT IN LINEAR WIRELESS SENSOR NETWORKS

5.1 Chapter Abstract

In the previous chapters we discussed the structural health monitoring of bridges. As an underlying assumption, we considered a monitoring of bridge with fixed length and equidistant sensor nodes deployed along one or both of its sides. As SHM applications produce high volume of sensing data, we used the network coding technique to decrease the volume of data.

However, in the previous chapter we showed that there is a certain degree of unfairness among different sensor nodes. Some nodes relay higher data loads than others. All nodes have the same data load in terms of number of necessary receptions. And as distances among adjacent nodes are fixed, this means that some nodes drain out their energy supply faster than other nodes. Once a node dies, its first-hop neighbours will be forced to transmit with the higher power level than it was initially planned to keep the network connected. Consequently, this accelerates the rate at which limited battery supply is used. Dying of nodes is a chain effect. Soon after a node dies its neighbours will die in attempt to reconnect the network. Once three adjacent nodes die, we consider a network to be disconnected (as remaining nodes that border the chain of failed nodes would be forced to increase the transmission power by more than 18 [dB]). Probably the worst consequence of this effect is the increased maintenance price. Technicians are forced to replace batteries at certain nodes quite often.

Thus, due to the uneven data loads the energy supply of some nodes drains out faster, consequently they die and so do their neighbours and the network becomes disconnected. In this chapter, we propose a methodology for the optimal placement of sensor nodes for linear network topologies (e.g., nodes deployed along the length of a bridge) that aims to minimise the link connectivity problems and maximise the lifetime of the network. Both simple packet relaying and network coding are considered for the routing of the collected data towards two sink nodes positioned at both ends of the bridge. Our mathematical analysis, verified by simulation results, shows that the proposed methodology leads to the significant energy saving and prolonged network lifetime. Results presented in this chapter are published at EUSIPCO'13 [72].

5.2 Related Work

Last few years there is an increased interest in the planning of WSNs. Researchers considered various objectives such are fault tolerance, coverage area, network connectivity, network lifetime, data fidelity, number of nodes, load balancing and energy efficiency. Zou et al. optimised the sensor nodes placement with the objective to increase the sensing coverage [60]. Their work measures the probability that an event would happen without being detected. An interesting criterion to use as an objective function is the network lifetime – the time until the network gets partitioned in a way that is impossible to collect the data from a part of that network [58]. Some papers used multiple objective parameters [59].

5.3 Motivation

In the previous chapters we proposed a network coding based protocol for wireless sensor networks used for structural health monitoring of bridges. Network coding increases the transmission rate without distorting the data and decreases the number of messages that should be transmitted. Therefore NC also decreases the power consumption throughout the network.

However, due to the linear type of the network topology and the nature of proposed algorithm some sensor nodes have to relay more traffic than others. As a result, the energy of those nodes drains out faster, and these sensors tend to die early. Thus, the network soon becomes disconnected. When a sensor node fails its immediate neighbours increase their transmission powers in order to preserve the connectivity of the network. However, the rate at which those sensors spend their energy supply increases sharply, as they attempt to achieve the same probability of successful decoding at twice the distance. Thus, those nodes also fail soon after. We will refer to this effect as *chain nodes failure effect*. We assume that the network is disconnected once three nodes that used to be immediate neighbours fail (drain out of battery supply).

In this chapter, we resolve the problem of the uneven data traffic distribution which creates discontinuities and bottlenecks in the network by controlling the positioning of sensor nodes (i.e., the distribution of the sensors on the bridge). In this way we maximise the total volume of data traffic that can be processed and maximise the overall lifetime of the network. More specifically we develop a numerical solution that calculates the required number of sensor nodes and their optimum positioning over a bridge given the length of the bridge and the required data throughput. Our numerical and simulation results indicate that the proposed method prolongs the network lifetime by up to 25% while at the same time it eliminates the bottlenecks.

5.4 Scenario

5.4.1 System Model

So far we assumed that we monitor a bridge of fixed length, 2 [km]. We assumed that sensor nodes are equidistantly placed along the line. We assumed that a perfect Medium Access Layer (MAC) is in use, so that no data losses due to the interference happen. We assumed that the channel is not lossless. The expressions we obtained for the number of transmissions are completely independent of the positions of sensor nodes. We have noticed that some nodes die faster than others, as a consequence of equal distances among nodes coupled with their uneven data loads.

In this chapter, we abandon the assumption of equidistant sensor nodes. We try to determine the ideal positions of sensor nodes, in order to make all nodes die at the same time. Our goal is to extend the network lifetime, decrease the maintenance cost and remove the bottlenecks in the network.

Therefore, the most important assumption of this chapter is that nodes are not equidistant any longer. We assume that nodes are deployed at the one side of a bridge, forming a line. We assume that all nodes report their data to both sink nodes that are located at the ends of the bridge. Sink nodes are connected to Internet and are used as a gateway.

We assume that channel is predominantly under the influence of path loss and limited interference. The interference we are taking into account is coming from limited number of nearest nodes, while the potential interference from further nodes is considered insignificant.

We assume that nodes fail only when they run out of power supply. Physical damage of nodes, or issues with nodes' electronics are neglected in this thesis.

5.4.2 Network Topology

Let us consider a linear wireless sensor network, so far referred as single-line layout of nodes. A set of nodes deployed in a network is identical (in terms of current, voltage, computational and communication capabilities). Two sink nodes (i.e., the nodes that aggregate information generated by the whole network and serve as a gateway to the Internet) are placed at the two ends of the line. This is a fair assumption as sensors are expected to be deployed along the length of the monitored bridge. All sensor nodes are collecting information related to the structural health of the bridge periodically and try to relay this information to both sink nodes. SHM applications require high reliability of collected data, and frequent failures caused by packet errors and link failures are not working in its favour. Thus, packets should be relied to the both sinks. This provides the redundancy against packet loss due to volatile wireless channel, interference or other network failures. The cost of using two sink nodes is low given the significance of reliable data collection which is vital to prevent infrastructure failure.

In order to maximise the operational lifetime of sensor nodes that have limited energy supply, sensor nodes need to transmit data at a limited power and hence form a multi-hop network [64]. Based on transmission power level, we observe three different network connectivity cases: one-hop, two-hop and three-hop direct-connectivity. Let us further assume omnidirectional transmissions, as previously explained. As we concluded earlier, the single-line layout is superior to the double-parallel-lines layout, in the following chapter we will focus solely on the single-line nodes layout.

5.4.3 Channel Model

In our analysis we consider interference from limited number of wireless nodes. We consider the interference power to be much higher than background noise level, and therefore noise can be neglected. We use the Shannon's capacity formula [48] to define the wireless link data rate as:

$$C = B\, log_2\,(1 + SINR)\ , \tag{5.1}$$

where *C* is the link capacity in [bits/sec/Hz], *B* is the channel bandwidths in [Hz], and *SINR* is the Signal to Interference plus Noise Ratio. We further consider frequency reuse factor to be equal to 3, as suggested in [45]. We allow sensors that are located three times the transmission range away from each other to transmit at the same time. For instance, in the 1-hop case, nodes that are located three-hops away may share the same timeslot. Similarly, in the 2-hop and 3-hop cases nodes that are six hops and nine hops away, respectively, will be allowed to use the same timeslot. Finally, we consider that significant interference comes only from the closest interfering nodes. For instance, in the 1-hop direct-connectivity case, we only consider potential interference from nodes located two-hops away from the receiver and we ignore the interference from nodes located four hops away or any other more distant simultaneous user of wireless channel.

$$SINR = \frac{P_{r,i}}{\sum P_{r,j}} = \frac{P_{r,i}}{P_{r,I}} \tag{5.2}$$

The receiving power is given by $P_r = P_t d^{-\gamma}$ where P_t is the transmitting power, d is the distance between transmitter and receiver and γ is the path-loss exponent, with a value between 2 and 5.

Very common problem in wireless communications is the hidden terminal problem. When a number of nodes are outside of each other's communication range, it happens that they send data at the same time to the same receiver, causing the data loss at the receiver and consequently retransmissions. These packets therefore collide at the receiver. Data collisions do not always result in data loss. If power level at the receiver of one transmitted packet is stronger than all others, the receiver will successfully decode it. This phenomenon is known as capture effect [49].

The proposed protocol is prone to collisions as nodes are equidistant, and they use the same transmission power level. In the ideal conditions, power level of two signals received from two equally distant nodes in the network is the same at the receiver. However, this is where the fading and shadowing boost the probability of successful reception at the receiver. Two signals arriving at the same time to the receiver will rarely have the same power level as they undergo different channel characteristics. In our work, we take into account the capture effect when signals originating from different nodes arrive at the same receiving node simultaneously, with the different power levels. If the difference in their power levels is larger than the certain threshold, the stronger signal will be decoded at the receiver despite the collision, while the weaker signal will have to be retransmitted. Otherwise, both of signals would be lost, and both packets would need to be retransmitted. We will discuss the probability of capture in a wireless channel that arises not only because of the different distances from the receiving node (near-far effect), but also because of the fluctuations in the signal level due to the random nature of shadowing and fading.

5.5 Mathematical Analysis

All nodes initially have the same amount of energy available. Energy spent for the receptions, E_{rec} , is equal for all nodes in the network, as all of them receive the same number of packets (nodes sleep when they do not expect a packet, and at this point the communication is relatively reliable). Therefore, all nodes have the same amount of energy available for the data transmission, E_{tot}. The total energy spent by node with the ID i on transmissions is given by:

$$E_{tot} = P_{t,i} N_{t,i}, \tag{5.3}$$

where $N_{t,i}$ is the total number of transmissions by node with the ID i:

$$N_{t,i} = \begin{cases} \left\lceil \frac{i}{h} \right\rceil, & \left\lfloor \frac{N}{2} \right\rfloor < i < N \\ \left\lfloor \frac{N-i}{h} \right\rfloor + 1, & 1 < i < \left\lfloor \frac{N}{2} \right\rfloor \\ 1, & otherwise \end{cases} \tag{5.4}$$

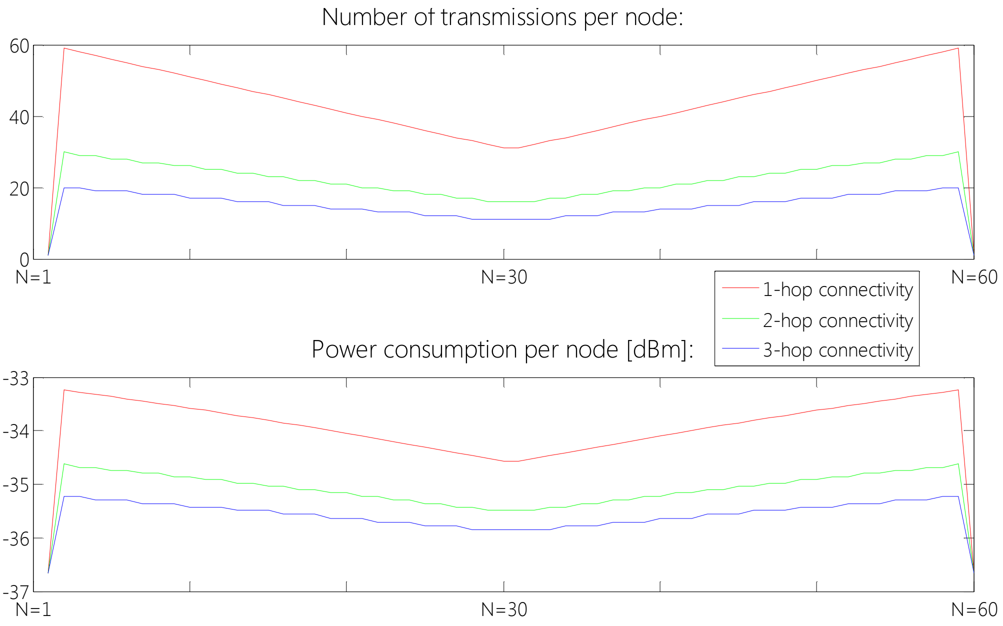


Fig. 5.1. The uneven power consumption distribution in the network.

The hop factor h takes values $\{1, 2, 3\}$ for the one-hop, two-hop and three-hop cases respectively. N denotes the total number of sensors deployed on bridge, and i is the index of the observed node ($i = 1, \ldots, N$).

In the further work, we assume that frequency reuse factor and transmission powers are chosen carefully enough to cancel out the probability of failed decoding. Thus, all nodes receive the same number of messages.

As shown, all nodes consume the same energy for their receptions. We only discuss the power consumed for transmissions. As all nodes started with the same battery supply, and all of them consume equally the power for receptions, the power available for transmissions is equal for all sensor nodes. Nodes located at the ends and in the middle consume less energy than the surrounding nodes. Nodes that have the most of load are those nodes who relay the most of packets via traditional forwarding. Sink nodes that transmit only once, and middle nodes that use only network coding based forwarding consume less power. Since nodes that relay messages to their neighbours via traditional forwarding have to relay more packets than middle and sink nodes, and all nodes transmit at the same power, they will drain out battery supply faster and die quicker. Once a node dies, its neighbours reconnect the network by increasing their transmission power level, which drains their energy much quicker. If three nodes in a row die, we consider the network disconnected.

All nodes with the high energy consumption represent spots of high risk. It is expensive to go to the field each time a node fails/a network gets disconnected. Ideally, all nodes would be replaced at the same time, driving down the cost of network maintenance.

The idea is to push nodes with the low energy consumption (those relying mostly on network coding based communication) to transmit their messages to the higher distances, pushing them to cover higher areas. While nodes with the high energy consumption, can use this to their advantage and transmit with the lower powers to shorter distances, decreasing their power consumption. Ideally, all nodes end up draining out of energy simultaneously. At the same time, as equation (5.1) indicates, the transmission power affects the data rate of wireless links. Since we require all nodes to transmit at the same data rate, the distance between nodes surrounding the centre must be smaller compared to the distance of the nodes at the very edges. In order to account for the aforementioned issues we impose the following two conditions in the network:

- C1: All nodes must drain out of energy at the same time. In this way the network will remain connected for as long as possible.

- C2: All sensor nodes must transmit at the same data rate. In this way we avoid any bottleneck in the data flow and we ensure the availability of packets for network coding.

In the following pages we calculate the optimal position of the sensor nodes such that the conditions C1 and C2 hold.

From eq. (5.1) we obtain:

$$SINR = 2^{\frac{C}{B}} - 1. \tag{5.5}$$

Using (5.2) and (5.3) and the assumed path loss model we obtain the following formula for the SINR which is independent of the transmission power.

$$SINR = 2^{C/B} - 1 = \frac{P_{t,s} d_s^{-\gamma}}{P_{t,I} d_I^{-\gamma}} = \frac{N_{t,I}}{N_{t,s}} \left(\frac{d_I}{d_s}\right)^{\gamma}, \tag{5.6}$$

where d_s is distance between a node that transmitted a signal that a receiver wants to decode, to which we will refer as *a useful signal*; d_I is the distance between the interferer and the receiver. $N_{t,I}$ stands for the number of transmissions from a node that is creating the interference, while $N_{t,s}$ stands for the number of transmissions from a node where the useful signal is originating, γ is the path loss exponent.

$$d_I - d_s \sqrt[\gamma]{(2^{C/B}-1)\frac{N_{t,s}}{N_{t,I}}} = 0 \cdot \tag{5.7}$$

Equation (5.6) can be generalized as:

$$d_j - d_{j-1} \sqrt[\gamma]{(2^{C/B}-1)\frac{N_{j-1}}{N_{j+1}}} = 0 \ , \tag{5.8}$$

where j can be any node with the ID $j = \{2, 3 \dots \text{N-1}\}$, while d_j is the distance between the nodes with ID j and $j+1$. Similarly, if the interference comes from left side in a line, and useful signal comes from right side, we have:

$$d_{j-1} - d_j \sqrt[\gamma]{(2^{C/B}-1)\frac{N_{j+1}}{N_{j-1}}} = 0 \, . \tag{5.9}$$

The final constraint is related to the total number of sensor nodes required to cover the whole length of the bridge (denoted by L)

$$\sum_{j=1}^{N-1} d_j = L \quad . \tag{5.10}$$

If we solve this set of equations, we obtain the recommended layout of nodes on the bridge. Following the same principle, in a general case we have:

$$\sum_{k=1}^{h} d_{j-k} - d_j \sqrt[\gamma]{(2^{C/B}-1)\frac{N_{j+1}}{N_{j-h}}} = 0 \ , \tag{5.11}$$

where j = {h+1,…N-1}. For j = {2,…N-h},

$$\sum_{k=0}^{h-1} d_{j+k} - d_{j-1} \sqrt[\gamma]{(2^{C/B}-1)\frac{N_{j-1}}{N_{j+h}}} = 0 \, . \tag{5.12}$$

Solution of the set of equations that is defined by (5.9), (5.10) and (5.11) provides the optimum distribution of sensor nodes on a bridge such that the bottlenecks are eliminated and the network lifetime is maximised.

This set of rules can serve as indicators during the sensor network deployment.

5.6 Simulation Testing

We run extensive simulations to verify the numerical analysis and assess the performance improvement of the proposed network deployment rules. We use linear programing to solve the optimisation problem. MATLAB has built in solvers for linear programming (linprog function).

Since radio link changes with the time, we adjust the transmission power at each node so that the transmitted signal can be decoded by immediate neighbours. We use the $SINR_{tresh}$=6 [dBW] as a threshold and a condition for a successful packet decoding. We chose this value as we found it in the specification of a sensor node. Nodes have possibility to choose between small number of different values for their transmission powers.

Each node is equipped with the AA alkaline battery (current I = 1700 [mAh] and voltage U = 1.5 [V]) that is commonly used for EYESIFXv2 sensor nodes. The transceiver in the low power mode uses the following set of values for current I = {4.1, 4.9, 6.8} [mA], and I = {9.4, 11.9, 14.6} [mA] in the high power mode. The available voltage values are U = {2.1, 3, 5} [V].

The receiving unit of these nodes uses the current of I = {9, 9.5} [mA] and voltage of U = {3, 5} [V]. We set the path loss exponent γ to 3, which is the value commonly used in the given environment. We assumed the packet length of 128 [B], and the link capacity of 250 [kbps]. The bridge we observe is 2[km] long.

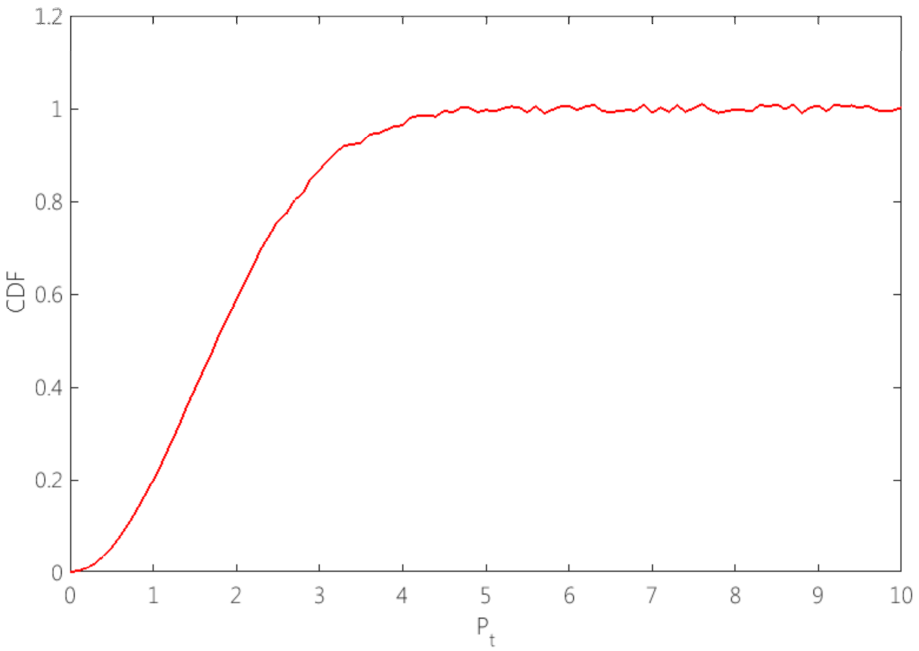


Fig. 5.2 Cumulative Distribution Function of Wireless Channel for various transmission power levels [dBm] and distance between adjacent nodes of 50 [m].

Figure 5.2 shows CDF of wireless channel for different transmission powers of wireless nodes and distance among adjacent nodes of 50 [m]. Figure 5.3 shows optimised layout of nodes in the network based on proposed algorithm. Nodes are unevenly spread along the bridge. Most of nodes are concentrated towards the middle of the bridge, and the rest are spread out towards the very ends. When our algorithm is used, nodes in the middle of bridge are exposed more to the interference. As they follow our algorithm they often have strong interferers from both sides.

We use MATLAB programming language to simulate the environment in which nodes are deployed. As losses both due to the combined path loss, shadowing and fading, and interference happen we have to use NACK messages (that would let know a node that it needs to retransmit a message as it was not successfully decoded). Nodes have protocol programmed, thus they know when they expect a message from which node. These messages decreased a throughput and slightly increased delays. Also, as there is a presence of random component Monte Carlo Simulations were used (100.000 of different runs), and results were averaged out to provide us with information how does the channel looks in average. Unfortunately, the distribution of losses has tails and small negative skew. Thus, in this simulation nodes will choose between different power levels depending on how far they want the message to be received. Also, linear programming solver will provide us with position of nodes so that the power level a node uses combined with the number of messages it sends on average evens out its power consumption with the power consumption of other nodes deployed in the network.

The distribution of nodes is the most uneven in the case of three-hop direct-receiving. In the worst case, the wanted signal arrives from the node that is three-hops away, whereas the interference comes from a node only one-hop away, from the opposite side of the line.

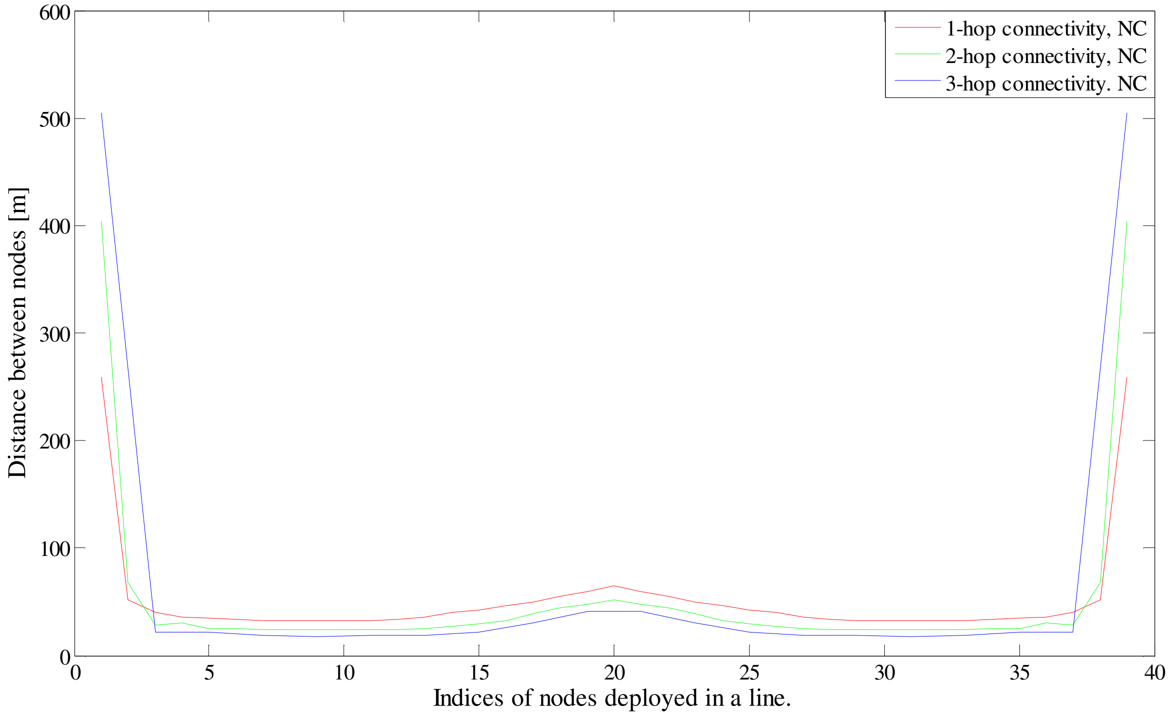


Fig. 5.3. The distribution of distances between adjacent nodes for different connectivity levels. The interference originating from limited number of neighbouring nodes is assumed. Network coding technique is applied.

If network coding is not used, nodes do not have as different loads in a form of messages they should transmit/relay, as they do when network coding is applied. Therefore, the distances among adjacent nodes are less uneven then when network coding technique is deployed. The distribution of nodes is dominantly influenced and dictated by the level of interference.

We explored how the varying number of nodes in network influences the distribution of optimal distances between nodes. When we compare distances between adjacent nodes for network that consists of 20 nodes, we see they are significantly higher than in the case when network comprises of 60 nodes, both distances between end nodes and between central nodes. Similarly, network that does not use network coding has more equalized burden in terms of the number of transmissions performed by each node (excluding end sink nodes that transmit only one message). The difference in distances between

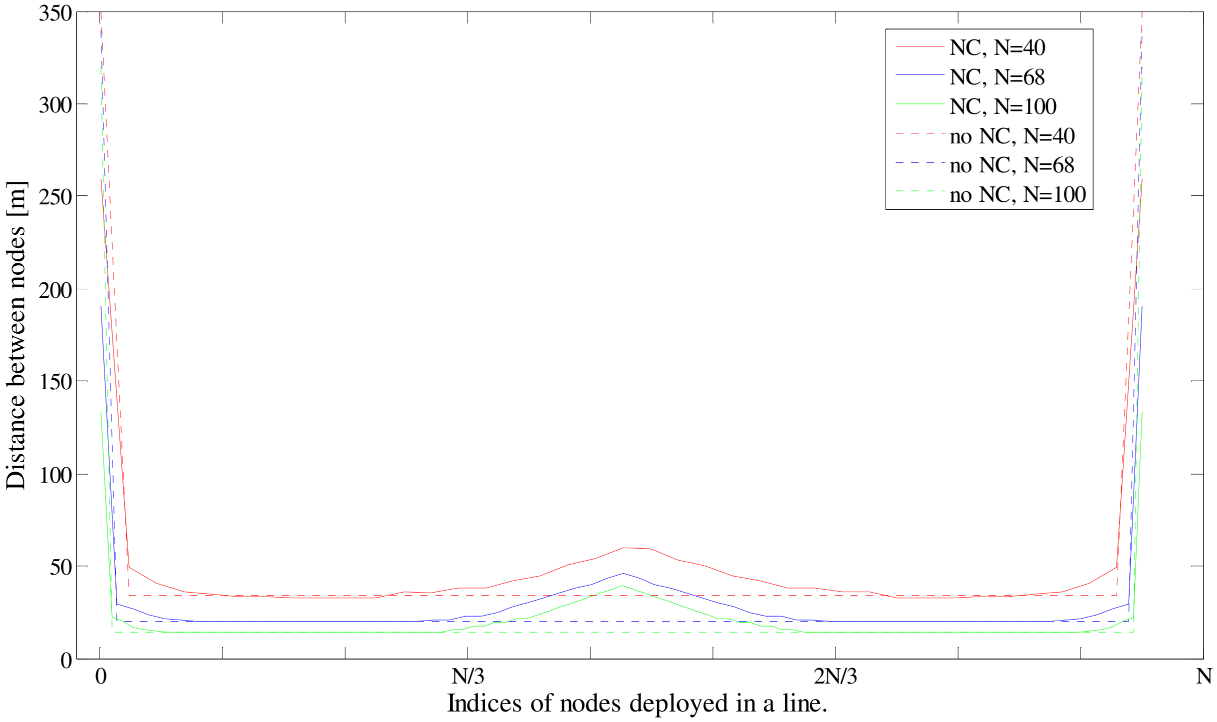


Fig. 5.4. The distribution of distances between adjacent nodes for different number of nodes covering the same distance. Comparison of network coding based algorithm and traditional data forwarding. The interference from limited number of neighbouring nodes is assumed.

adjacent nodes is smaller than in the case when network coding is applied. The interference remains major factor that determines the distances between adjacent nodes, Figure 5.4. In the previous section of this chapter we have seen that the distribution of distances is a function of target data rate *C* and available bandwidth *B*.

The effect of throughput is mild compared to the other factors. To conclude how the throughput impacts the optimal distribution of nodes, we had to severely change data rates. Figure 5.5 shows that if we target the lower throughput transmission rates will not be high and the effect of interference is under control. Therefore, for the lower data rates nodes are more evenly distributed. As available bandwidth and data rates are closely related, we will not discuss the effect of available bandwidth separately Figure 5.5.

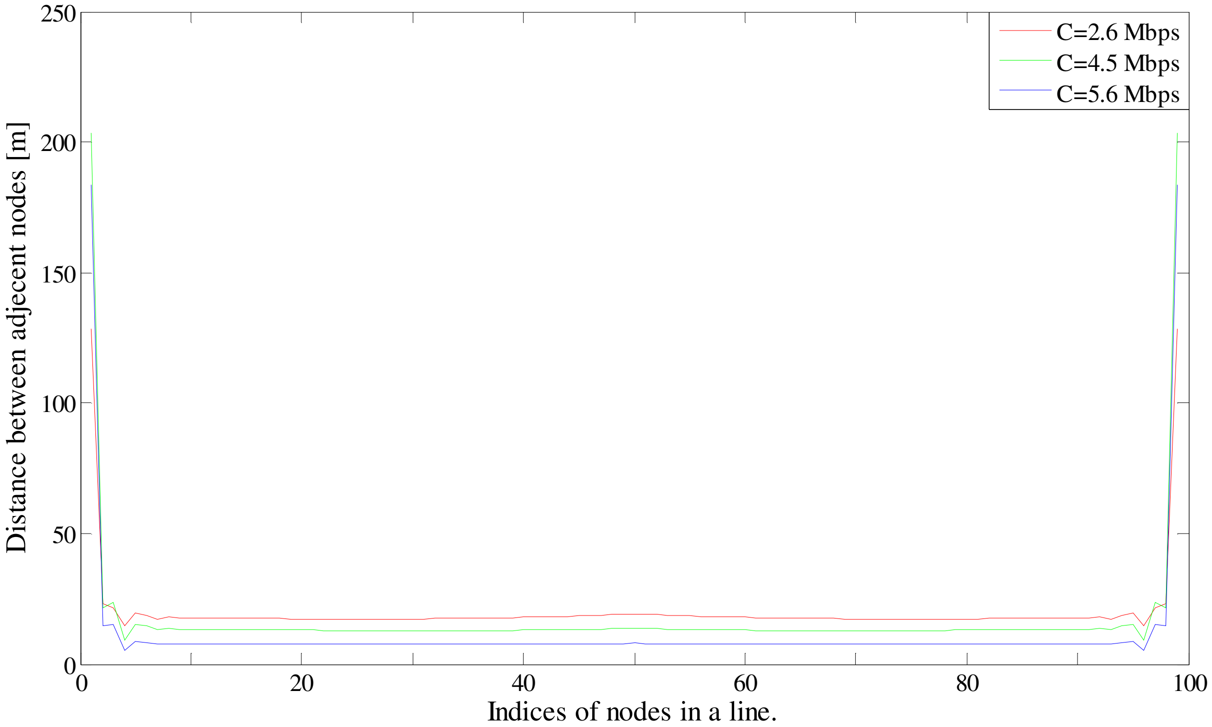


Fig. 5.5 The distribution of distances between adjacent nodes for different data rates. The interference from limited number of neighbouring nodes is assumed. Network coding technique is applied.

We assumed that initially all nodes have the same amount of energy. We showed that if network deploys our algorithm and no retransmissions are in use, all nodes consume the same amount of energy for receptions. Therefore, all nodes within the network have the same amount of energy available for transmissions. We suggested the optimisation approach that boosted the lifetime of network. Using proposed approach all nodes in the network die at the approximately same time, eliminating the bottlenecks in network. As an example we used one-hop receiving case, and tested the lifetime of network. When both sinks in the network receive all messages we consider that sensing cycle is finished. Simulations show that when sensor nodes are placed following our instructions, the lifetime of network is always significantly prolonged. For network of 20 nodes, the lifetime is extended by 25% using this approach. Figure 5.6 shows that network coding increases the network lifetime even for the small-span

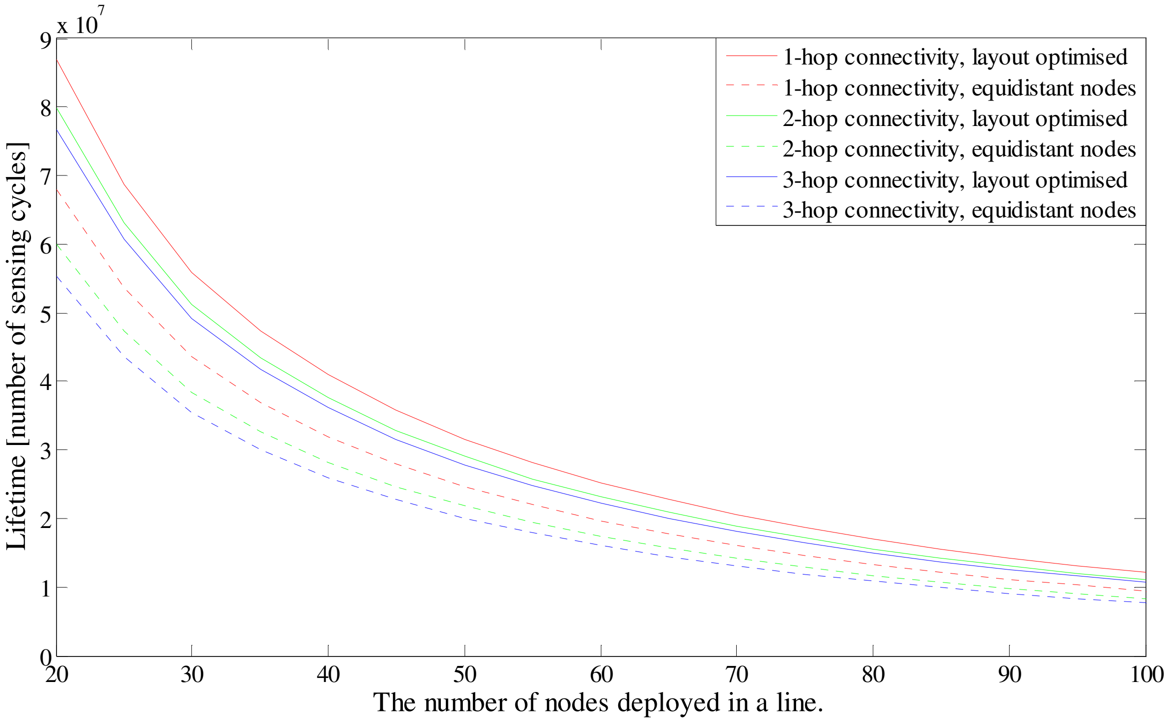


Fig. 5.6. The lifetime comparison of networks with various numbers of nodes and connectivity levels. Comparison of the equidistant nodes' layout and optimised layout of nodes.

networks. Network lifetime is longer for network of 20 nodes, than for the network of 60 nodes because each node has the message that should be relayed to sinks. Data rates are significantly lower for network of 20 nodes, Figure 5.6. It is shown that the lower degree of direct-connectivity results in the longer lifetime of network. However, the optimisation of nodes' layout is dominant factor compared to the connectivity level. The best case of not optimised layout (one-hop direct-connectivity) performed worse than the worst case of the optimal layout (three-hop connectivity). All six cases presented in Figure 5.7. used the network coding technique.

We are interested to see how the layout optimisation effects networks that use network coding technique compared to those that rely on traditional data forwarding. Such comparison is presented in Figure 5.7.

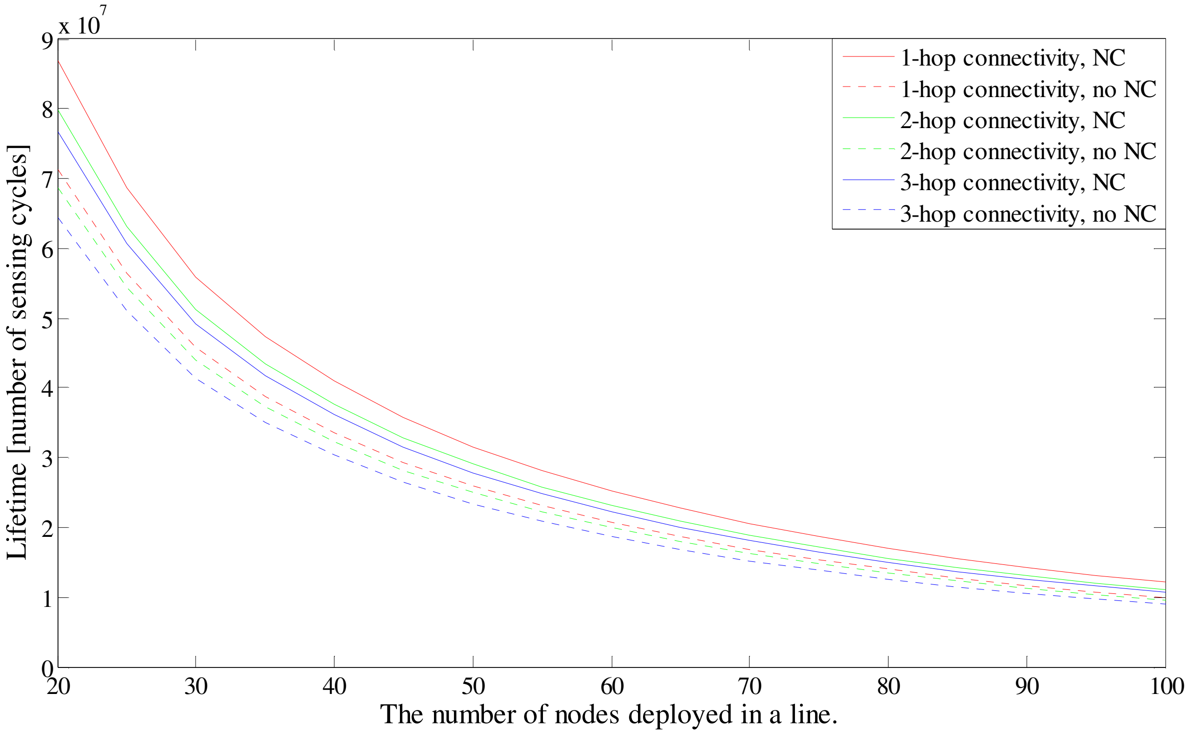


Fig. 5.7. The lifetime comparison of networks with various numbers of nodes and connectivity levels. Comparison of the optimised layout of nodes, for networks using network coding and networks using data forwarding.

We may notice that this technique is effective for all networks regardless of routing protocols that nodes use, as long as we are able to feed the model with the number of transmissions per node.

Interesting result arising from the comparison of the worst cases (the curve showing the shortest lifetime) in Figure 5.6 and Figure 5.7 is that the optimisation of the layout does more for lifetime than the network coding technique. The same picture also shows that this is specific to the three-hop direct-connectivity layout. For one-hop connectivity layout, network coding outweighs the benefits of nodes' layout optimisation.

Finally, we want to see what network coding and nodes' layout do to a network with arguably best layout and connectivity level (single-line, one-hop direct-connectivity). Figure 5.8 shows that the longest lifetime of network is achieved with the combination of network coding and optimal node placement.

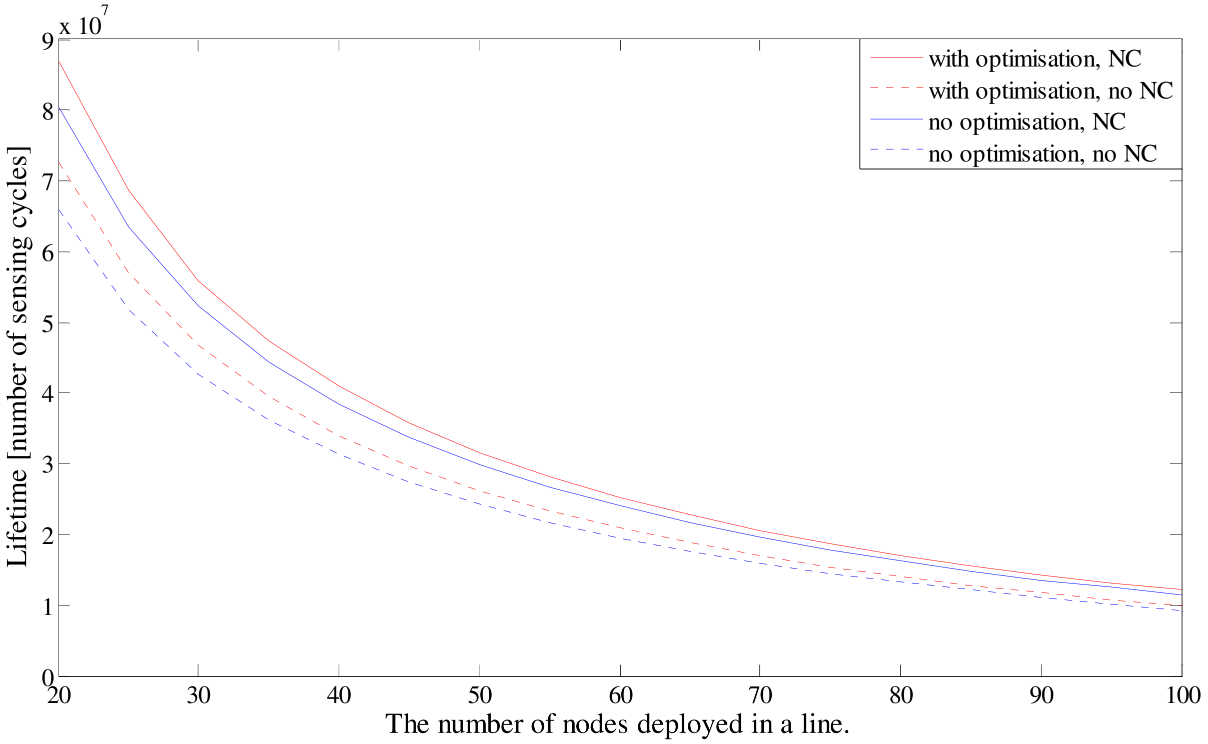


Fig. 5.8. The lifetime of a single-line, one-hop direct-connectivity network. The lifetime comparison of networks with various numbers of nodes, routing techniques and optional optimisation of nodes' layout.

5.7 Summary

In this chapter, we presented the method for nodes placement in a linear network based on network coding. We provided the guidelines for optimal nodes placement for three different levels of connectivity. The goal of this optimisation is to force all nodes to die at the same time, and to remove the bottlenecks in the network. All nodes in our network will die at the same time, as we place closer to each other nodes that are exchanging more messages.

As a constraint in optimisation problem, we use the same link capacity for all links in the network. Therefore, we eliminate the bottlenecks in the network. Network coding by itself was beneficial for network lifetime. As a result not only that overall network lives longer, but technicians do not need to go to the field often to replace the batteries every time a node runs out of battery. With the optimal sensor nodes' placement we further improved the network lifetime by 25%.

In this chapter we provided the analytical guidelines for the optimisation of nodes' layout. Then, we used MATLAB to solve the optimisation problem, applying linear programing technique. We compared result with the case when this algorithm is not applied. We showed that the longest lifetime is achieved combining network coding with the nodes' layout optimisation method.

The results presented in this chapter are published at EUSIPCO'13 [72].

6 DELAY ANALYSIS OF NETWORK CODING IN LINEAR WIRELESS SENSOR NETWORKS

6.1 Chapter Abstract

In previous chapters we talked about Structural Health Monitoring (SHM) of bridges. We observed monitoring of bridges of fixed length using equidistantly placed sensor nodes. As SHM applications produce high volume of sensing data, we use network coding technique to decrease the volume of data.

We assume that wireless medium is under combined effect of fading, shadowing, path loss and limited interference. We presented an optimisation technique that prolongs the network lifetime and decreases the maintenance cost of structural health monitoring.

In this chapter, we discuss delays for the first time. We analytically calculate packet delays obtained by the use of network coding in linear multi-hop wireless sensor networks. Moreover, we calculate the exact packet delays (from packet generation time until its delivery to sink nodes) as a function of location of the source sensor node within the linear network.

The derived packet delay distribution formulas have been verified by simulations and can serve as a benchmark for the delay performance of linear sensor networks.

We verify the derived packet delay distribution formulas via simulations. Our results can be used for the design of more efficient network coding based scheduling and routing algorithms in wireless sensor networks for structural health monitoring of linear structures.

Results presented in this chapter are published in [73].

6.2 Related Work

Some researchers have already worked on problem of delays in linear wireless sensor networks. However, they used a system model with only one sink, which is simpler and does not satisfy needs of SHM application. We use two sink nodes, one at the each end of linear structure.

Amdouni et al. focuses on a traffic-aware time slot assignment that minimises the schedule length for both three and linear topologies. A smaller schedule length decreases the network delays and reduces the energy consumption. They formulize the problem as a linear program and provide results on the optimal number of slots. They suggest a delay optimised algorithm with two heuristics that reduces energy consumption and storage capacity. However, their approach cannot be combined with the network coding technique. Also, they used only one sink node which is not useful for SHM of bridges particularly very long bridges [66].

A level based scheduling that minimises the schedule length is proposed in [64]. Firstly, a linear network is built from the initial network. Each node corresponds to a certain level in the original network. Afterwards, the schedule of the original network can be deducted using colouring of the original network. In this paper, each node has one packet to transmit and only nodes that cannot hear each other interfere. Their solution is also considering usage of only one sink node.

Gandham et al. compute the theoretical lower bound on number of slots required in linear, multi-line and tree network topologies. Their solution is close to lower bounds for linear and multi-line topologies [67]. Yu et al. jointly optimise source coding (data quantisation), routing and power control strategy at each sensor in order to obtain an efficient solution for the optimisation of overall network [68].

6.3 Motivation

Very common problem in wireless communications is *hidden terminal problem.* When a number of nodes are outside of each other's communication range, they might send data at the same time to the same receiver, causing the data loss and the retransmission. These packets therefore collide at the receiver. Hidden terminal problem can be solved using RTS/CTS schemes [74], however buttery-run WSNs cannot afford this kind of overhead. However, data collisions do not always result in data loss. If the power level at the receiver of one packet is stronger by a certain margin than power level of all other incoming signals, the receiver will successfully decode the strongest signal. This phenomenon is called capture effect [49].

Our proposed protocol is prone to collisions as nodes are equidistant, and use the same transmission power levels. In the ideal conditions, the power level of two same signals received from two equally distant nodes in the network would be the same at the receiver. However, this is where the fading and shadowing boost the probability of successful decoding. Two signals arriving at the same time to the receiver rarely have the same power level as they undergo different channel characteristics. In our work, we take into account the capture effect when signals originating from different nodes arrive to the same receiving node simultaneously, with different power levels. If the difference in their power levels is larger than certain threshold, the stronger signal will still be decoded at the receiver in spite of the collision. Otherwise, both of signals are lost, and both messages need to be retransmitted. We discuss the probability of capture in a wireless channel that arises not only because of different distances from the receiving node (near-far effect), but also because of the fluctuations in signal level due to the random nature of shadowing and fading. We will try to achieve the lowest possible delays, as the end-to-end delay in network is one of the parameters that determine the quality of structural health monitoring systems.

6.4 Scenario

6.4.1 System Model

Wireless sensor nodes are equidistantly placed along one side of bridge, forming a line. We assume that a perfect medium access layer is in use so that no data losses due to the interference happen. We assumed that the channel is not lossless.

The expressions we obtained for the number of transmissions per node are completely independent of positions of sensor nodes in such environment. We have noticed that some nodes die faster than others as a consequence of the equal distances among nodes coupled with the uneven data loads.

In this chapter, we will explore the impact of interference on performance of a network. We will assume that only nodes unable to hear each other can transmit at the same time slot. We observe the worst possible case in which all nodes that cannot hear each other transmit at the same time.

We analyse which nodes can transmit at the same time not causing the interference that would lead to the loss of other nodes' packets at designated receivers. Having this in mind, we derive the set of equations describing end-to-end delays and individual packet delays.

As an underlying assumption we consider that nodes are equidistant for the purpose of simplicity. If nodes were not equidistant, we would have to choose the minimum slot reuse factor that secures no losses due to interference in a network. In that case, slot reuse factor would be higher than the one mentioned in this chapter, and this solution would be sub-optimal resulting in higher delays.

6.4.2 Channel Model

We mount wireless sensor nodes along linear structure, e.g. a bridge. To increase the reliability of communication, we put sink nodes at the both ends of bridge. The ends of bridge are the most convenient places for the collection of data since they can be further forwarded via Internet without the interruption of bridge function.

If a packet is lost due to the temporarily problems on link and it cannot reach the designated sink node, it can likely be collected at the other sink node. The cost of using two sink nodes is low compared to the bridge failure.

Our proposed scheme can be applied to any linear network that requires high reliability of data transfer and delivery of non-distorted data. We consider three different connectivity subcases. In the first observed case, a node transmits with the lowest power level that enables only immediate neighbours of a transmitting node to decode a packet. This represents one-hop direct-connectivity case or one-hop receiving case as we will refer to it now on. For instance, if the node with the ID 4 transmits a packet, only nodes with the IDs 3 and 5 will be able to decode it, Figure 3.2a. When the same node transmits a packet with the next higher benchmark power level, enabling nodes with the IDs 2 and 6 to decode its message the two-hop receiving mode is achieved. Respectively, in the case of three-hop receiving, when node with the ID 4 transmits with the highest benchmark power level nodes with the IDs 1 and 7 will also decode the packet. The higher degree of receiving, the more opportunities for network coding.

We briefly describe the algorithm as we propose a scheduling solution. We use one-hop receiving case to explain the algorithm. Each node adjusts its power level to the minimum required for one-hop receiving.

Previously, we assumed perfect scheduling of transmissions at different nodes, providing a separate time slot for each transmission and, eliminating losses on MAC layer. In this paper, we consider the impact of interference.

A number of nodes will be able to transmit at a given time slot, provided that the interference level they create to each other at designated receivers is low enough to secure the capture effect of a useful signal can take place. Designated signal will be successfully decoded if the Signal to Interference and Noise

Ratio (*SINR*) at the destination is higher than capture threshold determined by the type of deployed sensor nodes. With the technology whose deployment by default we assume in this thesis, the required *SINR* is set to be 6 [dB]. For the simplicity, we consider equidistant nodes' layout ($d_i=dj$, $i \neq j$). As the path loss factor for observed environment is considered to be 3 [45], the power level decreases with the cube of distance between a transmitter and a receiver. We consider only the strongest interferers relevant.

Initially, each node uses its sensing unit and creates a message from the sensed data. Consequently, each node knows its own message, as shown in the top row in the Figure 6.1. Following that, each node broadcasts its message to the immediate neighbours. In the first time slot, nodes with the IDs 1 and 4 broadcast their messages. In the second and third time slots node with the IDs (2, 5), and (3, 6), broadcast their messages respectively.

At this point, each node knows its own message and messages created by its immediate neighbours, as shown at the second row of the Figure 6.1. Once a node receives a message (e.g., a message originated at a node *j*), it fills in the corresponding rectangle associated to that node. For instance, after the first two rounds, node with the ID 3 receives a message from its own sensors (message 3), and its two immediate neighbours, nodes with the IDs 2 and 4, as shown in the second row of rectangles in Figure 6.1. Notice that for instance the node with ID 2, transmits its message in the slot 2, whereas in the remaining two slots of this round it receives packets from its immediate neighbours (in the first time slot, it receives a packet from the node with the ID 1, and in the third time slot it receives a packet from a node 3). The scheduling should avoid collisions at the transceiver unit.

As it continues, each node XORs messages received by its immediate neighbours in the previous round, and broadcasts the coded packet in an appropriate time slot. For example, a node with the ID 2 XORs messages 1 and 3, and broadcasts the coded XORed message to nodes with IDs 1 and 3, in the fourth time slot. Node with the ID 3 performs XOR operation over its own message 3 and received coded packet. Therefore, it decodes the message from node with the ID 1. Likewise, the node with the ID 1 receives the coded packet, and performs XOR operation with its own message 1 – as a result, it can decode message from node with the ID 3. Along with the node 2, node 5 is allowed to use this slot for transmission of its messages. In the next round, nodes receive and decode messages from its second-hop neighbours, as shown in the third row of rectangles.

In the each round of this cycle, node XORs the received coded packet with the corresponding packet from its buffer to decode a new packet. Process continues in the same manner. In the last round of broadcasting no node receives a (XORed) message.

The proposed algorithm operates similarly in the case of two-hop receiving and three-hop receiving as explained in [70]. After we exploit all possibilities for the network coding, the remaining messages are forwarded using traditional routing.

Figure 6.2 presents the required information fields in every transmitted packet. Each packet has information about its node of origin, the IDs of nodes whose information is contained in DATA field, and the IDs of last two received packets at the node of origin.

Each node keeps two separate lists with packet IDs, so it can keep track which packets it should send to which neighbour. If a node received two packets in the past $p+1$ time-slots, it will participate in the exchange of coded packets in the next round. In that case, the node will XOR last two packets from its buffer (both are received within last $(p+1)T_s$).

As it learns that both of its neighbours received a XORed packet, it deletes both of packets used to obtain the received XORed packet, from its buffer. Each node that received only one packet in the previous $(p+1)$ slots adds the ID of new packet to the ‘to send list’ –that contains IDs of packets that will be sent to the other neighbour using traditional forwarding.

Once coded packets exchange finishes, nodes rely the remaining packets using conventional routing. Nodes add packets to the ‘to send list’ as they receive the new ones, and remove them from the list as they successfully forward them.

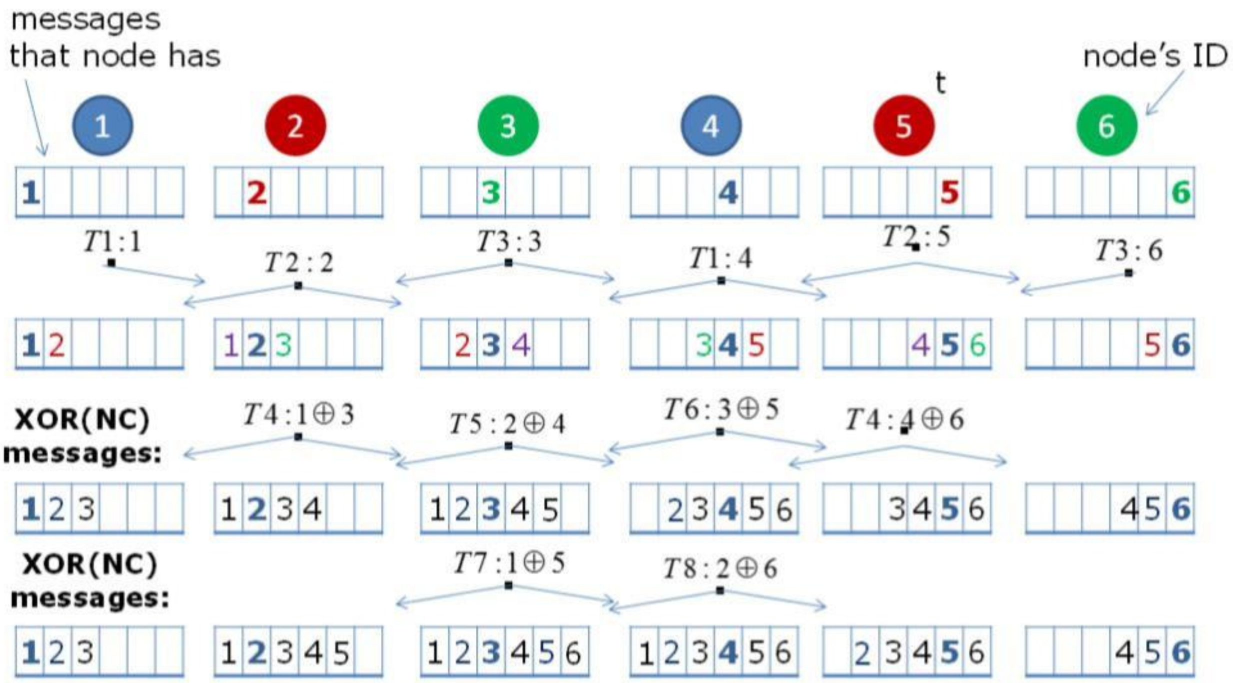
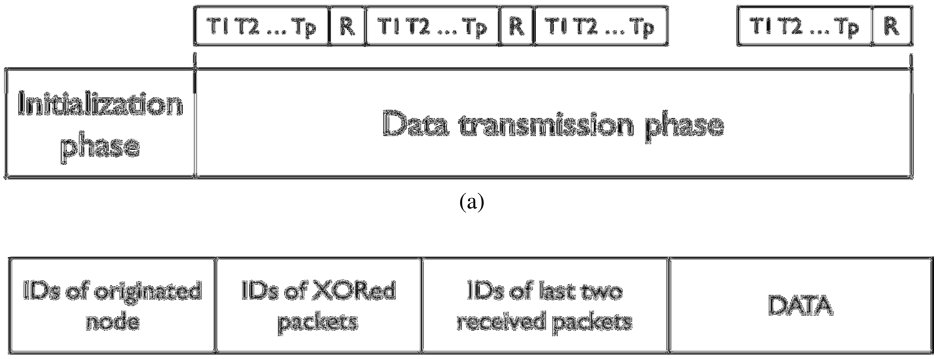


Fig. 6.1. Network coding based algorithm with the time-slot reuse and one-hop receiving.



(a)

(b)

Figure 6.2. Packet at MAC level. a) Phases of scheduling algorithm. b) Packet content.

6.5 Interference Levels at Nodes

If we apply this network coding algorithm and use the proposed scheme for nodes' optimised placement, some nodes will be exposed to the more interference than others. That is particularly true for the nodes in the middle as they have stronger interferers than other nodes. Usually two strongest interferers are at the distance $p * d_j$, while nodes at the end have only one interferer at that distance. In general, the middle nodes and the end nodes have the same number of interferers, but the average distance of an interferer is smaller for the middle nodes. As a result, the most critical part of the network, if nodes are equidistant will be in the middle, Figure 6.3. We can observe this phenomenon through a *SINR* level at nodes, Figure 6.4a. The middle nodes have the lower *SINRs* if all nodes in the network have the same transmission powers and are equidistant, Figure 6.4a.This kind of network is subject to the outage. If we set the probability of outage to 0.01%, the possible outage capacity at different nodes is different. Nodes in the middle have the lowest capacity levels, which is fine as the have the least messages to transmit. Nodes placed towards the ends of the bridge will have the highest achievable capacities, Figure 6.4b.

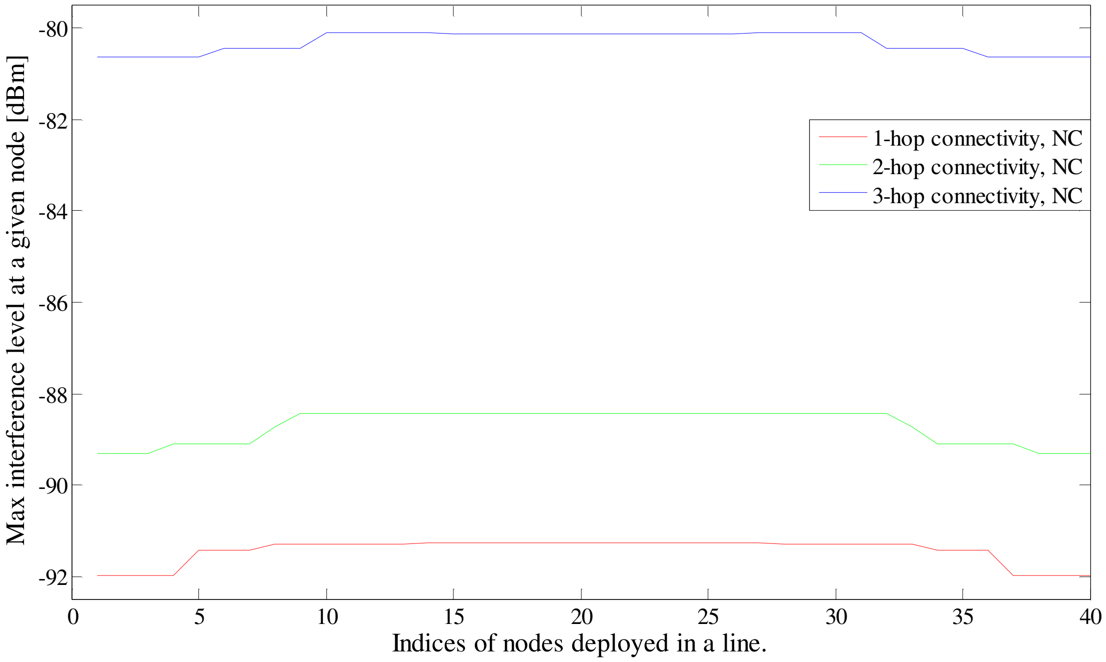
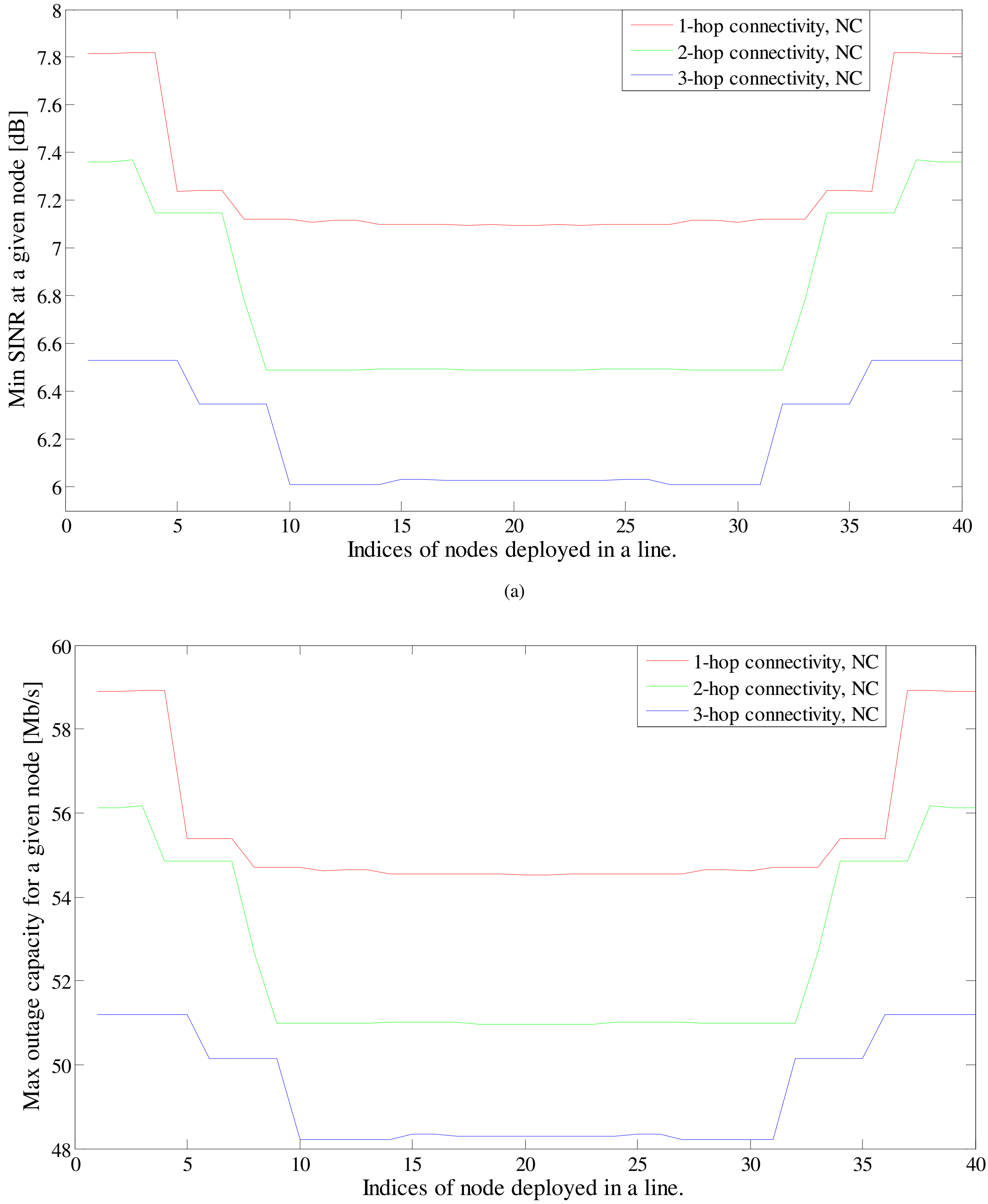


Figure 6.3. Maximum possible interference level at a node.



(a)

(b)

Fig. 6.4. a) Minimum possible SINR level at a node. b) Maximum outage capacity for a given node.

6.6 Mathematical Analysis

In this section we analytically calculate the exact packet delay (from the packet generation time to the time it is delivered to sink nodes) as a function of the location of source sensor node within the linear network. Let us define *p* as the time-slot reuse factor (e.g. if *p=3* nodes with IDs 1, 4, 7 can transmit their messages at the same time). *N* is the number of sensor nodes deployed in a linear network, and j as the index of the observed node. The hop factor *h* takes values {1, 2, 3} for one-hop, two-hop and three-hop receiving, respectively.

Data transmission phase consists of $k + l + 1$ transmission slots of duration T_s. The factor *k* represents the total number of rounds during the network coding phase, *l* represents the total number of rounds in the traditional forwarding phase, and 1 comes from the broadcasting round. We define the following functions/notations to simplify the presentation of our equations for the delay analysis:

$$A_i = i \; mod \; 2, B_i = i \; mod \; 4 \text{ and } C_i = i \; mod \; 6, \tag{6.1}$$

$$\{E, F, H, I, J, G\}(i) = \left\{\left\lceil\frac{i}{2}\right\rceil, \left\lfloor\frac{i}{2}\right\rfloor, \left\lceil\frac{i}{4}\right\rceil, \left\lfloor\frac{i}{4}\right\rfloor, \left\lceil\frac{i}{6}\right\rceil, \left\lfloor\frac{i}{6}\right\rfloor\right\}. \tag{6.2}$$

We define two factors: *l* and *k* as follows:

$$l = \left\lfloor\frac{2p-((N+h \; mod \; 2) \; mod \; 2h)}{2h}\right\rfloor, \tag{6.3}$$

$$k = \left\lfloor\frac{p-(N \; mod \; 2h)}{2h}\right\rfloor. \tag{6.4}$$

We define the factor n_j as the total number of rounds that are left to be completed from the moment a packet *j* reaches the sink to the moment when the last packet reaches the sink. That is period needed that one packet from each node (that has a packet ready for transmission) to be delivered to sink nodes. For the case of one-hop receiving:

$$n_j = E(j + A_{N+1}) - 1 \, . \tag{6.5}$$

For two-hop receiving case

$$n_j = A_{j+1} H\left(j - 2A_N A_{\frac{N-3}{2}} + A_{N+1} A_{\frac{N-2}{2}}\right) + A_j I(j + A_{N+1} A_{\frac{N-2}{2}} + A_N (2A_{\frac{N-1}{2}} + 3A_{\frac{N+2}{2}})\ . \qquad (6.6)$$

Whereas, in the case of three-hop receiving:

$$n_j = n_{e,j} + n_{o,j}, \qquad (6.7)$$

$$n_{e,j} = A_{j+A_N} J(j + A_N - C_{N+2}) + (-1)^{N+1} A_{j+1+A_N} A_N I(C_N), \qquad (6.8)$$

$$n_{o,j} = A_{j+1+A_N} \{G(j + 6 + A_N - C_N) - I(j + A_{N+2} - C_N)\}\ . \qquad (6.9)$$

We need to compute the number of messages transmitted in the each round, and the total number of rounds. Manipulating similarities in movements of those two values, we will come up with the number of transmissions. Observing those two values combined with the changes due to the frequency reuse factor, we will come up with the equations that describe delays. The total number of transmission slots necessary for the transfer of all messages to sinks is given by the following formula:

$$N_s = (k + A_N)(k + 1) - p(k - F(N) + (l + A_{N+1})(l + 1) + 2pF(N - 3 - l). \qquad (6.10)$$

For two-hop receiving case, the number of transmissions slots needed for the transmissions of all messages is:

$$N_s = N_{s,e} + N_{s,o}, \qquad (6.11)$$

$$N_{s,e} = B_N(k + 1) + 2k(k + 1) + p(I(h) - k) + A_{N+1} A_{\frac{N+2,}{2}} \qquad (6.12)$$

$$N_{s,o} = B_N(l + 1) + 2l(l + 1) + 2p(I(h) - (l + 1)). \qquad (6.13)$$

The total number of slots for the three-hop receiving case is given by (6.11), where components are:

$$N_{s,e} = (3k + C_N)(k + 1) + (3l + C_{N+1})(l + 1), \qquad (6.14)$$

$$N_{s,o} = p(3J(N) - 2l - 5) + I(C_{N-1})(2p + 1 + 2pI(C_{N-2}). \qquad (6.15)$$

If network coding is not applied the total number of designated transmission slots must be larger, since the total number of transmissions is higher. Basically, for each transmission of a network coding packet we would have to perform two separate transmissions. As, the number of packets transmitted via network coding doubles, the total number of transmissions slots in the case of one-hop receiving is:

$$N_{s,no_{NC}} = N_s + (k + A_N)(k + 1) - pk + pF(N). \tag{6.16}$$

As the two-hop receiving case exploits the message overhearing, the total number of transmissions will be lower compared to the total number of transmissions in the case of one-hop receiving. Therefore, the number of designated transmission slots in this case will be lower than for one-hop receiving:

$$N_{s,no_{NC}} = N_s + (B_N + 2k)(k + 1) + p(I(N) - k). \tag{6.17}$$

Similarly, in the three-hop receiving case the number of necessary transmissions decreases compared to the two-hop receiving, and therefore this will be the case that requires the least transmission slots:

$$N_{s,no_{NC}} = N_s + (C_N + 3k)(k + 1) + p(J(N) - k - 1). \tag{6.18}$$

We compute the delay of the packet j, T_j, and the end to end delay (when last packet reaches both sinks):

$$T_j = T_s(N_s - N_m) \text{ and } T_{delay} = T_s N_s. \tag{6.19}$$

N_s is the total number of transmission slots, and N_m determines in how many slots before the last transmission slot a packet j will reach the further sink. For the one-hop receiving and $n \leq l$:

$$N_m = n(n - 1) + A_j, \tag{6.20}$$

otherwise,

$$N_m = l(l + 1) + 2p(n - l) + A_j. \tag{6.21}$$

Similarly, in the case of two-hop receiving and $n \leq l$:

$$N_m = n(B_N + 2n - 2) + A_{N+1}A_{\underline{N+2}\atop 2} + \sum_{i=1}^{5} N_i \tag{6.22}$$

Otherwise,

$$N_m = (l+1)(B_N + 2l) + A_{N+1}A_{\underline{N+2}\atop 2} + \sum_{i=1}^{5} N_i, \tag{6.23}$$

$$N_1 = A_j A_{N+1}\left(\mathrm{F(j)}A_{\underline{N+2}\atop 2} + \mathrm{F}\big(B_{j+2}\big)A_{\underline{N}\atop 2}\right), \tag{6.24}$$

$$N_2 = A_j A_N \mathrm{F}(B_j)\left(2(A_{\underline{N+2}\atop 2}A_j + 3A_{\underline{N-1}\atop 2}A_j\right), \tag{6.25}$$

$$N_3 = A_{j+1}F\big(B_{j+2}\big)\left(3A_{\underline{N+2}\atop 2} + 2A_{\underline{N}\atop 2}\right) + A_j\mathrm{F}(B_{j+2})\left(3A_{\underline{N+1}\atop 2} + 2A_{\underline{N-1}\atop 2}\right), \tag{6.26}$$

$$N_4 = A_{j+1}A_{N+1}\mathrm{F}\big(B_j\big)\left(2A_{\underline{N+2}\atop 2} + 3A_{\underline{N}\atop 2}\right), \tag{6.27}$$

$$N_{5=}A_{j+1}A_N\left(F\big(B_j\big)A_{\underline{N+1}\atop 2} + 3F(B_{j+2})A_{\underline{N-1}\atop 2}\right). \tag{6.28}$$

For the three-hop receiving and $n \leq l$:

$$N_m = (\mathrm{n}-1)(3\mathrm{n} + C_{N+1}) + \mathrm{I}(C_{N-1}) + \sum_{i=1}^{3} N_i. \tag{6.29}$$

Otherwise,

$$N_m = (l+1)(3l + C_{N+1}) + I(C_{N-1}) + 2p(n-l-1) + \sum_{i=1}^{3} N_i, \tag{6.30}$$

$$N_1 = A_{j+1+A_N}\big[5G\big(2C_{j-A_N+4-C_N}\big) + 3G(2C_{j-A_N+C_{2N}})\big], \tag{6.31}$$

$$N_2 = A_{j+A_N}\left(G\left(2C_{j-1+A_{N+C_{2N}}}\right) + 4G\big(2C_{j+1-A_N+C_{2N}}\big)\right), \tag{6.32}$$

$$N_3 = 2A_{j+A_N}G(2C_{j+3-A_N-C_N}). \tag{6.33}$$

Should we decide to add the slot *R*, as an additional protection against the packet loss, all formulas will still stand, but:

$$N_s' = N_s + k + l + 1, \text{ and } N_m' = N_m + n \tag{6.34}$$

6.7 Simulation Testing

We assume that we monitor a bridge of fixed length (2[km]). To obtain the numerical results, we assume a linear network that consists of 100 nodes. All nodes generate their own data and at the same time operate as relays for the data generated by the other nodes in the network. They relay these data towards the sink nodes located at the two edges of the linear topology.

Figure 6.5 depicts the maximum delay in the network as a function of the time-slot reuse factor for the different levels of connectivity when network coding (or simple relaying) is used for the data routing. The maximum delay is defined as the time required for all packets generated in the network to be received by both sink nodes, assuming that all nodes generate a single packet.

It is shown that the maximum delay decreases for the smaller time-slot reuse factor. Thus, if possible a small slot reuse factor should be chosen for delay critical applications.

Figure 6.5 quantifies the delays for the network under consideration. The combination of the three-hop direct-connectivity with the low time-slot reuse factor and network coding results in the lowest delays.

For the large-span networks, the delay in the three-hop direct-connectivity case and small value of factor p can be up to 60% lower compared to the one-hop receiving case. However, smaller time-slot reuse factor p increases the interference in the network, as more simultaneous transmissions from sensor nodes located closer to each other are allowed in the network. This will result in occasional packet collisions, and consequently in additional delays if packet retransmissions are considered or packet losses if erroneous packets are dropped.

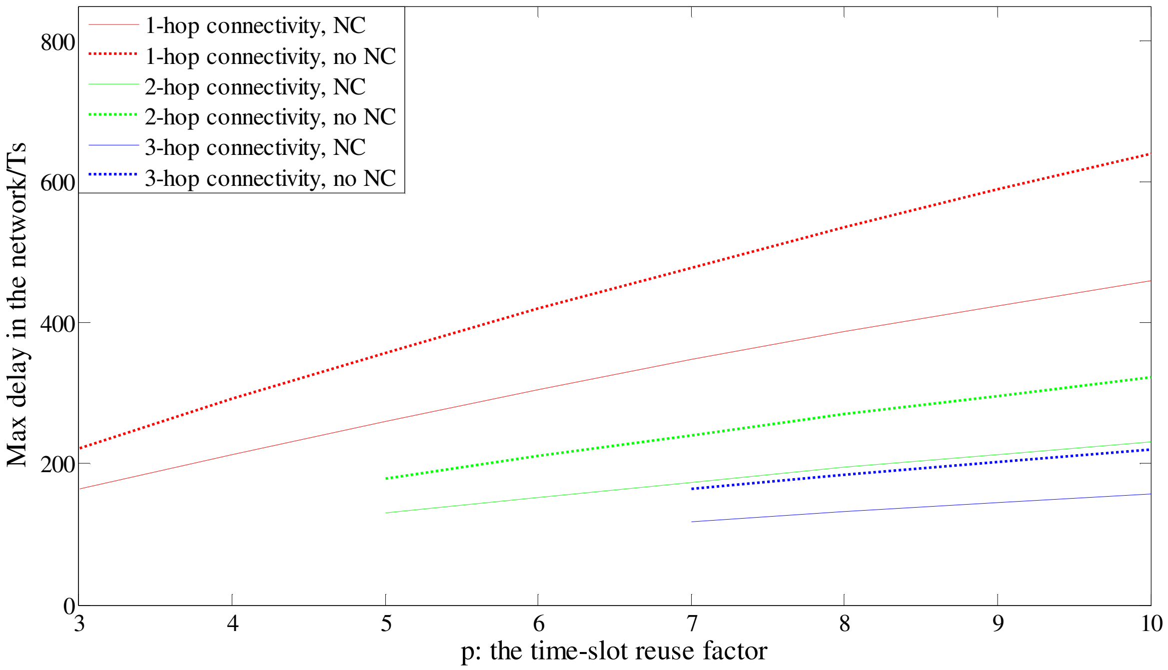


Fig. 6.5. The effect of time-slot reuse factor on network delay.

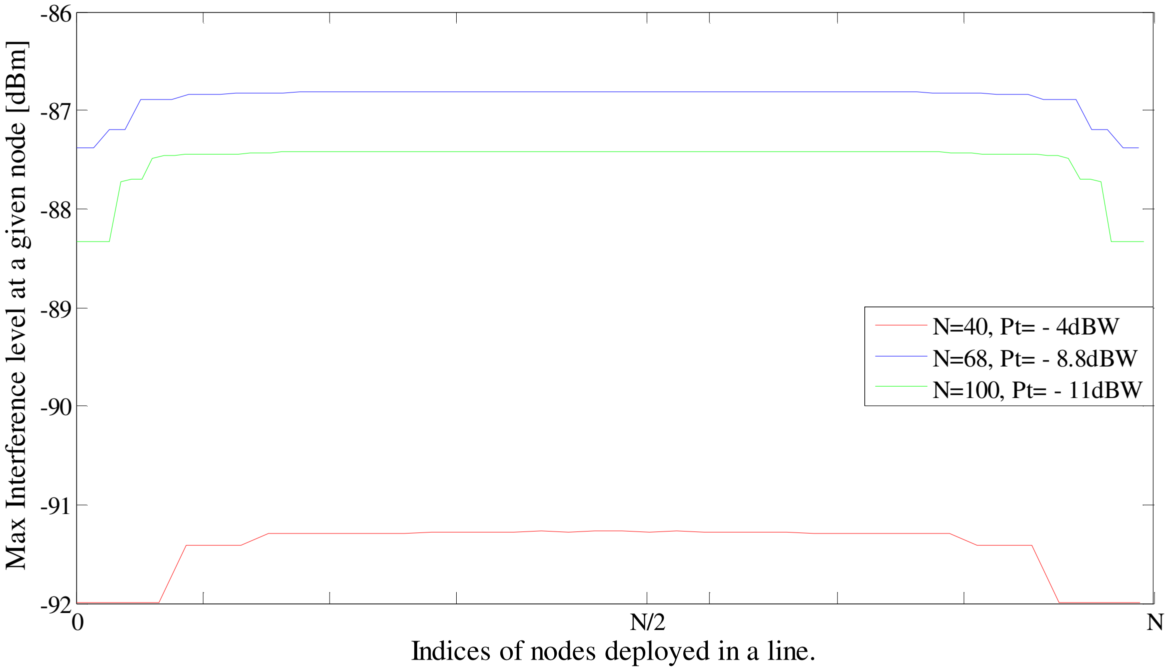
Therefore it is very important to choose the right value of the slot reuse factor *p* depending on the network and application requirements. The value of *p* should be the lowest one that does not create the strong enough interference that would cause losses to the other nodes. The minimum slot reuse factor *p* will be the one that assures:

$$\min \quad SINR_i = \frac{P_t}{P_I} \geq SINR_{capture} \tag{6.34}$$

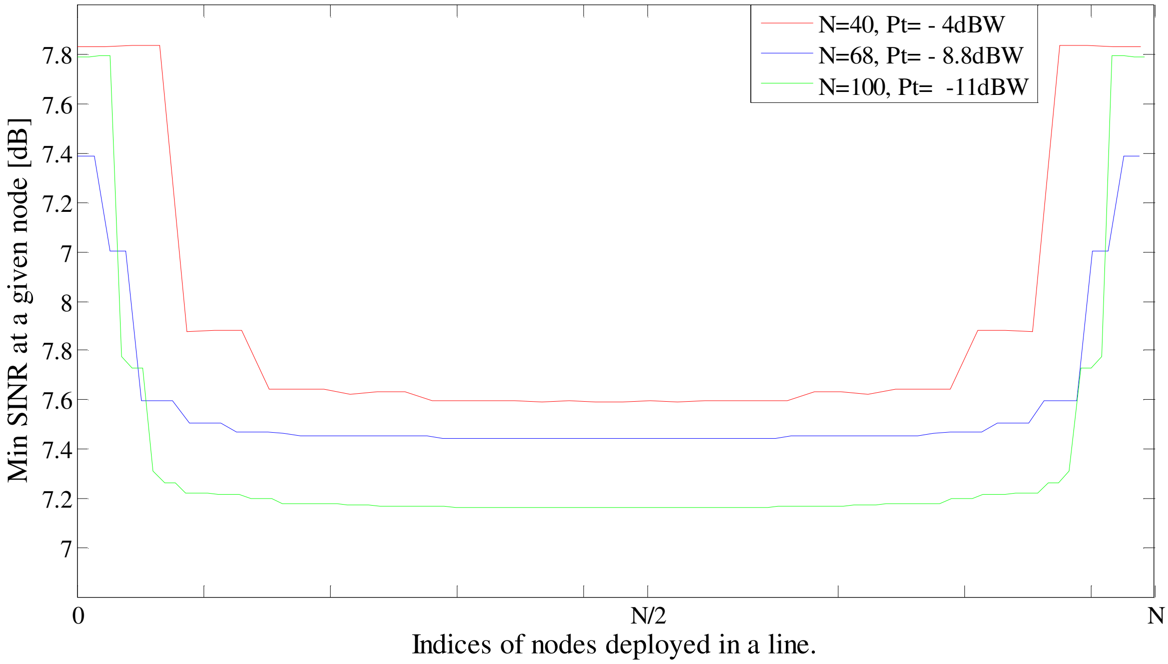
Where:

$$P_I = \sum_{j=0}^{i-1-k} P_{(h+pj)d} + \sum_{j=0}^{N-i-1-s} P_{(p-h+pj)d} \tag{6.35}$$

For interference coming from the right hand side $k = h$ and $s = p - h$, whereas for the interference coming from the left hand side $s = h$ and $k = p - h$. P_{ld} is a received power level at distance *l*d* from a transmitting node. Therefore, we choose the minimum slot reuse factor that guarantees no losses due to the interference. The received *SINR* as a function of the sensor location in the linear topology and one-hop receiving case is shown in the Figure 6.6.



(a)



(b)

Fig. 6.6. One-hop receiving. a) Maximum possible interference level at a node. b) Minimum possible SINR level at a node.

We assume that the channel state is the same, as described in the Chapter 4. We observe a 2[km] long bridge. We target *SINR* of 7 [dB], and we assume *p=3* for a single-line, one-hop receiving case.

We test the performance of a network consisting of 40, 68 and 100 nodes. Simulations show that in order to achieve 99.99% probability of successful reception, we have to use respectively -4 [dBW], -8.8 [dBW] and -11 [dBW] transmission powers. In all three cases we achieved at least the target SINR. Thus, time-slot reuse factor of 3 secures a successful operation of our proposed algorithm.

Similarly, we show that for the chosen power levels, and two-hop direct-connectivity case with p=5, and three-hop direct-connectivity case with *p=7* algorithm also operates in a way that controlled interference does not result in the packet loss at any occasion. We allow the controlled interference in the network in order to decrease the packet delays, as shown in the Figure 6.7.

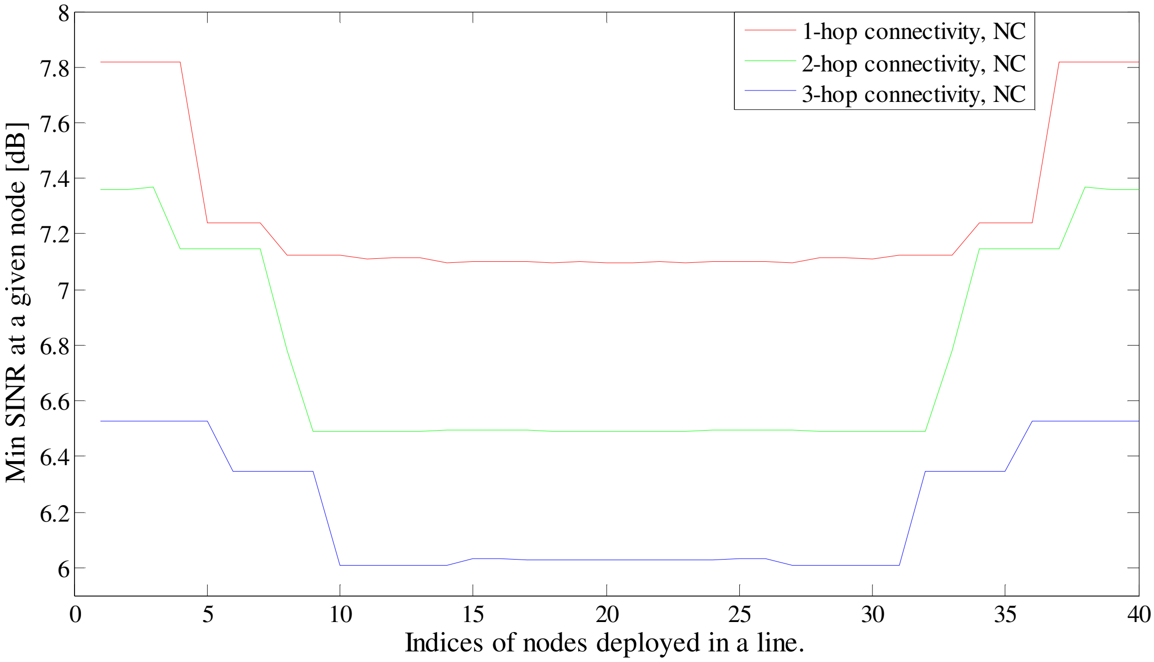
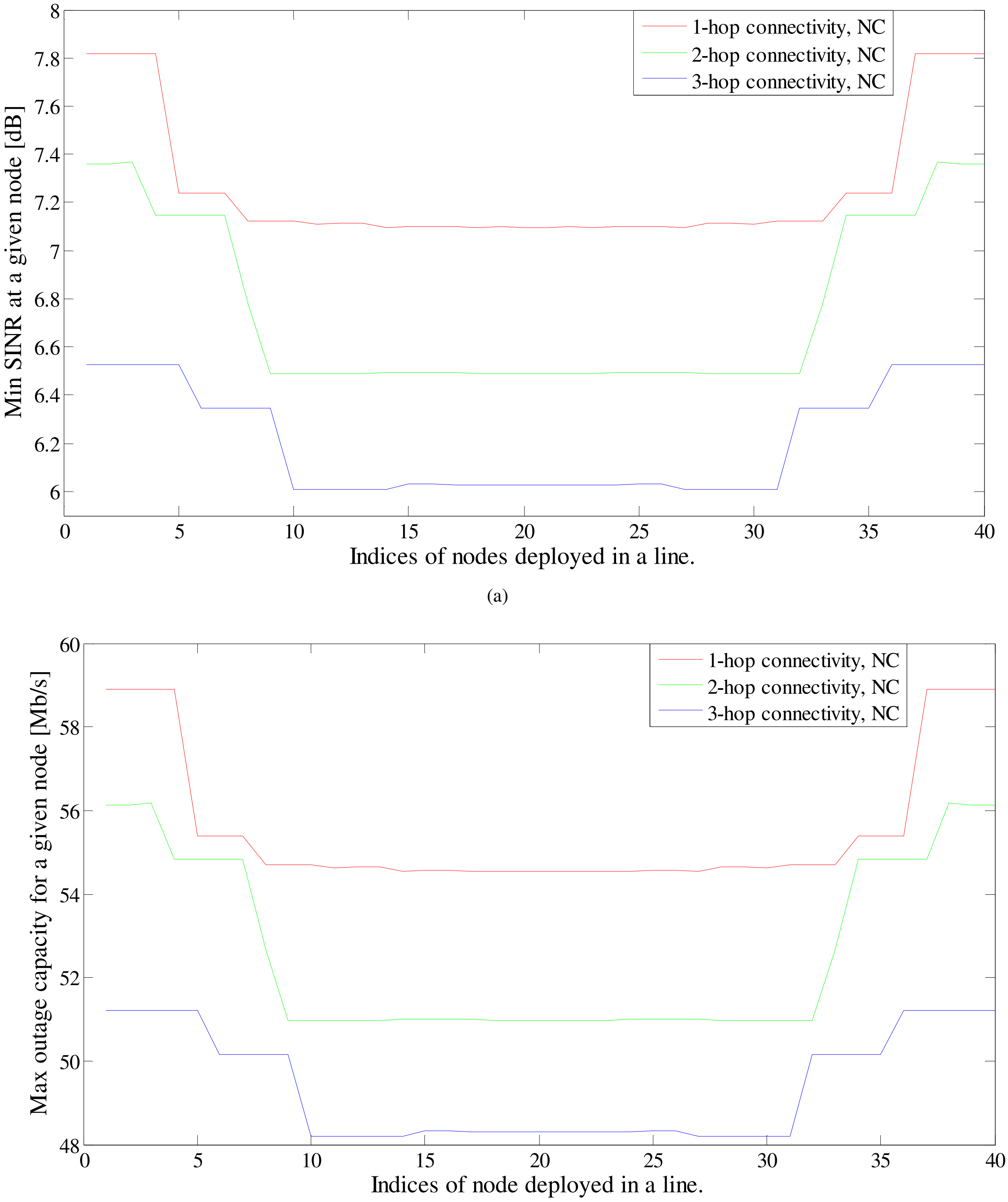


Fig. 6.7. Maximum possible interference level at a node. for a network with 40 nodes and various connectivity levels.



(a)

(b)

Fig. 6.8. A network with 40 nodes and various connectivity levels. a) Minimum possible SINR at a node. b) Capacity per node.

In the Figure 6.9, the distribution of maximum delay of packets coming from a given sensor node (x-axis) is presented. Maximum delay is the time that elapses from the packet generation at a given sensor, until this packet is received by both sink nodes. Maximum delay is presented as a function of the nodes' location within the linear network.

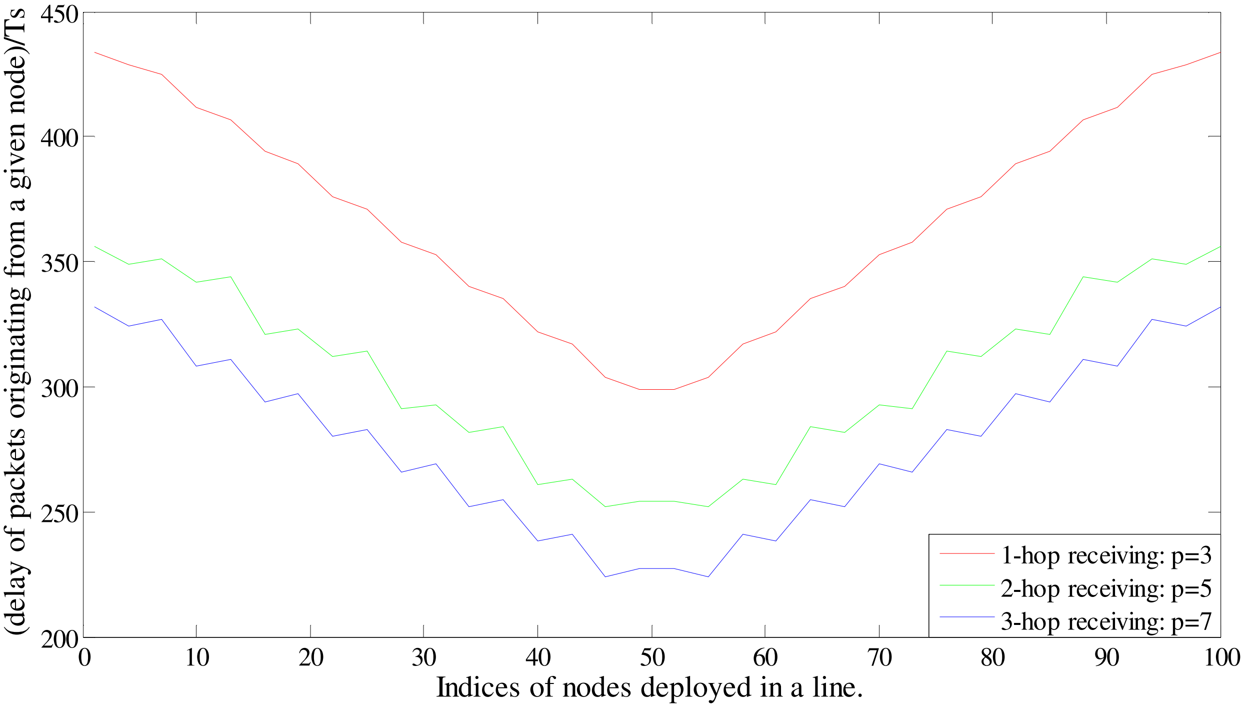


Fig. 6.9. The delay distribution of a packet originating from a given node (x-axis) in a network with the linear topology.

It is interesting to observe that messages from nodes located closer to the edges of the network suffer higher maximum delays. This is due to the fact that their packets need to be received by both sink nodes, and although the first sink will receive the packet immediately, the second sink is located on the opposite side from the originating sensor node. On the other hand, data packets generated by nodes located towards the centre of the linear topology face minimum delays. Figure 6.10 quantifies the benefits of network coding which reduces the packet delays in the network by effectively decreasing the number of packet retransmissions.

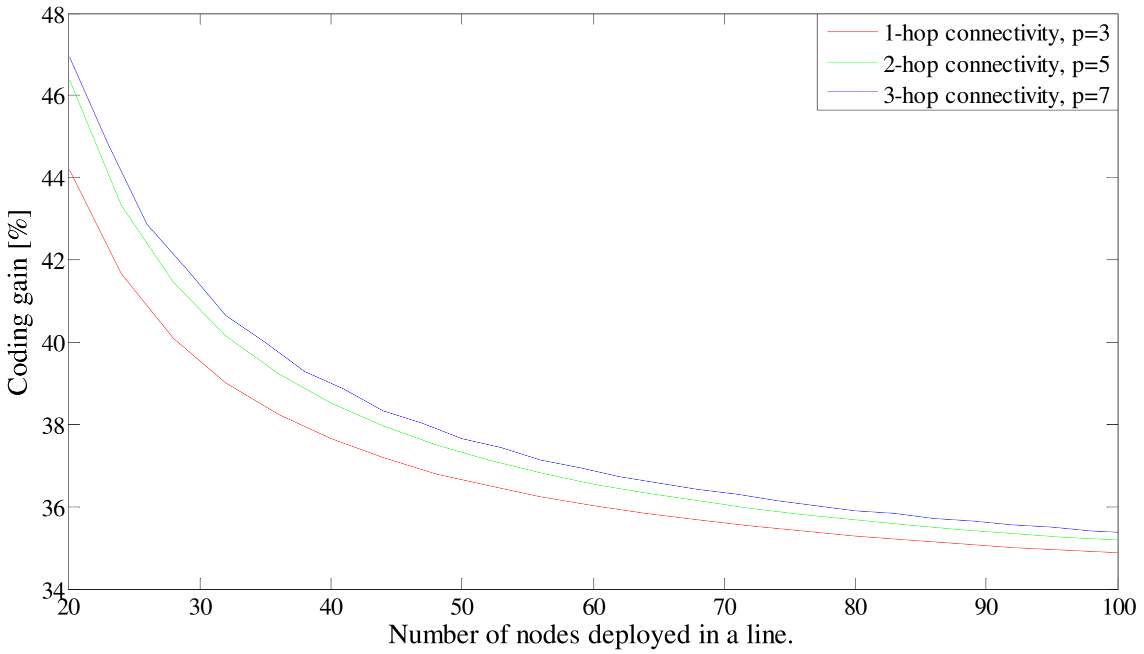


Fig. 6.10. The network coding gain in terms of delay.

It was shown in [71] that the network coding in linear topologies reduces the number of overall packet transmissions by a third. In order to better quantify the benefits of network coding compared to the simple packet relaying (i.e., the conventional routing) let us define the network coding and time slot reuse gain as:

$$Coding\ Gain = \ 100 * (\frac{MaxDelay(no\ NC\ applied\ \&\ slot\ reuse)}{MaxDelay(\ NC\ applied\ \&\ slot\ reuse)} - 1) \qquad (6.36)$$

The coding gain is demonstrated in Figure 6.10. The coding gain is higher for smaller number on nodes in the network and higher connectivity levels. For higher number of nodes in the network the coding gain converges to 36 %.

6.8 Summary

In this chapter we analytically calculated the gain in terms of packet delays, obtained by use of network coding in linear multi-hop wireless sensor network topologies. Moreover, we calculated the exact packet delays (from the packet generation time to the time it is delivered to the sink nodes) as a function of the location the source sensor nodes' within the linear network. The derived packet delay distribution formulae have been verified by simulations and provide a benchmark for the delay performance of a linear wireless sensor networks. Our results should be used for the design of more efficient network coding based scheduling and routing algorithms in wireless sensor networks for structural health monitoring of linear structures. We explained the distribution of interference in a network. We verified our results using extensive simulations. The results presented in this chapter are published at ICIWISE'13 [73].

7 ADAPTIVE NETWORK CODING BASED ROUTING ALGORITHM

7.1 Chapter Abstract

In the previous chapters we discussed various aspects of Structural Health Monitoring (SHM) of bridges. We use network coding based technique to cope with the high data loads, and decrease the data volume. We suggested a deterministic protocol, and analysed it both mathematically and through extensive simulations. We suggested the scheduling algorithm and analysed delays in a network. We suggested an optimisation method for the placement of sensor nodes in a network. The proposed technique prolonged network lifetime and decreased the cost of monitoring. We considered a wireless channel where signal is under the influence of interference, fading, shadowing and path loss.

In this chapter, we propose an adaptive version of network coding based algorithm. The proposed version is not prone to errors. In the case of packet loss, nodes do not retransmit messages as they are able to internally decide how to cope with the situation. The goal of this algorithm is to reduce the power consumption, and decrease delays whenever it is possible. This algorithm achieves delays similar to the three-hop direct-connectivity version of the deterministic algorithm, and consumes power almost like one-hop direct-connectivity version of the deterministic algorithm. In very poor channel conditions, this protocol outperforms deterministic algorithm both in terms of delays and power consumption.

We explain how nodes choose which messages will be transmitted in the next designated time slot. Then we test our algorithm in different channel conditions using extensive simulations. We show the distribution of delays and power consumption in the network. We also show end-to-end delays, total power consumption and network lifetime. Our results can be used for the design of more efficient network coding based scheduling and routing algorithms in wireless sensor networks for Structural Health Monitoring (SHM) of linear structures.

7.2 Related Work

Katti et al. worked on opportunistic wireless network coding. They suggested utilizing the opportunistic scheduling in the multiple unicast case in order to increase the throughput. Their method performs the optimal scheduling using the information about the state of neighbouring nodes. The method makes use of the overhearing capability, and based on it a node makes an optimally decodable network coding codes that can be decoded by the neighbours, which results in the improved throughput [26]. This is different from our work, as we do not use multiple unicast data streams, and we also use two sink nodes. Additionally, we do not rely on control messages to update information a node has about the state of other nodes.

One of the first network coded based protocols for WSNs was AdapCode. It achieves the high reliability due to its adaptive network coding scheme that uses the information it has about the link quality. AdapCode is not suitable for the structural health monitoring applications as it does not support multicast [27]. Additionally, it is very hard to keep the information about link state updated and accurate given the limited capability of sensor nodes.

Shwe et al. point out that AdapCode might be missing out on potential network coding opportunities as it includes only some of the neighbouring nodes in the network. They suggest an improved version of AdapCode that uses a power efficient protocol to discover all neighbours of a node. They are trying to develop a more efficient protocol as more opportunities for network coding result in the reduction of power consumption [69]. However, they still rely on link quality to choose the best messages for network coding, which is not good fit for relatively simple wireless sensor nodes.

7.3 Motivation

SHM applications produce high data rates. WSNs are not designed to transfer large data rates reliably. Therefore, there is a need for a data compression technique. We choose to use the network coding as it enables us to decrease the data volume, yet to preserve the data fidelity.

We came up with the network coding based algorithm that successfully decreased the number of messages, power consumption, delays and prolonged the network lifetime.

The algorithm we proposed and used in earlier chapters has difficulty dealing with the packet retransmissions, as it is deterministic and has to reserve time slots for the retransmissions up front. That decreases its efficiency. Because of retransmissions, there is a need for control messages (ACK/NACK). The overhead further decreases the efficiency of deterministic algorithm.

Also, different versions of this algorithm achieved either good delay characteristics or low power consumption. There is a need for an algorithm that would keep benefits of the network coding, but achieve low delays as the three-hop connectivity version and low power consumption like the one-hop direct-connectivity version of the previously proposed algorithm.

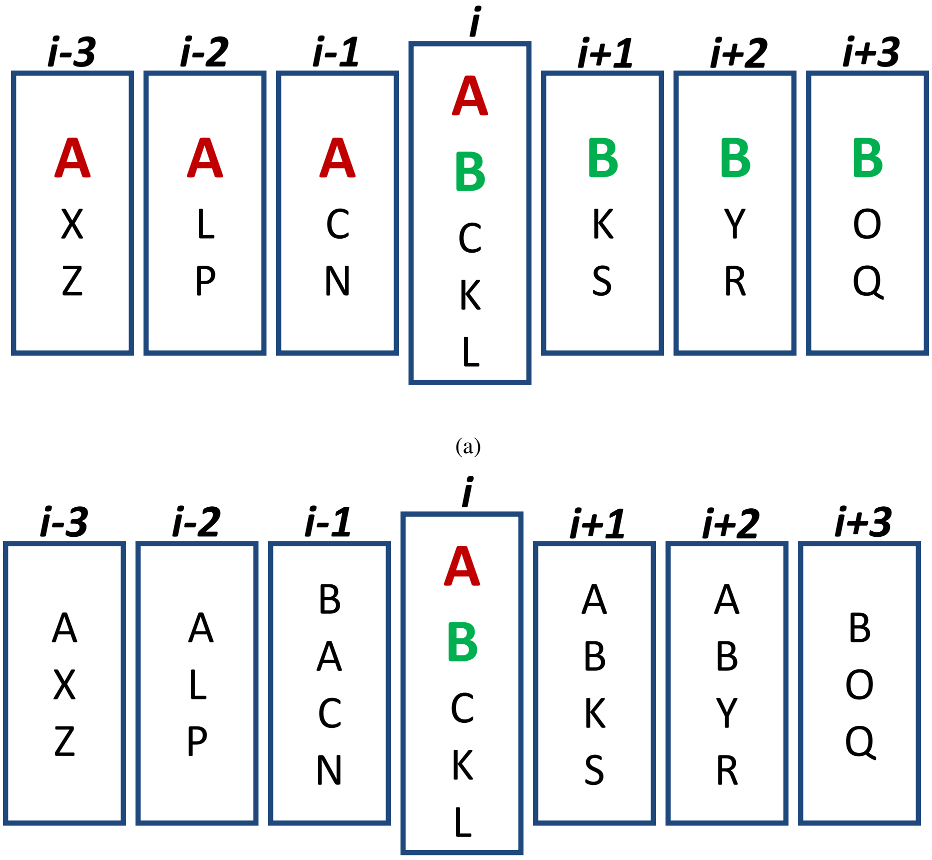
Therefore, we created an algorithm that takes the best from different versions of deterministic algorithm, and also deals with the losses in the wireless channel in a better way. The new version of algorithm does not use retransmissions as a way to make up for packet losses. The information about successful receptions is piggybacked which results in the decrease of data volume, particularly the part caused by control messages.

We show how nodes independently decide which message should they transmit at a given moment, simplifying the operation of algorithm.

7.4 Scenario

7.4.1 System Model

We assume that sensor nodes are equidistantly placed along one side of bridge. We assume that the channel is under influence of interference, fading, shadowing and path loss. Thus, channel is not lossless. Capture effect is also considered.



(a)

(b)

Fig. 7.1 Content of the buffer and lists of messages that six closest neighbours have. a) Node *i* transmits $A_{XOR}B$ message to the third-hop neighbours. b) Transmission is partly successful as nodes *i-3*, *i-2* and *i+3* did not decode any information due to the volatile channel.

A node broadcasts single (or network coded packet) towards a single (or a pair) of its up to three-hop neighbour(s) according to the utility function provided in the Table 1.

If a message is lost, it will be retransmitted should a deterministic protocol be in use. However, if adaptive protocol is used other nodes will just adapt to the loss, and consider it when they choose which messages should be transmitted in their next designated slot. Thus, when intended receivers fail to capture the packet(s) correctly, due to the broadcasting nature of the wireless channel some of other neighbouring nodes may capture packets and store them in their buffers. We will refer to this transmission as partly successful, as shown in the Figure 7.1.

In this chapter, we take into account the impact of the interference on performance of a network. We assume that only nodes unable to hear each other are able to transmit at the same time slot. Thus, as the maximum reception coverage is three-hops away, we will use conclusions from Chapter 6, regarding which packets can transmit at the same time.

We observed the worst possible case in which all nodes that cannot hear each other transmit at the same time. Also, three power levels will be available for each node, to enable a node to reach directly one-, two- or three-hop neighbour.

Each node will keep and regularly update six lists that contain information regarding messages that their nearest three neighbours from each side have. Thus, each node will individually choose to transmit a packet or a combination of packets from which its neighbour(s) will benefit the most.

Nodes regularly update their lists as the information concerning changes at their neighbours side is always piggybacked with the each packet a node transmits.

A node will also know how far will it send certain message and accordingly it will choose the transmission power level. If a combination of packets is sent, the transmission power level is chosen based on how far should a packet that needs to be forwarded further be received.

7.4.2 Hybrid relaying and network coding

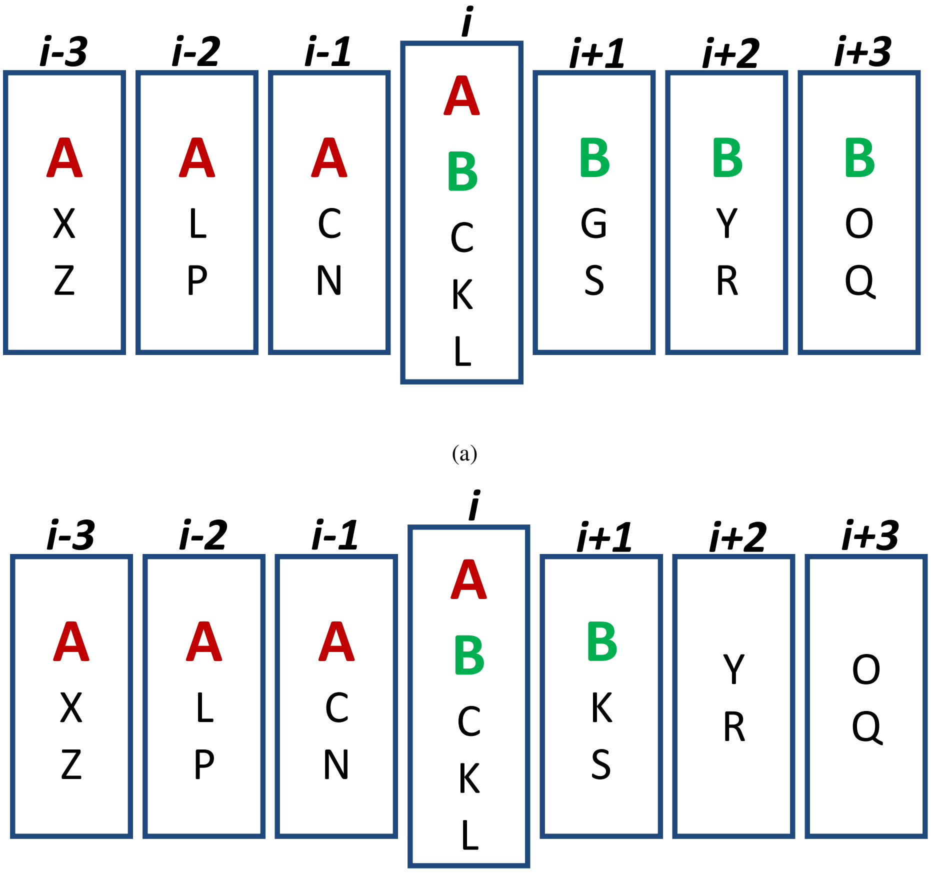
Each node maintains six different lists with the IDs of packets that each of its up to the three-hop neighbours have at the moment. For instance, a node with the ID 4 maintains six lists. List one contains the information concerning messages that node 1 has (third node from left side). List 2 contains IDs of messages known to the node with ID 2. List 3, contains IDs of messages node 3 already has (the first immediate neighbour from left side). Lists 4, 5 and 6 contain IDs of messages already received by respectively nodes with the IDs 5, 6, and 7.

If a node has less than six neighbours that are up to three-hops away, it will maintain a number of lists corresponding to its number of neighbours three or less hops away. Each message sent by a node piggybacks the update concerning messages that node received in the past transmission cycle. In general, a node with the ID i maintains six lists, for the nearest neighbouring nodes with the IDs i-3, i-2, i-1, i+1, i+2, i+3. For instance, a list for neighbour with the ID i-3 points out which packets node i-3 has at the moment. Those lists are updated continuously, as the updates are piggybacked. Each message sent by a node piggybacks the update concerning messages that node received in the last transmission cycle. Based on that information, a node with the ID i picks one or two messages from its buffer to transmit in the time slot assigned to that node, according to the criteria we will define in the following pages.

We define a set of criteria that a node uses to pick the best available packet/combination of packets for the transmission in its designated transmission slot. A node i compares a parameter u, which stands for utility, for different pairs of messages/individual messages from its buffer. The pair of messages or an individual message with the highest utility will be forwarded further. Should there be two or more pairs/messages with the highest utility, a node will consider the time they have spent in the buffer.

A node forms a pair of messages for transmission in occasions when one of the messages from its buffer should be forwarded to the nodes on its right hand side; and another message should be forwarded to the nodes on its left hand side (both of messages are known to the opposite sides). Node combines those two messages and forwards the coded packet to its neighbours. A node will pick a message or a combination of messages taking into account:

- how many nodes would profit from the information they would receive
- would one or two messages be forwarded
- how long is a message stored in the buffer (in the case of a combination of messages the packet that waited longer in the buffer would be chosen)



(a)

(b)

Fig. 7.2 Content of the buffer and lists of messages that six closest neighbours have. Utility: a) $\alpha = \alpha_A + \alpha_B = 3 + 3 = 6$. b) Utility: $\alpha = \alpha_A + \alpha_B = 1 + 3 = 4$.

Figure 7.2a shows the situation when node *i* has messages *A, B, C, K* and *L*. Message *A* is wanted by three nodes on its right side, while nodes on its left side already have it; similarly *B* is wanted by three nodes on its left side while nodes on its right side already have it, making network coding possible as all designated receptors will be able to extract the wanted message. Message *A* can be extracted by three nodes (with IDs *i+1*, *i+2*, *i+3*), thus its utility will be $\alpha_A = 3$. Message *B* can also be extracted by three nodes (with IDs *i-3*, *i-2*, *i-1*), thus its utility will be $\alpha_B = 3$.

When a node decides to send XORed message, utility metrics do not only observe how far is a message needed, it observes how far is it needed but also how far can it be decoded, Figure 7.2b. Even though a message *A* is needed by all three nodes on right hand side in the Figure 7.2b, if node *i* transmits message $A_{XOR}B$, only node *i+1* will decode a message *A* as it is the only one that knows message *B*. The utility for message *A* will therefore be $\alpha_A = 1$.

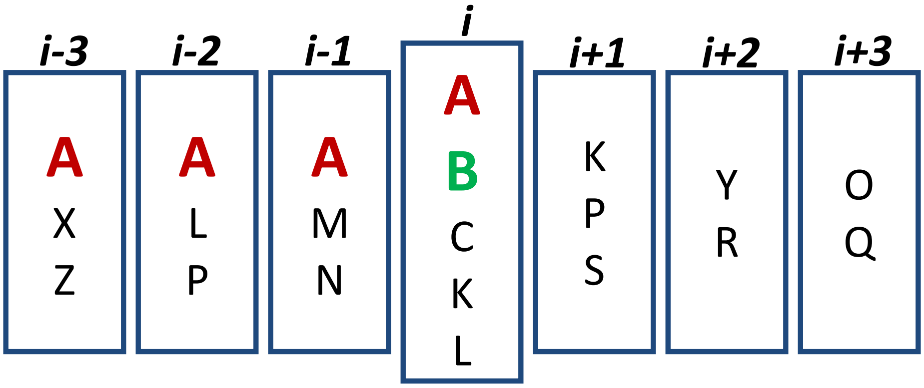


Fig. 7.3 Content of buffer and lists of messages belonging to six closest neighbours. Utility: $\alpha_A = 3$ and $\alpha_B = 3$.

We use the parameter $\boldsymbol{\alpha}$ as the utility that determines how far we can relay the information with one transmission. It will be interesting to explore how prioritising of messages influences the network performance. We first transmit XORed couples or single messages with the highest value of utility $\boldsymbol{\alpha}$. If two or more packets or combinations of packets have the same highest value of utility, we will decide what to transmit in a given slot based on parameter $\boldsymbol{\beta}$, that describes the time when packet arrived in buffer. The utility $\boldsymbol{\beta}$ prioritises packets that are waiting for the longest time in the buffer. The lower the $\boldsymbol{\beta}$, the longer is packet waiting for the transmission. The combination with the lowest value of parameter $\boldsymbol{\beta}$

and the highest value of $\boldsymbol{\alpha}$ is chosen for the network coding pair or a single message to be transmitted in a given time slot. An XORed packet will have $\boldsymbol{\beta}$ equal to the sum of $\sum_i \beta_i$ of its elements.

If there are no candidates for network coding, it should be determined which messages have the higher priority: those originating from other nodes that should be relied via traditional forwarding or nodes' own messages that should be broadcasted, Figure 7.3. If we assume that each node has a number of its own messages from its own sensors to transmit, we have to think if we want to prioritise those messages over the messages that a node is only relaying to other nodes.

It is a good idea for the total delay that a node prioritises its own messages and forwards them the first to sink nodes, and only then starts relaying the other nodes' messages if there are no candidates for network coding. This would mean that in the example from Figure 7.3, message $\boldsymbol{B}$ will be broadcasted. Otherwise, at the beginning of the protocol all nodes would end up relying messages and many opportunities for the network coding would be lost, their own messages must be relied on their own anyways.

We also need to discuss should we prioritise utility $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$? Parameter $\boldsymbol{\alpha}$ serves to help a node decide which message to send so that routing is the most efficient and results in the lowest number of transmissions. Technically, parameter $\boldsymbol{\alpha}$ helps node to use optimally the available battery power. Parameter $\boldsymbol{\beta}$ on the other hand takes care of the delay in the network. For the purpose of structural health monitoring, we will consider parameter $\boldsymbol{\alpha}$ to be more important, as shown in the Table 7.1. Only when parameter $\boldsymbol{\alpha}$ cannot resolve the situation, parameter $\boldsymbol{\beta}$ will be used to decide which message/set of messages will win the given time slot. This is because power consumption is usually more critical than end-to-end delay for SHM applications. The algorithm we use to choose the set of messages for transmission is showed in the Figure 7.4.

Based on this discussion, we create the table of utility values, Table 7.1. Different message prioritising rules will have impact on network performance. We decided to prioritise network coded messages as in the beginning there are only messages from the broadcasting phase so they will be transmitted the first instead of nodes getting involved in relaying of other nodes' messages. However, between the network coded messages and messages for simple forwarding, we chose to transfer the first the network coded messages as other messages do not create a potential XOR pair now, but they might be used to create a pair for network coding in the future. It is possible to choose a power level that enables node to achieve one-hop, two-hops or three-hops direct-connectivity, and node will adopt power level to the chosen transmission pair and distance at which it supposed to be received. It should be enough to enable the reception at the further designated node.

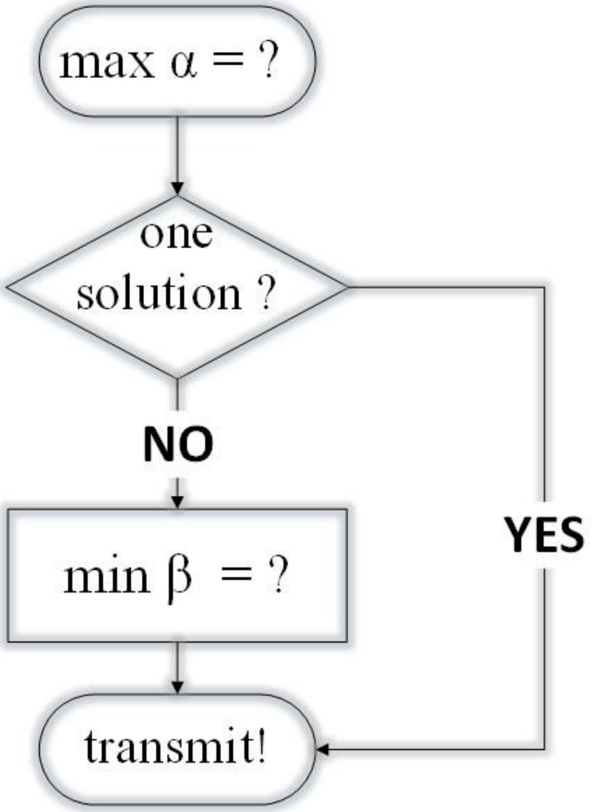


Fig 7.4. Adaptive protocol. Decision making process. What message(s) will a node transmit?

# hops on right (α_A) + # hops on left (α_B)	$\alpha_{1} = \alpha_{A} + \alpha_B$ [packet*hops]	α_2 *[NC or forwarding correction factor]*	$\alpha = \alpha_1 + \alpha_2$ [UTILITY WE USE]	**B** *[ordinary no in buffer]*
3+3	6	4	10	$\beta = A_\beta + B_\beta$
3+2	5	4	9	$\beta = A_\beta + B_\beta$
2+3	5	4	9	$\beta = A_\beta + B_\beta$
2+2	4	4	8	$\beta = A_\beta + B_\beta$
3+1	4	4	8	$\beta = A_\beta + B_\beta$
1+3	4	4	8	$\beta = A_\beta + B_\beta$
2+1	3	4	7	$\beta = A_\beta + B_\beta$
1+2	3	4	7	$\beta = A_\beta + B_\beta$
3-hop, its own message	3	3	6	$\beta = A_\beta + B_\beta$
2-hop, its own message	2	3	5	$\beta = A_\beta + B_\beta$
1-hop, its own message	2	3	4	$\beta = A_\beta + B_\beta$
4 hops, but only one side needs it	3	0	3	$\beta = A_\beta + B_\beta$
2-hops, but only one side needs it	2	0	2	$\beta = A_\beta + B_\beta$
1-hop, but only one side needs it	1	0	1	$\beta = A_\beta + B_\beta$

Table 7.1. The table of decision making rules. Which packet should have the priority?

Each message transmitted by a node, piggybacks the information about messages node received in the previous transmission cycle. If a node sent message to its two-hop neighbour only once this neighbour transmits it will piggyback that information, so that third-hop neighbour becomes aware of the changes. This way all nodes will be able to update their lists regularly. Once a node, and both of its 3rd hop neighbours have the same message, that message will be deleted from buffer in order not to occupy unnecessarily the memory. Each node uses its designated transmission slot to update others about its own situation. Thus, lists are dynamically updated. There is no need for ACK messages, as node will find out if and which of its neighbours successfully received the message via piggybacked information (from the list of messages' IDs a node received in the previous transmission cycle).

If a transmission was unsuccessful or partially successful (e.g. only nodes 1-hop or 2-hops away decoded it, while a node 3-hops away did not) there is no need for retransmission. Simply, that information will be considered in the next transmissions round, when packets compete for the transmission slot.

This approach is very practical as it solves the situations when packets do not come in order or when there are no network coding opportunities (as it still uses the slot for forwarding). It removes the overhead coming from retransmissions, ACK and NACK messages, it is simple and convenient. This approach is developed in a manner that tackles both problems of power consumption and end-to-end delays.

7.5 Simulation Testing

We compare the adaptive protocol to the deterministic version of it discussed in the previous chapter for single-line layout of nodes. We will consider the following parameters for the network of 40 nodes. Transmission powers used to achieve respectively one-, two-, and three-hop direct-connectivity are P_{t1}=-3.35 [dBm], P_{t2}= 3.1 [dBm], P_{t3}=8.08 [dBm]. A node is able to receive and decode a signal if its transmission power at the reception is above $P_{treshold} = -103$ [dBm]. We considered packets with $L=128$ [B]. The data rate of $V_b=250$ [kbps]. The length of observed bridge with 40 nodes deployed in a line is $l=2$ [km].

As explained in the previous chapters, we compute the energy spent for a reception of one message, using following values for current and voltage at reception, $I = 9[mA]$ and $U = 3[V]$. We obtain the reception power in [dBm] using

$$P_r=10*\log(U*I/0.001). \tag{7.1}$$

The energy spent for receptions can be obtained from:

$$E_r=I*U*T \ . \tag{7.2}$$

The battery used by sensor nodes has current $I_{tot}=1.7*3600$ [mAh], and voltage of $U_{tot}=1.5$ [V].

Thus, the total available energy is:

$$E_{tot}=I_{tot}*U_{tot} \ . \tag{7.3}$$

As there are no more retransmissions, and the order of packets in the buffer is taken into account, adaptive protocol outperforms deterministic protocol. It is mostly so because of the absence of retransmissions and the efficient use of transmission slots by this version of protocol that is better even than three-hop direct-connectivity case in terms of delay.

Maximum delay is defined as the higher of times that takes a packet to reach each sink node. $D_{i,j}$ stands for the time it takes a packet with the ID i to reach the node j. A maximum delay of packet is:

$$Max\ D_i = \max(D_{i,1}, D_{j,N}). \tag{7.4}$$

Figure 7.5 shows that adaptive protocol performs almost as well as the three-hop deterministic network coding based protocol in terms of maximum delay of a packet originated from a given node (x-axis). This

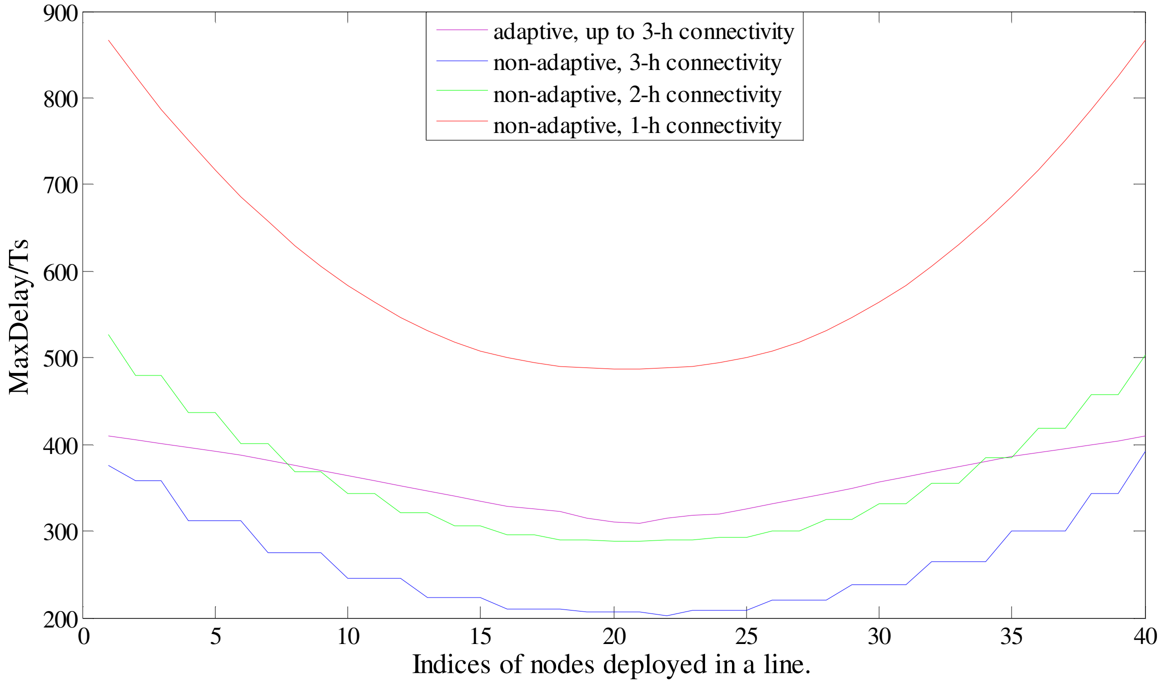


Fig. 7.5 Adaptive protocol. Maximum end-to-end delay.

happens in spite of the fact that the adaptive version benefits from far less overhearing (as some of the messages are relied only to the one-hop or two-hops distance). However, the absence of retransmissions and ACK/NACK messages helps decrease the delay significantly.

We may go a step further and define the minimum delay D_i as the time when a packet from the node *i* reaches the first sink node (usually the nearest sink node) as:

$$Min\ D_i = \min(D_{i,1}, D_{j,N}). \tag{7.5}$$

We see that nodes from the ends in general reach the first sink node before the middle nodes, while middle nodes reach both sinks before end nodes do, Figure 7.6.

In the case of deterministic protocol, maximum delay was a better measure of delay. However, because of the nature of adaptive protocol minimum delay should be used as a measure since there is guarantee that a message will be received at the sink node.

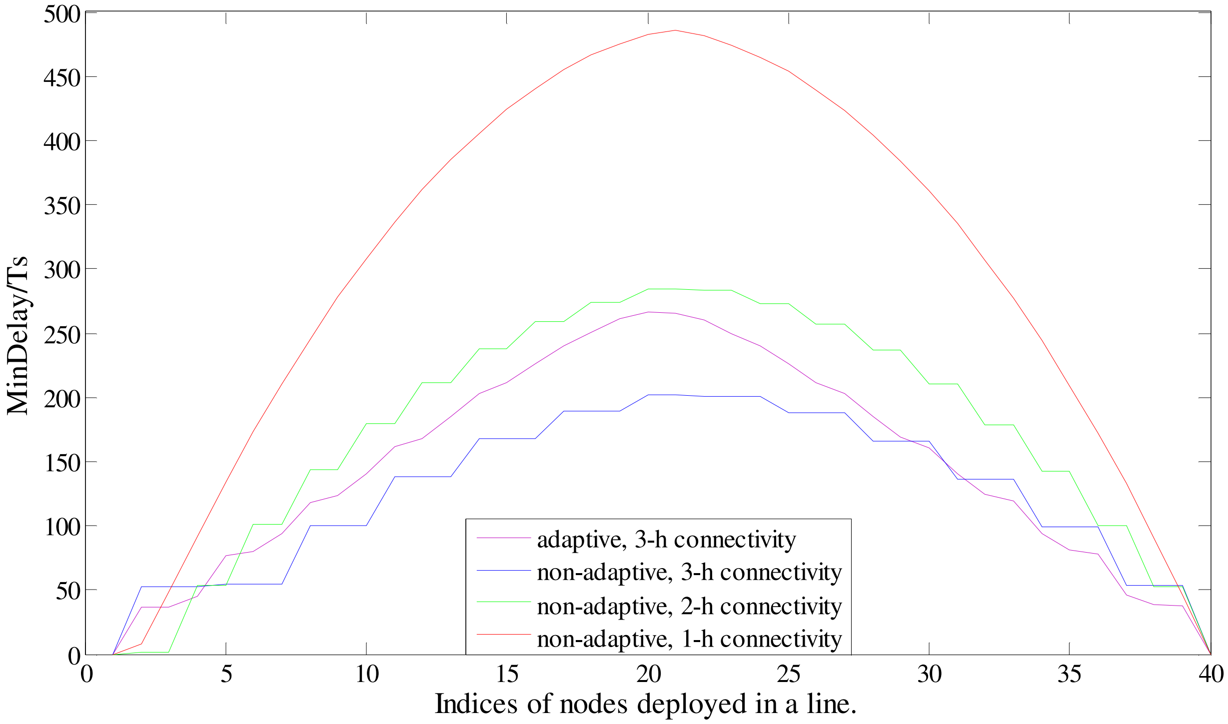


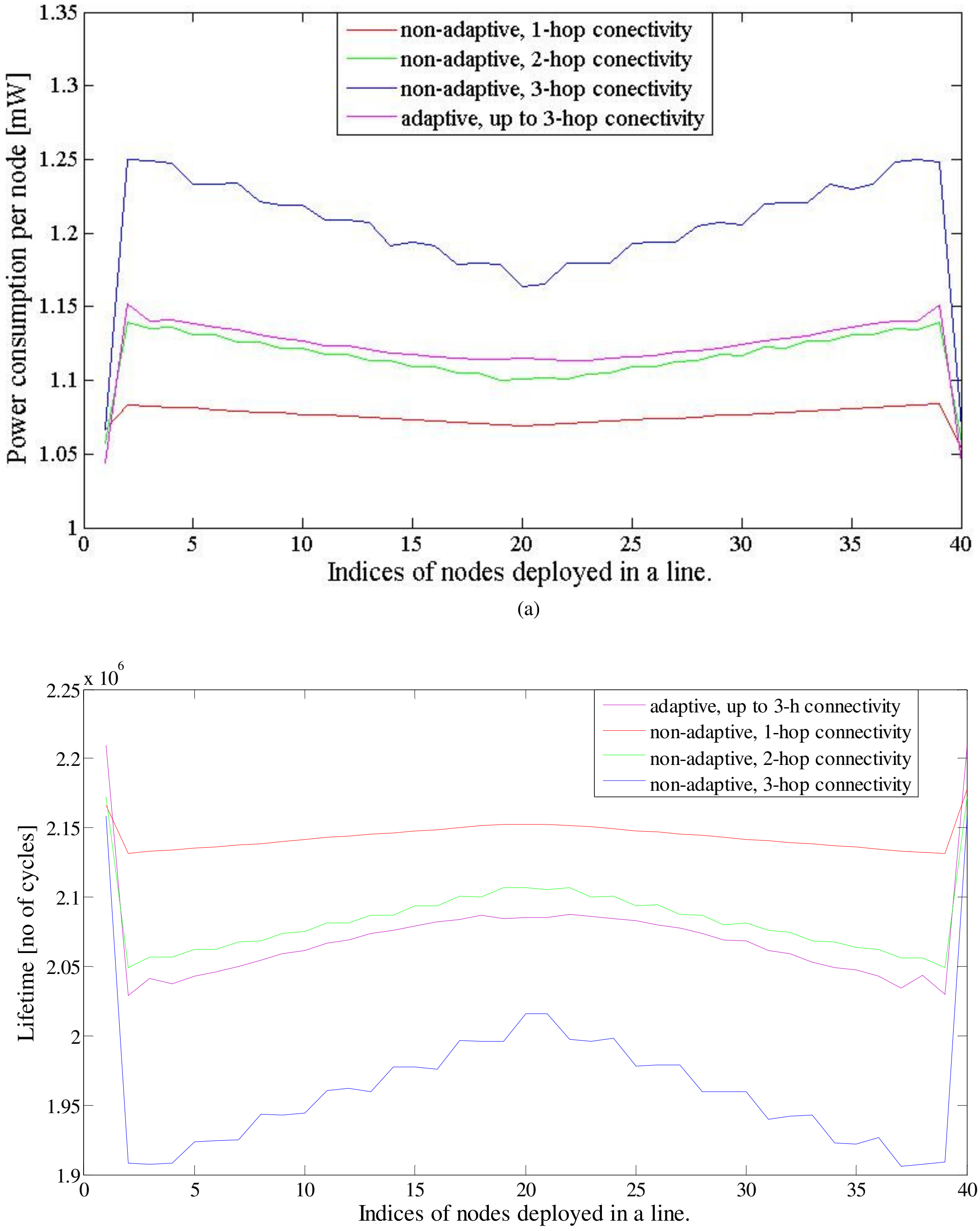
Fig. 7.6 Adaptive protocol. Minimum end-to-end delay.

Combining previous two measures, we might also be interested in the average delay in network. We define average delay as:

$$Average\ D_i = \frac{Min\ D_i + Max\ D_i}{2}. \tag{7.6}$$

We are interested to see how the adaptive version of protocol performs in terms of the power consumption. As some of transmissions are targeting 3rd -hop neighbours, the total power consumption is likely going to be higher than in the case of a one-hop direct-connectivity. As transmission slots are used efficiently and there are no retransmissions, the protocol likely performs better than in the case of three-hops direct-connectivity regardless the same transmission powers.

In Figure 7.7a, we compared power consumption of nodes deployed on the bridge.



(a)

(b)

Fig. 7.7. Network with 40 nodes. a) Total power consumption per node for the deterministic and adaptive protocol. b) Lifetime distribution for nodes deployed in a line for the deterministic and adaptive protocols.

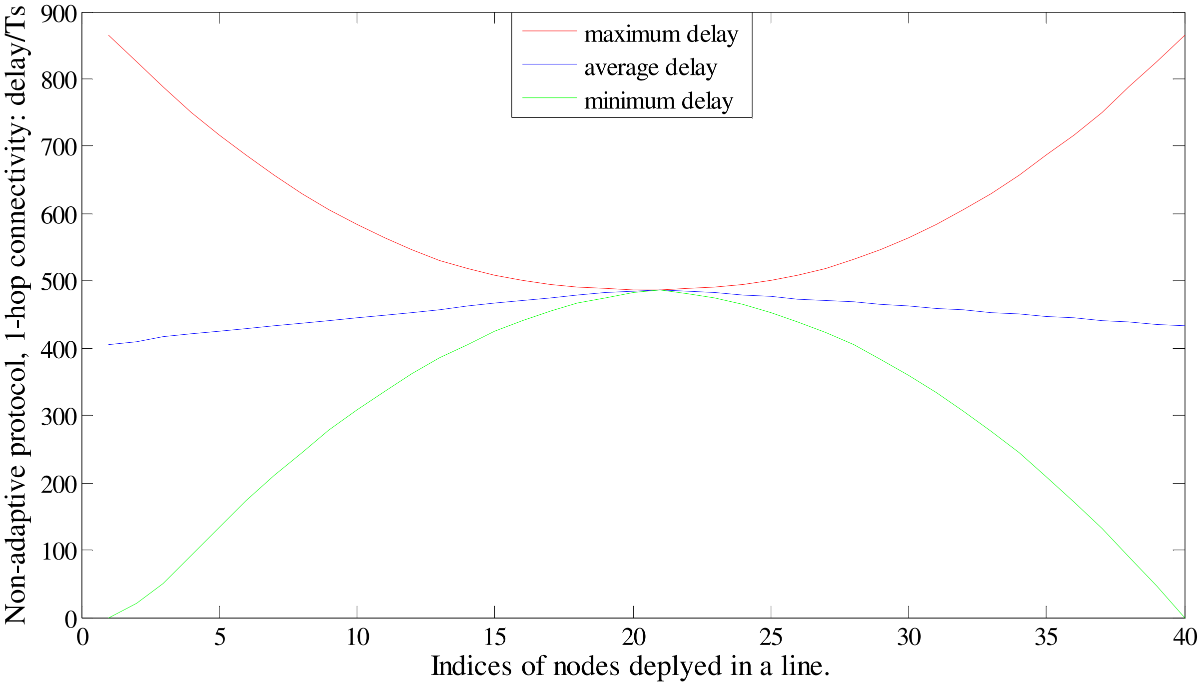
We see that adaptive protocol performs almost as well as two-hop direct-connectivity network coding based routing, in spite of being 3-hop based. This is mainly because no retransmissions or overhead in form of ACK/NACK messages exist in the network.

We are interested to see what is the lifespan of network based on this protocol and how does it compare to the previous deterministic versions of this protocol. If a node dies, we assume that its neighbours will increase their transmission powers in order to keep the network connected. However, this would impose the new load on already limited batteries of these nodes, and their energy supply would drain out at a faster rate.

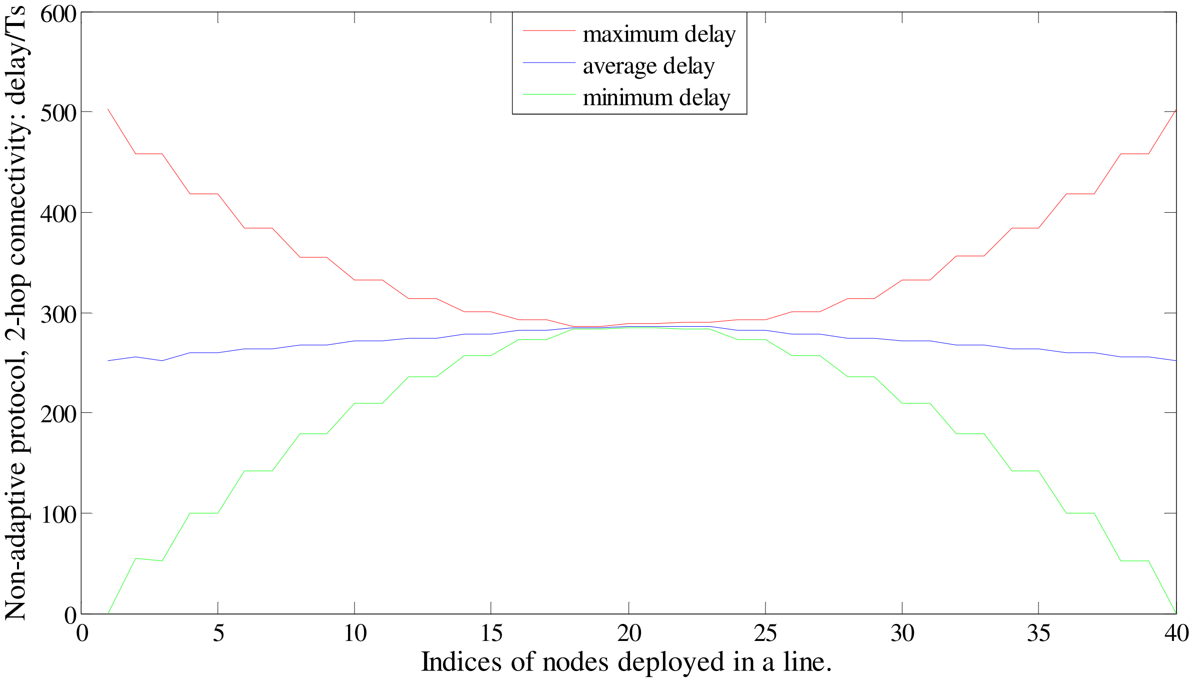
Once three nodes in a row die, we consider a network to be disconnected. As expected in terms of lifetime, the adaptive protocol performs similarly to the one-hop direct-connectivity case of the deterministic network coding based routing, as shown in the Figure 7.7b. Thus, adaptive protocol outperforms both two-hop and three-hop direct-connectivity network coding based routing protocols.

From the Figure 7.7b we see that the network with one hop-connectivity has the longest lifetime. It is unfair to compare this to up to the three-hop adaptive protocol that has a much better delay characteristic. The fair comparison is between three-hop adaptive and deterministic protocols. Adaptive protocol outperforms the deterministic version both in terms of delays and power consumption.

Let us observe the delay characteristics of deterministic protocols with one- and two-hops direct-connectivity. If we observe Figures 7.8 and 7.9 we notice that delay distribution always has similar shape. Nodes in the middle have very similar values of maximum and minimum delay, while end nodes have very high maximum delays and their minimum delays are close to zero. Delays are the highest in the case of deterministic protocol with one-hop connectivity, while deterministic protocol with the three-hop direct-connectivity case has the lowest delays. Adaptive protocol with up to the three-hop connectivity has a slightly different delay distribution than others. Maximum delays are more balanced than in the case of deterministic protocol.

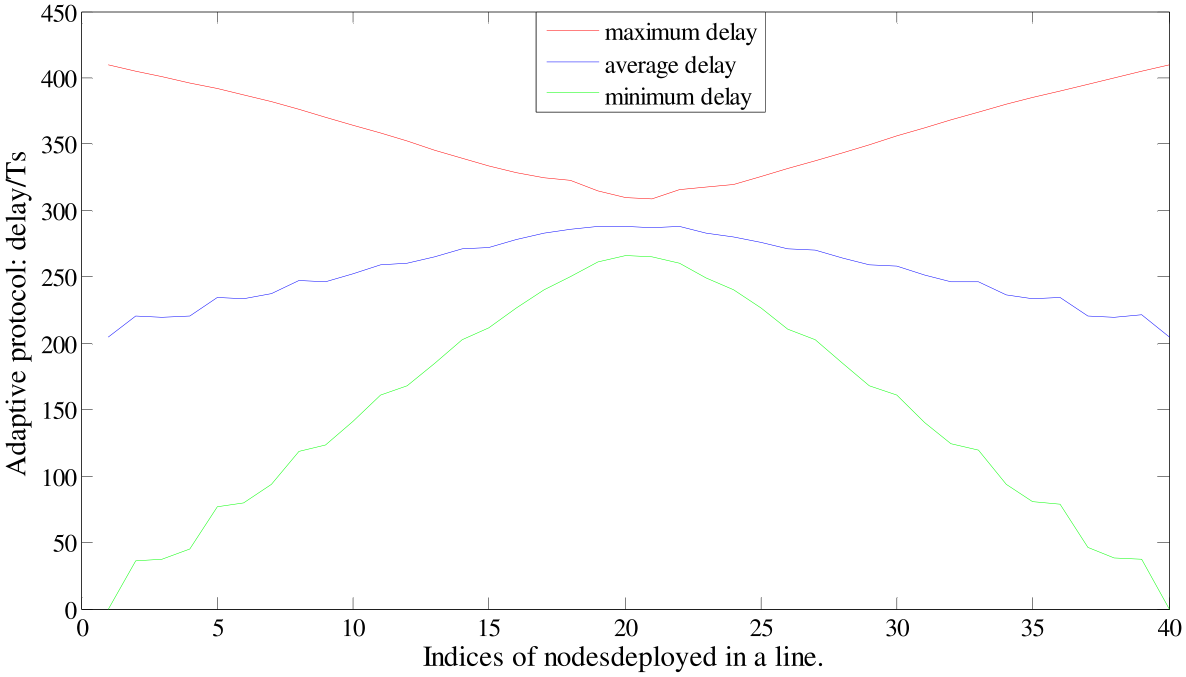


(a)

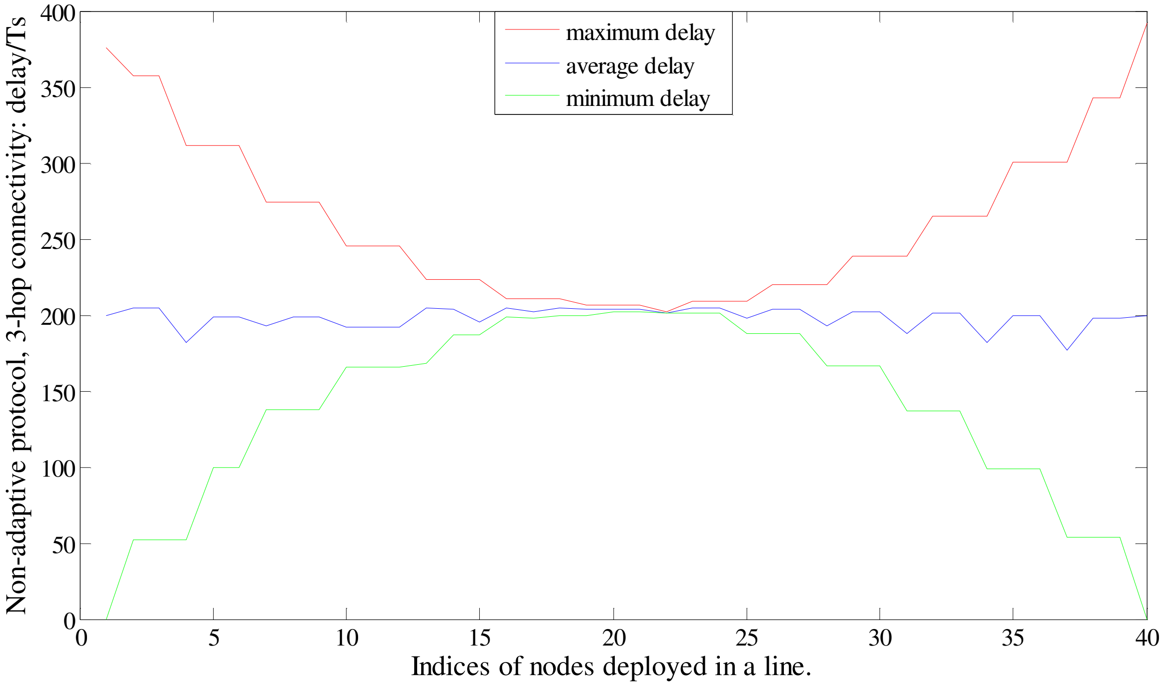


(b)

Fig. 7.8 Minimum, average and maximum delay of deterministic protocol with a) one-hop connectivity b) two-hop connectivity.



(a)



(b)

Fig. 7.9. Minimum, average and maximum delay of a) adaptive protocol b) deterministic protocol with three-hop connectivity.

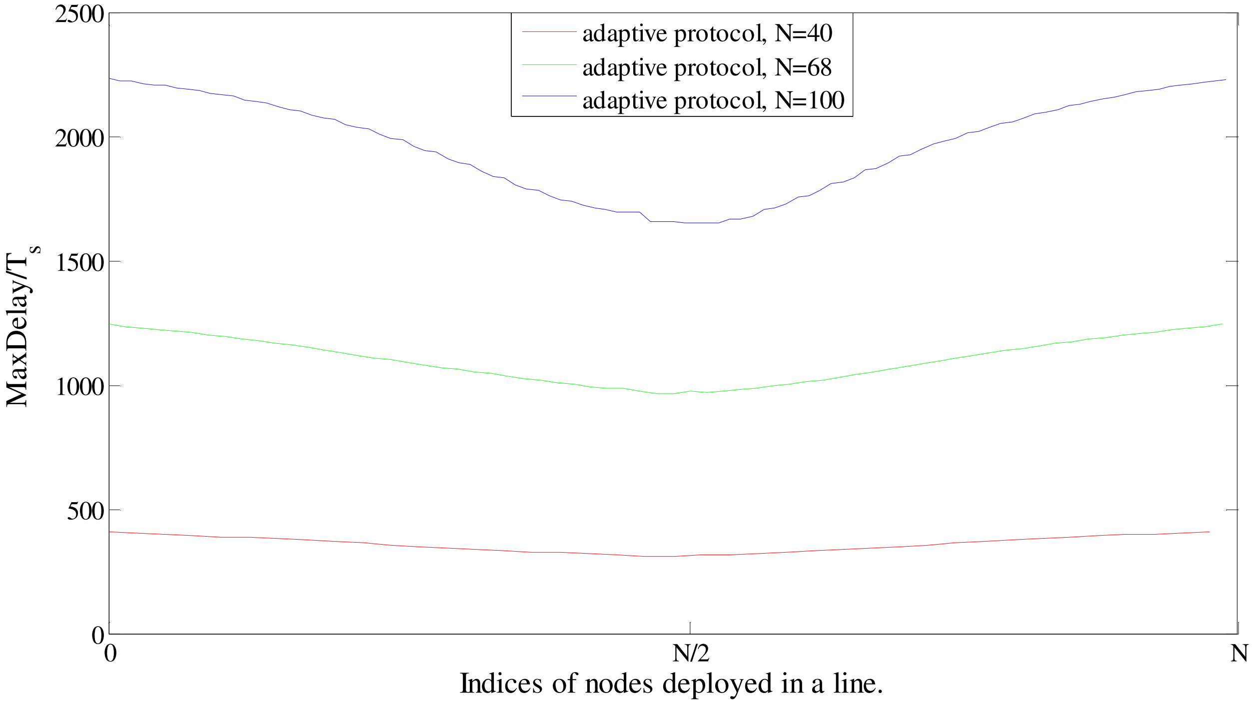
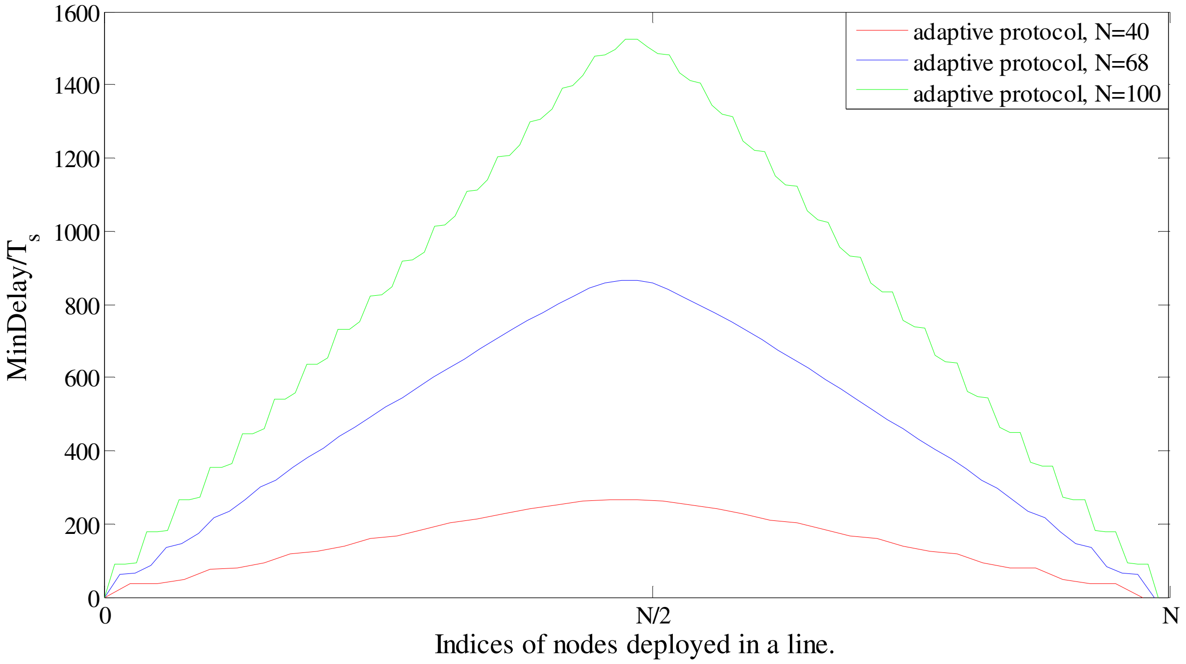
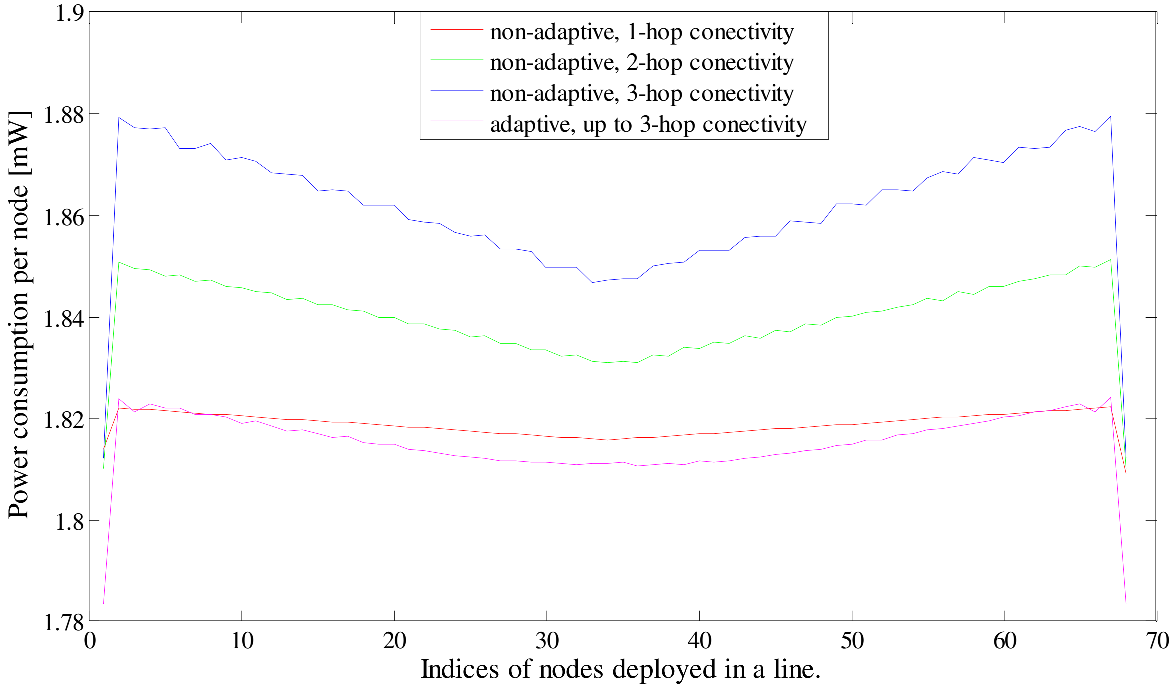


Fig. 7.10. The maximum delay characteristic of the networks using adaptive protocol.

If we observe how the maximum delay characteristic of adaptive protocol changes with the number of nodes deployed in the network, we realise that this protocol tends to make the maximum delay distribution curvier. It prioritises messages generated at the middle nodes, since most of them use network coding. The minimum delay distribution of adaptive protocol has the peak in the middle and zeroes at the ends. The moment a sink node creates a packet (from data received from its sensors), that packet is already at one sink. If packet transfer was not prone to errors, this curve would be completely relevant. But because we are interested in collection of information at the both sink nodes in order to increase the reliability, we are mostly interested in the moment a packet is collected at both sink nodes. Therefore, having no retransmissions is an advantage of the adaptive version of this protocol. Nodes using deterministic protocols are exposed to the combined Rayleigh fading, shadowing, path loss and interference, and they are forced to retransmit their messages a number of times. We want to show that results are not dependant on number of nodes deployed in the network.

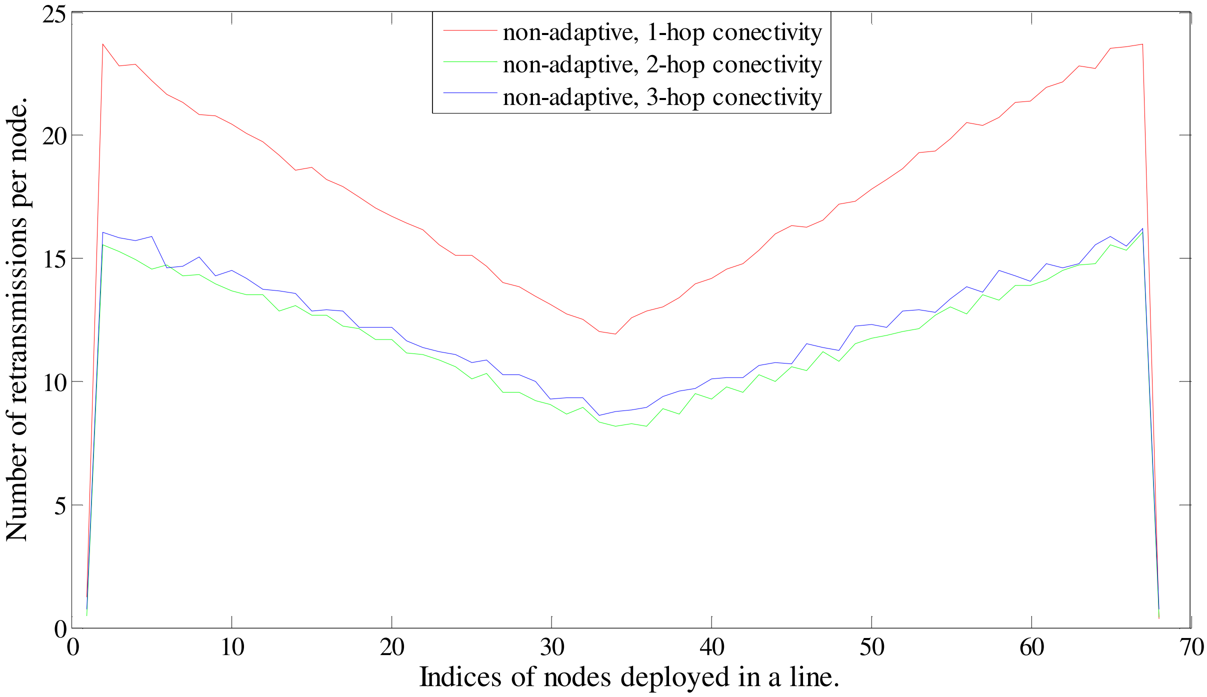


(a)

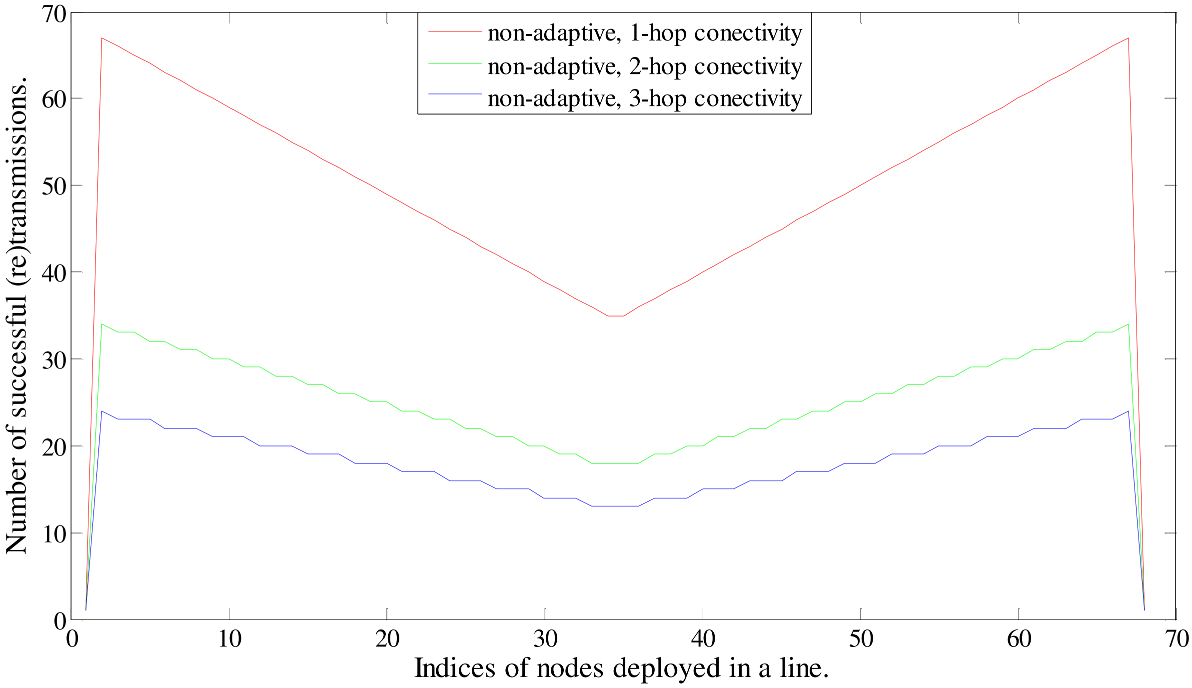


(b)

Fig. 7.11. a) Minimum delay characteristic of networks using adaptive protocol. b) Total power consumption per node of the network using deterministic and adaptive protocols. Given network consists of 68 nodes.



(a)



(b)

Fig. 7.12. Adaptive protocol. a) Total number of retransmissions. b) Total number of successful (re)transmissions.

We see that the power consumption of adaptive protocol can be even better compared to the deterministic protocol using one-hop connectivity. This is coming from the elimination of retransmissions, and the decrease in message overhead, Figure 7.12.

In general, the power consumption curve of adaptive protocol tends to fluctuate between the curves depicting deterministic cases with two-hop connectivity and one-hop connectivity depending on number of retransmissions in the network deterministic protocol is exposed to.

We chose transmission powers that show two extremes of the power consumption of adaptive protocol. In the case shown in the Figure 7.11b, number of retransmissions is very high as shown in the Figure 7.12a. In the Figure 7.12, the upper picture shows the total number of retransmissions per node, whereas the lower shows the total number of successful transmissions. The sum of these two values represents the total number of transmissions needed to successfully relay messages to the sink nodes, Figure 7.11.

How does this situation reflect on delays in the network? The delay for some nodes is even higher than it would be if they used two-hop direct-connectivity deterministic protocol. This is because the poor conditions in wireless channel effectively cause losses for the adaptive protocol, so the greater proportion of messages will be delivered through two-hop relying.

Thus, while the power consumption characteristic is better than in the case of deterministic protocol, that happens at the expense of the end-to-end delay, Figure 7.14.

In the same channel conditions, protocols have slightly different performance. Figure 7.13a shows the percentage of the transmissions successful at the first attempt, compared to the the percentage of the retransmissions. Messages whose decoding was partly successful are not going to be retransmitted. Simply, the furtherst node that receives a message forwards it further. The longer distance a message is trying ot reach, the more often it fails.

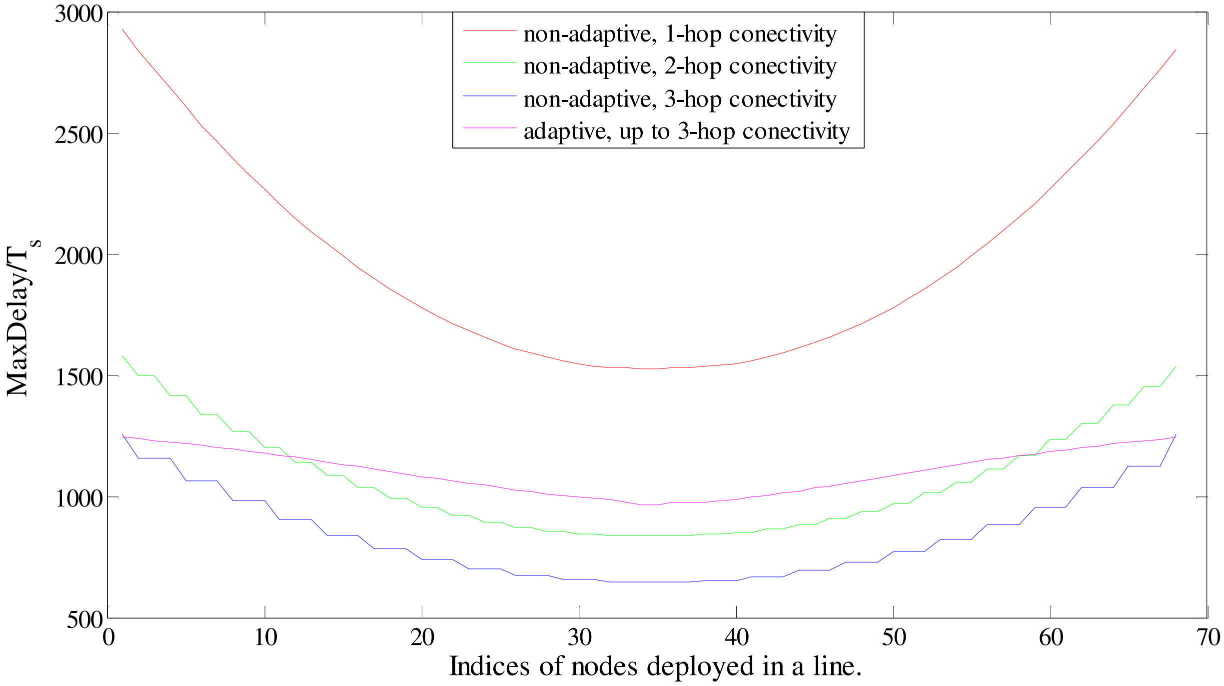


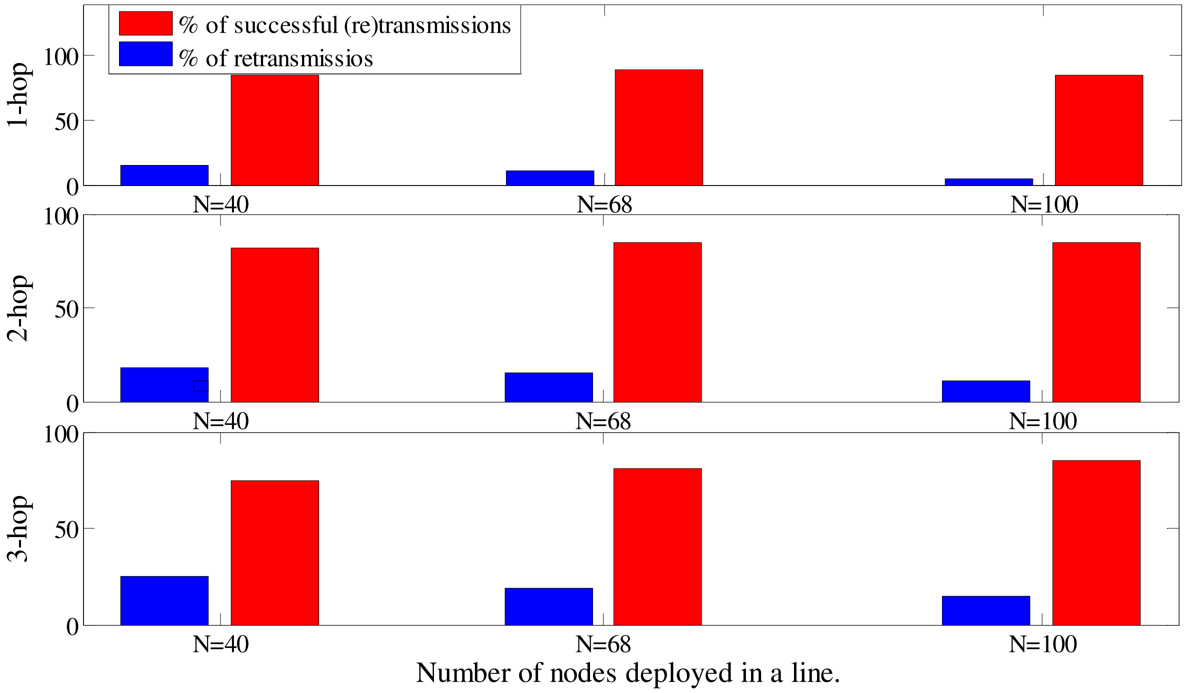
Fig. 7.13 Comparison of the end-to-end delays in a poor channel for deterministic and adaptive protocols.

Figure 7.13b shows the percentage of messages that are directly relied to their 3-hop neighbors in the same channel, compared to the percentage of messagess relied to the 2-hop and 1-hop neighbors.

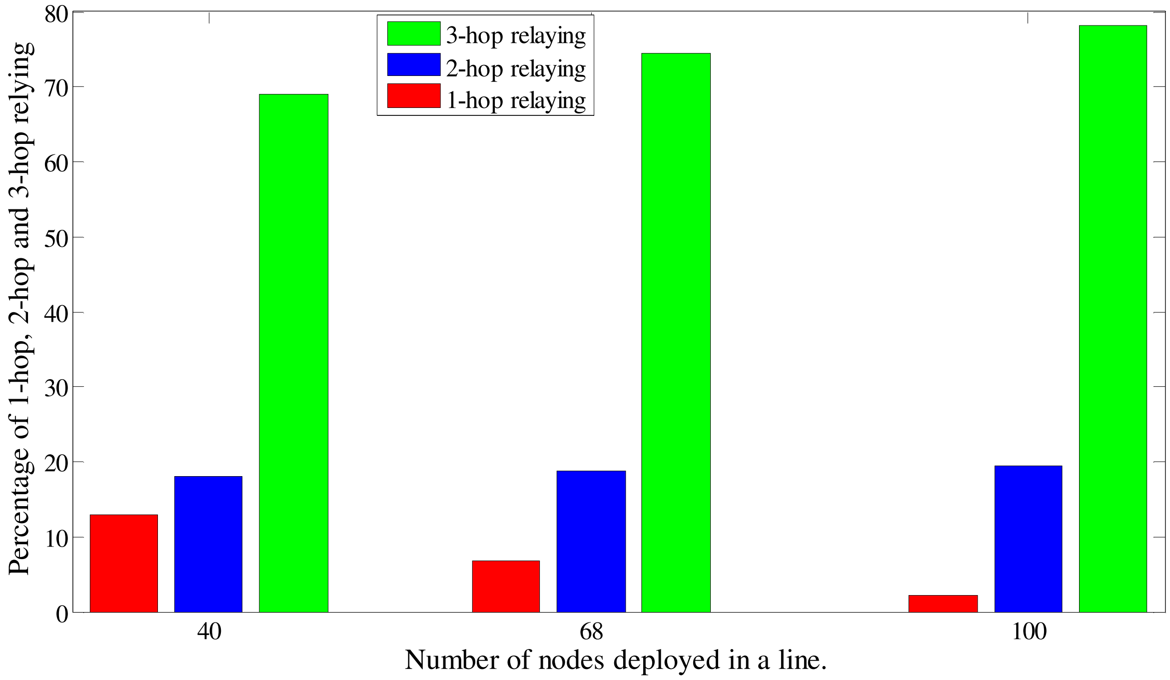
As both deterministic and adaptive versions of protocol are using the same wireless channel, they have a similar percentage of lost messages that are relied to the one-hop, two-hops or three-hops the distance among adjacent nodes in a line.

The difference in Tables 7.2 and 7.3 is due to the fact that adaptive protocol inherently chooses to transmit a portion of imessages via one-hop, two-hops or three-hops relying. On top of that there are losses in the channel to which the protocol will adapt.

Any time a transmission is partly successful, adaptive protocol will not retransmit the message. Instead the furthest node that successfully decoded the message will relay that message further.



(a)



(b)

Fig. 7.14 Success rate statistics. a) Percentage of successful transmissions and retransmissions for determinitc protocol. b) Percentage of transmissions that are successfully decoded at 1-hop, 2-hop and 3-hop distance for adaptive protocol.

DETERMINISTIC PROTOCOL		N=40	N=68	N=100
1-hop direct-connectivity	% of successful first time transmissions	85	89	95
	% of retransmissions	15	11	5
2-hop direct-connectivity	% of successful first time transmissions	82	85	89
	% of retransmissions	18	15	11
3-hop direct-connectivity	% of successful first time transmissions	75	81	85
	% of retransmissions	25	19	15

Table 7.2. Percentage of successful transmissions and retransmissions for the deterministic protocol.

ADAPTIVE PROTOCOL	N=40	N=68	N=100
% of messages relied to the 3-hop neighbour	69	74.52	78.2
% of messages relied to the 2-hop neighbour	18.04	18.7	19.52
% of messages relied to the 1-hop neighbour	12.96	6.78	2.28

Table 7.3. Percentage of messages relied via 3-hop, 2-hop and 1-hop relying.

When we observe the numbers in the Table 7.3, we should bare in mind that as there are no retransmissions in the case of adaptive protocol sometimes messagges are on purpose sent to the 2-hop and 1-hop distance and it was sucessful. However, some percentage is due to the fact that transmissions can be only partly successful (e.g. a message was sent to distance $3d_i$ but it was decoded only by nodes at distances d_i and $2d_i$).

We are interested to explore how does the number of sensor nodes deployed at a bridge of a fixed length reflects on the performance of network. Figure 7.14 shows the total number of transmissions (including the retransmissions) in a network for a given protocol and connectivity level. As expected, adaptive protocol results in the lowest number of transmissions as there are no retransmissions and control mesages overhead.

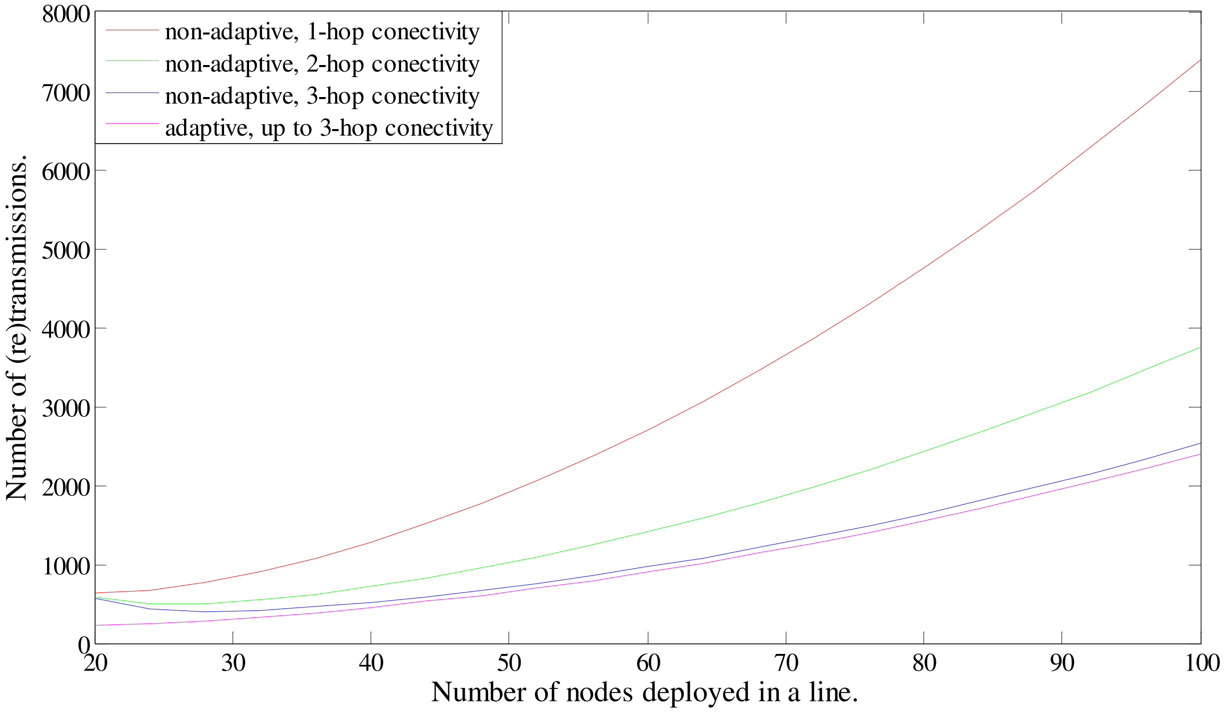
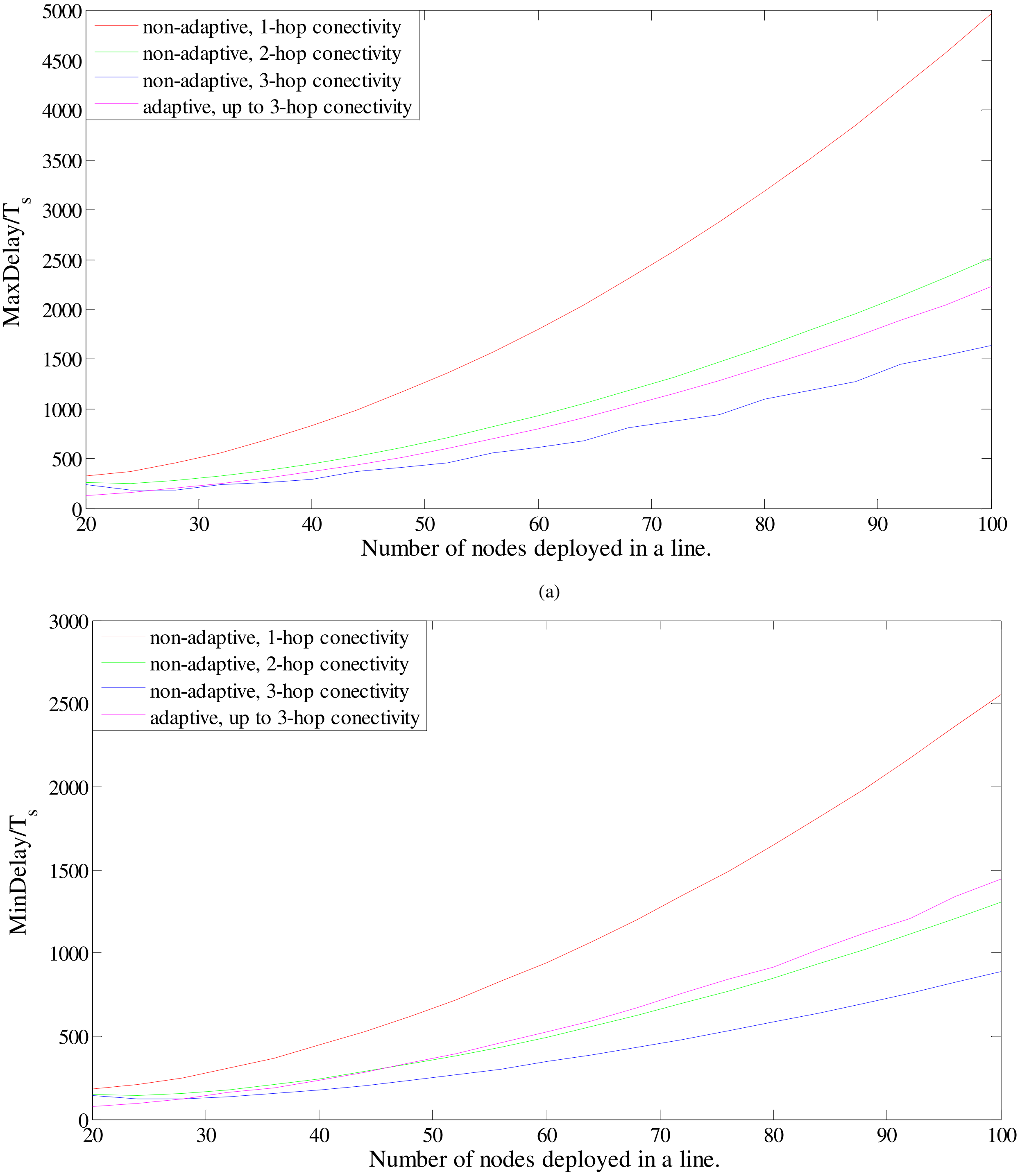


Fig. 7.15 Total number of (re)transmissions in a network using deterministic or adaptive protocol for varying number of nodes.



(a)

(b)

Fig. 7.16 Network with different number of nodes a) Maximum end-to-end delay. b) Minimum end-to-end delay.

We are interested in the moment when all messages generated at sensor nodes will be delivered to the both sink nodes. We define this parameter as a maximum end-to-end-delay:

$$Max\ D_{tot} = \max_i(Max\ D_i). \qquad (7.7)$$

$Max\ D_i$ is defined in the formula (7.4). This newly defined parameter represents the time when the last packet reaches both sink nodes. Maximum delay is shown in the Figure 7.14a. We may see that the adaptive protocol performs better than 1-hop and 2-hop direct-connectivity deterministic protocol. It performs worse than 3-hop direct-connectivity deterministic protocol, as for the same channel conditions signals undergo similar losses. Adaptive protocol transmits smaller proportion of messages with the highest power level, enabling them to be decoded at the three-hop distance. The fact that only about 75% of messages are relied via 3-hop receiving is a disadvantage over the deterministic case, as nodes benefit less from overhearing. The lack of retransmissions offsets this issue for adaptive protocol to some extent.

Some researchers prefer using other parameter, e.g. the minimum end-to-end delay. It represents the time when the last packet is delivered to at least one sensor node. We define it as:

$$Min\ D_{tot} = \max_i(Min\ D_i). \qquad (7.8)$$

Where $Min\ D_i$ is defined in the formula (7.5). The value of this parameter is presented in the Figure 7.14b. For the given number of nodes this value is always lower than $Max\ D_{tot}$. We see that for the larger number of nodes the minimum end-to-end delay is better for deterministic 2-hop direct-connectivity case. This is because adaptive protocol is designed to primarily take care of the efficient usage of power resources, while delay has the lower priority.

However, the minimum end-to-end delay is not a parameter of concern for the Structural Health Monitoring applications. It is important that both sink nodes receive the message to consider it relevant as the reliability is the main requirement of this application.

Now, we observe the total power consumption of the network. That concerns all power consumed in the network from the beginning of sensing cycle until all messages are delivered to both sink nodes. The adaptive protocol performs almost as well as one-hop direct-connectivity deterministic protocol. This is because the power consumption was priority in the protocol design phase. It benefits from no retransmissions and the control message overhead. All information needed is piggybacked.

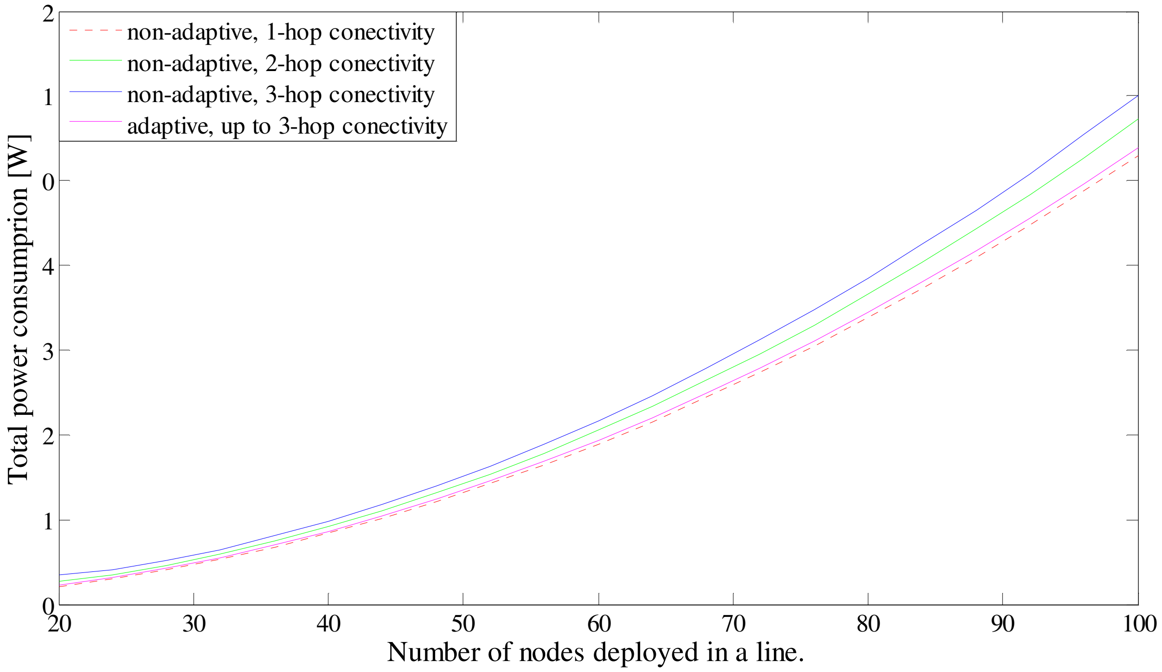


Fig. 7.17. Total power consumption of network using adaptive and deterministic algorithms.

We want to explore the lifetime of a network using adaptive protocol, and compare it to the lifetimes of a network that relies on deterministic protocol. We assume all cases are tested in the same channel, and thus they undergo similar losses due to the interference, fading, shadowing and the path loss. We consider a network to be disconnected only when three sensor nodes in a row die. As discussed in the previous chapters, once a node dies its neighbours have to increase their transmission powers in order to keep the network connected. Nodes that reconnected the network are spending their power supply at a much faster rate, and as a result die themselves soon after. This effect is known as chain dying of nodes. Once three consecutive nodes drained out their power supply the network is disconnected.

Figure 7.18 shows the lifetime of a network using different number of nodes. It turns out that adaptive protocol performs almost as well as the deterministic protocol with the one-hop direct-connectivity.

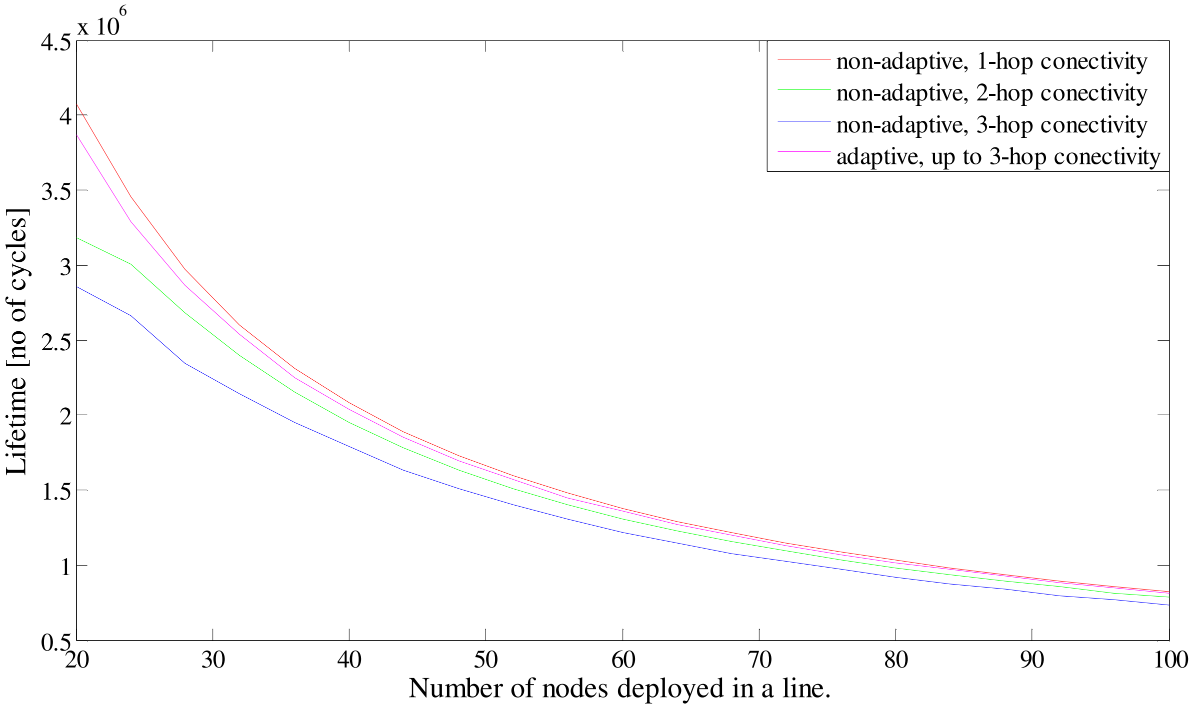


Figure 7.18. The lifetime of network for adaptive and deterministic algorithms and different number of nodes deployed in a network.

In general, the adaptive protocol is a good trade-off between one-hop and three-hop direct-connectivity versions of the deterministic protocol. In a reasonable channel it performs almost as well as the one-hop direct-connectivity deterministic algorithm in terms of power consumption. In terms of end-to-end delays it is almost as good as the three-hop direct-connectivity version. In the extremely poor channel conditions it outperforms all versions of deterministic protocol both in terms of delays and power consumption.

7.6 Summary

In this chapter we propose an adaptive network coding based algorithm. Nodes internally decide what messages they will transmit at a given time slot using their best knowledge regarding the information their neighbors have at a certain moment. Nodes broadcast single or network coded packets towards a single or a pair of its up to three-hop neighbor(s), using a novel utility function. A protocol attempts to minimise the energy consumption while keeping delays in the network low.

In the case of packet loss, nodes do not retransmit messages. Instead, that information will be used in the further choices regarding which message/pair of message a node will transmit in its designated time slot. The goal of this algorithm is to reduce the power consumption and decrease delays whenever possible.

Thus, the intended receiver may fail to capture the packet(s) correctly, however due to the broadcasting nature of the wireless channel some other neighboring nodes may capture the packet(s) and store them in their buffers (partly successful transmission).

This algorithm achieves delays similar to the three-hop direct-connectivity version of the deterministic algorithm, and consumes power similar to the one-hop direct-connectivity version of the deterministic algorithm. In very poor channel conditions, this protocol outperforms all versions of deterministic algorithm both in terms of delays and power consumption.

We show how nodes choose which messages will be transmitted in the next designated time slot. We test our algorithm in different channel conditions using extensive simulations. We show the distribution of delays and power consumption in the network, end-to-end delays, total power consumption and network lifetime. The adaptive algorithm is compared with the deterministic version.

Our results can be used for the design of more efficient network coding based scheduling and routing algorithms in the wireless sensor networks for Structural Health Monitoring (SHM) of linear structures.

8 CONCLUSIONS

We have proposed and analysed the performance of a new deterministic algorithm that employs the network coding technique for structural health monitoring of bridges. Wireless sensors are deployed at bridges to take (sense) a variety of measurements. The key idea is to use the network coding to reduce the number of transmissions needed to forward sensing data to the sink nodes. We have analysed the performance gain of the network coding technique. We have also considered adjusting the transmission power to cope with the propagation environment characterized by fading, shadowing and path loss to assure the high probability of successful reception. We evaluated the power consumption by all nodes in the network. It has been found that the network coding method can always reduce the power consumption for a given node layout and degree of connectivity. However, special attention has to be paid to the layout and connectivity if the objective is to minimise the overall power consumption (i.e. to maximise the network lifetime) when the network coding technique is applied. For the numerical examples considered, the scenario with the single-line layout, one-hop connectivity and network coding consumes the least amount of power. It is so because networking coding gain depends on the node layout and connectivity in some complicated ways. In terms of the future work, we plan to design the medium-access protocol to support the proposed algorithm and to investigate the interference cancellation techniques that could improve the algorithm performance as nearby sensor nodes may transmit data simultaneously.

We evaluate performance of to the best of our knowledge, the only algorithm that deploys network coding for structural health monitoring of bridges. We considered the power control technique to cope with the propagation environment effected by fading, shadowing and path loss; and assure the high probability of successful reception. We evaluate the power consumption of each node, and overall power consumption. We compared the result with the case when evaluated algorithm is not applied. We concluded that network coding should be carefully applied. Transmitting with the higher powers creates the overhearing opportunities and increases the network coding gain. However, in order to transmit the signal with high enough power to reach its second-hop direct-neighbour consumes too much of energy. Thus, we show that single-hop direct-connectivity is better in terms of the power consumption. Even though more messages should be transmitted, total power consumption in the case of one-hop direct-connectivity is higher than in the case of two-hop direct-connectivity.

We presented a method for optimum sensor nodes placement in the linear network-coding based networks that maximises the operational lifetime of the network. More specifically, we developed a numerical solution that calculates the optimum positioning of the sensors over a bridge given its length and required data throughput. Our numerical and simulation results indicate that the proposed method prolongs the network lifetime by up to 25% while at the same time it eliminates the bottlenecks.

We presented the scheduling method that supports our network coding based algorithm, reducing the maximum delay up to 48%. We analytically calculated the packet delay gain obtained by the use of network coding in the linear multi-hop wireless sensor network topologies. Moreover we calculated the exact packet delay (from the packet generation time to the time it is delivered to the sink nodes) as a function of the location of the source sensor node within the linear network. The derived packet delay distribution formulas have been verified by simulations and can provide a benchmark for the delay performance of linear sensor networks. Our results will be used for the design of more efficient network coding based scheduling and routing algorithms in the wireless sensor networks for Structural Health Monitoring (SHM) of linear structures.

We presented the adaptive algorithm based on network coding that improved end-to end delays and power consumption of nodes in a network. It increases the lifetime of network. This approach avoids all the overhead as nodes adapt to the conditions of the channel. In the case of losses in the channel, this protocol simply takes into consideration the loss instead of retransmitting the packet. The transmission slots are optimally used, and so is the limited internal battery.

9 FUTURE WORK

In the future work we plan to use the Wi-Fi technology as underlying 802.11TM. We plan to use a combination of FDMA and TDMA technologies. If we observe the Figure 3.7, any row (so far called a transmission cycle) would take place in one transmission slot while different nodes within that slot would use different frequencies. Let us observe the Figure 8.1.

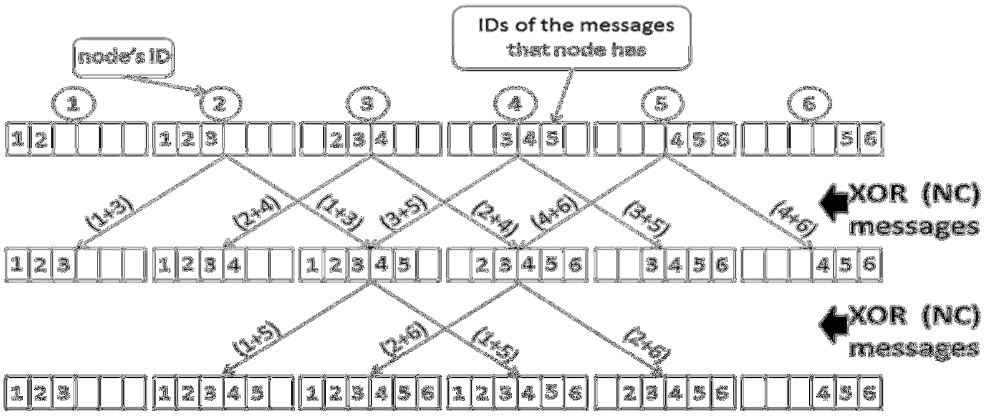


Fig 8.1. FDMA/TDMA network coding based algorithm for the single-line layout, 1-hop receiving case.

Nodes with the IDs 1, 2, 3, 4, 5 and 6 will be able to transmit their own messages (from their own sensors) at the same time using different frequencies. Frequencies nodes with the IDs 1, 3, 6 use for transmissions, nodes with the IDs 2, 4, 6 use for receptions. Frequencies nodes with IDs 2, 4, 6 use for transmissions nodes with IDs 1, 3, 6 use for receptions. In the next time slot (the next row of the rectangles), all nodes that received two messages in the previous time slot will transmit the XOR-ed combination of those messages. In the third time slot, nodes that received two messages in the previous time slot will transmit their XORed combination. This way, we expect a significant decrease in the delays in the network. We dedicated 20MHz for transmissions, and 20 MHz for receptions.

The only limitation is that 64 nodes can transmit at the same time without causing interference to each other. Thus, this approach we will use for the networks that deploy up to the 64 nodes. Sensors in the simulation access the medium within one time slot and they start sending their packets in different times (so the clock does not have to be synchronized). They each first send their pilot signals that are followed by data (as many symbols as we want).

REFERENCES

[1] C. R. Farrar, K. Worden, “An Introduction to Structural Health Monitoring”, *Philosophical Transactions of the Royal Society A*, 365, pp. 303-315, 2007.

[2] M. Celebi, “Seismic instrumentation of buildings (with emphasis on federal buildings)”, *Report No. 0-7460-68170, United States Geological Survey (USGS)*, Menlo Park, CA, 2002.

[3] P. Cheng, W. J. Shi, W. Zheng, “Large Structure Health Dynamic Monitoring Using GPS Technology”.

[4] Zhuo Liu, Bingwen Wang, Wenjun Yang, “Design and Deployment of Bridge Structural Health Monitoring System Based on Wireless Sensor Network”, *WiCOM’10*, pp. 1–4, Chengdu City, China, 2010.

[5] J. M. Caicedo, J. Marulanda, P. Thomson, S. J. Dyke, “Monitoring of Bridges to Detect Changes in Structural Health”, *in the Proc. of the 2001 American Control Conference*, Arlington, Virginia, 2001.

[6] S. N. Pakzad, G. L. Fenves, S. Kim, D. E. Culler, “Design and Implementation of Scalable Wireless Sensor Network for Structural Monitoring”, *Journal of Infrastructure Systems*, 14(1), 89-101, 2008.

[7] M. J. Whelan, K. D. Janoyan, “Design of a robust, high-rate wireless sensor network for static and dynamic structural monitoring”, *Journal of Intelligent Material Systems and Structures*, 20(7), 849-863, 2009.

[8] J. P. Lynch, Y. Wang, K. Loh, J. H. Yi, C. B. Yun, “Performance monitoring of the Geumdang Bridge using a dense network of high-resolution wireless sensors”, *Smart Materials and Structures*, 15(6), 1561-1575, 2006.

[9] J. P. Lynch, K. H. Law, A. S. Kiremidjian, E. Carryer, C. R. Farrar, H. Sohn, D. W. Allen, B. Nadler, J. R. Wait, “Design and performance validation of a wireless sensing unit for structural monitoring applications”, *Structural Engineering and Mechanics*, 17(3-4), 393-408, 2004.

[10] K.C. Lu, Y. Wang, J. P. Lynch, C. H. Loh, Y.J. Chen, P. Y. Lin, Z. K. Lee, “Ambient vibration study of the Gi-Lu cable-stay bridge: application of wireless sensing units”, *in Proc. of the SPIE – The International Society for Optical Engineering*, San Diego, CA, 2006.

[11] N. Kurata, B. F. Jr. Spencer, B.F., M. Ruiz-Sandoval, “Risk monitoring of buildings with wireless sensor networks”, *Structural Control Health Monitoring*, 12, 315–327, 2005.

[12] T. Kijewski-Correa, M. Haenggi, P. Antsaklis, “Wireless sensor networks for structural health monitoring: A multi-scale approach”, *In Proc. of the 2006 ASCE Structures Congress, 17th Analysis and Computation Specialty Conference*, pp. 1-16, 2006.

[13] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, J. Anderson, “Wireless Sensor Networks for Habitat Monitoring”, *In ACM International Workshop on Wireless Sensor Networks and Applications (WSNA ’02)*, 2002.

[14] J. Liu, J. Liu, J. Reich, P. Cheung, F. Zhao, “Distributed Group Management in Sensor Networks: Algorithms and Applications to Localization and Tracking”, *Telecommunication Systems*, 26(2):235–251, 2004.

[15] D. Estrin, A. Sayeed, M. Srivastava, “Wireless Sensor Networks”, *Tutorial at MobiCom’02*, 2002.

[16] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, M. Turon, “Health monitoring of civil infrastructures using wireless sensor networks”, *In Proc. of the 6th International Symposium on Information Processing in Sensor Networks*, pp. 254-263, 2007.

[17] J. P. Lynch, K. J. Loh, “A summary review of wireless sensors and sensor networks for structural health monitoring”, *The Shock and Vibration Digest*, 38(2):91-128, 2006.

[18] J. Paek , K. Chintalapudi , R. Govindan , J. Caffrey , S. Masri, “A wireless sensor network for structural health monitoring: performance and experience”, *In Proc. of the 2nd IEEE workshop on Embedded Networked Sensors*, pp. 1-9, 2005.

[19] Y. Lin, B. Liang, B. Li, “Passive Loss Interference in Wireless Sensor Networks based on Network Coding”, *In Proc. of IEEE INFOCOM*, 2005.

[20] G. Hartl, B. Li, “Loss inference in wireless sensor networks based on data aggregation”, *in Proc. of IPSN*, 2004.

[21] E. Goldoni, P. Gamba, “PRISM: a Novel Protocol for Real-Time Synchronous Acquisitions in WSNs”, *International Journal of Sensors, Wireless Communications and Control (JSWCC)*, Vol 1, Issue 2, 2012.

[22] A. G. Dimakis, P. B. Godfrey, M. J. Wainwright, K. Ramchandran, “Network coding for distributed storage systems”, *In Proc. of IEEE INFOCOM*, 2007.

[23] A. E. Kamal, A. Ramamoorthy, L. Long, S. Li, “Overlay protection against link failures using network coding”, *IEEE/ACM Transactions on Networking*, 19(4): 1071-1084, 2011.

[24] C. Fragouli, E. Soljanin, “Network coding fundamentals”, *Foundations and Trends in Networking*, Hanover, USA, 2(1): 1-133, 2007.

[25] B. Kasireddy, Y. Wang, J. Liu J, “Energy conservation using network coding in grid wireless sensor Networks”, *In Proc. 3rd International Workshop on Intelligent Systems and Applications (ISA)*, pp. 1-4, 2011.

[26] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, J. Crowcroft, “XORs in the air: practical wireless network coding”, *In Proc. of SIGCOMM’06*, Pisa, Italy, 2006.

[27] I. H. Hou, Y. E. Tsai, T. F. Abdelzaher, I. Gupta, “AdapCode: adaptive network coding for code updates in wireless sensor networks”, *In Proc. of IEEE INFOCOM*, Phoenix, Aariz, USA, 2008.

[28] O. Yilmaz, S. Demirci, Y. Kaymak, S. Ergun, A. Ildirim,” Shortest hop multipath algorithm for wireless sensor networks”, *Computer Communications*, 63, 48–59, 2012.

[29] R. Ahlswede, N. Cai, S. R. Li, R. W. Yeung, “Network information flow”, *IEEE Trans. on Information Theory*, 2000.

[30] S.-Y. R. Li, R. W. Yeung, N. Cai, “Linear network coding", *IEEE Transactions on Information Theory*, vol. 49, pp. 371-381, 2003.

[31] R. Koetter, M. Medard, “An Algebraic Approach to Network Coding", *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782–795, 2003.

[32] M. Nagajothy, S. Radha, “Network lifetime enhancement in wireless sensor network using network coding”, *International Conference on Control, Automation, Communication and Energy Conservation (INCACEC)*, pp. 1–4, 2009.

[33] Y. Hong, J. Xu, C. Jiang, “Lifetime Maximisation in Wireless Sensor Networks with Network Coding”, *International Conf. on Wireless Communications, Networking and Mobile Computing (WiCom)*, 2007.

[34] P.M. Glatz, K.B. Hein, R. Weiss, “Energy Conservation with Network Coding for Wireless Sensor Networks with Multiple Crossed Information Flows”, *10th International Symposium on Pervasive Systems, Algorithms, and Networks (ISPAN)*, 2009.

[35] Y. Wang, I.D. Henning, D.K. Hunter, “Efficient Information Exchange in Wireless Sensor Networks using Network Coding”, *Fourth Workshop on Network Coding, Theory and Applications (NetCod)*, 2008.

[36] P.M. Glatz, L.B. Hormann, C. Steger, R. Weiss, “Implementing autonomous network coding for wireless sensor network applications”, *18th International Conference on Telecommunications (ICT)*, 2011.

[37] M. Sikora, J. N. Laneman, M. Haenggi, D. J. Costello, Jr. T. Fuja, “On the Optimum Number of Hops in Linear Wireless Networks”, *Information Theory Workshop*, Texas, USA, 2004.

[38] Y. Zhang, L. Cheng, “Cross-Layer Optimisation for Sensor Networks”, *New York Metro Area Networking Workshop*, New York, 2003.

[39] C.-K. Toh, “Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks”, *IEEE Communications Magazine*, 2001.

[40] R. Cravets, P. Krishnan, “Power Management Techniques for Mobile Communication”, *In Proc. of MobiCom '98*, Dallas Texas USA, 1998.

[41] V. Shah-Mansour, V.W.S. Wong, “Link Loss Interference in Wireless Sensor Networks with Randomized Network Coding”, *IEEE GLOBECOM*, 2010.

[42] G. Buta, E. Coca, A. Graur, "Path Loss Exponent Influence on Distance Estimation between Wireless Sensor Nodes," *Advances in Electrical and Computer Engineering*, vol. 10, no. 1, pp. 110-115, 2010.

[43] F. Perez Fontan, P. Marino Espineira. Modelling the Wireless Propagation Channel: A Simulation Approach with Matlab, Wiley, 2008.

[44] M. Zorz, R.R. Rao, "Capture and Retransmission Control in Mobile Radio", *IEEE Journal on Selected Areas in Communications*, Vol. 24, No. 12, pp. 2341-2342, 1994.

[45] T.S. Rappaport. Wireless Communications - Principles and Practice, Prentice Hall, 2002.

[46] H. Suzuki , "A statistical model for urban radio propagation", *IEEE Trans. Communication*, COM-25, pp. 673-680, 1977.

[47] H. Hasemi, "Simulation of the rural radio propagation channel", *IEEE Trans. Veh. Technol.*, VT-28, pp. 213-224, 1979.

[48] C. E. Shannon, "Communication in the presence of noise", *In Proc. of the Institute of Radio Engineers*, vol. 37, pp. 10–21, 1949.

[49] K. Leentvaar, J. Flint, "The capture effect in FM receivers", *IEEE Transactions on Communications*, 24(5):531–539, 1976.

[50] M. Sikora, J. N. Laneman, M. Haenggi, D. J. Costello, T. E. Fuja, "Bandwidth and power efficient routing in linear wireless networks", *Joint Special Issue of IEEE Trans. Inform. Theory and IEEE Trans. Networking*, vol. 52, pp. 2624-2633, 2006.

[51] A. K. Rathore, A. K. S. Bhat, R. Oruganti, "A Comparison of Soft Switched DC-DC Converters for Fuel Cell to Utility Interface Application", *IEEJ Transactions on Industry Applications*, 128(4), pp. 450-458, 2008.

[52] K. H. Law, J. P. Lynch, Y. Wang, "Wireless Structural Sensors using Reliable Communication Protocols for Data Acquisition and Interrogation", *In Proc. of 42nd Allerton Conference on Communication, Control and Computing*, Allerton, IL, USA, 2004.

[53] Y. Wang, K. H. Law, J. P. Lynch, ”A Wireless Structural Health Monitoring System with Multithreaded Sensing Devices: Design and Validation“, *In Structure & Infrastructure Engineering: Maintenance, Management, Life-Cycle Design & Performance*, Florence, KY, USA, 2005.

[54] C. D. E. Lakeman, P. F. Fleig, J. L. DeGreeff, J. T. Trainor, “Hybrid power for wireless sensors”, *In Proceedings of SPIE 6231, Unattended Ground, Sea and Air Sensor Technologies and Applications* VIII, Bellingham, WA, USA, 2005.

[55] S. Blom, C. Bellettini, A. Sinigalliesi, L. Stabellini, M. Rossi, G. Mazzini, “Transmission power measurements for wireless sensor nodes and their relationship to the battery level”, *In 2nd International Symposium on Wireless Communication Systems*, pages 342–345, Siena, Italy, 2005.

[56] J. Yuan, W. Yu, “Joint source coding, routing and power allocation in wireless sensor networks”, *IEEE. Trans. On Communications*, 2005.

[57] X. Liu, J. Cao, S. Lai, C. Yang, H. Wu, Y.L. Xu, “Energy Efficient Clustering for WSN-based Structural Health Monitoring”, *In Proc. of IEEE INFOCOM*, 2011.

[58] K. P. Ferentinos, T. A. Tsiligiridis, “Adaptive design optimisation of wireless sensor networks using genetic algorithms”, *Computer Networks*, vol. 51, no. 4, pp. 1031–1051, 2007.

[59] D. B. Jourdan, L. de Weck, “Layout optimisation for a wireless sensor network using a multi-objective genetic algorithm”, *in Proc. of IEEE VTC’04*, 2004.

[60] Y. Zou, K. Chakrabarty, “Sensor deployment and target localization based on virtual forces,” *INFOCOM ‘03*, 2003.

[61] T. Fu, A. Ghosh, E. Johnson, B. Krishnamachari, "Energy- efficient deployment strategies in structural health monitoring using wireless sensor networks," *Structural Control and Health Monitoring*, 2011.

[62] H. Wang, Y. Yang, M. Ma, J. Xe, X. Wang, "Network lifetime maximisation with cross-layer design in wireless sensor networks", *IEEE Trans. Wireless Communications*, vol. 7, no. 10, pp. 3759-3768, 2008.

[63] Y. E. Sagduyu, A. Ephremides, “On joint MAC and network coding in wireless Ad Hoc networks”, *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3697-3713, 2007.

[64] S.C. Ergen, P. Varaiya, “TDMA scheduling algorithms for wireless sensor networks”, *Wireless Networks*, vol. 16, pp. 985-997, 2010.

[65] M. O. Diaz, K. K. Leung, “Randomized Scheduling Algorithm for Data Aggregation in Wireless Sensor Networks”, *European Wireless Conference*, 2010.

[66] I. Amdouni, R. Soua, E. Livolant, P. Minet, “Delay optimised time slot assignment for data gathering applications in wireless sensor networks”, *ICWCUCA*’12, France,2012.

[67] S. Gandham, Y. Zhang, Q. Huang, “Distributed time-optimal scheduling for converge cast in wireless sensor networks”, *Computer Networks*, vol. 52, pp. 610-629, 2008.

[68] W. Yu, J. Yuan, “Joint source coding, routing, and resource allocation for wireless sensor networks” *In Proc. IEEE ICC*, 2005.

[69] H. Y. Shwe, X. Jiang, S. Horiguchi, ”Efficient network coding for power saving in wireless sensor networks”, *15th Asia-Pacific Conference on Communications, APCC 2009*, Page(s): 36 – 39, Shanghai, 2009.

[70] J. Skulic, K.K. Leung, “Application of Network Coding in Wireless Sensor Networks for Bridge Monitoring,” PIMRC 2012, Australia, 2012.

[71] J. Skulic, K. K. Leung, “Monitoring of bridge using a wireless sensor network based on network coding”, *IABMAS 2012*, Italy, 2012.

[72] J. Skulic, A. Gkelias, K. K. Leung, "Node Placement in Linear Wireless Sensor Networks ", *EUSIPCO 2013*, Morocco, 2013.

[73] J. Skulic, A. Gkelias, K. K. Leung, "Delay Analysis of Network Coding in Linear Wireless Sensor Network", *ICWiSe'13*, Malaysia, 2013.

[74] F. A. Tobagi, L. Kleinrock, "Packet Switching in Radio Channels: Part II - The Hidden Terminal Problem in Carrier Sensing Multiple Access and Busy Tone Solution", *In IEEE Trans. on Commun.*, volume 23. IEEE, 1975.

[75] J. Skulic, A. Gkelias, K.K. Leung, "Adaptive Network Coding Based Cross layer Protocol for Wireless Sensor Networks", to be submitted.

[76] D. Gay, P. Levis, R. Von Behren, M. Welsh, E. Brewer and D. Culler, NesC language: A holistic approach to network embedded systems, *Proc. Of the ACM SIGPLAN*, 2003.

[77] P. Levis, S. Mdden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer and D. Culler, TinyOS: An Operating System for Wireless Sensor Networks *In Ambient Intelligence. New York: Springer-Verlag*, 2005.

[78] R. Kiviluoma, G. Feltrin, J. Meyer, T. Aho, A. Kilpela, V. Lyori, , Sb -8.2 demonstration of bridge monitoring, *Prepared by Sustainable Bridges, FP6 EU Project*, 2007.

[79] BluetoothTM. Specificaton of the Bluetooth System; *IEEE: Piscataway*, New Jersey, 2004.

[80] M. Win, R. Scholtz, Impulse radio: How it works. *IEEE Communications Letters*, 36–38, 1998.

[81] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, M. Win, Ranging with ultra wide bandwidth signals in multipath environments. *Proc. IEEE 2009*, 97, 404–426.

[82] IEEE 802.15.4a Standard. Part 15.4: Wireless MAC and PHY Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs): Amendment to add alternate PHY (Draft). *IEEE: Piscataway*, NJ, USA, 2006.

[83] C. Buratti, R. Verdone, Performance analysis of IEEE 802.15.4 non-beacon enabled mode. *IEEE Trans. Vehicular Technology*, 4096, 884–893, 2009.

[84] C. Buratti, A Mathematical Model for Performance of IEEE 802.15.4 Beacon-Enabled Mode. In Proceedings of IEEE IWCMC, Leipzig, Germany, 2009.

[85] Aliance, Z. Zigbee Specifications. Zigbee Standard Organisation: San Ramon, CA, USA, 2008.

[86] C. Buratti, A. Conti, D. Dardari, and R. Verdone, “An overview on wireless sensor networks technology and evolution,” *Sensors, vol. 9, no. 9, pp. 6869–6896*, 2009.