M 2015

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

SENSOR NETWORKS POWERED BY SOLAR ENERGY USING MULTIPLE RADIO CHANNELS

XAVIER DA SILVA ARAÚJO
DISSERTAÇÃO DE MESTRADO APRESENTADA
À FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO EM
ÁREA CIENTÍFICA

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Sensor networks powered by solar energy using multiple radio channels

Xavier da Silva Araújo

FOR JURY EVALUATION

MESTRADO INTEGRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

Supervisor: Doutora Tânia Cláudia dos Santos Pinto Calçada

Co-Supervisor: Professor Doutor Manuel Alberto Pereira Ricardo

July 30, 2015

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

MIEEC - MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES **2014/2015**

A Dissertação intitulada

"Sensor Networks Powered by Solar Energy Using Multiple Radio Channels"

foi aprovada em provas realizadas em 24-07-2015

o júri Ricardo Santos Morla

Presidente Professor Doutor Ricardo Santos Morla
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Susana Isabel B. M. Sargento

Professora Doutora Susana Isabel Barreto de Miranda Sargento
Professora Associada c/ Agregação do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Tânia Cláudia dos Santos Pinto Calçada

Doutora Tânia Cláudia dos Santos Pinto Calçada
Investigadora Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.

Xavier Silva Araújo

Autor - Xavier da Silva Araújo

Faculdade de Engenharia da Universidade do Porto

Abstract

This dissertation addresses the problems of capacity and lifetime of Wireless Multimedia Sensor Networks (WMSNs). More specifically, it focuses on low-cost WMSNs, powered by solar energy and based on Wi-Fi, for video-surveillance purposes. To solve the problems related with the capacity and the lifetime of these networks, this dissertation adopts a multi-channel approach. After the analysis of the state of the art solutions, it was concluded that this multi-channel approach allows to increase the capacity of the network, by reducing its levels of interference, and also allows to extend its lifetime, by using a proper channel assignment algorithm, capable of avoiding traffic forwarding through nodes with low energy level.

Thus, this dissertation proposes eTILIA, a dynamic, centralized and energy-aware channel assignment algorithm for WMSNs, capable of extending their lifetime. This algorithm selects the best channel to every node in the network, using solely the information regarding the network topology and regarding the energy level of the nodes. Experimental evaluation through NS-3 simulations shows that eTILIA can provide a more efficient management of the energy resources of random topologies of 16 nodes, when compared to TILIA channel assignment algorithm and to a random channel assignment procedure. This experimental evaluation also shows that eTILIA can slightly increase the network lifetime of random topologies of 36 nodes, when compared to TILIA channel assignment algorithm, and can provide a more efficient management of the energy resources of random topologies of 36 nodes, when compared to a random channel assignment procedure.

This dissertation also proposes an architecture for a low-cost video-surveillance system, powered by solar energy and based on Wi-Fi. This proposal is based on the WMSN concept, and covers the physical components to be used, the protocol stack to be adopted, and the mode of operation of the network.

Resumo

Esta tese aborda os problemas relacionados com a capacidade e o tempo de vida das Redes de Sensores Multimédia Sem Fios (WMSNs - Wireless Multimedia Sensor Networks). Mais especificamente, foca-se em WMSN de baixo custo, alimentadas a energia solar e baseadas em Wi-Fi, para fins de video-vigilância. Para resolver os problemas relacionados com a capacidade e o tempo de vida destas redes, esta tese adopta uma abordagem multi-canal. Após a análise das soluções do estado da arte, concluiu-se que esta abordagem multi-canal permite aumentar a capacidade da rede, ao reduzir os seus níveis de interferência, e pode também prolongar o seu tempo de vida, ao usar um algoritmo de atribuição de canais apropriado, capaz de evitar o encaminhamento de tráfego através de nós com nível de energia reduzido.

Assim, esta tese propõe eTILIA, um algoritmo de atribuição de canais dinâmico, centralizado e energeticamente eficiente para WMSNs, capaz de prolongar os seus tempos de vida. Este algoritmo selecciona o melhor canal para cada nó da rede, utilizando apenas a informação sobre a topologia da rede e sobre o nível de energia dos nós. Avaliação experimental através de simulações em NS-3 mostram que o eTILIA consegue providenciar uma gestão mais eficiente dos recursos energéticos de topologias aleatórias com 16 nós, quando comparado com o algoritmo de atribuição de canais TILIA e com um procedimento de atribuição de canais aleatório. Esta avaliação experimental também mostra que o eTILIA consegue aumentar superficialmente o tempo de vida da rede de topologias aleatórias com 36 nós, quando comparado com o algoritmo de atribuição de canais TILIA, e consegue providenciar uma gestão mais eficiente dos recursos energéticos de topologias aleatórias com 36 nós, quando comparado com um procedimento de atribuição de canais aleatório.

Esta tese também propõe uma arquitectura para um sistema de video-vigilância de baixo custo, alimentado a energia solar e baseado em Wi-Fi. Esta proposta é baseada no conceito de WMSN, e cobre os componentes físicos a serem usados, a pilha protocolar a ser adoptada, e o modo de operação da rede.

Acknowledgments

First of all I would like to thank Dra. Tânia Calçada for all her support and guidance. Throughout this dissertation, she was capable of aiding me during the most difficult times. Her availability and willingness to hear me and support me, were essential for the success of this dissertation. I could not have chosen a better supervisor.

Secondly, I want to thank Prof. Manuel Ricardo who was always available to clarify my doubts about this dissertation. I also want to thank Sérgio Tafula for all his support regarding the Roundabout virtual machine, and to Agostinho Ferraz for his support regarding the components of the proposed system.

I want to thank my father, Manuel Araújo, and my mother, Ana Araújo. Throughout my whole life they had been there whenever I needed, and for that I thank them. They provided all the required tools for me to succeed. Their love, support, encouragement and inspiration were crucial. I also want to thank to my family which was always by my side, and contributed to keep me motivated.

I want to thank to Álvaro Ferreira, Catarina Santos, Cátia Silva, Elisabete Gonçalves, Fábio Silva, Luciano Duarte, Luís Úngaro, Patrícia Oliveira, Sónia Coelho and Tiago Teixeira. They were with me in the good times, and more important, they were with me in the bad ones. I can proudly say that they are truly my friends. All the moments that we had together, during all these years, were essential to my success. To them, I leave this: *"Never forget the days I spent with you. Continue to be my friend, as you will always find me yours"*, Ludwig van Beethoven

Last but not least, I thank to Beatriz Andrade for all her support. She stayed always by my side during this dissertation, and gave me the strength and motivation to finish it.

Thank you all! This dissertation is dedicated to you!

Xavier da Silva Araújo

"Tired of lying in the sunshine
Staying home to watch the rain
You are young, and life is long
And there is time to kill today
And then one day, you find
Ten years have got behind you
No one told you when to run
You missed the starting gun"

Pink Floyd - Time

Contents

List of Figures

List of Tables

Abbreviations

CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access Collision Avoidance
CSI	Camera Serial Interface
EMRP	Energy-Aware Mesh Routing Protocol
ETE	Expected Transmission Energy
HDMI	High-Definition Multimedia Interface
HWMP	Hybrid Wireless Mesh Protocol
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
kbps	Kilobits per Second
LB-MCP	Load Balancing Multi Channel Protocol
Mbps	Megabits per Second
MIMO	Multiple Input Multiple Output
NS-3	Network Simulator 3
OFDM	Orthogonal Frequency Division Multiplexing
QoS	Quality of Service
SSCH	Slotted Seeded Channel Hopping
TCP	Transmission Control Protocol
TDCC	Topology Discovery and Channel Change
TDMA	Time Division Multiple Access
TH-UWB	Time-Hopping Ultra-Wide Band
UDP	User Datagram Protocol
USB	Universal Serial Bus
WMN	Wireless Mesh Network
WMNs	Wireless Mesh Networks
WMSN	Wireless Multimedia Sensor Network
WMSNs	Wireless Multimedia Sensor Networks
WSN	Wireless Sensor Network
WSNs	Wireless Sensor Networks
μ	Expected Value
σ^2	Variance

Chapter 1

Introduction

1.1 Context

Future Cities is a project that is currently being implemented in the city of Porto, in Portugal. The main idea of this project is to transform the city of Porto in an urban-scale living lab, where new technologies, services and products, can be developed, tested and evaluated. These technologies, services and products can explore several subjects, such as sustainable mobility, urban-scale sensing or even the quality of life of the citizens.

One of the challenges of this project is the creation of low-cost, solar powered video-surveillance systems, based in Wi-Fi, to cover large unconnected areas such as parks or beaches. These systems fit in the definition of Wireless Multimedia Sensor Network (WMSN). WMSNs are networks of interconnected wireless devices, that allow to retrieve multimedia content, like video streams, audio streams or static images. WMSNs are a recent technology that emerged from Wireless Sensor Networks (WSN), which in its turn emerged from Wireless Mesh Networks (WMN). As stated in [1], these networks can be useful in numerous applications such as person locater systems, traffic avoidance systems, control systems, environmental monitoring systems and surveillance systems.

1.2 Problem Characterization

The goal of this thesis is the designing of a networking solution to improve the performance of Carrier Sense Multiple Access (CSMA) based WMSNs, regarding two major factors: the network capacity and the network lifetime. The network capacity is a major limitation because the WMSN has to be capable of transmitting large amounts of data, extracted from the video-surveillance system, to the appropriate destination, within a maximum pre-defined delay time. Thus, the network capacity can clearly limit the network performance, if not taken into account. The network lifetime is also a major limitation, since the deployment of a WMSN powered by solar energy, can greatly reduce the network lifetime, restraining the network operation.

It is essential to come up with solutions to overcome these limitations, in order to deploy functional, reliable and effective WMSNs. This is important since this emerging type of network can bring great advantages, as stated in [1]. This work focuses on the designing of a single solution that, at the same time, increases the network capacity and extends the network lifetime. By doing this, it is possible to improve the performance of WMSNs, allowing them to transmit a higher amount of multimedia data, within a maximum pre-defined delay time, and increasing the period in which they can collect information. This allows the deployment of WMSNs with better characteristics in terms of performance and reliability.

There are several strategies to increase the capacity of the network, and to extend its lifetime. This thesis adopts a multi-channel approach. With this approach, it is possible to increase the capacity of the network, by reducing its interference levels, and it is also possible to extend the network lifetime, by using a proper dynamic channel assignment procedure, capable of avoiding traffic forwarding through nodes with low energy level. Thus, this thesis proposes a centralized and energy-aware channel assignment algorithm, which takes into account the energy level of the nodes of the WMSNs. The reason behind the centralized approach has to do with the low processing and memory resources, owned by the WMSN nodes.

The energy-aware channel assignment algorithm developed is an improvement of the channel assignment algorithm named TILIA, described in [2]. The proposed algorithm, eTILIA, adds an energy parameter to the TILIA algorithm, in order to adapt it to situations where energy availability is a strong limitation.

1.3 Contributions

The main contribution of this thesis is eTILIA, a dynamic, centralized and energy-aware channel assignment algorithm for WMSNs, capable of extending their lifetime. This algorithm take as input the network graph, with all the gateways and all the connections between the nodes of the network, and the energy level of each node. The output of the algorithm is the channel assigned to each node. The evaluation of eTILIA was based on computer simulations, using the Network Simulator 3 (NS-3) tool. These simulations relied mainly on the mesh network model and energy model of NS-3.

A secondary contribution is the proposal of a video surveillance system, based on the WMSN concept. This proposal encompasses the components of the system, the protocol stack adopted and its mode of operation.

1.4 Structure

This document is organized in four more chapters. Chapter 2 presents the state-of-the-art solutions to increase the network capacity, and to achieve energy efficiency in wireless multi-hop networks. Chapter 3 presents the proposed video surveillance system, based on the WMSN concept, and the proposed eTILIA algorithm. This chapter starts by giving an overview of the system and a description of its components, protocol stack and mode of operation, and then presents the specification and implementation of eTILIA. Chapter 4 describes the simulation tools which were used, the topologies simulated, the simulation methodology adopted and the results obtained. Chapter 5 presents the final conclusion of this dissertation.

Chapter 2

State-of-the-Art

The first section of this chapter, presents a set of procedures that can be used to increase the capacity of wireless multi-hop networks. The second section of this chapter exposes a set of approaches proposed to save the energy resources of wireless multi-hop networks. This chapter concludes with an analysis relating the solutions capable of increasing the network capacity, with the energy saving solutions.

2.1 Capacity of Wireless Multi-Hop Networks

The retrieving of multimedia data normally generates large volumes of data. This requires a high capacity network, capable of transmitting all the retrieved information to its destination, within a certain time interval. The network capacity depends on many factors, such as the network architecture, network topology, traffic patterns, network node density, number of communication channels used for each node, transmission power level and node mobility, as stated in [3]. Since there are many factors that influence the network capacity, there are also several approaches designed to increase it. This section presents the relevant approaches and solutions, regarding this topic. A special attention is given to TILIA algorithm due to its importance on this dissertation.

2.1.1 Capacity Improvement Approaches

There are numerous ways to improve the capacity of wireless multi-hop networks, being the following the most adopted approaches:

Reduce Interference: In order to increase the network capacity is possible to design solutions to reduce, or to completely avoid, the interference between simultaneous communications. By doing this, the number of possible simultaneous communications increases, allowing to achieve a higher network capacity. For instance, in [4] is proposed a routing metric, designated iAWARE, which aims to reduce the interference in the network, increasing its throughput. In [5] the authors propose a solution that, based on a channel assignment procedure, modifies the network topology to minimize the interference, increasing the throughput and the QoS of the network.

Design Routing Protocols/Metrics: It is possible to take advantage from routing protocols and metrics to achieve a higher network capacity. In [6] is studied the performance gain, in terms of throughput, obtained by making routing decisions with the awareness of network coding. In [7] is proposed a set of metrics to enable the routing protocol to find paths with low levels of interference, reliability in terms of packet success rate, and high available transmission rate.

Using Multiple Communication Channels: One of the most common approaches to increase the network capacity is to use multiple communication channels in the same wireless multi-hop network. This approach enables to have a higher amount of simultaneous communications, which substantially increases the overall throughput of the multi-hop network. This type of approach is referred in [8] and in [9].

Using Multiple Network Interfaces: Using multiple networks interfaces can also be used to increase the capacity of a wireless multi-hop network. By combining this approach with the multiple channel approach, it is possible to achieve a much more better performance in terms of capacity. In [10] is specified a channel assignment algorithm, to be used in multi-radio WMNs, that avoids interference by trying to assign non-overlapping channels to nodes which are near from each other. In [11] is proposed a network model for analysing the capacity of multi-radio and multi-channel WMNs.

2.1.2 Capacity Improvement Solutions

The combination of the approaches above described, led several researchers to come up with concrete solutions to enhance the capacity of wireless multi-hop networks. In [12] is proposed a link-layer protocol, named SSCH, that uses frequency diversity, with orthogonal channels, to increase the network capacity of the IEEE 802.11 standard. The idea of the protocol is to switch the channels of the nodes that want to establish communication, in order to overlap them. At the same time, the protocol avoids to interfere with the nodes that are not interested in that particular communication, by assign them non-overlapping channels. To allow the communication between all the neighbours, each node has a frequency hoping pattern which is regularly broadcasted. SSCH can be applied in both single-hop and multi-hop wireless networks, and requires only a single radio interface per node.

In [13] is theoretically demonstrated that is feasible to increase substantially the capacity of interference-limited wireless networks, by using antenna spatial diversity (multiple antennas) together with optimum combining. By making use of multiple antennas in the same communication, it is possible to improve the reliability of a wireless link, because, even if one the antennas receive a weak signal, it is likely that one of the other antennas receives the signal in good conditions.

With optimum combining is possible to join all the different received signals and obtain a reliable representation of the original signal.

In [14] the authors exposed a solution to increase the throughput of wireless networks, based on a radio technology designated Pulsed Time-Hopping Ultra-Wide Band. With this technology is possible to strictly limit the radiated power, without sacrificing the acceptable data rate required. Instead of using protocols like CSMA/CA or TDMA to manage interference and multiple-access, this solution adopts a rate control strategy. By taking advantage of the pulse nature of TH-UWB is possible to reduce the impact caused by the interferences, significantly increasing the network throughput.

In [15] is demonstrated that by introducing one dimensional mobility in the nodes of some ad-hoc networks, is possible to significantly improve the network capacity. The idea that the mobility of the nodes can enhance the capacity of the network is also present in [16].

In [17] is disclosed a routing protocol, designated LB-MCP, to be used in wireless multi-hop networks, which aims to extend infrastructure networks that own several access points. The wireless multi-hop network must have a multi-channel architecture, and each node of the network must have only one network interface. This routing protocol tries to balance the traffic load in each one of the network channels, enhancing their utilization and hence increasing the capacity of the network. Each node discover several routes to the access points, and choose the one that originates a more balanced traffic load, maintaining all the other routes for backup purposes.

In [18] is stated the fact that the use of multiple channels in a wireless network improves the network capacity, and is presented a routing protocol specified to multi-hop networks, with multiple channels and multiple interfaces in each node, and an algorithm to assign the channels to the nodes interfaces. The use of multiple channels in order to increase the capacity is also present in the standard 802.11a, which offers 12 non-overlapping channels as is described in [19]. In [20] is proposed a multi-channel WMN architecture, designated Hyacinth, that equips each one of the WMN nodes with multiple 802.11 network interface cards. Together with the architecture of the network, the authors presents also a distributed channel assignment and routing algorithm, which uses only the local traffic load information to dynamically assign channels and route packets. In [21] is exposed a link layer protocol and a routing protocol for increasing the capacity in multi-channel networks. The link layer protocol was designed to be implemented over 802.11 hardware, and the routing protocol was designed to be used in multi-channel and multi-interface wireless networks. In [22] is stated that if the number of network interfaces on the nodes is smaller than the number of available channels, there will be a degradation in the network capacity in many scenarios.

2.1.3 TILIA Algorithm

This sub-section focuses on the TILIA algorithm, described in [2]. It starts by giving a detailed description of the mode of operation of TILIA, and then presents a detailed description of its implementation.

2.1.3.1 TILIA Description

TILIA is a centralized channel assignment algorithm for single radio WMNs, which tries to improve the performance of multi-channel single radio WMNs, by assigning the best channel to each node, using solely the network topology information. It adopts a centralized approach because WMNs are normally formed by low cost nodes, with small memory resources and reduced processing capability. So the main goal of TILIA is to centrally assign the channels in which the WMNs nodes will operate, optimizing the network performance and avoiding to disconnect it. To do this, TILIA uses a breadth-first tree growing technique, but instead of growing a single tree, it grows a forest, which is composed by several trees rooted at each gateway. Each tree operates in a different radio channel, avoiding interference between them. The growth of the trees is simultaneous and their union spans the network. In Table 2.1 is presented the meaning of some terms used to describe TILIA.

Term	**Meaning**
Gateway	Special node of the network which is the destination of most of the traffic generated by the other nodes. It is usually connected to an infra-structured network.
Tree	Set of nodes that communicate with the same gateway, using the same communication channel.
Forest	Set of trees that span the WMN.
Parent	Next or previous hop in the path between a node and the gateway, for upstream or downstream traffic, respectively.
Tree Load	Assuming that the total traffic of each node is constant (λ), the tree load is given by $\sum_{v \in Vgi} \lambda d(v,gi)$, where v represents a node, Vgi represent the set of nodes that are attached to the tree i and $d(v,gi)$ represents the hop count between the node v and its gateway gi. Since the total traffic of each node is assumed to be constant, it is possible to remove this parameter from the tree load expression: $\sum_{v \in Vgi} d(v,gi)$.
1st Ring	Set of nodes which are only one hop count away from the gateway.

Table 2.1: TILIA terms

TILIA requires, as input, the network graph, with all the nodes and links between them, and the location of the existent gateways. Given the required input, it starts by initialize a tree in each one of the existent gateways, and analyses the nodes, one by one, attaching them to the best tree. Every time a node is attached to a certain tree, TILIA carefully selects the next node to be analysed. First the algorithm chooses the nodes which are neighbours from a previously attached node, and selects the ones with the lower hop count to the closest gateway. From these nodes it picks the nodes which have the smallest number of nearby channels and then it picks the nodes which have the smallest number of nearby parents. Then it selects the nodes with the lower number of hidden links and randomly chooses one of these nodes. For the selected node TILIA determines the most appropriate channels. In order to do this it finds the channels which were previously assigned to the neighbours of the selected node, and selects the ones that belong to the trees with the lower traffic load. After that, based on the set of channels selected, TILIA determines the best parent candidates, for the selected node. To do this it starts by selecting the neighbours operating in one of the channels of the set. Then it chooses the ones with the lower hop count to their respective gateway, and, from this set, picks the ones which are attached to the trees with the smallest traffic load. Finally, from the set of candidate parents obtained, are picked up the ones that present less problems due to hidden nodes. If after this selection only remains a single candidate parent, the node is attached to the tree of this candidate and selects him as his parent. If the set of candidate parents is greater than one but there is only one parent in the first ring, the node selects him as his parent. If the set of candidate parents is greater than one, and there isn't any candidate in the first ring, the node randomly selects one of the candidates to be his parent. If the set of candidate parents in the first ring is greater than one, the TILIA algorithm employs a recursive procedure to explore all the possible alternative forests. By using recursion, the TILIA algorithm allows to create alternative forests, and then selects the forest that present the best characteristics, according to a certain metric. This is done because after several computer simulations it was discovered that the network topology near the gateways, had great impact in the overall performance of the network. This procedure is repeated for every node in the network until all the nodes have been assigned with a channel, and belong to a certain tree. In the end, the TILIA algorithm uses a pre-defined metric, denominated *tmet*, to determine which forest leads to a better network performance. This metric is exposed in Equation 2.1.

$$tmet: \quad \theta = k_l\theta_l + k_lb\theta_lb + k_r1\theta_r1 + k_r1b\theta_r1b + k_m\theta_m \tag{2.1}$$

The *tmet* metric is composed by five components that enable to measure the performance of the forest. These five components are:

Total Load (θ_l): This component represents the ratio between the minimum load of the network and the sum of all the loads of the trees that constitute the forest. In the best case $\theta_l = 1$ and this occurs when the load on the forest equals the minimum load of the network.

Total Load Balancing (θ_lb): This component represents the fairness of the load distribution among all the trees that constitute the forest. In the best case $\theta_lb = 1$ and this occurs when all the trees have exactly the same load.

Total Number of 1st Ring Nodes (θ_r1): This component represents the ratio between the sum of the connectivity degree of the trees and the size of the set that contain the gateways neighbours of the original network. In the best case $\theta_r1 = 1$ and this occurs when all the nodes in the neighbourhood of gateways of the original network are assigned to one of their closest gateways.

1st Ring Balancing (θ_r1b): This component represents the fairness of the distribution of 1st ring nodes among the gateways. In the best case $\theta_r1b = 1$ and this occurs when the sizes of the 1st ring of each tree are equal.

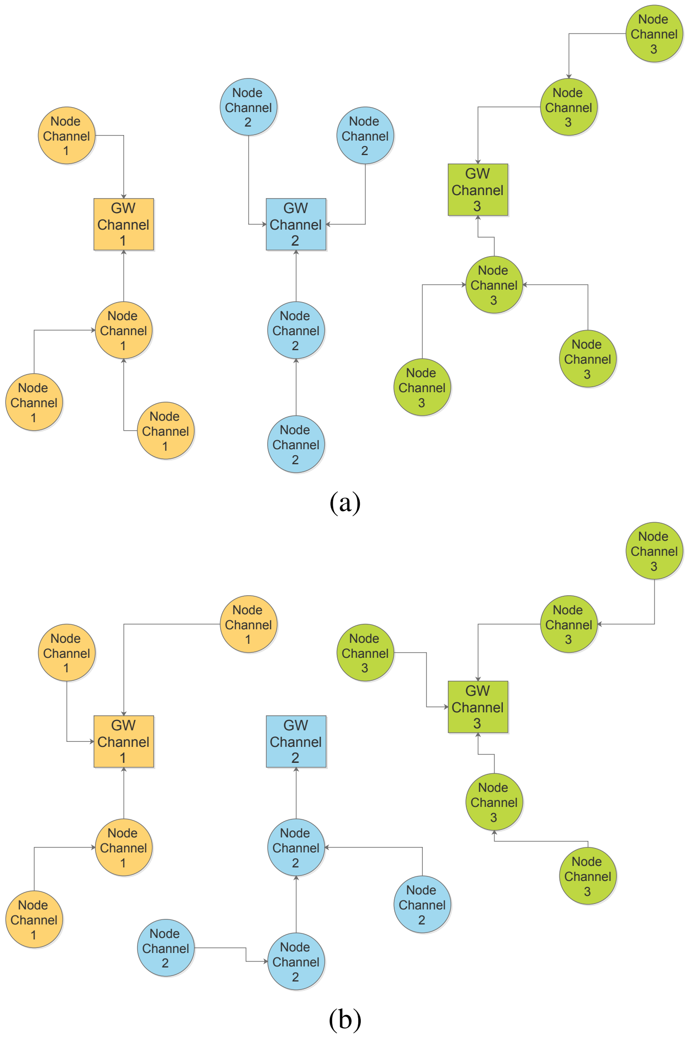


Figure 2.1: TILIA Forests

1st Ring Miss Ratio (θ_m): This component measures the hidden node problem on the gateways neighbourhood. In the best case $\theta_r m = 1$ and this occurs when there are no hidden nodes in the gateways neighbourhood.

To each component is attached a weight, which represents the importance of that particular component to the network performance. The higher the weight the more important is the component associated. So, given all the possible forests, TILIA algorithm selects the one with the higher *tmet* metric, since that forest is probably the one that achieves greater performance. In Figure 2.1 (a) is presented a possible forest, with the respective trees, for a WMN with three gateways and nine nodes. In Figure 2.1 (b) is presented an alternative forest, that result from the TILIA recursive procedure, for the same WMN.

It is important to notice that since TILIA algorithm forces each node to choose a parent, it can be classified as a joint channel assignment and routing procedure, because every node can reach the gateway through its parent node.

2.1.3.2 TILIA Implementation

The implementation of TILIA was made using the Python scripting language. The functions that compose the TILIA script are now exposed and explained:

***main()*:** Read the name of the *.dot* file from the command line arguments. This is the file which contains the information about the topology to be analysed. On this file are presented the positions of all the nodes, the connections between them, and are also identified the nodes which act as gateways. Then it calls the function *assign_channels()*, using the name of the *.dot* file as the function argument.

***assign_channels()*:** Creates a graph of the network using the Pygraphviz tool, referred in [23], and reorders the nodes of the graph, putting the gateways on the first positions of the graph. Reads the information contained in the *.dot* file and initializes some parameters of the nodes. Initializes the global variable *COLORS*, which assigns a colour to each channel, through the execution of the function *channels()*. Specifies the ring of each one of the nodes through the execution of the function *assign_rings_to_nodes(G)*. It then calls the channel assignment function *tilia()* and prints the result of the assignment to a *.tcl* file through the execution of the function *printBestSchemeToFile()*.

***channels()*:** Assigns a colour to each communication channel used, through the global variable *COLORS*, and return a list containing all the existent channels. This is useful to draw a graphical representation of the network after the channel assignment procedure, using for example the *neato* tool.

***assign_rings_to_nodes()*:** Specifies the ring to which each node belongs. A node which is one hop away from the gateway is assigned to the first ring, a node which is two hops away from the gateway is assigned to the second ring, and so on.

***printBestSchemeToFile()*:** Prints the result of the channel assignment algorithm to a *.tcl* file. This file contains the information about the channel of each node, its position and its download and upload parent. It also contains information about the load of each tree after the assignment.

***draw_graph()*:** This function draws the network graph, after the channel assignment procedure, in a *.pdf* and *.png* file using the *neato* tool.

***tilia()*:** This function obtains all the possible alternative trees for the network in analysis, through the execution of the function *recursive_tilia()*, and calculates the *tmet* metric for each one of them. Then it returns the one with the best metric.

***recursive_tilia()*:** This function implements the most part of the TILIA algorithm. It starts by discovering which nodes have not yet a channel assigned, and reorders these nodes using the function *reorderNodeList()*. Then it chooses the best candidate channels and the best candidate parents to that node. It uses recursion if there are multiple candidate parents in the first ring. This way it allows to generate several alternative trees which are then evaluated through the *tmet* metric, referred in Equation 2.1.

***assign_parent_to_node()*:** Establishes the connection between a node and the node which was elected its parent.

***reorderNodeList()*:** Reorders the nodes of the network. It starts by putting the nodes with the lowest hop count to the gateway first. Then it organizes the nodes with the same hop count by the lowest number of gateways choices. After this, it organizes the nodes with the same number of gateway choices by the lowest number of parent choices. Finally it organizes the nodes with the same number of parents choices by the lowest number of hidden links.

***min_up_hidden_nodes()*:** Returns the candidate parents which present less problems with hidden nodes. This is useful to select the best candidate parents.

***remove_edges_between_channels()*:** After the channel assignment procedure, this function removes the graph links between nodes with different communication channels.

***nodes_ring()*:** Returns the nodes which are in a given ring.

***check_children_alternatives()*:** Returns the number of neighbours with a lower hop count to the gateways. Depending on the parameter *samechan* it counts only the nodes in the same channel or it counts all the nodes with a lower hop count to the gateway independently of their channel.

***node_comitted_load()*:** Returns the load of a certain node due to its descendants.

***min_load_gw()*:** Returns the gateway which will have a lower load after attaching a certain node.

***fairness()*:** Calculates the Jain fairness index, referred in [24], of a certain sequence.

***tilia_fix_paths()*:** This method is executed after the channel assignment procedure. It analyses the trees that were built and checks if there is any possibility of improving the load balance between them.

***nodes_on_channel()*:** Returns the number of nodes which were assigned to a given channel.

***hidden_nodes_of_a_link()*:** Returns the number of hidden nodes of a given graph link.

***hidden_nodes_ratioR1()*:** Returns the miss ratio on the gateways neighbourhood which is a measure of the hidden node problem.

2.2 Lifetime of Wireless Multi-Hop Networks

In order to deploy wireless multi-hop networks it is necessary to have an energy source to power the nodes, since the capacity of the existing batteries has strong limitations. If the network is deployed in an outdoor environment, the best option to obtain the necessary power for the network operation is to take advantage from solar energy. However, the uncertainty related to the solar energy availability can be a major problem, and so it is necessary to assure that the network does not breakdown during scenarios of low availability of solar energy. In this section are presented some of the existent solutions to extend the lifetime of wireless multi-hop networks.

2.2.1 Lifetime Extension Approaches

To decrease the energy consumption of wireless multi-hop networks is possible to modify several aspects of their functioning. As stated in [25] the main energy-efficient mechanisms are:

Radio Optimisation: The radio is the component which mostly affects the battery depletion of the nodes in wireless multi-hop networks. So by optimising the radio parameters, a great improvement can be obtained, in terms of energy efficiency. It is feasible to do this by optimising the radio modulation, transmission power, the type of antennas or by adopting an cooperative communication scheme. In [26] is showed that is possible to minimize the energy consumption, satisfying some given throughput and delay requirements, by optimizing the transmission time. In [27] is presented a study about the energy efficiency of three distinct modulations schemes. In [28] is described an algorithm for transmission power control, which improves the energy efficiency. In [29] is presented a study about the use of directional antennas for energy efficiency. In [30] is stated that is possible to achieve better energy performance, in sensor networks, by using cooperative MIMO techniques.

Data Reduction: Energy-efficiency can be achieved by reducing the amount of traffic that is transferred on the network. In order to do this, it is conceivable to use information aggregation techniques such as adaptive sampling, network coding algorithms or compression methods. A survey about data aggregation techniques is available in [31]. In [32] is presented a study that adjusts the sampling frequency, in a human activity recognition application, according to the level of movement. In [33] is described a network coding algorithm that improves the energy efficiency of the network, and a survey about data compression techniques can be consulted in [34].

Sleep/Wakeup Schemes: To minimize the amount of energy wasted by the nodes, it is viable to use schemes that temporarily put the nodes in sleep mode, when they are not in an active mode. A device that is capable of implementing a passive wake-up radio sensor network, is presented in [35]. In [36] is presented a solution capable of minimizing the energy consumption of a WSN by activating only a subset of the existent nodes.

Energy-Efficient Routing: The energy drained from the nodes can increase significantly, if they are systematically chosen to forward packets that are destined to other nodes. Thus the routing algorithm could really affect the network lifetime, if not done carefully. In [37] are described two energy aware cost based routing algorithms, and in [38] is presented a multipath routing protocol.

Energy Harvesting: The use of rechargeable batteries in the nodes can really improve the network energy efficiency. By using solar energy, wind energy or even wireless charging to recharge these batteries, it is possible to increase the network lifetime. A survey about various aspects of energy harvesting sensor systems is available in [39]. In [40] is presented a study about wireless charging in WSNs.

2.2.2 Lifetime Extension Solutions

Based on the mechanisms above referred, several researchers come up with energy-efficient solutions that can be applied in wireless multi-hop networks, in order to increase their lifetime. In [41] the authors present a study about the behaviour of batteries, which declares that the discharged power of the batteries is higher than the power actually needed. The study also states that this over-discharged power can be recovered, if the battery has a sufficiently long recovery time. Based on this battery behaviour, the authors developed two algorithms to extend wired network infrastructures: the coverage algorithm and the back-haul routing algorithm. The main idea of the coverage algorithm, is to adjust the transceivers radius of the nodes, in a collaboratively way, allowing them to recover the over-discharged power, while providing the necessary network coverage for the network clients. As for the back-haul routing algorithm, the objective is to forward the packets of the nodes, to their neighbours with the lower over-discharged power. With these two algorithms it is viable to improve the energy efficiency of the network, by providing the necessary recovering time to the nodes with higher over-discharged power.

In [42] was created an algorithm which aim to turn off the largest possible number of radio interfaces, while maintaining a certain level of performance. So when the network load decreases the algorithm turn off some radio interfaces, and when the network load increases the algorithm turn on more radio interfaces. When a radio interface is turned on, its channel is carefully selected to better use the network resources. By doing this the algorithm can save power since the unnecessary radio interfaces are turned off. This algorithm assumes that the channel assignment and routing decisions were already made, and only tries to optimize the energy efficiency of the network.

In [43] was created a new routing metric for WSNs which takes in account the energy factor. To do this the authors take the HWMP airtime metric, defined in IEEE 802.11 standard, which reflects the amount of channel resources consumed by transmitting the frame over a certain wireless link, and add an energy factor to this metric. The authors assume that every node is considered equally important in the WMN and the objective is to ensure that each one of them consumes a similar quantity of energy. Thus they come up with a metric, named ETE, that takes into account the remaining energy (after transmission) and the initial energy of the nodes along the route.

In [44] is defined a new routing algorithm destined to WMNs which use solar energy and wind energy as a power supply. In order to save the power of the nodes, this routing algorithm specifies a new routing metric based on the traditional minimum hop metric. The idea is to add a hop penalty factor to the minimum hop metric, based on the remaining energy of the nodes. For every possible route between two nodes, a routing cost value is calculated. Each node in the route adds a penalty, based on its remaining energy, and the route with the lowest routing cost value is selected as the best route.

In [45] is proposed a new routing algorithm, for WSNs, named EMRP. This algorithm divides the network in several clusters and in each cluster, a cluster head is selected. This cluster head is responsible for aggregate the information of every other element of the cluster, and send it to a base station. The base station is the destination node of the information of all the nodes in the network. The EMRP is an event-driven cluster based algorithm, which means that clustering and data transmission to the base station, only happens when a certain event occurs. To save the power of the network the cluster head is chosen based on the energy available in the nodes. Thus the nodes with higher energy are the ones which are selected. That way the algorithm avoids using the nodes with low energy level to forward information, of other nodes, to the base station. In what concerns the routing problem there are two different situations: when the cluster head is in the transmission range of the base station and when the cluster head is out of the transmission range of the base station. In the first case the cluster head aggregates the information of the cluster nodes, and sends it directly to the base station. In the second case the cluster head tries to discover two relay nodes to the base station: a relay node and a backup relay node. This relay node discovering is repeated by the nodes elected by the cluster head, until a relay node in the transmission range of the base station is found. Then, by monitoring the energy of each of the relay nodes, this algorithm can switch between the two paths in order to achieve energetic efficiency. By adopting this switching strategy this algorithm provides reliability of routing paths, even load balance and energy efficiency.

2.2.3 Network Failure Point

To analyse the proposed solutions is necessary to precisely define the circumstances in which is considered that the network has failed. This is equivalent to define the meaning of the term network lifetime. This definition is crucial in order to compare two distinct algorithms, since the comparison is only valid if both algorithms are evaluated by the same criteria. In [41] the lifetime of the network is defined as the duration between the network set up and the moment when the routers can no longer cover the entire area. In [43] the network is considered active while there is, at least, one active node. However the author keeps track of the moment when each node becomes inactive, which allows to have a detailed record of the network behaviour regarding its lifetime. In [45] the simulation is executed until all the nodes which have the base station in their transmission range, become inactive. In [37] the lifetime of network is considered to be the amount of time between the network set up and the moment when the first node depletes its energy. There are also approaches that define the lifetime as the period between the network set up and the moment when a percentage of the nodes depletes its energy, as is referred in [46].

From the examples above referred it is noticeable that the definition of lifetime is not universal, and is very dependent of the type of application of the network. Thus it is necessary to precisely define the meaning of the term lifetime before evaluate a certain energy efficient solution. In this dissertation is adopted the definition proposed by [45]. We consider that the network is active until all the nodes which have the gateway in their transmission range become inactive. This approach makes sense in the context of this dissertation, because when all the nodes that can reach the gateways become inactive the video-surveillance system fails, since not even a single node is capable of transmitting the video information to the appropriate destination.

2.3 Summary

The first section of this chapter described a set of approaches to improve the capacity of wireless multi-hop networks, such as reducing the interference, designing new routing protocols and metrics, using multiple communications channels and using multiple network interfaces. Given these approaches, it were exposed several solutions, proposed by different authors, to surpass this problem. A special attention is given to the TILIA algorithm, due to its importance in this dissertation. In the second section of this chapter it were presented the most important parameters that affect the lifetime of wireless multi-hop networks. Given these parameters, it were exposed some solutions capable of reducing the energy waste in these networks.

By analysing all the possible approaches and solutions capable of increasing the capacity of wireless multi-hop networks we concluded that the best option is to use multiple communication channels. Using multiple communication channels allows us to have several simultaneous communications without having interference between them, and the complexity associated to this type of approach is reasonably acceptable. This type of approach is very dynamic since it is possible to add a new communication channel, if there is an increment in the number of nodes of the network, or to remove some of the existing communication channels, if there is a decrement in the number of nodes of the network. In what concerns energy efficiency, we can use a channel assignment algorithm, capable of modifying the network topology according to the energy of the nodes, to extend the lifetime of the network. If this channel assignment procedure is done regularly it is possible to adapt to changes in the energy levels of the network, avoiding, for example, to forward traffic through nodes with low remaining energy.

To conclude we can point out that the research regarding the state of the art, in what concerns capacity and energy efficiency of wireless multi-hop networks, allowed us to determine the best approach to solve the proposed problem.

Chapter 3

Proposed System

This chapter presents the proposed video surveillance system, which is based on the Wireless Multimedia Sensor Network (WMSN) concept, and the proposed eTILIA algorithm. The first section describes the addressed scenario and the proposed WMSN. The second section details the components of the system and presents off-the shelf alternatives. The third section exposes the protocol stack of the network, with a detailed description of the protocols used at each layer. The fourth section describes the mode of operation of the network. Finally, the fifth section presents the specification and implementation of the proposed channel assignment algorithm, eTILIA.

3.1 System Overview

In this section is proposed a low-cost, solar powered video-surveillance system, based in Wi-Fi, to cover large unconnected areas such as parks or beaches, similar to the one depicted in Figure 3.1. The idea of this system is to allow the surveillance of these isolated areas, from a remote surveillance centre. The system is composed by several video-surveillance cameras spread around a certain region. Only a few of these cameras can access the Internet, since most of them are placed in zones without network coverage. This way, in order to deploy a functional system, it is necessary to create a wireless multi-hop network, allowing the video transmission of the cameras located in places without network coverage. More specifically, it is necessary to deploy a WMSN, due to the multimedia nature of the network traffic.

Thus, this system can be represented through a network graph, such as the one exposed in Figure 3.2, in which a G letter represents a gateway and a N letter represents a common node. The nodes are responsible for gathering the desired data and for sending it to one of the gateways. The gateways are responsible for delivering the received information to the surveillance centre. To build a proper WMSN, the deployment of the network should be made in order to allow each node to reach every other node of the network. This means that it must be possible to represent the network through a connected graph. The main goal of the system is to create a WMSN capable of guaranteeing that every camera can successfully deliver its own information to these gateways. After that, the reliability of the wired connection between the gateways and the surveillance centre, allows to ensure the correct delivery of the information. The system is expected to have around 40 nodes.

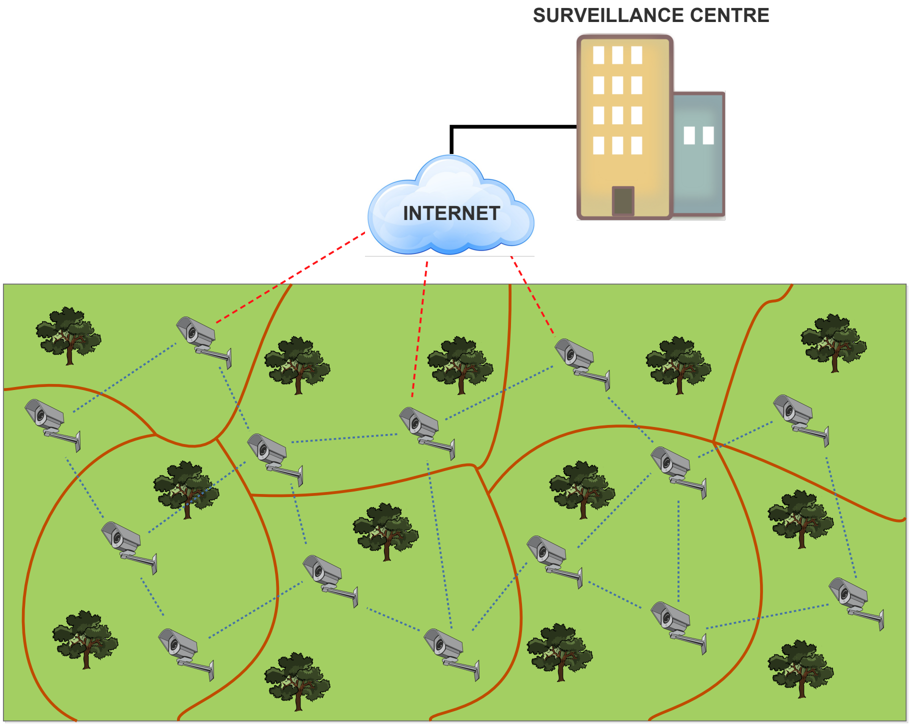


Figure 3.1: Video-surveillance system in a park

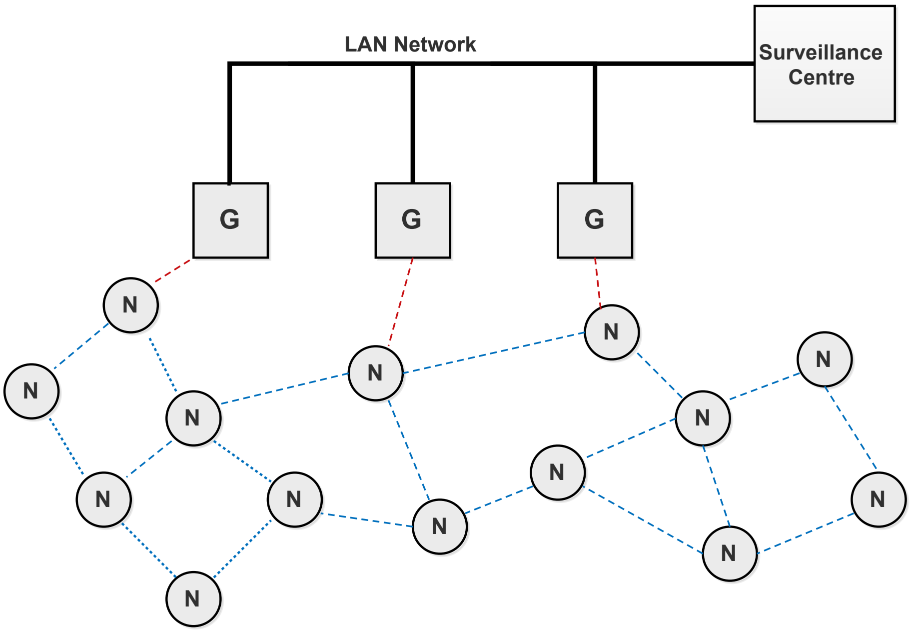


Figure 3.2: Network Graph

3.2 System Components

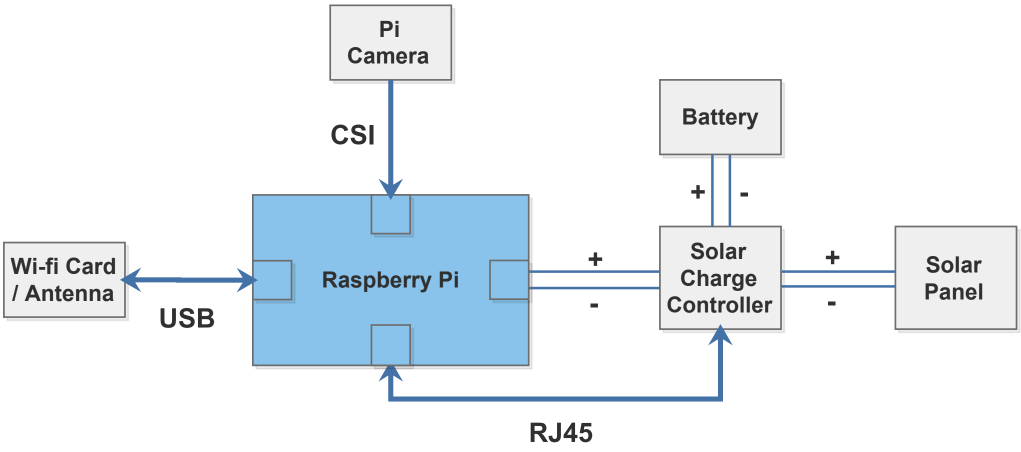


Figure 3.3: Connections schematic

This section details the components of the system and presents off-the shelf alternatives. In Figure 3.3 is presented a schematic of the connections between all the components. The components required to fulfil the specifications of the system are now presented:

Processing Unit: One of the fundamental components of the nodes is a processing unit capable of doing the required processing for the system to work properly. One possible alternative for this component is the Raspberry Pi processing unit, which is illustrated in Figure 3.4. Some of the advantages of the Raspberry Pi are the low price, the good energy efficiency and the good processing power. The compatibility with common standards such as USB, Ethernet or HDMI is also a great advantage. It is possible to obtain more information about the Raspberry Pi in [47].

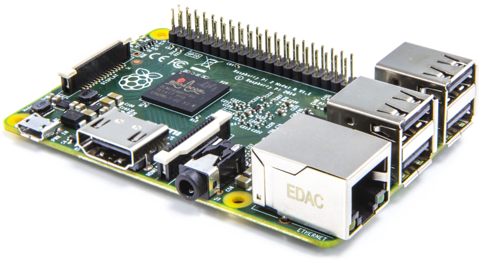


Figure 3.4: Raspberry Pi

Wi-fi Card/Antenna: Each node must be equipped with a Wi-Fi card and an antenna, in order to be able to communicate with the other nodes. The Wi-Fi card must be compatible with the IEEE 802.11 standard. One alternative would be to use a wireless USB adapter such as the TP-LINK TL-WN722N, depicted in Figure 3.5. This wireless USB adapter can achieve traffic rates up to 150Mbps, and is compatible with the wireless standards IEEE 802.11n, IEEE 802.11g and IEEE 802.11b. The possibility of adding an external antenna to improve the radio coverage is also an advantage. It is possible to obtain more information about the TP-LINK TL-WN722N in [48].

Figure 3.5: Wireless USB Adapter TP-LINK TL-WN722N

Multimedia Sensor: Each node must be equipped with a multimedia sensor, which is responsible for retrieving video footage from the surrounding environment and deliver it to the processing unit. One possible alternative for this component is the Raspberry Pi camera, which is illustrated in Figure 3.6. The full compatibility of this camera with the Raspberry Pi is an advantage. It is possible to obtain more information about the Raspberry Pi camera in [49].

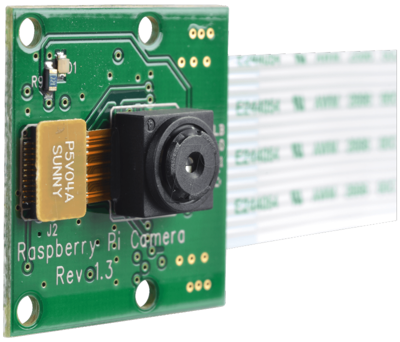


Figure 3.6: Raspberry Pi Camera

Battery: Each node must be equipped with a battery capable of supplying the necessary power for the node operation. The battery must be adequate to the power needed by the node. This require an analysis of the current drawn by each component of the nodes, in order to acquire an adequate battery. One possible alternative for this component is the battery DGY12-7.5EV, which is illustrated in Figure 3.8. It is possible to obtain more information about the battery DGY12-7.5EV in [50].

Figure 3.7: Battery DGY12-7.5EV

Solar Panel: Each node must have a dedicated solar panel to allow the energy harvesting. This avoids the necessity of an electrical infrastructure to provide power to the nodes. The size of the solar panel must be carefully calculated, to guarantee that the node has always the required power to operate. One possible alternative for this component is the solar panel GSAP6, which is illustrated in Figure 3.8. It is possible to obtain more information about the solar panel GSAP6 in [51].

Figure 3.8: Solar Panel GSAP6

Solar Charge Controller: Each node must also be equipped with a charge controller, which will be responsible for the connection between the battery and the solar panel. This charge controller prevents the overcharging of the battery and it may protect it against over-voltage. This is very important since these events can reduce the performance and the lifespan of the battery. Besides this, the charge controller can also allow to obtain information regarding the status of the battery, such as its energy level. One possible alternative for this component is the solar charge controller Tracer-1210RN, illustrated in Figure 3.9, which provides a RJ45 connection to obtain information about the status of the battery. It is possible to obtain more information about the solar charge controller Tracer-1210RN in [52].

Figure 3.9: Solar Charge Controller Tracer-1210RN

3.3 System Protocol Stack

This section exposes the protocol stack adopted in this system. In order to do this it is used the TCP/IP model, which divides the protocol stack in four main layers: Application Layer, Transport Layer, Internet Layer and Network Interface Layer. In Figure 3.10 are visually exposed the protocols chosen at each layer.

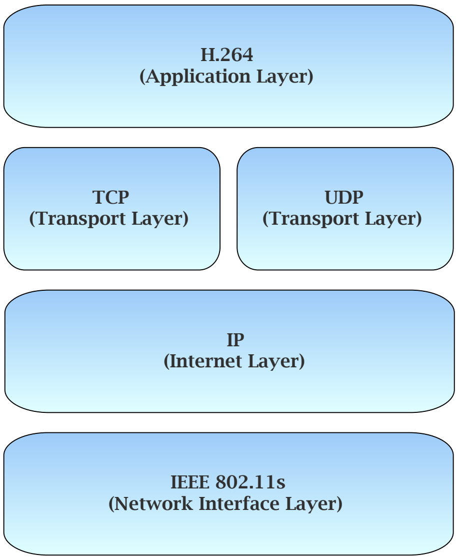


Figure 3.10: Protocol Stack

The protocol used at the application layer is the standard responsible for coding the video footage retrieved by the multimedia sensor. These video coding standards are a normalized format to represent digital video content, and usually adopt techniques to reduce the bit-rate of the video. By eliminating redundant information, such as spatial and temporal redundancy, and by removing less important information, it is possible to decrease the bit-rate, maintaining, at the same time, an acceptable image quality, as can be seen in [53]. There are many standards capable of doing this task, as for example: MPEG-2 Part 2, MPEG-4 Part 2, H.264, HEVC, Theora. The protocol to be used at this layer must represent a compromise between the compression ratio, the quality of the image and the computational power required by the encoder. A very high compression ratio could save the network resources, but could simultaneously lead to a very bad image quality. A very good image quality would be desirable for surveillance purposes, but could generate an unbearable traffic rate, leading to the network breakdown. The use of an excellent video coding standard, capable of generating good quality video with low traffic rate, could require a great computational power, which could lead to a quickly consumption of the energy resources. This way it is possible to see that it is necessary to find a compromise between these three characteristics, in order to deploy a fully functional system. One of the current best video coding standard is the H.264. So this was the video coding standard chosen to be used in this system. This standard is composed by several profiles, which can have a higher or lower compression ratio and a better or worse image quality. The profile to be used should be chosen taking into consideration the system in which will be applied. So the most appropriate profile to our particular system is the *Baseline Profile*. The *Baseline Profile* is the profile which requires less computational resources, and which has the lower latency. Besides the profiles, the H.264 also defines several levels which can be combined with the profiles. Each level within a profile specifies the maximum picture resolution, frame rate, and bit-rate that a decoder may use. So taking into consideration the system to be implemented, the chosen level to be combined with the *Baseline Profile* was the level *1b*. With this combination, the resolution of the recorded video is 176x144, and the maximum bit-rate is 128 kbit/s, as can be seen in [54].

At the transport layer there are two main alternative protocols: TCP or UDP. TCP provides a reliable, ordered, error-checked delivery of a data stream, and requires the establishment of a connection before sending any messages. However its reliability mechanisms could lead to great transmission delays. UDP on the other hand is an unreliable protocol which does not guarantee the packet delivery, packet ordering or duplicate packet protection. However is suitable for purposes where error checking and correction is either not necessary or is performed by the application, avoiding the overhead of such processing at the transport layer. Giving the characteristics of the two alternatives, the choice must be made taking into consideration the final goal of the information. If the information is intended to be saved in the surveillance centre then the most suitable protocol to be used is the TCP, due to its reliability. If the system is supposed to provide live video streams to the surveillance centre, then the most suitable protocol to be used is the UDP, since dropping packets is preferable than waiting for delayed packets.

At the internet layer the chosen protocol is the IP. This protocol enables the packet delivery from one host in the Internet to any other host, even if the hosts are on different networks. This is the most common protocol used at this layer.

At the network interface layer is used the standard IEEE 802.11s. This standard is an IEEE 802.11 amendment for mesh networking which defines how wireless devices can interconnect to create a WLAN mesh network. As stated in [55], this standard introduces wireless frame forwarding and routing capabilities at the MAC layer, allowing multi-hop communications. 802.11s depends on one of the following standards for carrying the actual traffic: 802.11a, 802.11b, 802.11g or 802.11n. The default routing protocol used by this standard is the HWMP.

It is important to refer that the security issues of the system are out of the scope of this dissertation.

3.4 System Operation

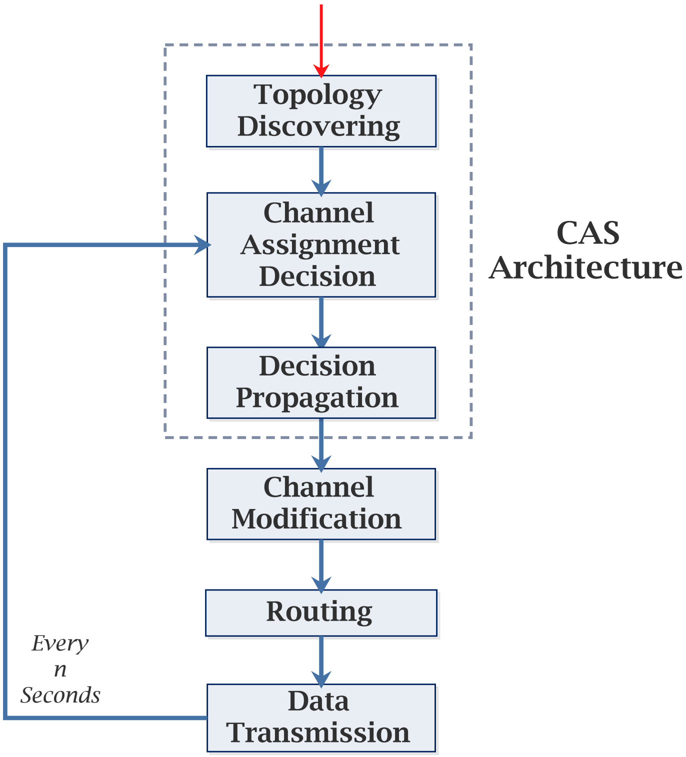


Figure 3.11: Network Operation Flowchart

In Figure 3.11 is presented the flowchart of the network operation. At the starting point all the nodes of the network have the same channel. The first stage of the network operation is to discover the topology of the network, and the energy of every single node. This is necessary since the eTILIA algorithm, referred in the next section, requires this information as input. After the successful delivery of all this information to the surveillance centre, it is possible to execute the channel assignment procedure. Once the channel assignment algorithm defines the channel to be assigned to each node, it is necessary to propagate the decision to every node, allowing the nodes to change their channel accordingly. After every node change its own communication channel, it is necessary to execute a routing algorithm in order to discover a route, so that every node can be capable of reaching a gateway to deliver its traffic. This routing procedure could be done using, for example, the HWMP protocol or using the information provided by eTILIA regarding the parent of each node. Once this route is discovered the nodes collect data and continuously send it to the surveillance centre through the appropriate gateway. After a certain amount of time a new channel assignment procedure is initiated in order to allow the network to adapt to changes in the energy levels of the nodes, leading to an improvement of the lifetime of the network.

To be able to discover the network topology, centrally assigned the channels to the nodes and propagate this information to the nodes, it is necessary to use some kind of mechanism or subsystem. One alternative would be to use the Channel Assignment Subsystem (CAS) architecture defined in [56]. This architecture works as follows:

1 - Topology Discovery: To discover the topology of the network this architecture uses a procedure similar to the TDCC protocol defined in [57]. This protocol discovers the network topology, independently of the channel in which each node is operating, and delivers that information to a master node, which could be, for example, the network gateway. To do this each node creates a table with its own neighbours, appends this information to the routing messages, and then forwards it to the master node.

2 - Information Decoding: After gathering all the relevant information it is necessary to decode it.

3 - Graph Generation: After the information decoding, the graph of the network is created.

4 - Channel Assignment Decision: After the network graph creation, it is possible to execute the centralized channel assignment procedure. In this phase the system uses the channel assignment algorithm proposed in this dissertation, eTILIA, which is described in the next section.

5 - Decision Propagation: After the channel assignment the procedure similar to TDCC protocol is used again, in order to inform all the nodes about the channel which was assigned to them.

3.5 eTILIA

This section focuses on the description of the eTILIA algorithm, the main contribution of this dissertation. eTILIA is a dynamic, centralized and energy-aware channel assignment algorithm, based on the TILIA algorithm, referred in Section 2.1.3, which efficiently manages the energy resources of WMSNs. eTILIA take as input the network graph, with all the gateways and all the connections between the nodes of the network, and the energy level of each node. The output of the algorithm is the channel assigned to each node. This section starts by presenting the specification of eTILIA, and then exposes a high level description of its implementation.

3.5.1 eTILIA Specification

In this dissertation we focus on extend the lifetime of WMSNs, by avoiding to forward traffic through nodes with low energy level, taking advantage from a multi-channel network architecture. Since the transmission of information is responsible for a significant amount of the wasted energy of the nodes, we can improve the energy efficiency by avoiding to constantly forward traffic through the same nodes. The idea of the eTILIA algorithm is to avoid to forward traffic through nodes with low energy, by carefully assign the channels to the nodes. This decision can be done in one of two ways:

Strict Decision: In this mode of operation we would strictly avoid to forward traffic through nodes with low energy level. This means that the decision would be made without analysing the consequences. This type of decision could significantly increase the lifetime of the network, but could also lead to isolated nodes. If we had a network as the one illustrated in Figure 3.12, where node A could only reach the gateway through node B, assigning to node A a different channel from the channel previously assigned to node B, with the goal of avoiding the traffic forwarding through this node, would isolate node A. This mean that node A would not be capable to send its traffic to the gateway.

Flexible Decision: In this mode of operation we would avoid to forward traffic through nodes with low energy level in a flexible way. This means that the decision would be made only after analysing its consequences. This type of decision could prevent us from having disconnected networks. If, for example, we had a network where node A could only reach the gateway through node B, then the channel of node A would have to be the same as the channel of node B, independently of the energy level of node B. Otherwise it would be impossible for node A to send its traffic to the gateway.

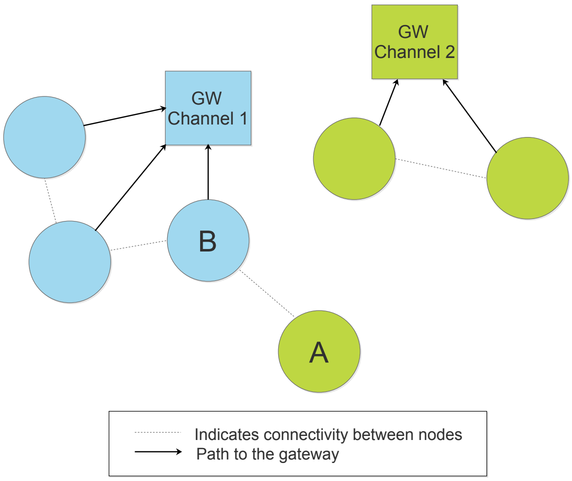


Figure 3.12: Strict decision consequences

Taking into account the two types of decision above referred, in the specification the eTILIA algorithm we chose the flexible decision type. This type of decision allows a better trade-off between the improvement of the network lifetime and the consequences in the network topology.

Another important detail in the specification of the algorithm was the decision of when it would be appropriate to take into consideration the energy level of the nodes, in the channel assignment procedure. So after analysing all the possibilities we ended up with these alternatives:

1: Define a fixed energy threshold. If the energy of a certain node decreases beyond this threshold, the algorithm tries to avoid traffic forwarding through that node. This alternative only takes into consideration the energy of the node itself, and do not consider the energy of the other nodes of the network. Also it only takes the energy of the nodes into consideration if their energy drop beyond a certain value.

2: Analyse the energy level of all the nodes in the network and avoid to forward traffic through the nodes which present the lowest values. With this alternative the algorithm has a global view from the energy level of the network, and the definition of low energy depends on the energy level of all the nodes in the network. To do this the algorithm could, for example, calculate the mean energy of all the network nodes and declare the ones with an energy level below the average as the low energy nodes.

3: Another alternative it is a combination of the two alternatives above referred. The algorithm declares that all the nodes have a high energy level until they reach some specified remaining energy value. After the nodes cross this energy threshold, the algorithm could declare the ones with the lowest energy values as the low energy nodes.

4: The last alternative is similar to the second alternative but it does not define a threshold from which the nodes are considered to be with low energy. Instead it always tries to forward traffic through the node with higher amount of energy.

After carefully analyse each alternative, we decided that the second alternative and the fourth alternative would be the most appropriate, since the strategy adopted in those alternatives allow us to manage the energy levels of the network right from the start of the network operation, and not only after a certain energy threshold being crossed. After this first decision we had to choose between one of these two alternatives. In the second alternative it would be possible to compare the energy levels of all the nodes of the network, and declare a set of them as the low energy nodes. This way if we had multiple nodes with an energy level higher than the defined low energy threshold, we could choose to forward traffic through the node which led to the better network throughput. In the fourth alternative this choice does not exist because we always choose to forward traffic through the node with higher energy. So by picking the fourth alternative we are making a trade-off between the throughput of the network and the improvement of its lifetime. Taking all of this into consideration we chose to adopt the fourth alternative since we are willing to trade-off the network throughput for an improvement of the lifetime of the network.

By combining this alternative with a flexible decision mode of operation, we can try to avoid the traffic forwarding through nodes with low energy level. The goal is to do this without ever affecting the connectivity of the network. This is very important since WMSNs requires each node to be able to deliver its own traffic to the appropriate destination.

3.5.2 eTILIA Implementation

In order to implement the eTILIA algorithm we took advantage from the TILIA algorithm, described in Section 2.1.3. After a detailed study of the TILIA algorithm, it was necessary to understand which modifications should be done in order to adapt it to the specifications defined in Section 3.5.1. After a thorough analysis it was concluded that the modifications should be done in the parent choice procedure of the TILIA algorithm. By carefully choosing the most appropriate parent it is possible to change the network topology to achieve a better energy efficiency. The idea is making this decision based on the energy levels of the candidate parents. This way it is possible to avoid choosing the nodes with low energy level as parent nodes. By doing the modification at this stage of the TILIA algorithm it is also possible to guarantee the network connectivity, because if the node has only one candidate parent, then that parent is selected independently of its energy level. The energy of the nodes is only taken into consideration if there are multiple candidate parents.

To implement this modification it was necessary to understand the python script which implements the TILIA algorithm. After a meticulous analysis of the script it was concluded that the modification should be done in the function *recursive_tilia*. As referred in Section 2.1.3.2, this function discovers all the nodes which do not have a channel assigned, and selects a parent and a channel to them. To fulfil the specifications of the eTILIA algorithm it was necessary to adapt this function in order to take in consideration the energy of the nodes during the parent choice procedure. In the left column of the Figure 3.13 is exposed a high level description of the original *recursive_tilia* function of the TILIA algorithm. In the right column of the Figure 3.13 is exposed a high level description of the *recursive_tilia* function of the eTILIA algorithm, after the modification. The python code of this specific part of the script can be seen in Appendix A.

As can be seen in Figure 3.13, the modification was introduced after the eTILIA algorithm select the neighbours with the lowest hop count which operate in one of the best channels. If the modification was introduced before this stage we could overload a certain tree, or originate longer traffic flows paths. If the modification was done after this stage, we would restrict too much the set of possible parents. Thus, we concluded that this was the most appropriate stage to place this modification.

With this modification, the recursion procedure of TILIA was removed from eTILIA. Instead of using recursion, the eTILIA algorithm simply chooses the candidate parent with the highest remaining energy. By doing this, the eTILIA algorithm possibly sacrifices an improvement of the network throughput for an improvement in the network lifetime. This decision is acceptable since this algorithm was specified to be used in networks with reduced energy resources, in which the concern about the lifetime of the network is far more important than the concern about the network

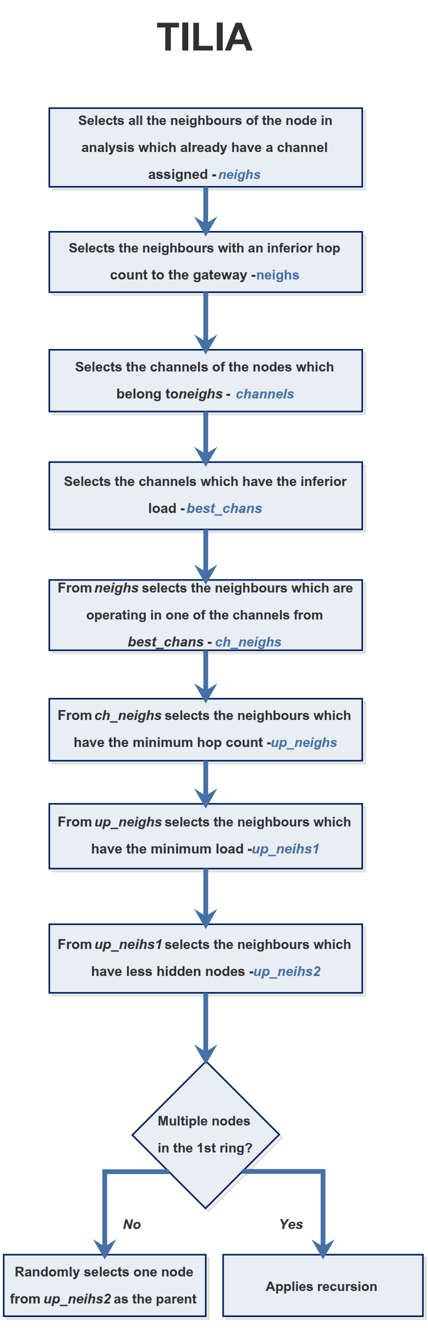


Figure 3.13: High level description of the function *recursive_tilia* in both TILIA and eTILIA script

throughput. Another positive consequence of removing the recursion procedure is the decrease of the computer complexity required by the algorithm.

3.6 Summary

In this chapter was exposed the architecture of the system explored in this dissertation. In the first section it was described the addressed scenario and the proposed WMSN. The addressed scenario is related with a a low-cost, solar powered video-surveillance system, based in Wi-Fi, to cover large unconnected areas such as parks or beaches. The idea is to allow the surveillance of these isolated areas, from a remote surveillance centre. Regarding the network topology it was seen that there are regular nodes, responsible for collecting the information, gateways, responsible for forwarding the information to the final destination, and a surveillance centre, which is the final destination of the information.

In the second section were detailed the components of the system and presented off-the shelf alternatives. In what concerns the node components it was seen that each node is composed by a processing unit, a Wi-Fi card/antenna, a multimedia sensor, a battery, a solar panel and a solar charge controller. It was also presented a schematic of the connections between the components.

In the third section was exposed the protocol stack of the system, based on the TCP/IP model. At the application layer was chosen the H.264 protocol due to its efficient video coding. At the transport layer there are two alternatives: TCP and UDP. If the goal of the system is to save the video information on the surveillance centre, then TCP should be used due to its reliability. If the goal of the system is to provide a live stream to the surveillance centre then UDP should be used. At the internet layer the protocol chosen was IP since it is the most common protocol used at this layer. At the network interface layer it was adopted the IEEE 802.11s protocol to be able to deploy a mesh network.

In the fourth section it was described the network operation which consists in six steps: topology discovering, channel assignment decision, decision propagation, channel modification, routing procedure and data transmission.

In the fifth section was presented the proposed channel assignment algorithm, eTILIA. First it was exposed the specification of eTILIA, together with the main decisions that were made regarding the algorithm operation. After this, it was presented a high level description of the eTILIA implementation. In this section it was seen that eTILIA avoids to forward traffic through nodes with low energy level, in order to extend the lifetime of the network.

Chapter 4

Evaluation and Results

In order to evaluate the channel assignment algorithm proposed in this dissertation, eTILIA, a set of simulations were carried out, using the Network Simulator NS-3. This chapter starts by describing, in the first section, the tools and models that were used in the simulations. The second section describes the regular and random network topologies used in the simulations. Then, the third section presents a description of the methodology adopted. Finally, the fourth section exposes the results obtained.

4.1 Network Simulator 3

To test the algorithm proposed in this dissertation we took advantage from computer simulations. This is a very common procedure in the network research area, since it allows to verify and validate new network algorithms without having to deploy a complete test-bed, which could be very costly and time-consuming. By using computer simulation we were able to create networks with the required dimension to evaluate the ETILIA algorithm, without having to worry about the monetary costs of setting up a real network. So, the computer simulations were an important resource to the validation of the algorithm developed.

The simulation tool used in this dissertation was the Network Simulator 3 (NS-3), publicly available in [58]. NS-3 is a discrete-event network simulator, implemented in C++, built as a library which can be linked to a C++ main program that defines the topology and the mode of operation of the network to be simulated. NS-3 is an open-source tool which means that everyone can easily access its source code and modify it. This way everyone can add new simulation models to the existent ones. In the simulations performed in this dissertation it was used the version 3.22 of NS-3. To be able to do the desired simulations it was necessary to make some modifications in the source code of NS-3. These modifications are exposed in Appendix B

4.1.1 Simulation Models

To simulate the required network it was necessary to use some NS-3 models. In this section are presented and described the required models: Mesh Network Model, Energy Model and Flow

Monitor Model.

4.1.1.1 Mesh Network Model

To simulate the desired networks it was used the wireless mesh network model, already existent in NS-3 source code, referred in [59]. This model implements an IEEE 802.11s wireless mesh network and focuses on three important aspects of this standard:

Addressing and Forwarding: The addressing of the mesh stations is done using a 48-bit MAC address. The forwarding of the frames is made based on a 6-address scheme.

Peering Management: The peering management protocol is used to open, maintain and close links with neighbours mesh stations. In order to do this, each station regularly transmit a small one-hop management frame, known as a beacon. This is very important since a mesh station is not allowed to exchange data frames with other mesh stations before a link is established between them being.

Routing Protocol: The routing protocol is necessary to discover multi-hop paths in the network. One of the supported routing protocols is the HWMP protocol. This protocol allows two distinct modes of operation: on demand routing or proactive routing. With on demand routing the path is built by exchanging messages (PREQ: Path Request, PREP: Path Reply) between the source node and the destination node. With proactive routing the path is established using an intermediate node, which knows the path to every node in the network. These two modes of operation can be executed simultaneously

4.1.1.2 Energy Model

To simulate the energetic behaviour of the network it was used the tool exposed in [60]. This energy tool is an energy framework which is mainly composed by two parts: the energy source model and the energy consumption model. The energy source model represents the power supply of the nodes, and the energy consumption model captures how the the energy is consumed by the nodes.

The energy source model allows to simulate both linear and non-linear battery models. The linear battery model is easier to configure and set up, but gives less accurate results comparing to the non-linear battery model, since it does not takes into account some non-linear effects that greatly affect the battery lifetime, such as Rate Capacity Effect and Recovery Effect defined in [61]. The non-linear model allows to take this factors into consideration, and it also enables to model the discharge curves of specific batteries.

The energy consumption model focuses on modelling the energy consumption of the radio, since this it is the component which contributes the most to the energy depletion of the nodes. This model allows to specify the energy wasted by the radio in the different operating states, being the energy consumption calculated by multiplying the power wasted in each state by the time spent in that same state. However, this is not a very accurate solution, since it does not take into consideration the energy wasted by other node components, such as sensors or processors.

This energy framework is based on the assumptions that a node can be powered by a single energy source or by multiple energy sources, that the energy sources supply power at a constant voltage and that the operation of all devices of the node is state-based and that each one of this state has a specific load current value associated with it.

In this dissertation we chose to use the linear energy source model, because this model represents a compromise between the required computational power for the simulation, and the accuracy of the simulation results. This energy framework allowed us to test the energy efficiency of the E-TILIA algorithm.

4.1.1.3 Flow Monitor Model

To keep track of all the packets of the simulation it was used the flow monitor tool provided by NS-3, and described in [62]. With this model, a flow is characterized by its source/destination address, source/destination port and protocol, and each flow is independently monitored. Using this tool it is possible to determine the time of the first and last transmitted or received packet, the delay sum of all received packets or even the amount of transmitted, received, lost and dropped packets for a certain flow. In these simulations we focused on analysing the number of transmitted/lost packets and the delay characteristics of each topology. This was really important to guarantee that the WMSNs simulated were capable of deliver most of the traffic to the appropriate gateways.

4.2 Topologies

The algorithm that was developed in this dissertation was tested with both regular and random topologies. These topologies are described in the next two sub-sections.

4.2.1 Regular Topologies

The first tests of the eTILIA algorithm were done in a grid topology. The simulations were performed in a 3x3, 4x4 and 5x5 grid topology, and each topology was tested with 2, 3 and 4 gateways/communication channels. The gateways were always positioned in the corners of the grid, and the horizontal and vertical distance between consecutive nodes was of 100 meters. In Appendix C is exposed the graphical representation of all the regular topologies simulated, before and after the first eTILIA channel assignment procedure. In the images on the left, the red circles represent the gateways and the black squares represent the ordinary nodes. The numbers identify the node, and the lines between the nodes represent link layer connectivity. In the images on the right, the nodes with the same colour have the same communication channel, and the lines with an arrow allow to visualize the trees constructed by the eTILIA algorithm. The nodes with the same colour connected by a line without an arrow can also establish communication.

These regular topologies allowed to test intermediate versions of the eTILIA algorithm. However, the simulation results obtained with these topologies are not exposed in this dissertation, since regular topologies do not correspond to real world scenarios.

4.2.2 Random Topologies

To test the eTILIA algorithm with random topologies, it was necessary to develop a random topology generator. Thus, it was created a C++ algorithm capable of generating random topologies with appropriate characteristics. The procedure adopted is now described:

1: At the beginning of the procedure it is necessary to specify the number of nodes to generate and the number of gateways of the network.

2: After this it is defined a maximum value for the random coordinates of the nodes, taking into consideration the number of the nodes of the network. This is necessary in order to generate connected graphs. The maximum value is defined according to the formula $\sqrt{NumberOfNodes * \pi * (Range/3)^2}$, where the term *Range* is the approximate transmission range of the nodes. The logic of this formula is the following: if all the nodes were placed in a regular topology the coverage area was given by $NumberOfNodes * \pi * (Range)^2$. This would mean that in a square are the measure of each axis would have to be equal to $\sqrt{NumberOfNodes * \pi * (Range)^2}$, to result in a total area of $NumberOfNodes * \pi * (Range)^2$. Finally the *Range* it is divided by a factor of three to guarantee a connected graph.

3: Taking into consideration the maximum value defined are randomly generated the coordinates of each node. The position of each node it is only valid if there is no neighbours at a distance lower than a certain imposed limit. If this requisite is not fulfilled a new random position is generated. The first nodes are elected as the gateways of the network.

4: Then it is created a short NS-3 simulation with the goal of discovering the neighbours of each node. In this simulation each node sends a traffic flow to all the other nodes. The packets of this flow are configured to have a Time-To-Live equal to 1 which means that they only can reach the nodes which are one hop away from the source. The nodes that receive more than 75% of the packets of a certain traffic flow are considered neighbours of the node which originated that flow.

5: After getting the information about the neighbours of each node, the topology is evaluated to see if there is no isolated nodes. If there are no isolated nodes, the topology is accepted. Otherwise, the topology is discarded and a new one is created.

The flowchart of the algorithm above explained is exposed in Figure 4.1. With this random topology generator it was possible to create appropriate topologies to test eTILIA. In Figure 4.2 it is exposed an example of a random topology, composed by 16 nodes and 2 gateways, generated by this algorithm. The gateways are represented as red circles and the lines between the nodes mean that they are close enough to communicate directly. In Appendix D is exposed the graphical representation of all the random topologies simulated, before and after the first eTILIA channel assignment procedure. In the images on the left, the red circles represent the gateways and the black squares represent the ordinary nodes. The numbers identify the node, and the lines between the nodes represent link layer connectivity. In the images on the right, the nodes with the same colour have the same communication channel, and the lines with an arrow allow to visualize the trees constructed by the eTILIA algorithm. The nodes with the same colour connected by a line without an arrow can also establish communication.

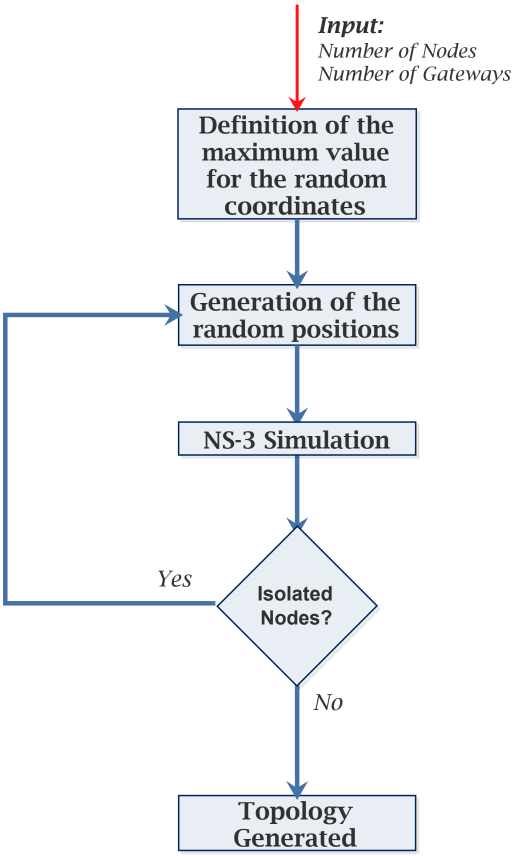


Figure 4.1: Flowchart of the algorithm for the random topology generation

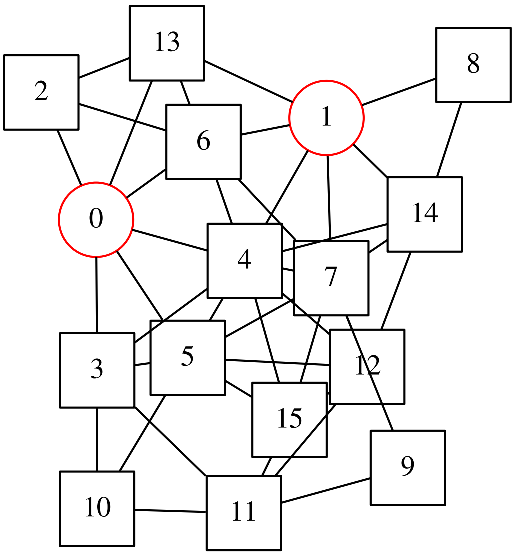


Figure 4.2: Random topology with 16 nodes and 2 gateways

4.3 Simulation Methodology

To evaluate the eTILIA algorithm were carried out a set of simulations, using NS-3. As referred above, these simulations were only performed with random topologies, since these random topologies have a higher correspondence with real world scenarios. It were used two types of topologies: topologies of 16 nodes and 2 gateways, and topologies of 36 nodes and 2 gateways. The algorithm was tested with 20 topologies of 16 nodes and with 20 topologies of 36 nodes. The graphical representations of the random topologies simulated are exposed in Appendix D. Each topology was simulated 5 times in order to obtain statistically relevant results. To guarantee that the results obtained from each simulation were independent we used the sub-stream capability provided by NS-3, instead of using a different seed for each simulation. We used this strategy because by using a different seed for each simulation it is not possible to guarantee that the streams produced by each seed will not overlap. By using the sub-stream feature, the seed is the same for each simulation, but the sub-stream of the random number generator is distinct, guaranteeing statistically independent simulations. This strategy allows for a maximum of $2.3x10^5$ independent replications, which is more than enough taking into account the number of simulations that were performed.

The procedure adopted in the simulations was the following:

1: The topology to be simulated is created and configured. Each node starts with an energy level of 1000 Joules.

2: Before any node starts sending its traffic, the eTILIA channel assignment algorithm is executed in order to assign an appropriate channel to each node.

3: After the first channel assignment procedure each node was configured to send a traffic flow, with a constant bit-rate of 100kbps, to a certain gateway. This traffic flow runs for 100 seconds.

4: Then, after these 100 seconds, the eTILIA channel assignment algorithm is executed again, and the channels of the nodes are modified accordingly. This allows the adaptation of the eTILIA algorithm to changes in the energy levels of the network. This step is repeated every 100 seconds, until the network fails. This occurs when all the nodes one hop away from the gateways, lose their energy, as stated in Section 2.2.3.

In Table 4.1 are presented some of the parameters used on the simulations. The parameters that are not referred were configured with the default value assigned by NS-3.

Parameter	**Value**
Wi-Fi physical standard	802.11a
Data mode	OFDM 6Mbps
Control mode	OFDM 6Mbps
Packet size	1200 bytes
RTS/CTS mechanism	Yes
Minimum packet size required to use RTS/CTS mechanism	1200 bytes
Traffic rate	100kbps
Transport protocol	UDP
Routing protocol	Static routing based on eTILIA information
Maximum number of packets retained while waiting for ARP reply	1000 packets
Maximum number of packets retained in wifi output queue	10000 packets
Initial energy of the nodes	1000 Joules
Mobility Mode	Constant Position Mobility Model

Table 4.1: Simulation parameters

4.4 Simulation Results

In this section are presented the results obtained with the simulations described in Section 4.3.

4.4.1 Simulation Metrics

The metrics used to evaluate eTILIA were the following:

Network Lifetime: Time interval between the start of the network operation and the moment of its failure. As described in Section 2.2.3, it is considered that the network fails when all the nodes which have the gateway in their transmission range, become inactive.

Mean Node Lifetime: Mean of the lifetime of the nodes which ran out of energy during the simulation. The gateways are not considered.

Alive Nodes: Number of nodes which still have energy resources at the moment of failure of the network.

Packets Received: Sum of the successfully received packets of each one of the traffic flows. It is expressed in terms of percentage.

Packets Lost: Sum of the lost packets of each one of the traffic flows. It is expressed in terms of percentage.

Mean Packet Delay: Mean of the delay of all the successfully received packets.

4.4.2 Results

In Figure 4.3, Figure 4.4 and Figure 4.5 are exposed the simulation results regarding the random topologies of 16 nodes, and in Figure 4.6, Figure 4.7 and Figure 4.8 are exposed the simulation results regarding the random topologies of 36 nodes. Figure 4.3 and Figure 4.6 show the gains of eTILIA regarding the energy management metrics. The left and right columns show, respectively, the improvements in relation to TILIA (Figure 4.3(a) and Figure 4.6(a)) and to a random channel assignment procedure (Figure 4.3(b) and Figure 4.6(b)). Figure 4.4 and Figure 4.7 show the absolute values of metrics shown in Figure 4.3 and Figure 4.6. Figure 4.5 and Figure 4.8 show the improvement of eTILIA regarding the capacity metrics. The left and right columns show, respectively, the absolute values and the gains. Figure 4.9 exposes the average gain for each metric, calculated over all the topologies simulated. The results obtained are presented with full detail in Appendix E.

To calculate the gain of eTILIA regarding the metrics described in the previous subsection, it was used the method described in [2]:

$$gain_s^\eta = k\frac{a_{eTILIA}^\eta - a_s^\eta}{a_s^\eta} \tag{4.1}$$

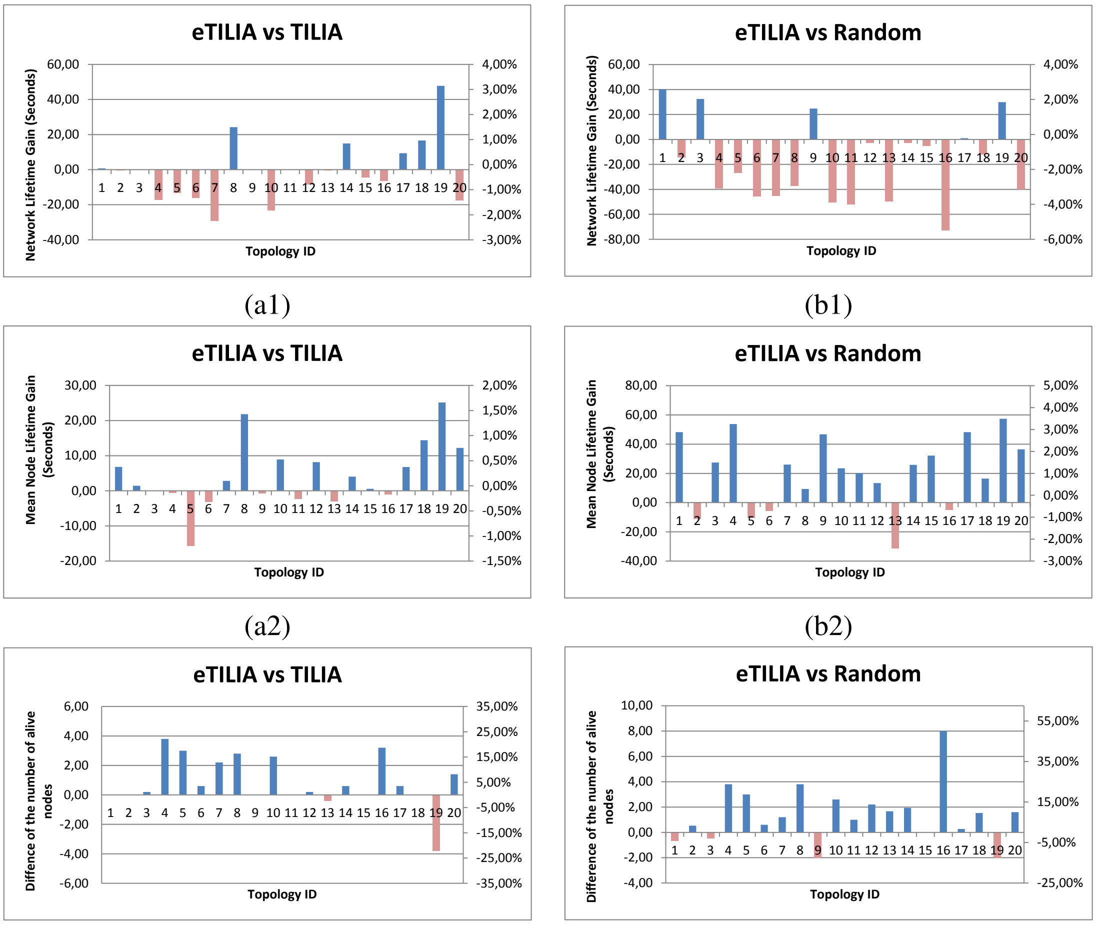
In Equation 4.1 $\eta \in \{$Network Lifetime, Mean Node Lifetime, Alive Nodes, Packets Received, Packets Lost, Mean Packet Delay$\}$, $s \in \{$TILIA, random procedure$\}$, a_{eTILIA}^η and a_s^η are the average values of the metric η for the networks operating respectively with eTILIA and with strategy s. $k = 1$ for network lifetime, mean node lifetime, alive nodes and packets received, and $k = -1$ for packets lost and mean packet delay.

By comparing eTILIA with TILIA, regarding the random topologies topologies of 16 nodes, it is possible to see that the improvements in the network lifetime, Figure 4.3(a1), and in the mean node lifetime, Figure 4.3(a2), are reduced. However, with the eTILIA algorithm the number of alive nodes, at the moment of the network failure, is significantly higher than with TILIA algorithm, as can be seen in Figure 4.3(a3). This allows to conclude that in the 16 nodes random topologies the eTILIA algorithm provides a better management of the energy resources of the network. However, this does not translate in an improvement of the lifetime. From Figure 4.5(a2) and Figure 4.5(b2) it is also possible to conclude that eTILIA leads to a worse network behaviour regarding the received/lost packets. This makes sense since eTILIA tries to make a trade-off between the performance provided by TILIA and a better management of the energy resources of the network.

By comparing eTILIA with a random channel assignment procedure, regarding the random topologies of 16 nodes, it is possible to see that the improvement in the network lifetime is reduced, as can be seen in Figure 4.3(b1). However, with eTILIA, there is an improvement in the mean node lifetime, and the number of alive nodes, at the moment of the network failure, is considerably superior, as can be seen in Figure 4.3(b2) and Figure 4.3(b3). Thus, it is possible to conclude that eTILIA that in the random topologies of 16 nodes the eTILIA algorithm leads to better results than the random assignment procedure. From Figure 4.5(a2), Figure 4.5(b2) and Figure 4.5(c2) it is also possible to conclude that eTILIA leads to an improvement of the number of received packets, of the number of lost packets and of the packets delay. This makes sense since the random procedure is not concerned with the improvement of the network throughput/delay characteristics, unlike eTILIA.

By comparing eTILIA with TILIA, regarding the random topologies of 36 nodes, it is possible to see some improvements in the number of alive nodes, at the moment of the network failure. However, these improvements are not as significant those of the random topologies of 16 nodes. In the random topologies of 36 nodes the main improvements originated by eTILIA are related with the network lifetime and the mean node lifetime, as can be seen in Figure 4.6(a1) and Figure 4.6(a2). However these improvements are not very significant. From Figure 4.8(a2) and Figure 4.8(b2) it is also possible to conclude that eTILIA leads to a worse network behaviour regarding the received/lost packets, which is similar to what occurs in the random topologies of 16 nodes.

By comparing eTILIA with a random channel assignment procedure, regarding the random topologies of 36 nodes, it is possible to conclude that the improvements in the network lifetime are reduced, as can be seen in Figure 4.6(b1). However the improvements in the mean node lifetime, Figure 4.6(b2), and in the number of alive nodes, Figure 4.6(b3), at the moment of the network failure, are significant. From Figure 4.8(a2), Figure 4.8(b2) and Figure 4.8(c2) it is also possible that eTILIA leads to a better network behaviour regarding the received/lost packets and the delay/characteristics, which is similar to what occurs in the random topologies of 16 nodes.

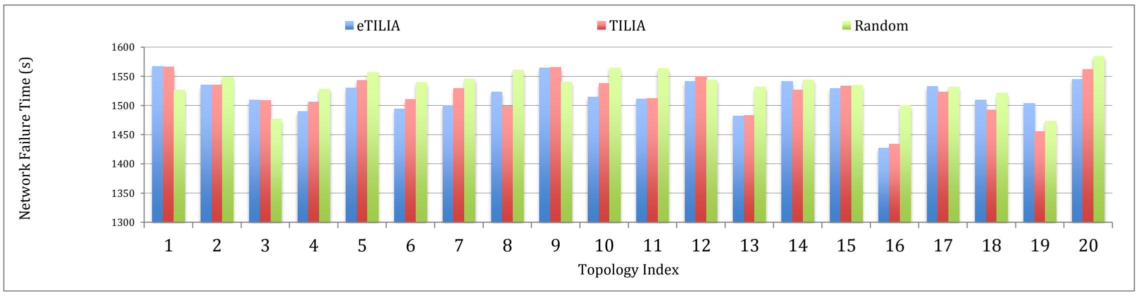


(a1) (b1)

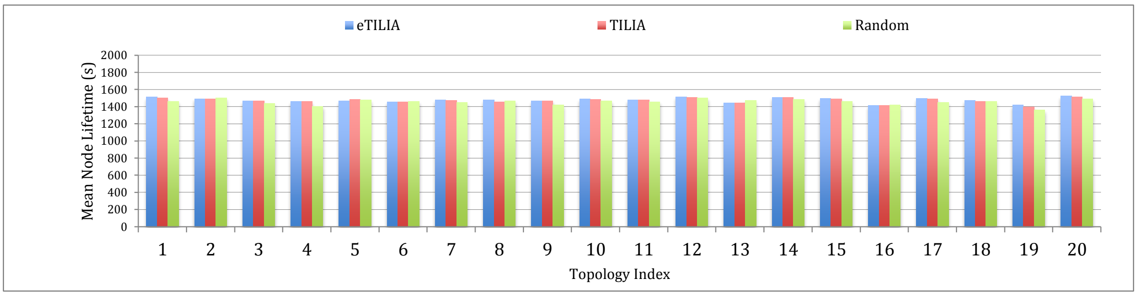
(a2) (b2)

(a3) (b3)

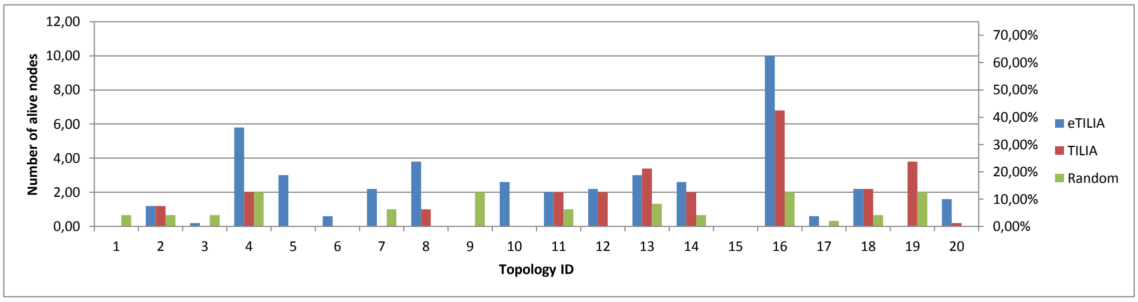
Figure 4.3: Simulation results with random topologies of 16 nodes (network lifetime, mean node lifetime, number of alive nodes)



Network lifetime

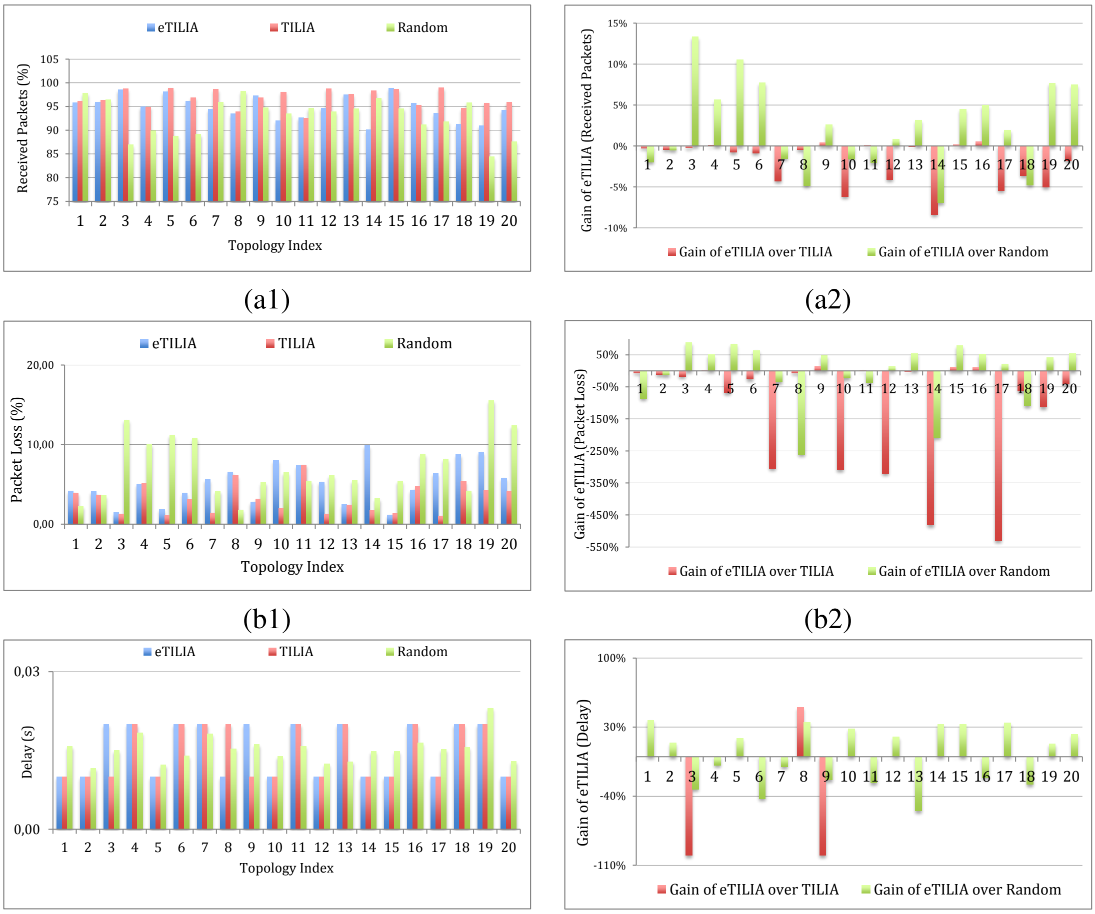


Mean node lifetime



Number of alive nodes

Figure 4.4: Simulation results with random topologies of 16 nodes (network lifetime, mean node lifetime, number of alive nodes). Absolute values.

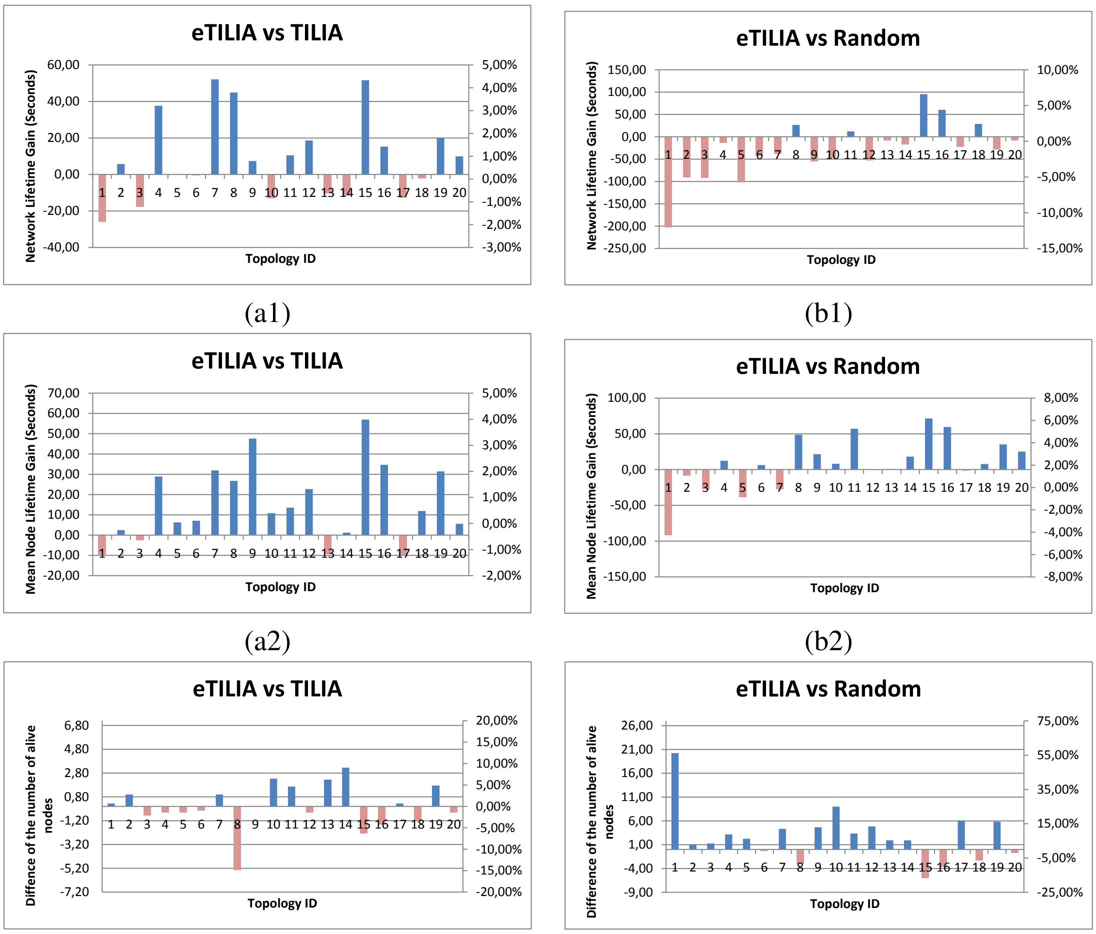


(a1) (a2)

(b1) (b2)

(c3) (c3)

Figure 4.5: Simulation results with random topologies of 16 nodes (received packets, lost packets, delay, gain average)

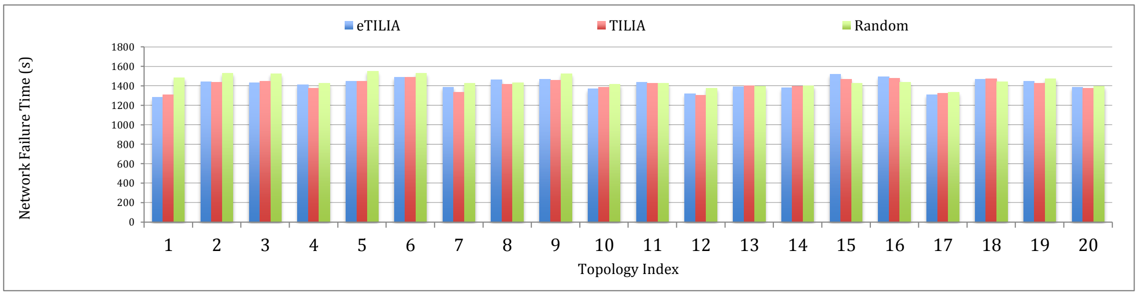


(a1) (b1)

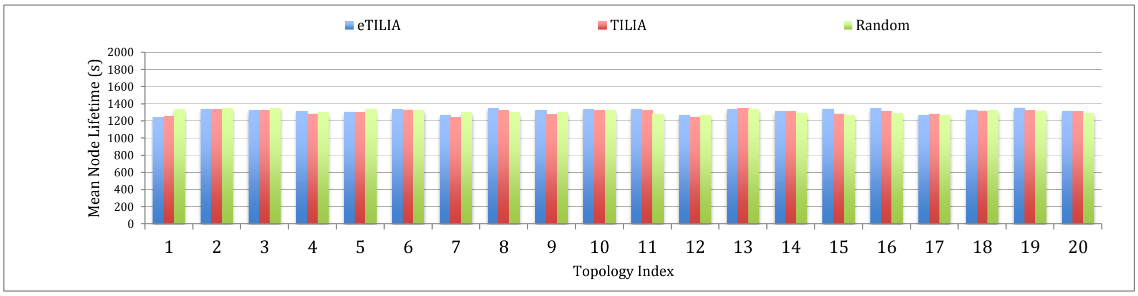
(a2) (b2)

(a3) (b3)

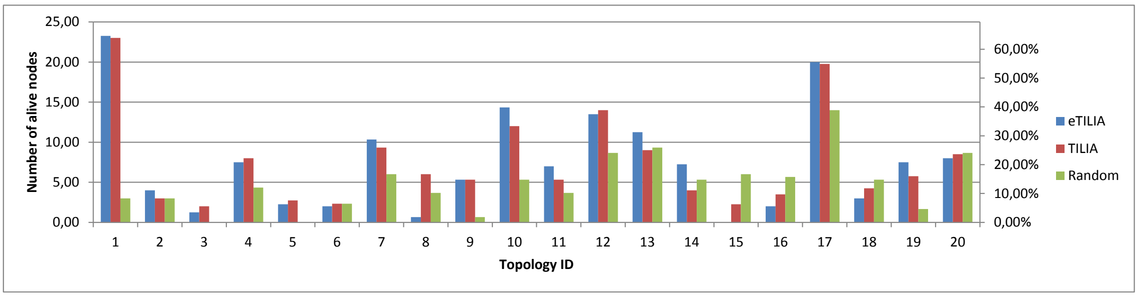
Figure 4.6: Simulation results with random topologies of 36 nodes (network lifetime, mean node lifetime, number of alive nodes)



Network lifetime



Mean node lifetime



Number of alive nodes

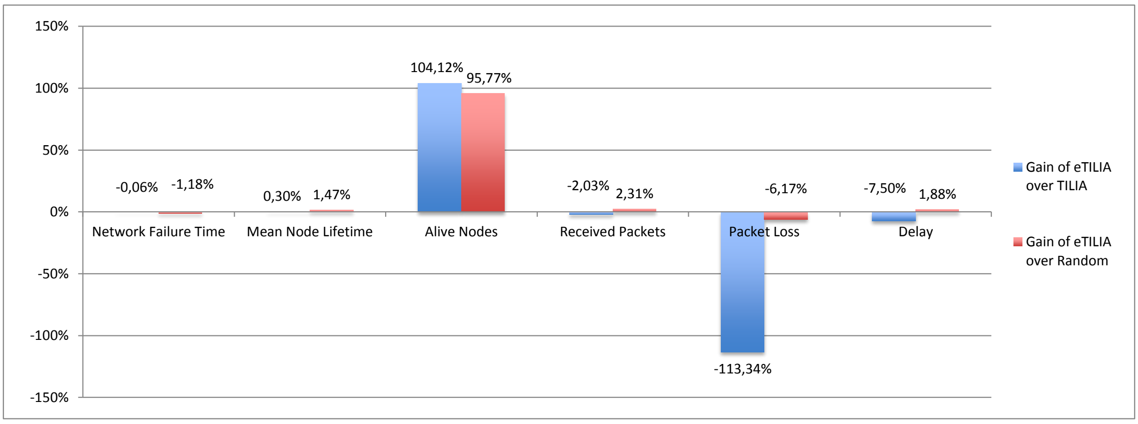
Figure 4.7: Simulation results with random topologies of 36 nodes (network lifetime, mean node lifetime, number of alive nodes). Absolute values.

(a1) (a2)

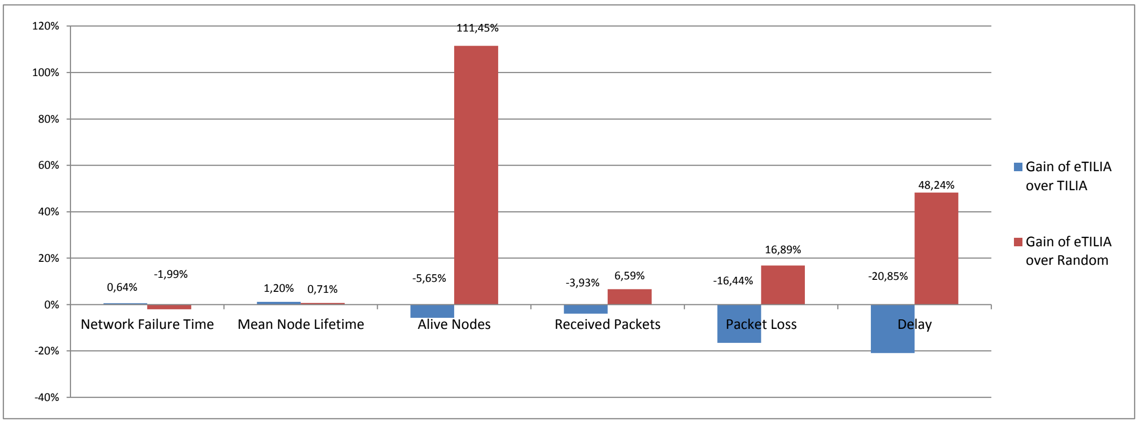
(b1) (b2)

(c1) (c2)

Figure 4.8: Simulation results with random topologies of 36 nodes (received packets, lost packets, delay, gain average)



Random Topologies of 16 nodes



Random Topologies of 36 nodes

Figure 4.9: Average Gains

4.5 Summary

In this chapter were presented the tools and methodology adopted to evaluate eTILIA algorithm, and were exposed the results obtained.

In the first section was described the NS-3 simulation tool. Besides this description, it were also presented and described the main models that were used, namely the mesh network model, the energy model and flow monitor model. The mesh network model implements an IEEE 802.11s wireless mesh network and focuses on three important aspects of this standard: adressing and forwarding, peering management and routing protocol. The energy model allows to simulate the energetic behaviour of the network and is composed by two parts: the energy source model and the energy consumption model. The energy source models represent the power supply of the network nodes, and the energy consumption model captures how the the energy is consumed by the nodes. The flow monitor model allow to keep track of all the packets of the simulation. With this mechanism a flow is characterized by its source/destination address, source/destination port and protocol, and each flow is independently monitored.

In the second section were described the regular and random topologies used in the simulations, and was exposed the algorithm created to generate the random topologies.

In the third section was described the methodology adopted in the simulations. The algorithm was tested with 20 random topologies of 16 nodes and with 20 random topologies of 36 nodes. Each simulation started by the creation and configuration of the topology. After this the eTILIA channel assignment algorithm was executed in order to assign an appropriate channel to each node. Then each node of the simulation was configured to send a traffic flow, with a constant bit-rate of 100kbps, to a certain gateway, during a time interval of 100 seconds. After these 100 seconds, the eTILIA channel assignment algorithm was executed again, and the channels of the nodes were modified accordingly. This step was repeated every 100 seconds, until the network fail.

In the fourth section were exposed and explained the results obtained. The main metrics evaluated were the network lifetime, the mean node lifetime, the number of alive nodes at the moment of the network failure, the percentage of packets received, the percentage of packets lost and the mean packet delay. The results of the topologies of 16 nodes and of the topologies of 36 nodes were separately aggregated. The main conclusion taken from these results is that eTILIA can improve the management of the energy resources of random topologies of 16 nodes, and can slightly improve the management of the energy resources of random topologies of 36 nodes.

Chapter 5

Conclusion and Future Work

This dissertation presents a networking solution to increase the capacity and the lifetime of CSMA based WMSNs. From the analysis of the state of the art solutions, regarding these topics, it was possible to see the different existent approaches to achieve these goals. The strategy followed in this dissertation to surpass these problems, was the adoption of a multiple communication channel scheme. This strategy allowed to increase the capacity of the network, by reducing its interference levels, and it also allowed to extend the network lifetime, by using the channel assignment algorithm proposed in this dissertation, eTILIA. eTILIA is a dynamic, centralized and energy-aware channel assignment algorithm, based in the TILIA algorithm described in [2], which makes the channel assignment decision based on the energy levels of the nodes, trying to avoid the traffic forwarding through nodes with low energy level.

Besides eTILIA, this dissertation also proposed an architecture for a low-cost, solar powered video-surveillance systems, based in Wi-Fi, to cover large unconnected areas such as parks or beaches. This proposal was based on the WMSN concept, and covered the physical components, the protocol stack and the mode of operation of the network.

The evaluation of the algorithm was made through computer simulation, using the NS-3 tool. This approach allowed to verify and validate the eTILIA algorithm, without the deployment of a complete test-bed, which would be very costly and time-consuming. The evaluation showed that eTILIA can provide a more efficient management of the energy resources of the network, in random topologies of 16 nodes, when compared to TILIA channel assignment algorithm and to a random channel assignment procedure. The evaluation also showed that eTILIA can slightly increase the network lifetime of random topologies of 36 nodes, when compared to TILIA channel assignment algorithm, and can provide a more efficient management of the energy resources of the network in random topologies of 36 nodes, when compared to a random channel assignment procedure.

5.1 Future Work

The first thing to do to improve this work would be to test eTILIA with a more significant number of topologies and traffic rates. This would allow to identify the characteristics of the topologies in which eTILIA could bring advantages. This would also allow to compare the topologies which present the best and the worst results, and understand what they have in common. The number of topologies simulated in this work was reduced. It were only simulated 40 topologies using a single traffic rate. The reason for this was the large time that each simulation required. Due to this the results obtained in this work are limited.

Test eTILIA in a real world test-bed would also improve this work. As it is common knowledge, computer simulation can provide accurate results, but cannot be compared to real world testing. Computer simulation is based on mathematical models of the real world, and since these models are not perfect, the simulation results obtained can often be different from the results obtained in real world conditions. Thus, the testing of eTILIA in a real world situation would really improve this work. This would require the deployment and configuration of a WMSN. To do this it is possible to take advantage from the architecture described in Chapter 3, to deploy a low-cost, solar powered video-surveillance system. With this system it would be possible to obtain the required data to evaluate the eTILIA algorithm. Then, it could be done a statistical analysis of the data, in order to precisely characterize the behaviour of the eTILIA algorithm.

Another improvement to this work would be to upgrade eTILIA. One modification that could be made in the eTILIA algorithm would be to take into consideration, not only the energy level of the nodes, but also the ratio between the energy spent and the energy being harvested at a certain moment. This could bring advantages in terms of energy efficiency.

Appendix A

eTILIA Python Script

This appendix exposes the main modification made in the TILIA python script in order to adapt it to eTILIA. Additional changes were also required to integrate the energy parameter in the eTILIA script, and to adapt the script to the NS-3 simulations. These changes are not here exposed.

A.1 TILIA script

```python
# Selects all the neighbours of the node in analysis which already have a channel assigned
a = [n for n in G.nodes() if G.node[n]["channel"]!=-1]
neighs = [b for b in G.neighbors(nid) if b in a]

# From the set 'neighs' selects the neighbours with an inferior hop count to the gateway (upstream neighbours)
hc = G.node[nid]["hop_count"]
if hc != -1:
  neighs = [b for b in neighs if G.node[b]["hop_count"] <= hc]

# Selects all the channels of all the upstream neighbours
channels = list(set([G.node[n]["channel"] for n in neighs]))

# From the set 'channels' previously created selects the channels which have the minimum load
best_chans = min_load_gw(G, channels,nid)

# From the set 'neighs' selects the nodes which operate in one of the channels of the set 'best_chans'
ch_neighs=[n for n in neighs if G.node[n]["channel"] in best_chans]

# From the set 'ch_neighs' selects the ones which have the a minimum hop count
hc = min([G.node[n]["hop_count"] for n in ch_neighs])
up_neighs = [n for n in ch_neighs if G.node[n]["hop_count"]==hc]

# From the set 'up_neighs' selects the ones which have the a minimum load
up_neihs1 = min_load_gw(G, up_neighs,nid, [], parent_type)

# If at this point there is more than one parent alternative
if len(up_neihs1) > 1:
  up_neihs2 = min_up_hidden_nodes(G, up_neihs1,nid, direction)
else:
  up_neihs2 = up_neihs1


if (len(up_neihs2)>1) & (hc <=1) :

  list_of_new_graphs = []

  for idxparent, parent in enumerate(up_neihs2):
    list_of_new_graphs.append(G.copy())
    list_of_new_graphs[-1].name = str(nid)+arr+str(parent)
    for n,p in zip(rec_node,rec_parent):
      list_of_new_graphs[-1].name+=' '+str(n)+arr+str(p)
```

```
  for idxparent, p in enumerate(up_neihs2):
    J = list_of_new_graphs[idxparent]

    rn =  [n for n in rec_node]
    rp = [n for n in rec_parent]
    if nid not in rec_node:
      rn.append(nid)
      rp.append(p)

        nG = newG + idxparent
        a = [n for n in assigned_nodes]
        nl = recursive_tilia(J, nG, direction, rn, rp, a )
        PossibleG.extend(nl)
  return PossibleG

else:
  assign_parent_to_node(G, up_neihs2[0], nid,parent_type)
```

A.2 eTILIA script

```
# Selects all the neighbours of the node in analysis which already have a channel assigned
a = [n for n in G.nodes() if G.node[n]["channel"]!=-1]
neighs = [b for b in G.neighbors(nid) if b in a]

# From the set 'neighs' selects the neighbours with an inferior hop count to the gateway (upstream neighbours)
hc = G.node[nid]["hop_count"]
if hc != -1:
  neighs = [b for b in neighs if G.node[b]["hop_count"] <= hc]

# Selects all the channels of all the upstream neighbours
channels = list(set([G.node[n]["channel"] for n in neighs]))

# From the set 'channels' previously created selects the channels which have the minimum load
best_chans = min_load_gw(G, channels,nid)

# From the set 'neighs' selects the nodes which operate in one of the channels of the set 'best_chans'
ch_neighs=[n for n in neighs if G.node[n]["channel"] in best_chans]

# From the set 'ch_neighs' selects the ones which have the a minimum hop count
hc = min([G.node[n]["hop_count"] for n in ch_neighs])
up_neighs = [n for n in ch_neighs if G.node[n]["hop_count"]==hc]

energyLowHopNeigh=[]
k=0
for n in up_neighs:
  energyLowHopNeigh.append([float(G.node[n]["energy"]),int(n),k])
  k=k+1

energyLowHopNeigh=sorted(energyLowHopNeigh)

indiceHighestEnergyParent=energyLowHopNeigh[len(energyLowHopNeigh)-1][2]
# Selects the highest energy candidate parent as the parent node
assign_parent_to_node(G, up_neighs[indiceHighestEnergyParent], nid,parent_type)
```

Appendix B

NS-3 Source Code Modifications

In order to simulate the desired networks it was necessary to modify the source code of NS-3. The modifications focused mainly on the NS-3 Mesh Model, referred in section 4.1.1.1, and on the NS-3 Energy Model, referred in section 4.1.1.2. These modifications are now exposed:

Modification 1: In the beginning of the simulation process it was not possible to deploy a functional multi-hop network. After searching the reason for this problem in the appropriate discussion forums it was possible to conclude that the problem was related with the zero processing time of the node's packets, which led to constant collision. The discussion about this bug can be seen in [63]. To solve this problem we introduce the following modifications in the *src/mesh/model/mesh-wifi-interface-mac.cc* file:

Original Version

```
m_edca[ac]->Queue (packet, hdr);
```

Modified Version

```
double uniformVar= rand() / (RAND_MAX + 1.0) * (100 - 0) + 0;
Time delay = MicroSeconds(uniformVar);
Simulator::Schedule(delay, &EdcaTxopN::Queue,m_edca[ac],packet,hdr);
```

Original Version

```
if (hdr.GetAddr1 () != Mac48Address::GetBroadcast ())
{
  m_edca[AC_VO]->Queue (packet, header);
}
else
{
  m_edca[AC_BK]->Queue (packet, header);
}
```

Modified Version

```
if (hdr.GetAddr1 () != Mac48Address::GetBroadcast ())
{
  double uniformVar= rand() / (RAND_MAX + 1.0) * (100 -
       0) + 0;
  Time delay = MicroSeconds(uniformVar);
  Simulator::Schedule(delay, &EdcaTxopN::Queue,m_edca[
       AC_VO],packet,header);
}
else
{
  double uniformVar= rand() / (RAND\_MAX + 1.0) * (100 -
       0) + 0;
  Time delay = MicroSeconds(uniformVar);
  Simulator::Schedule(delay, &EdcaTxopN::Queue,m_edca[
       AC_BK],packet,header);
}
```

Modification 2: Add the files GraphNode.c and GraphNode.h to the Mesh Model. In these files it is defined a new class to represent a graph node. The most important characteristics of the node are the node id, the node position, the type of node and its neighbours. In these files it is also defined a method, which given a set of graph nodes can create a .dot file with the network topology information. This file can then be processed by the Pygraphviz tool, referred in [23], to produce a visual representation of the network graph.

Modification 3: Another important modification was to adapt the source code to enable the installation of the energy model in the mesh radios. To do this we adapted the code which installs the energy model on wifi-radios, to install it on mesh radios. So we created the file

To do this we created the files *src/energy/helper/mesh-radio-energy-model-helper.cc* and *src/energy/helper/mesh-radio-energy-model-helper.h*, in which we declare a new class named *MeshRadioEnergyModelHelper*. These files are identical to the files *src/energy/helper/wifi-radio-energy-model-helper.cc* and *src/energy/helper/wifi-radio-energy-model-helper.h*, and had to be declared in the file *src/energy/wscript*. The main modification that was made is now presented:

wifi-radio-energy-model-helper.cc

```
wifiPhy->RegisterListener (model->GetPhyListener ());
if (m_txCurrentModel.GetTypeId ().GetUid ())
{
  Ptr<WifiTxCurrentModel> txcurrent = m_txCurrentModel.
        Create<WifiTxCurrentModel> ();
  model->SetTxCurrentModel (txcurrent);
}
return model;
```

mesh-radio-energy-model-helper.cc

```
Ptr<MeshPointDevice> mp = DynamicCast<MeshPointDevice> (
      device);
std::vector<Ptr<NetDevice> > interfaces = mp->
      GetInterfaces();
for (std::vector<Ptr<NetDevice> >::const_iterator i =
      interfaces.begin (); i != interfaces.end (); i++)
{
  Ptr<WifiNetDevice> wifiDevice = (*i)->GetObject<
        WifiNetDevice> ();
  Ptr<WifiPhy> wifiPhy = wifiDevice->GetPhy ();
  wifiPhy->RegisterListener (model->GetPhyListener ());
}
```

Modification 4: Another modification was made in the file *src/mesh/model/dot11s/hwmp-protocol.cc* in order to modify the minimum TTL value possible from 2 to 1. This was required to be able to implement the neighbour discovery procedure referred in section 4.2.2

Original Version

```
.AddAttribute ( "MaxTtl",
  "Initial value of Time To Live field",
  UintegerValue (32),
  MakeUintegerAccessor (
 &HwmpProtocol::m_maxTtl),
  MakeUintegerChecker<uint8_t> (2)
)
```

Modified Version

```
.AddAttribute ( "MaxTtl",
  "Initial value of Time To Live field",
  UintegerValue (32),
  MakeUintegerAccessor (
 &HwmpProtocol::m_maxTtl),
  MakeUintegerChecker<uint8_t> (1)
)
```

Modification 5: To access to the routing table of the HWMP protocol, in order to make the required modifications, it was necessary to create a new method. To create this method it was necessary to modify the files *src/mesh/model/dot11s/hwmp-protocol.h* and *src/mesh/model/dot11s/hwmp-protocol.cc*. The created method is now exposed:

hwmp-protocol.h

```
Ptr<HwmpRtable> Get_m_rtable();
)
```

hwmp-protocol.cc

```
Ptr<HwmpRtable> HwmpProtocol::Get_m_rtable()
{
  return m_rtable;
}
```

Modification 6: To analyse the results of the simulation it was necessary to create a method to print the routing table of the HWMP protocol. To create this method it was necessary to modify the files *src/mesh/model/dot11s/hwmp-rtable.h* and *src/mesh/model/dot11s/hwmp-rtable.cc*. The created method is now exposed:

hwmp-rtable.h

```
void Print_m_routes();
```

hwmp-rtable.cc

```
void HwmpRtable::Print_m_routes(){
  uint8_t bufferOrigin[6],bufferRetransmitter[6];

    for(int i=0; i<(int) m_routes.size();i++){

      std::map<Mac48Address, ReactiveRoute>::iterator it = m_routes.begin();
      std::advance(it, i);
      Mac48Address address = it->first;
      dot11s::HwmpRtable::ReactiveRoute route = it->second;

      (address).CopyTo(bufferOrigin);
      printf("Destination: %02X:%02X:%02X:%02X:%02X:%02X,",bufferOrigin[0],bufferOrigin
            [1],bufferOrigin[2],bufferOrigin[3],bufferOrigin[4],bufferOrigin[5]);

      (route.retransmitter).CopyTo(bufferRetransmitter);

      printf("interface: %i, ",route.interface);
      printf("metric: %i, ",route.metric);
      printf("nextHop: %02X:%02X:%02X:%02X:%02X:%02X, ",bufferRetransmitter[0],
            bufferRetransmitter[1],bufferRetransmitter[2],bufferRetransmitter[3],
            bufferRetransmitter[4],bufferRetransmitter[5]);
      printf("seqnum: %i, ",route.seqnum);
      printf("timeToExpire: %f\n",(route.whenExpire).GetSeconds());


    }
}
```

Appendix C

Regular Topologies

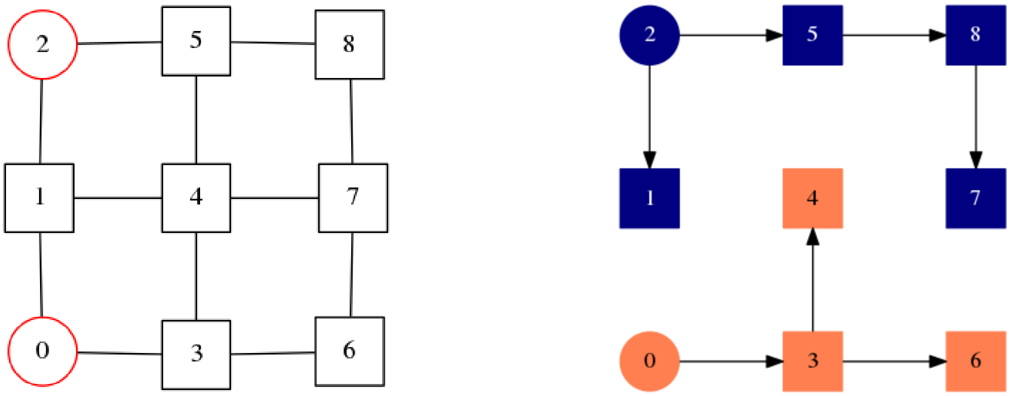


Figure C.1: 9 Nodes, 2 Gateways: Original Topology vs First eTilia Topology

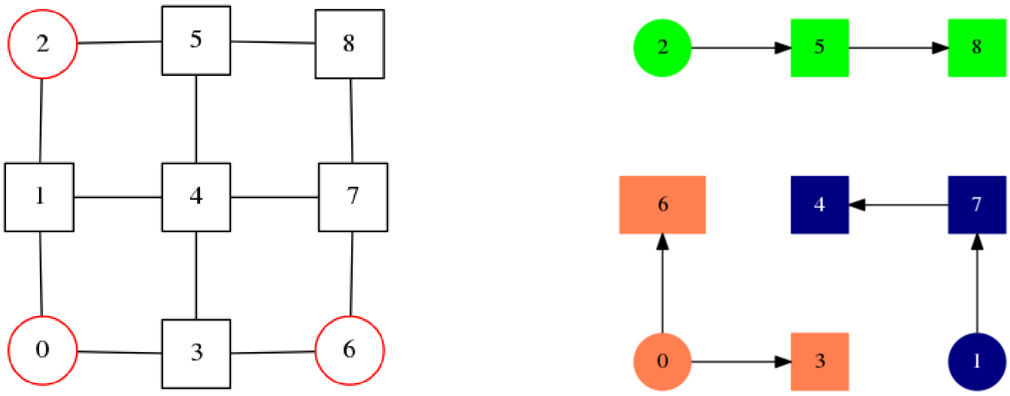


Figure C.2: 9 Nodes, 3 Gateways: Original Topology vs First eTilia Topology

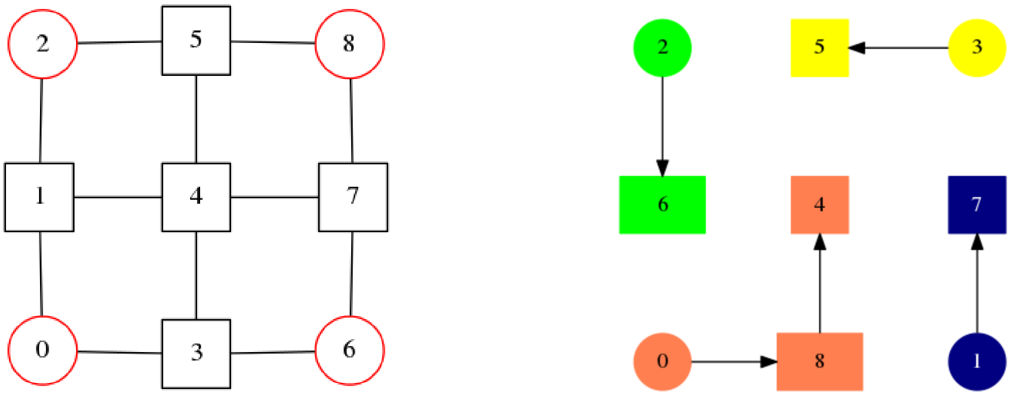


Figure C.3: 9 Nodes, 4 Gateways: Original Topology vs First eTilia Topology

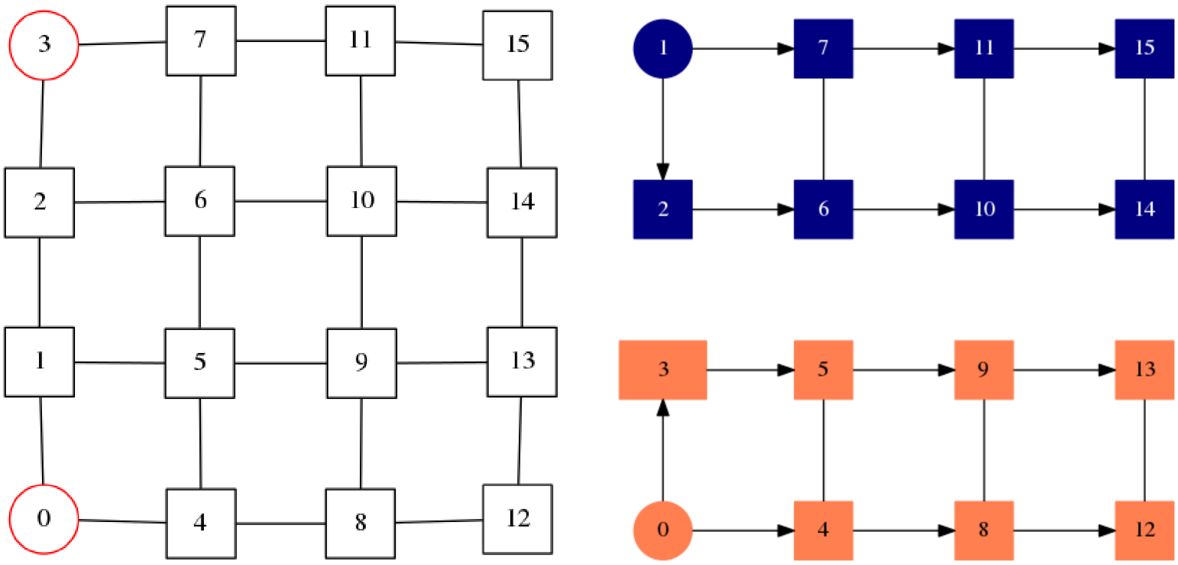


Figure C.4: 16 Nodes, 2 Gateways: Original Topology vs First eTilia Topology

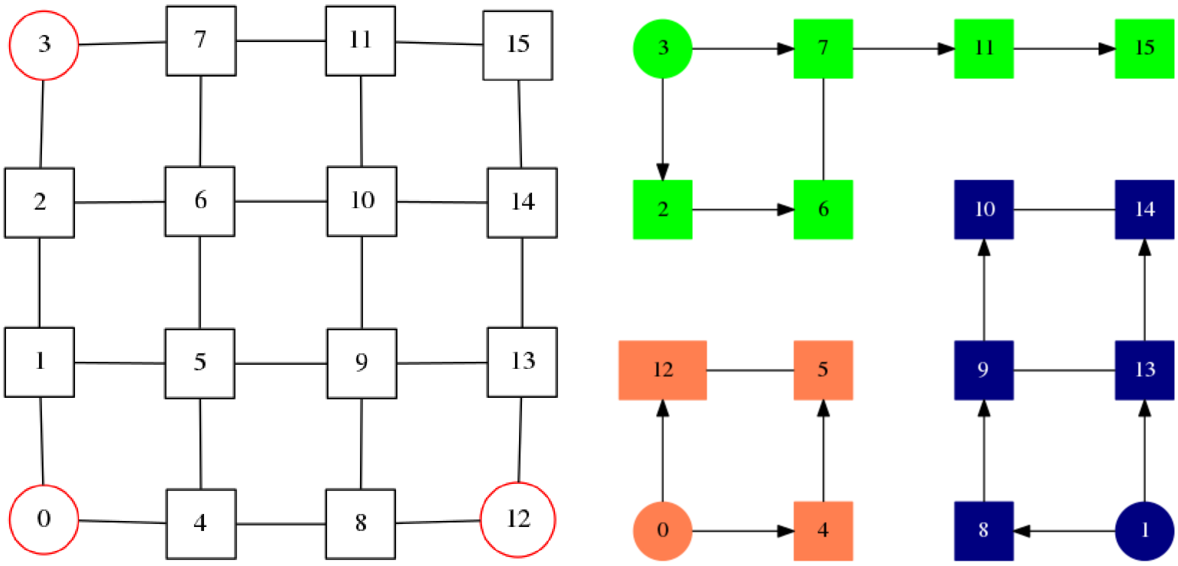


Figure C.5: 16 Nodes, 3 Gateways: Original Topology vs First eTilia Topology

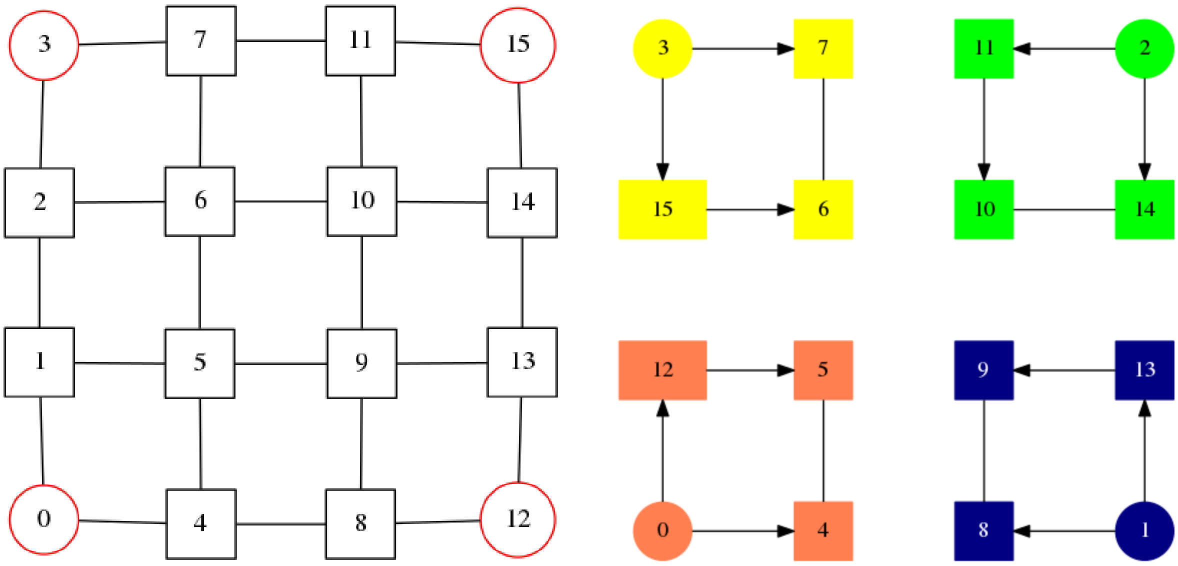


Figure C.6: 16 Nodes, 4 Gateways: Original Topology vs First eTilia Topology

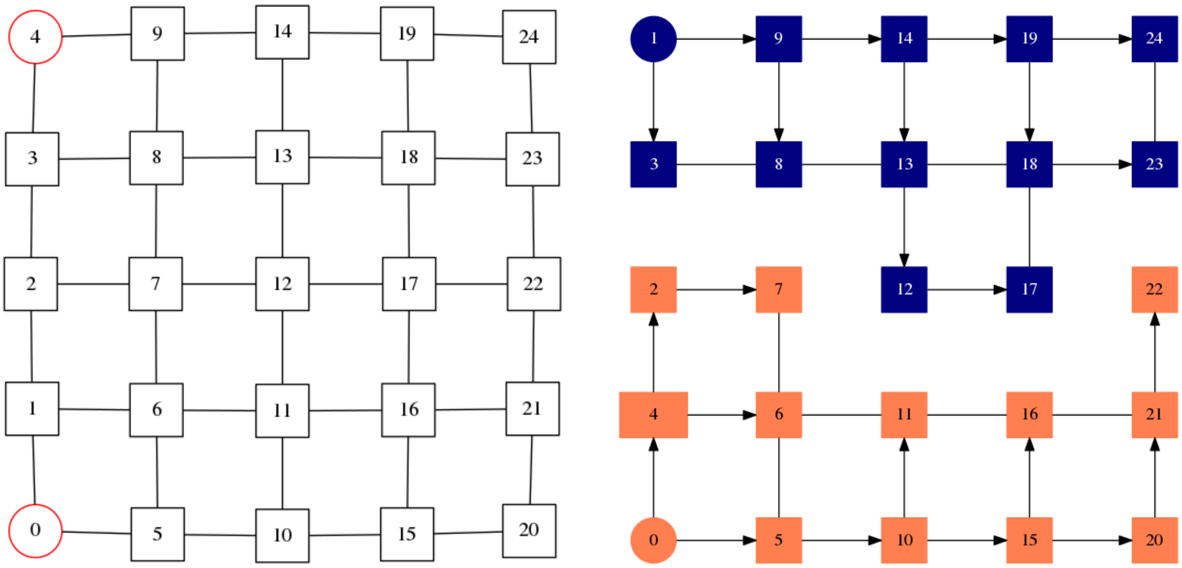


Figure C.7: 25 Nodes, 2 Gateway: Original Topology vs First eTilia Topology

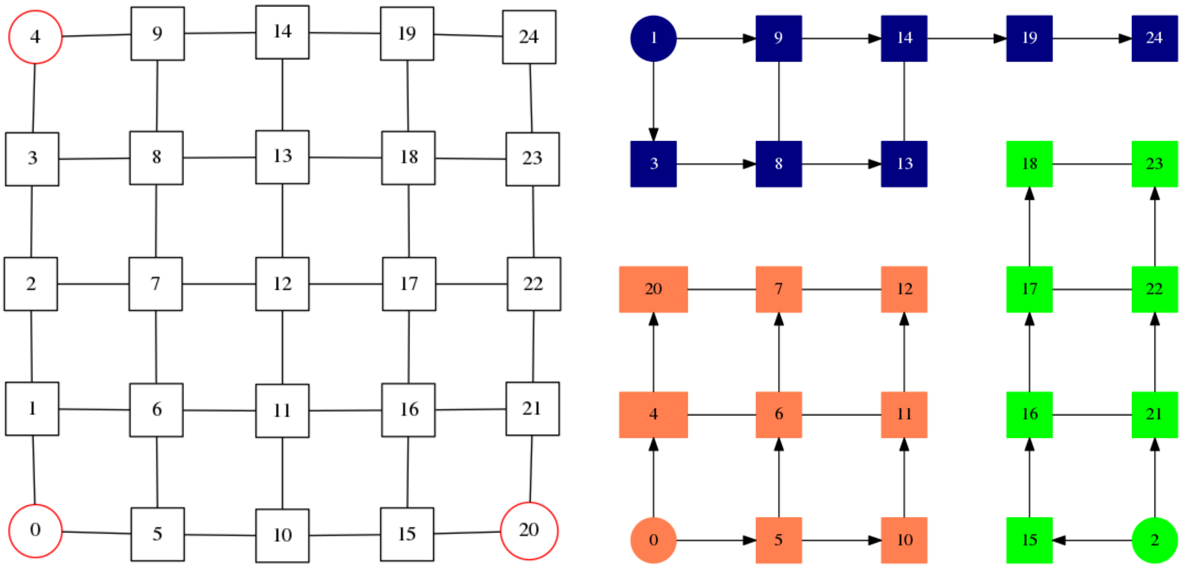


Figure C.8: 25 Nodes, 3 Gateway: Original Topology vs First eTilia Topology

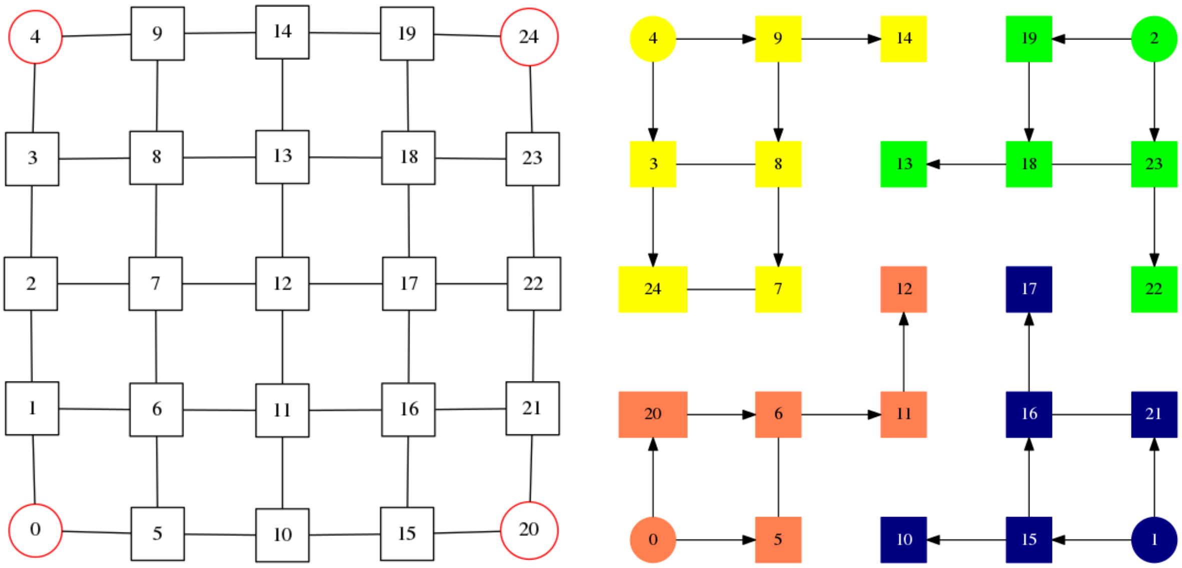


Figure C.9: 25 Nodes, 4 Gateway: Original Topology vs First eTilia Topology

Appendix D

Random Topologies

D.1 16 Nodes Topologies

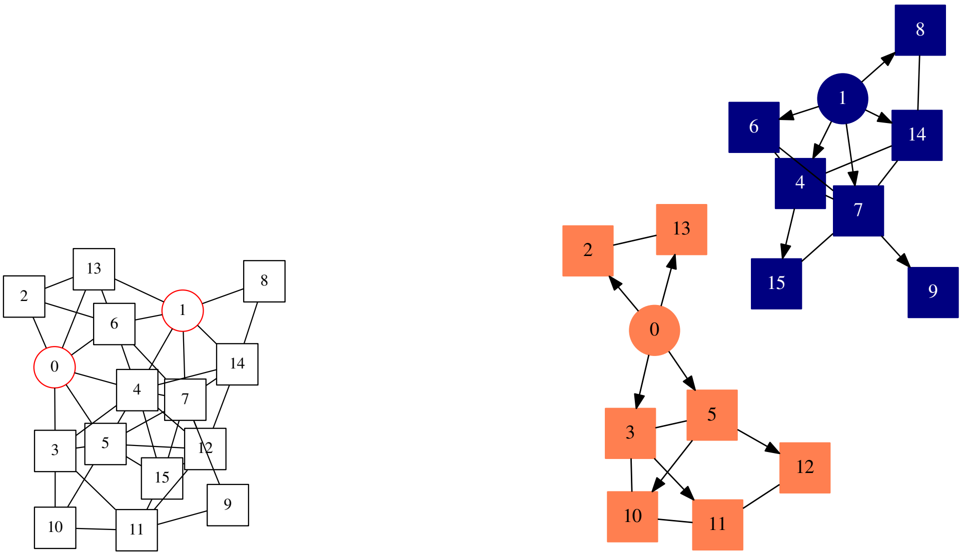


Figure D.1: 16 Nodes, 2 Gateways, Topology 1: Original Topology vs First eTilia Topology

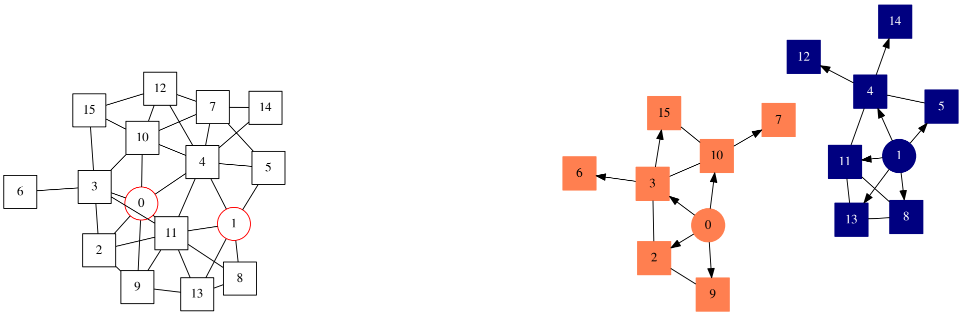


Figure D.2: 16 Nodes, 2 Gateways, Topology 2: Original Topology vs First eTilia Topology

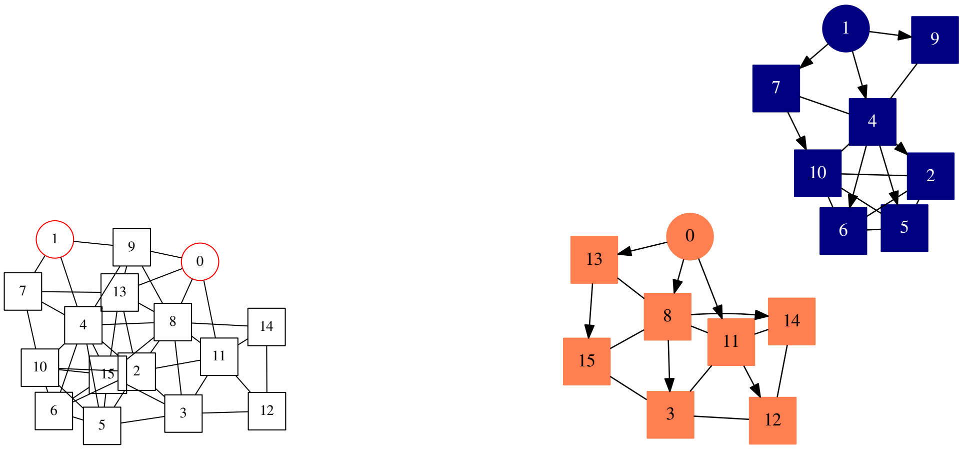


Figure D.3: 16 Nodes, 2 Gateways, Topology 3: Original Topology vs First eTilia Topology

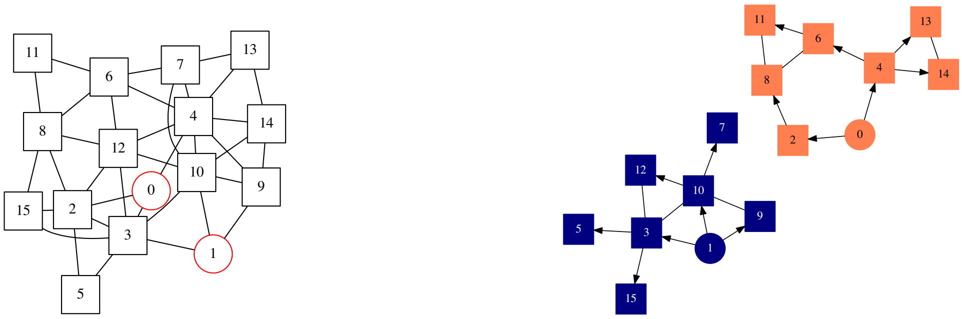


Figure D.4: 16 Nodes, 2 Gateways, Topology 4: Original Topology vs First eTilia Topology

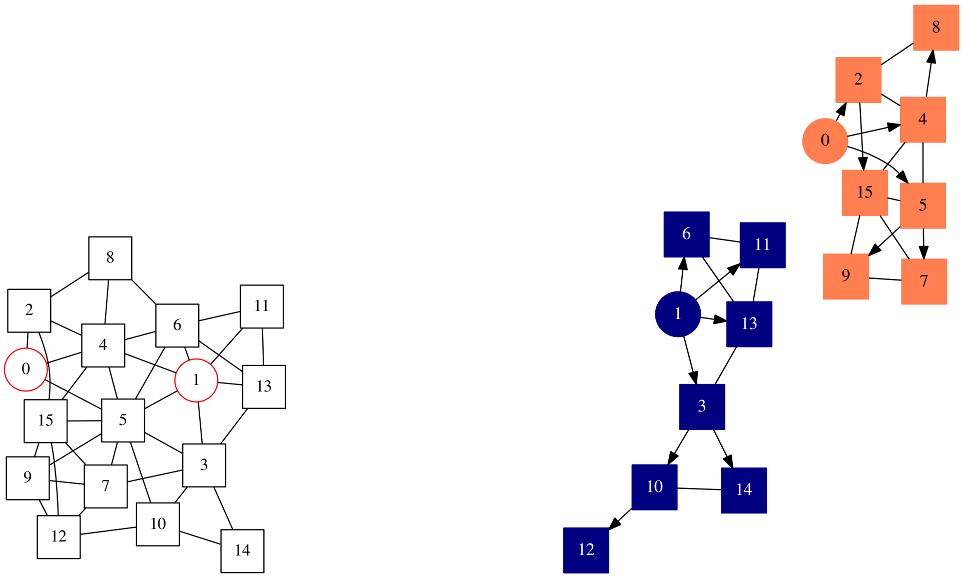


Figure D.5: 16 Nodes, 2 Gateways, Topology 5: Original Topology vs First eTilia Topology

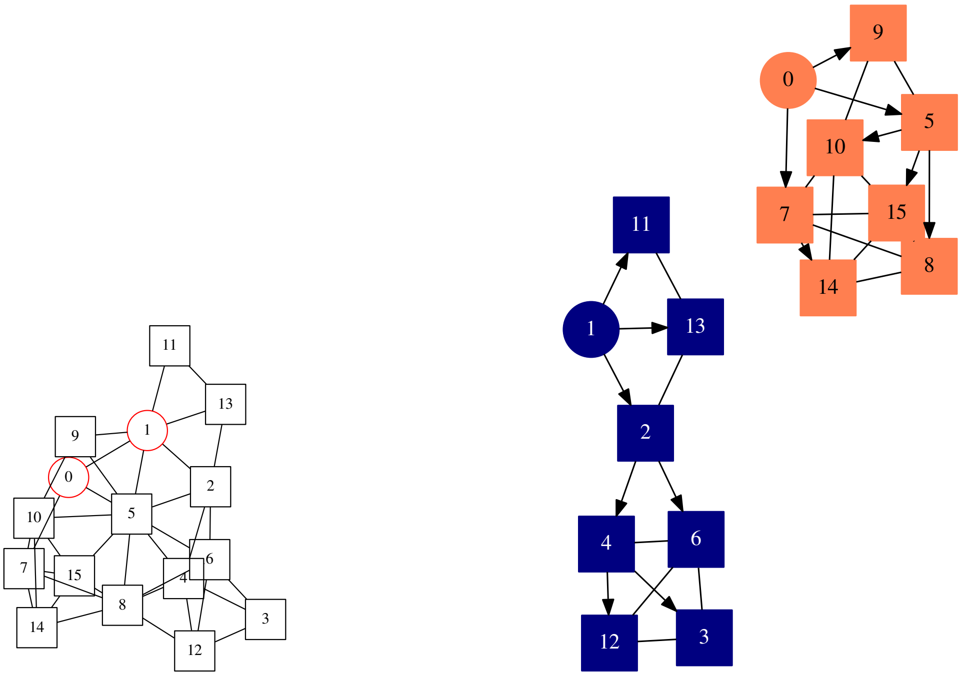


Figure D.6: 16 Nodes, 2 Gateways, Topology 6: Original Topology vs First eTilia Topology

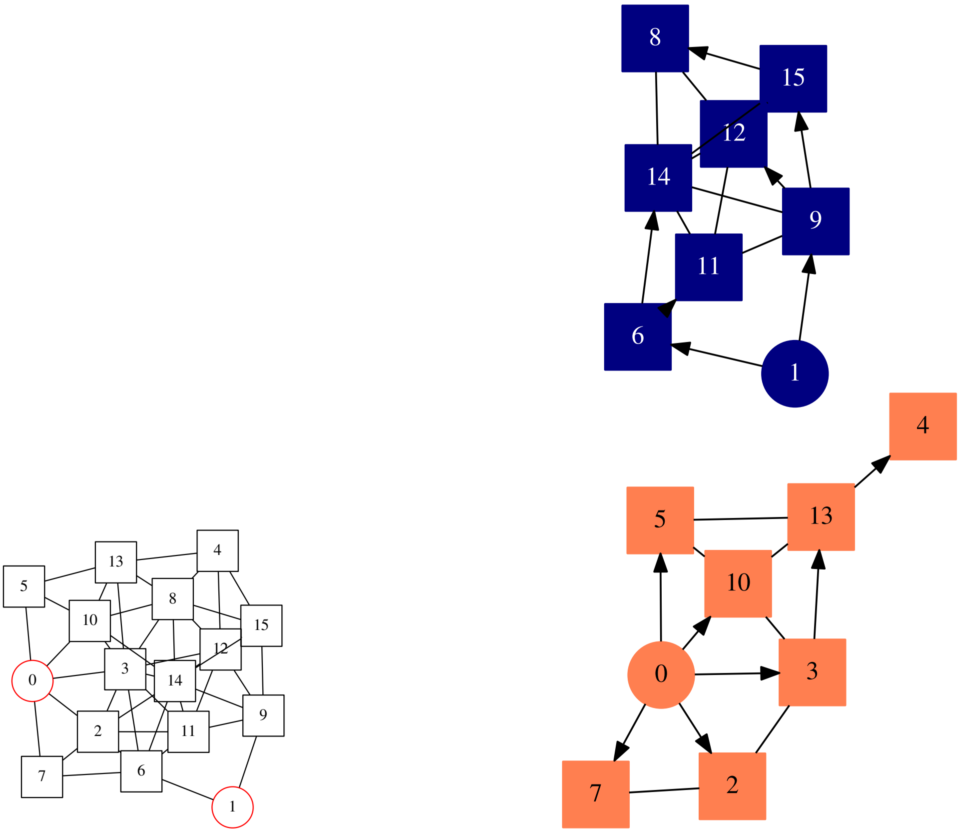


Figure D.7: 16 Nodes, 2 Gateways, Topology 7: Original Topology vs First eTilia Topology

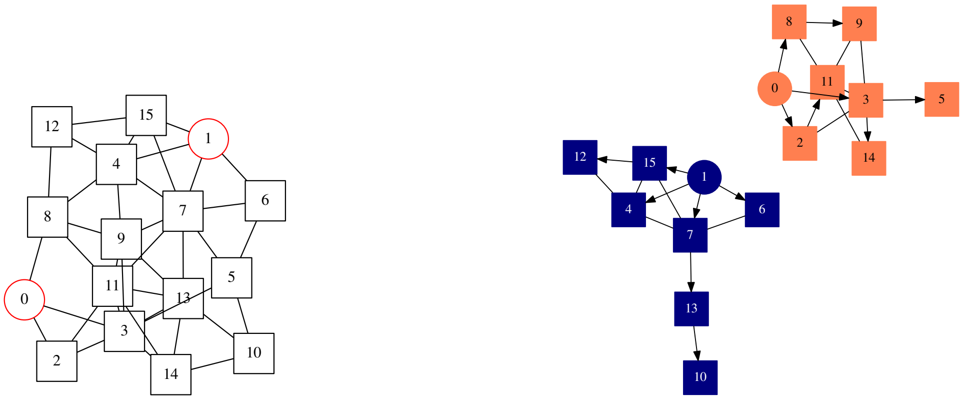


Figure D.8: 16 Nodes, 2 Gateways, Topology 8: Original Topology vs First eTilia Topology

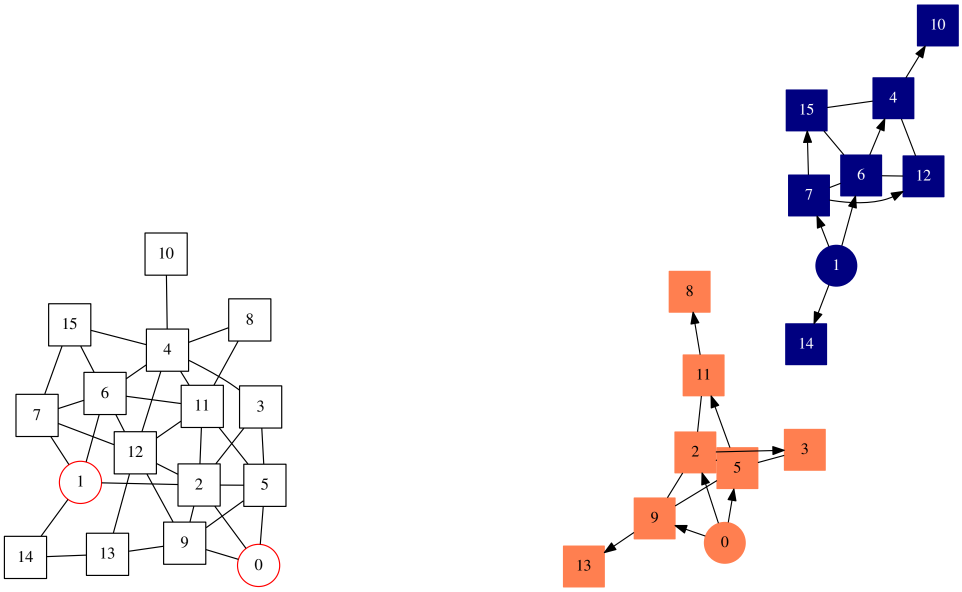


Figure D.9: 16 Nodes, 2 Gateways, Topology 9: Original Topology vs First eTilia Topology

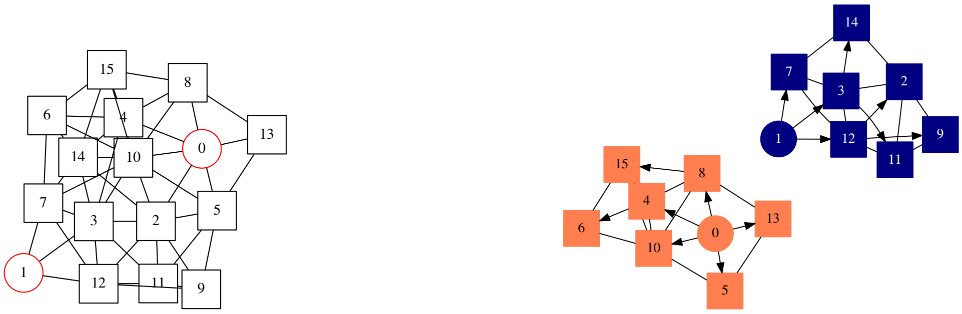


Figure D.10: 16 Nodes, 2 Gateways, Topology 10: Original Topology vs First eTilia Topology

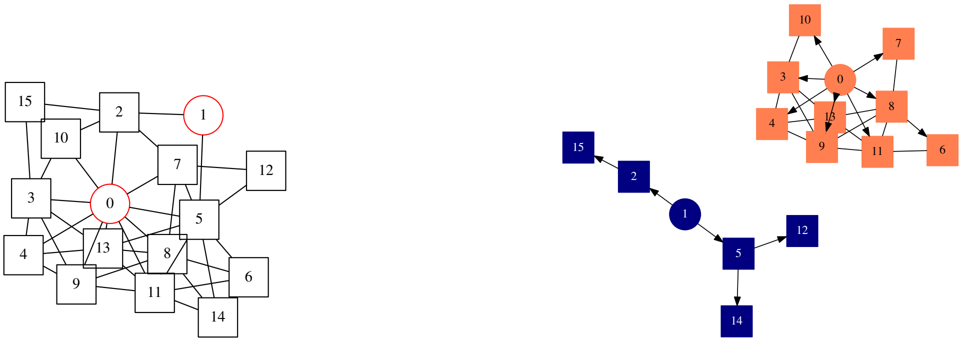


Figure D.11: 16 Nodes, 2 Gateways, Topology 11: Original Topology vs First eTilia Topology

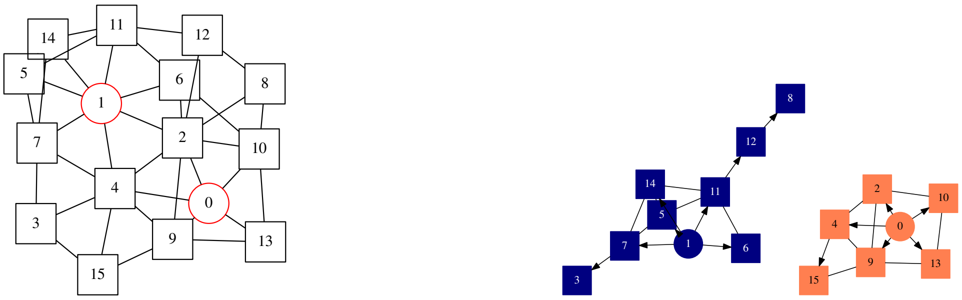


Figure D.12: 16 Nodes, 2 Gateways, Topology 12: Original Topology vs First eTilia Topology

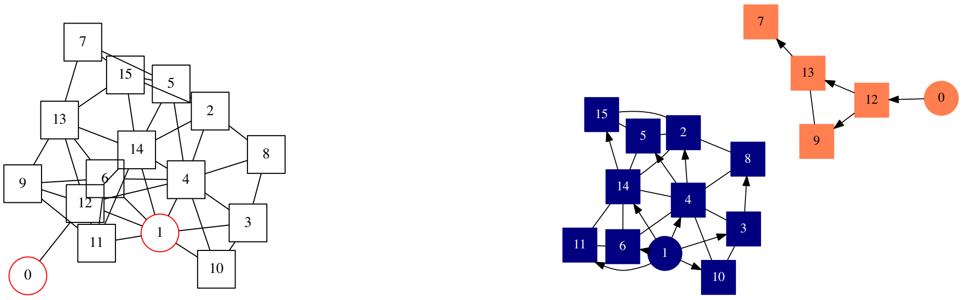


Figure D.13: 16 Nodes, 2 Gateways, Topology 13: Original Topology vs First eTilia Topology

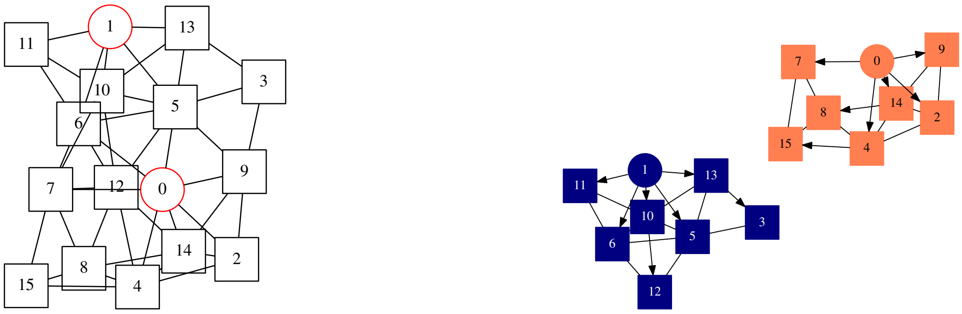


Figure D.14: 16 Nodes, 2 Gateways, Topology 14: Original Topology vs First eTilia Topology

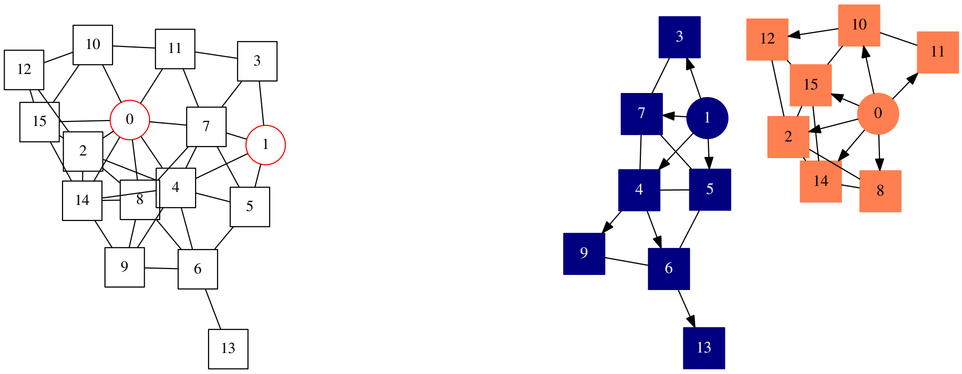


Figure D.15: 16 Nodes, 2 Gateways, Topology 15: Original Topology vs First eTilia Topology

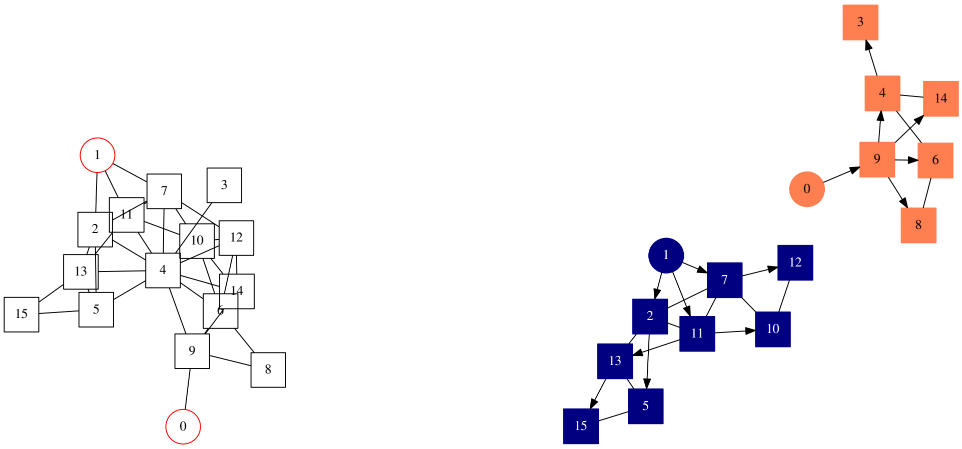


Figure D.16: 16 Nodes, 2 Gateways, Topology 16: Original Topology vs First eTilia Topology

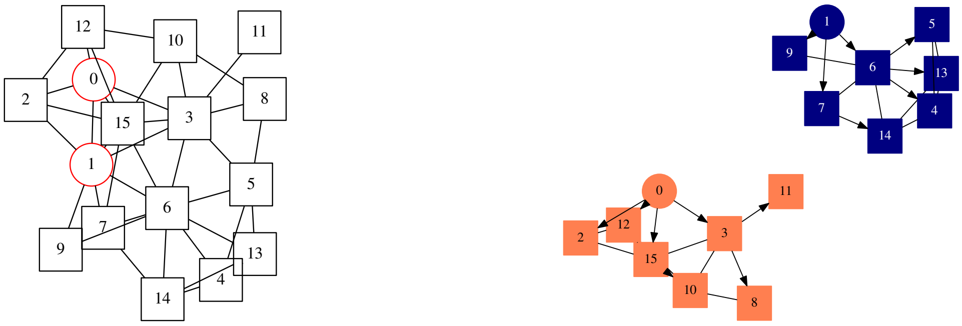


Figure D.17: 16 Nodes, 2 Gateways, Topology 17: Original Topology vs First eTilia Topology

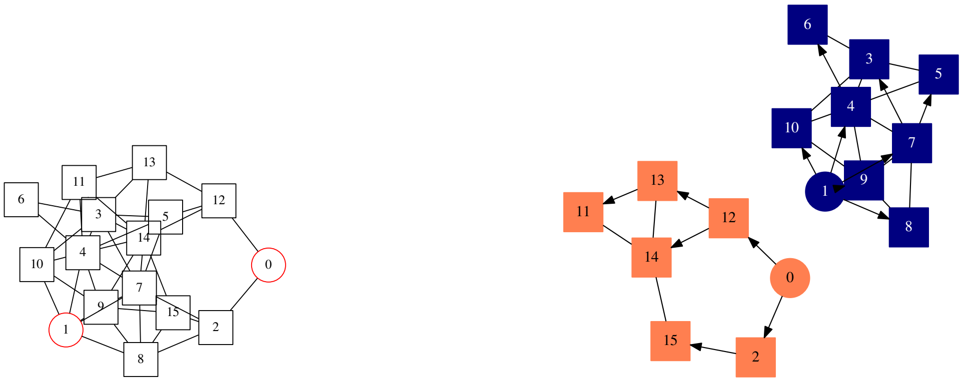


Figure D.18: 16 Nodes, 2 Gateways, Topology 18: Original Topology vs First eTilia Topology

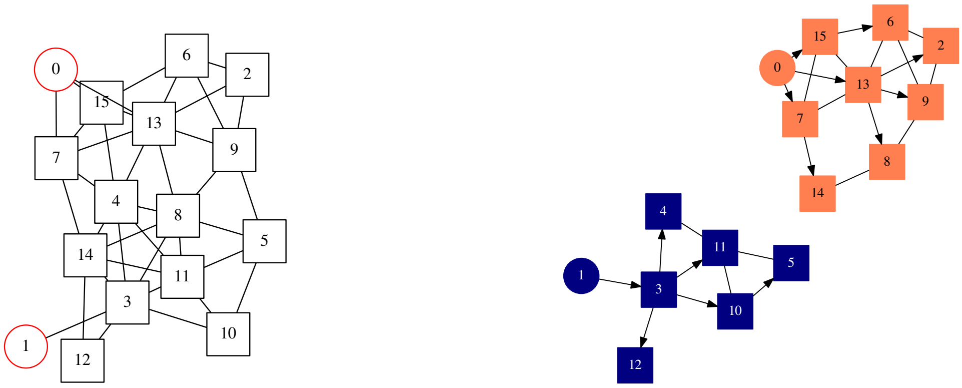


Figure D.19: 16 Nodes, 2 Gateways, Topology 19: Original Topology vs First eTilia Topology

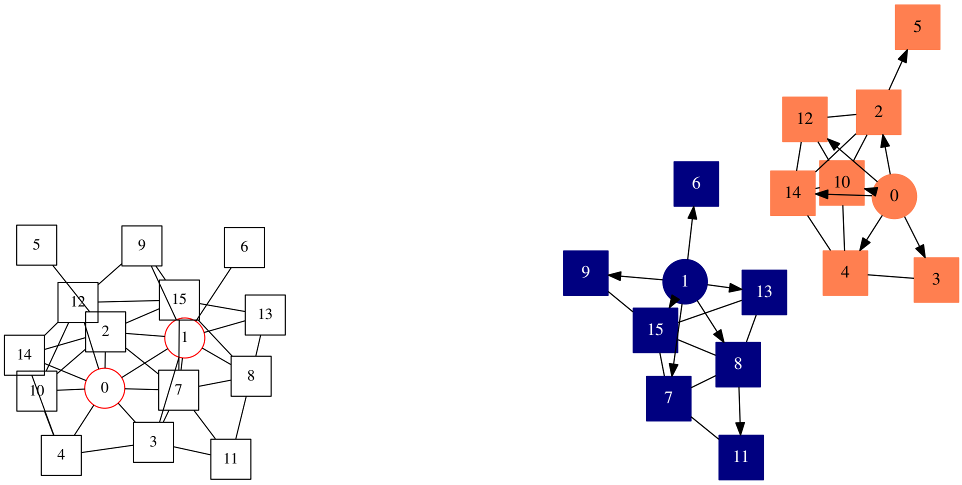


Figure D.20: 16 Nodes, 2 Gateways, Topology 20: Original Topology vs First eTilia Topology

D.2 36 Nodes Topologies

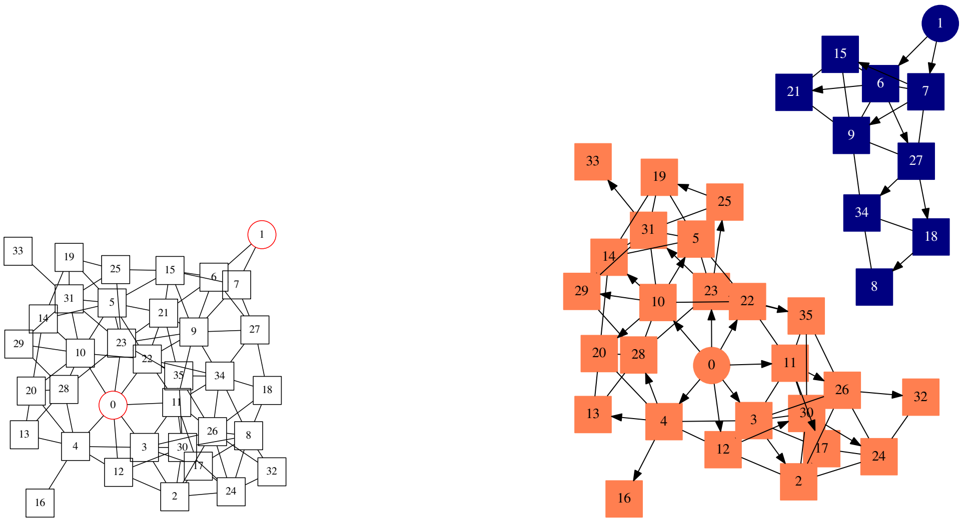


Figure D.21: 36 Nodes, 2 Gateways, Topology 1: Original Topology vs First eTilia Topology

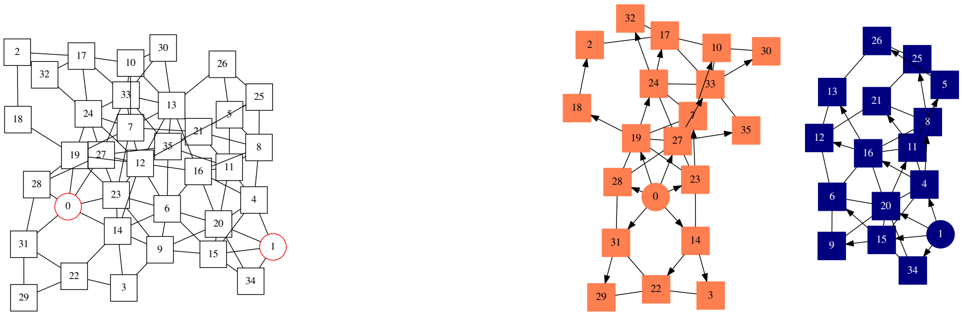


Figure D.22: 36 Nodes, 2 Gateways, Topology 2: Original Topology vs First eTilia Topology

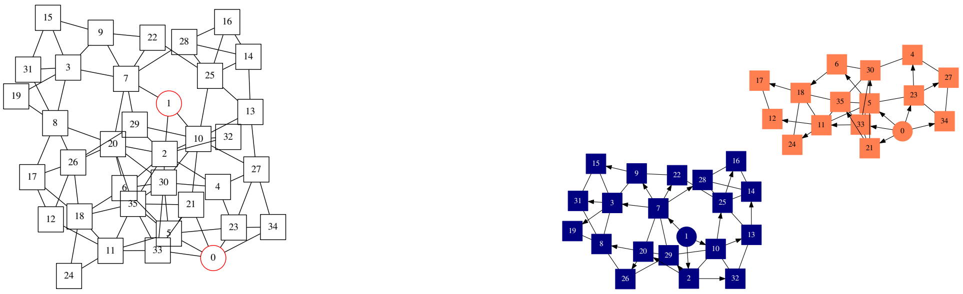


Figure D.23: 36 Nodes, 2 Gateways, Topology 3: Original Topology vs First eTilia Topology

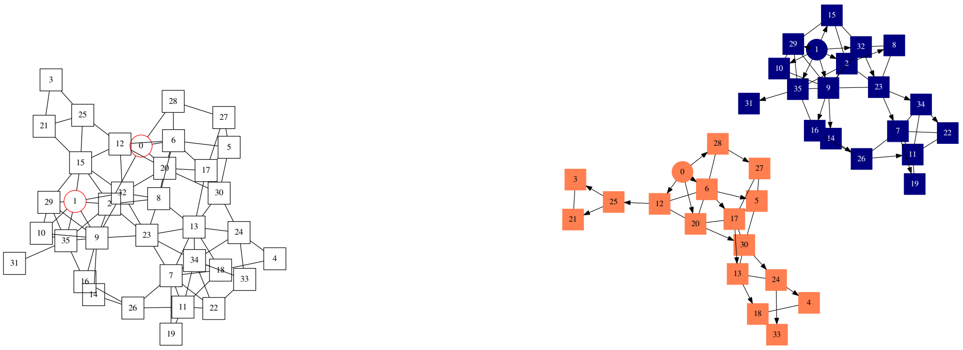


Figure D.24: 36 Nodes, 2 Gateways, Topology 4: Original Topology vs First eTilia Topology

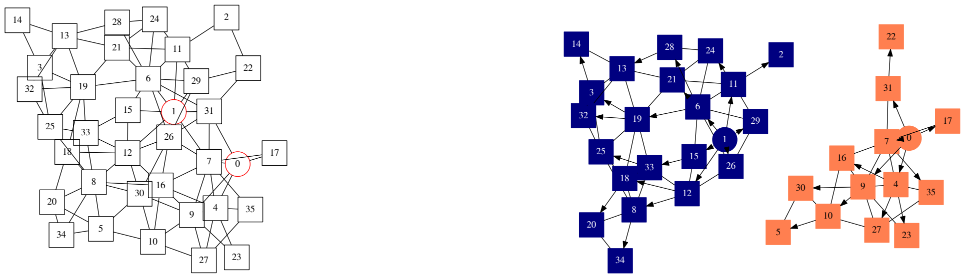


Figure D.25: 36 Nodes, 2 Gateways, Topology 5: Original Topology vs First eTilia Topology

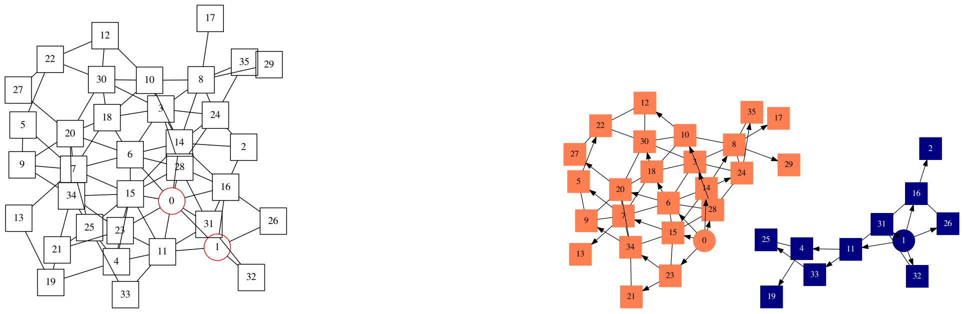


Figure D.26: 36 Nodes, 2 Gateways, Topology 6: Original Topology vs First eTilia Topology

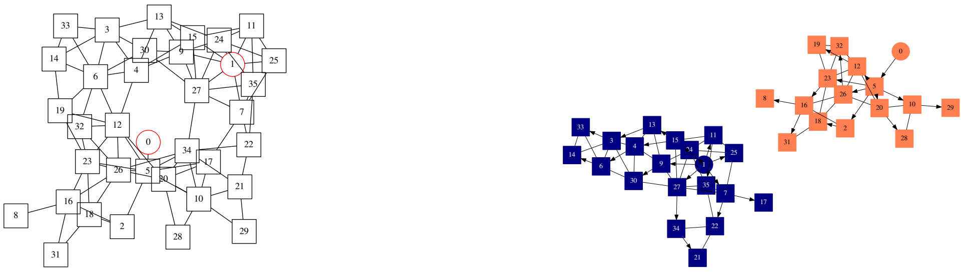


Figure D.27: 36 Nodes, 2 Gateways, Topology 7: Original Topology vs First eTilia Topology

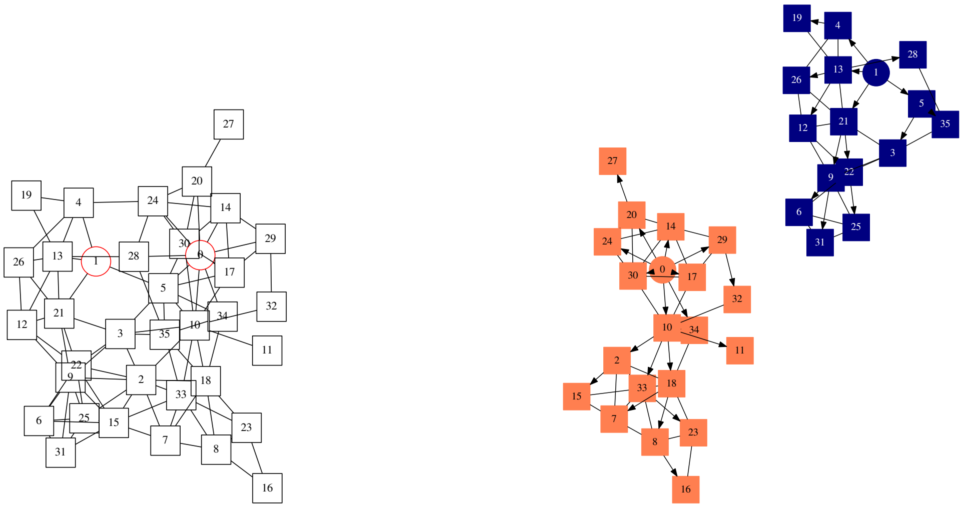


Figure D.28: 36 Nodes, 2 Gateways, Topology 8: Original Topology vs First eTilia Topology

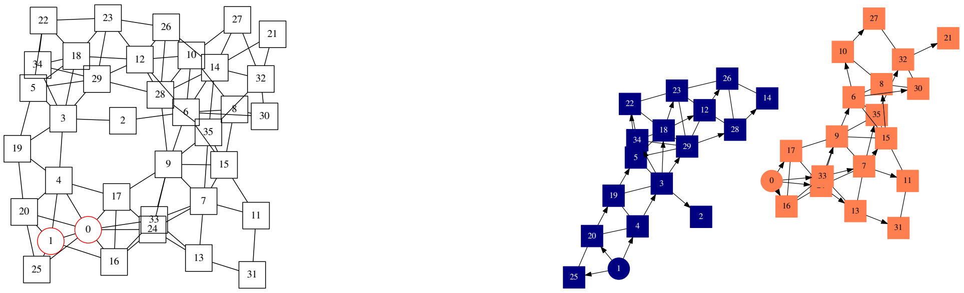


Figure D.29: 36 Nodes, 2 Gateways, Topology 9: Original Topology vs First eTilia Topology

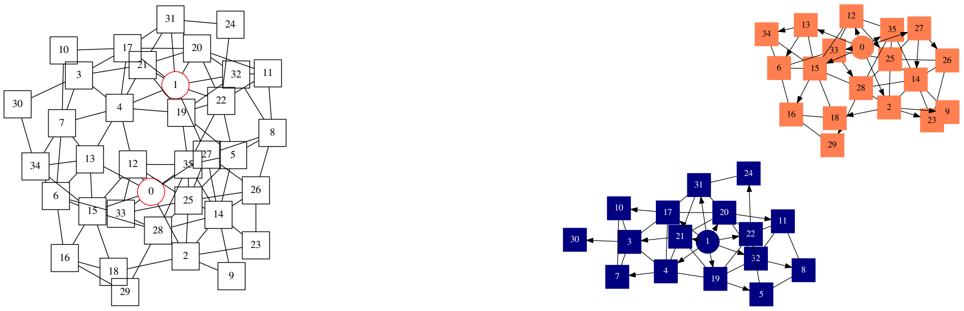


Figure D.30: 36 Nodes, 2 Gateways, Topology 10: Original Topology vs First eTilia Topology

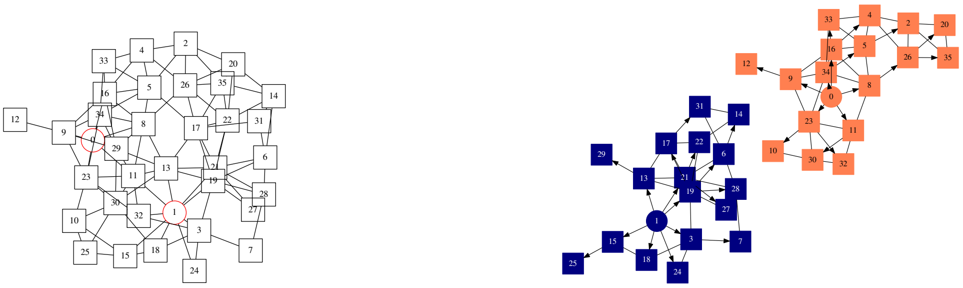


Figure D.31: 36 Nodes, 2 Gateways, Topology 11: Original Topology vs First eTilia Topology

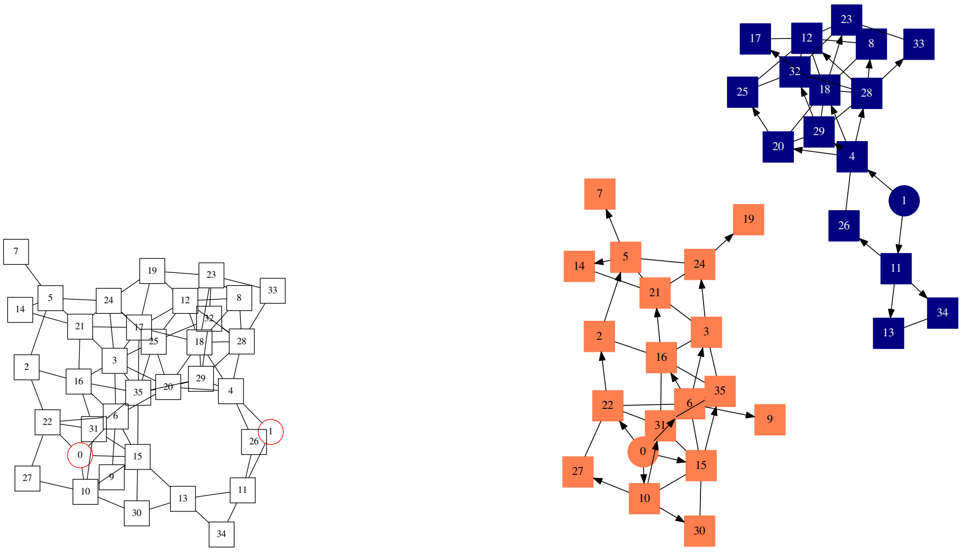


Figure D.32: 36 Nodes, 2 Gateways, Topology 12: Original Topology vs First eTilia Topology

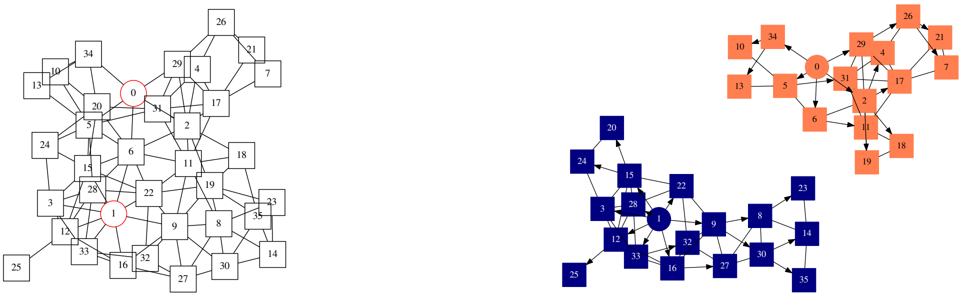


Figure D.33: 36 Nodes, 2 Gateways, Topology 13: Original Topology vs First eTilia Topology

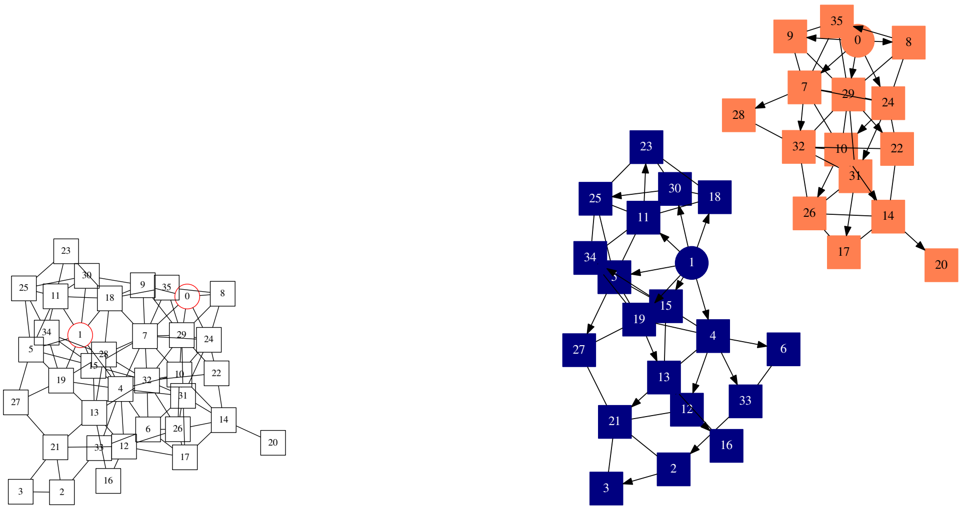


Figure D.34: 36 Nodes, 2 Gateways, Topology 14: Original Topology vs First eTilia Topology

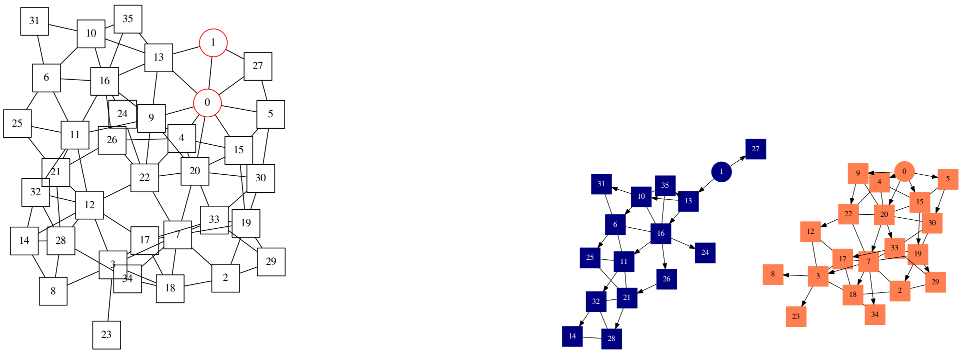


Figure D.35: 36 Nodes, 2 Gateways, Topology 15: Original Topology vs First eTilia Topology

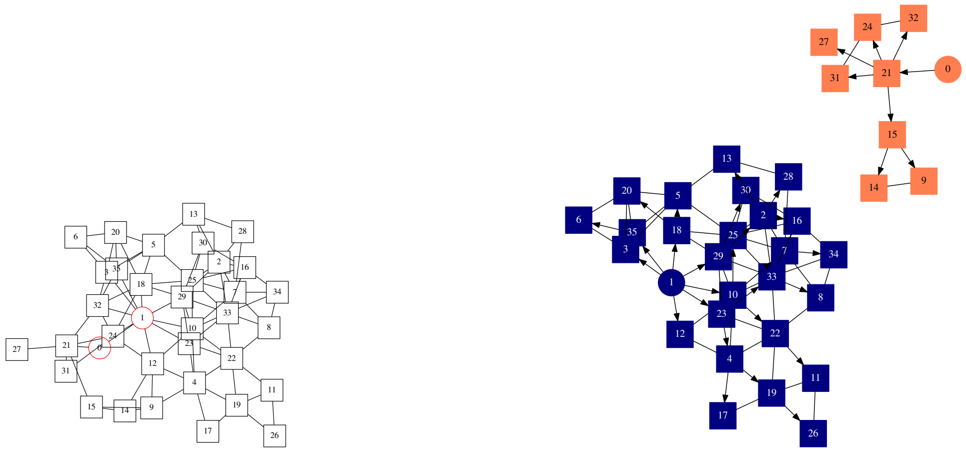


Figure D.36: 36 Nodes, 2 Gateways, Topology 16: Original Topology vs First eTilia Topology

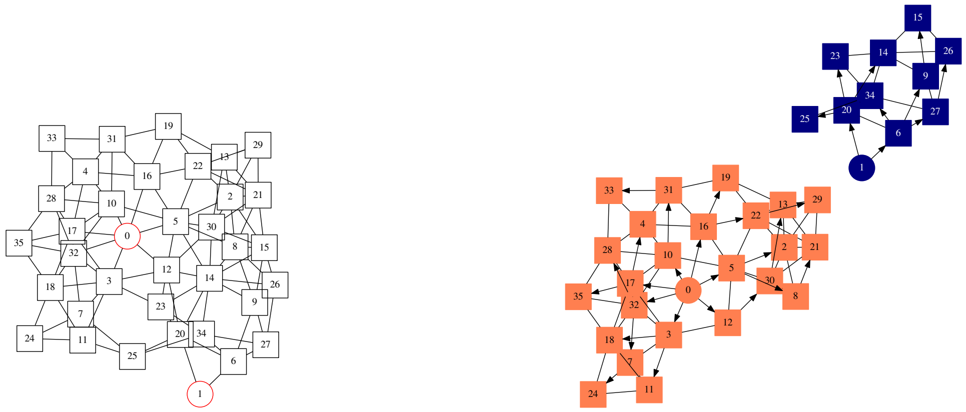


Figure D.37: 36 Nodes, 2 Gateways, Topology 17: Original Topology vs First eTilia Topology

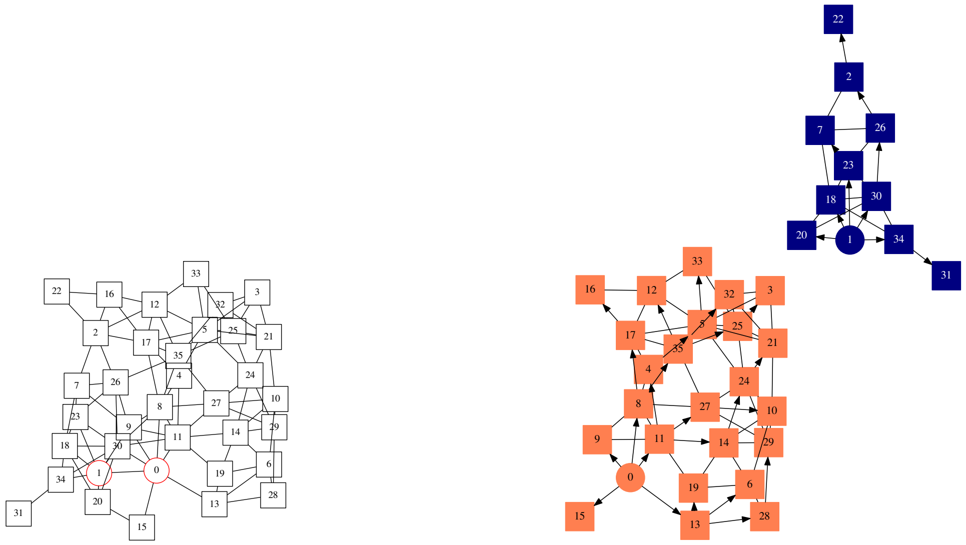


Figure D.38: 36 Nodes, 2 Gateways, Topology 18: Original Topology vs First eTilia Topology

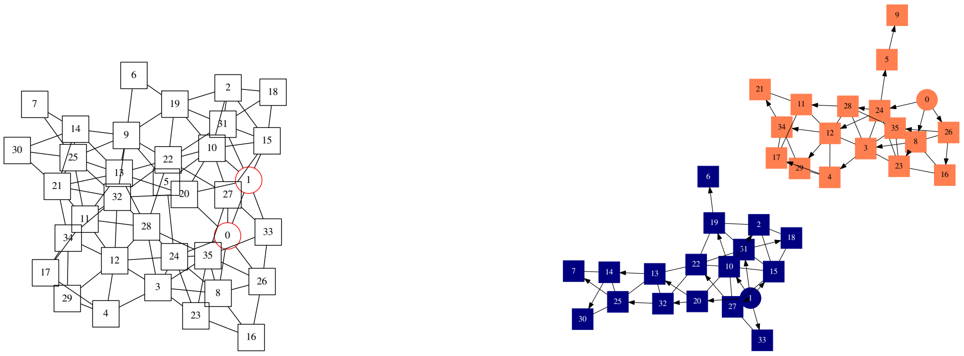


Figure D.39: 36 Nodes, 2 Gateways, Topology 19: Original Topology vs First eTilia Topology

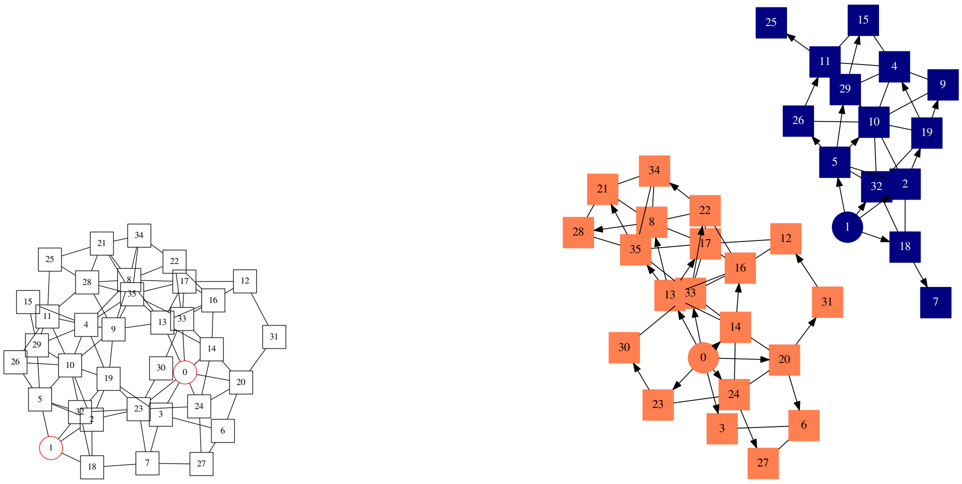


Figure D.40: 36 Nodes, 2 Gateways, Topology 20: Original Topology vs First eTilia Topology

Appendix E

Simulation Results

Table E.1: Simulation results: eTILIA algorithm, 16 nodes, 2 gateways

	eTILIA Algorithm					
Topology ID	**Network Failure Time (s) (u/v)**	**Mean Node Lifetime (s) (u/v)**	**Alive Nodes (u/v)**	**Packets Received (%) (u/v)**	**Packets Lost (%) (u/v)**	**Mean Packet Delay (s) (u/v)**
1	1567.19 / 0.10	1511.16 / 5.28	0.00 / 0.00	95.83 / 1.24	4.17 / 1.24	0.01 / 0.00
2	1534.85 / 0.27	1493.22 / 3.50	1.20 / 0.16	95.89 / 0.15	4.11 / 0.15	0.01 / 0.00
3	1509.18 / 0.40	1468.11 / 1.53	0.20 / 0.16	98.51 / 0.23	1.48 / 0.24	0.02 / 0.00
4	1489.39 / 1.50	1458.58 / 4.73	5.80 / 0.16	94.98 / 0.34	5.00 / 0.34	0.02 / 0.00
5	1529.78 / 0.14	1468.43 / 3.36	3.00 / 0.00	98.12 / 0.65	1.87 / 0.66	0.01 / 0.00
6	1493.79 / 82.07	1455.01 / 17.74	0.60 / 0.64	96.09 / 10.16	3.90 / 10.15	0.02 / 0.00
7	1500.21 / 4.02	1476.90 / 5.56	2.20 / 0.16	94.40 / 3.09	5.58 / 3.09	0.02 / 0.00
8	1523.43 / 23.43	1478.60 / 46.08	3.80 / 0.16	93.45 / 10.61	6.53 / 10.50	0.01 / 0.00
9	1564.83 / 0.03	1469.26 / 0.52	0.00 / 0.00	97.24 / 0.07	2.76 / 0.07	0.02 / 0.00
10	1514.35 / 26.31	1490.93 / 30.70	2.60 / 0.24	92.01 / 4.06	7.97 / 4.00	0.01 / 0.00
11	1511.66 / 0.06	1476.42 / 0.32	2.00 / 0.00	92.66 / 0.01	7.34 / 0.01	0.02 / 0.00
12	1541.50 / 76.07	1515.95 / 23.62	2.20 / 0.56	94.69 / 2.24	5.30 / 2.24	0.01 / 0.00
13	1482.17 / 0.83	1442.16 / 0.78	3.00 / 0.00	97.52 / 0.23	2.48 / 0.23	0.02 / 0.00
14	1541.56 / 7.09	1510.20 / 26.12	2.60 / 0.24	90.07 / 3.22	9.89 / 3.21	0.01 / 0.00
15	1529.22 / 0.01	1493.69 / 1.11	0.00 / 0.00	98.84 / 0.03	1.16 / 0.03	0.01 / 0.00
16	1427.31 / 17.19	1415.19 / 7.01	10.00 / 0.00	95.71 / 0.60	4.26 / 0.57	0.02 / 0.00
17	1532.64 / 38.02	1496.83 / 52.10	0.60 / 0.24	93.62 / 10.57	6.37 / 10.51	0.01 / 0.00
18	1509.40 / 4.48	1475.21 / 15.99	2.20 / 0.16	91.23 / 4.21	8.74 / 4.11	0.02 / 0.00
19	1503.30 / 7.38	1419.61 / 7.14	0.00 / 0.00	90.94 / 1.47	9.06 / 1.47	0.02 / 0.00
20	1544.44 / 35.43	1525.13 / 11.51	1.60 / 0.24	94.20 / 2.59	5.78 / 2.47	0.01 / 0.00

Table E.2: Simulation results: TILIA algorithm, 16 nodes, 2 gateways

	TILIA Algorithm					
Topology ID	**Network Failure Time (s) (u/v)**	**Mean Node Lifetime (s) (u/v)**	**Alive Nodes (u/v)**	**Packets Received (%) (u/v)**	**Packets Lost (%) (u/v)**	**Mean Packet Delay (s) (u/v)**
1	1566.45 / 0.22	1504.34 / 1.49	0.00 / 0.00	96.08 / 0.46	3.92 / 0.46	0.01 / 0.00
2	1535.13 / 0.16	1491.79 / 1.39	1.20 / 0.16	96.32 / 0.41	3.68 / 0.41	0.01 / 0.00
3	1509.08 / 0.06	1468.16 / 0.19	0.00 / 0.00	98.74 / 0.04	1.26 / 0.03	0.01 / 0.00
4	1506.53 / 0.04	1459.20 / 0.17	2.00 / 0.00	94.91 / 0.03	5.08 / 0.03	0.02 / 0.00
5	1543.12 / 0.02	1484.19 / 0.06	0.00 / 0.00	98.89 / 0.02	1.11 / 0.02	0.01 / 0.00
6	1510.00 / 0.00	1458.23 / 3.59	0.00 / 0.00	96.91 / 0.57	3.09 / 0.57	0.02 / 0.00
7	1529.39 / 0.03	1474.06 / 0.29	0.00 / 0.00	98.62 / 0.01	1.38 / 0.01	0.02 / 0.00
8	1499.20 / 0.08	1456.76 / 1.24	1.00 / 0.00	93.89 / 0.09	6.11 / 0.09	0.02 / 0.00
9	1564.85 / 0.07	1470.07 / 1.67	0.00 / 0.00	96.83 / 0.26	3.17 / 0.26	0.01 / 0.00
10	1537.44 / 0.00	1481.99 / 0.55	0.00 / 0.00	98.05 / 0.05	1.95 / 0.05	0.01 / 0.00
11	1511.68 / 0.25	1478.83 / 0.13	2.00 / 0.00	92.54 / 0.02	7.46 / 0.02	0.02 / 0.00
12	1549.74 / 0.59	1507.79 / 0.64	2.00 / 0.00	98.74 / 0.22	1.26 / 0.22	0.01 / 0.00
13	1482.57 / 2.47	1445.23 / 2.85	3.40 / 0.24	97.58 / 0.77	2.42 / 0.77	0.02 / 0.00
14	1526.67 / 0.11	1506.18 / 0.26	2.00 / 0.00	98.30 / 0.06	1.70 / 0.06	0.01 / 0.00
15	1533.58 / 0.00	1493.13 / 2.00	0.00 / 0.00	98.69 / 0.15	1.31 / 0.15	0.01 / 0.00
16	1433.70 / 0.85	1416.26 / 0.10	6.80 / 0.56	95.23 / 0.41	4.73 / 0.39	0.02 / 0.00
17	1523.28 / 0.16	1490.05 / 0.24	0.00 / 0.00	98.99 / 0.05	1.01 / 0.05	0.01 / 0.00
18	1492.80 / 4.28	1460.79 / 0.18	2.20 / 0.16	94.66 / 0.07	5.34 / 0.07	0.02 / 0.00
19	1455.57 / 15.47	1394.46 / 5.19	3.80 / 0.16	95.74 / 0.42	4.24 / 0.41	0.02 / 0.00
20	1561.93 / 0.00	1512.90 / 0.81	0.20 / 0.16	95.90 / 0.14	4.09 / 0.12	0.01 / 0.00

Table E.3: Simulation results: Random channel assignment procedure, 16 nodes, 2 gateways

	Random channel assignment procedure					
Topology ID	**Network Failure Time (s) (u/v)**	**Mean Node Lifetime (s) (u/v)**	**Alive Nodes (u/v)**	**Packets Received (%) (u/v)**	**Packets Lost (%) (u/v)**	**Mean Packet Delay (s) (u/v)**
1	1527.06 / 53.78	1462.96 / 180.21	0.67 / 0.89	97.76 / 0.12	2.24 / 0.12	0.02 / 0.00
2	1549.26 / 164.26	1504.19 / 23.46	0.67 / 0.22	96.39 / 11.91	3.61 / 11.91	0.01 / 0.00
3	1476.73 / 169.00	1440.67 / 48.70	0.67 / 0.22	86.91 / 27.73	13.07 / 27.94	0.02 / 0.00
4	1528.62 / 72.13	1404.85 / 35.68	2.00 / 2.00	89.92 / 3.65	10.06 / 3.55	0.02 / 0.00
5	1556.39 / 57.53	1478.64 / 15.06	0.00 / 0.00	88.79 / 8.35	11.20 / 8.30	0.01 / 0.00
6	1539.36 / 537.78	1460.58 / 102.10	0.00 / 0.00	89.19 / 32.19	10.81 / 32.19	0.01 / 0.00
7	1545.39 / 43.84	1450.86 / 33.59	1.00 / 0.00	95.88 / 0.69	4.12 / 0.69	0.02 / 0.00
8	1560.68 / 1.68	1469.28 / 151.23	0.00 / 0.00	98.19 / 0.03	1.81 / 0.03	0.02 / 0.00
9	1540.06 / 3.30	1422.56 / 76.40	2.00 / 0.00	94.76 / 0.34	5.22 / 0.35	0.02 / 0.00
10	1564.81 / 0.00	1467.50 / 49.42	0.00 / 0.00	93.50 / 1.33	6.50 / 1.33	0.01 / 0.00
11	1563.69 / 0.00	1456.29 / 133.23	1.00 / 0.00	94.60 / 6.29	5.40 / 6.29	0.02 / 0.00
12	1544.09 / 3.33	1502.53 / 56.27	0.00 / 0.00	93.91 / 5.35	6.09 / 5.35	0.01 / 0.00
13	1531.74 / 105.46	1473.31 / 216.85	1.33 / 0.22	94.54 / 8.10	5.46 / 8.10	0.01 / 0.00
14	1544.04 / 170.41	1484.34 / 137.58	0.67 / 0.22	96.78 / 0.14	3.21 / 0.14	0.01 / 0.00
15	1534.30 / 16.06	1461.50 / 362.68	0.00 / 0.00	94.59 / 1.29	5.41 / 1.29	0.01 / 0.00
16	1500.16 / 121.40	1419.96 / 204.54	2.00 / 0.67	91.19 / 8.32	8.81 / 8.32	0.02 / 0.00
17	1531.65 / 53.43	1448.60 / 287.04	0.33 / 0.22	91.84 / 1.80	8.16 / 1.80	0.02 / 0.00
18	1521.16 / 78.76	1458.76 / 22.90	0.67 / 0.22	95.81 / 3.45	4.19 / 3.45	0.02 / 0.00
19	1473.42 / 2.97	1362.25 / 136.05	2.00 / 0.00	84.45 / 19.70	15.55 / 19.70	0.02 / 0.00
20	1584.43 / 29.43	1488.71 / 24.40	0.00 / 0.00	87.63 / 20.58	12.37 / 20.58	0.01 / 0.00

Table E.4: Simulation results: eTILIA algorithm, 36 nodes, 2 gateways

	eTILIA Algorithm					
Topology ID	**Network Failure Time (s) (u/v)**	**Mean Node Lifetime (s) (u/v)**	**Alive Nodes (u/v)**	**Packets Received (%) (u/v)**	**Packets Lost (%) (u/v)**	**Mean Packet Delay (s) (u/v)**
1	1280.64 / 4.98	1241.94 / 1.72	23.25 / 0.19	81.06 / 0.04	18.81 / 0.03	0.09 / 0.00
2	1441.64 / 5.67	1337.26 / 4.05	4.00 / 0.67	86.51 / 0.13	13.49 / 0.13	0.03 / 0.00
3	1430.94 / 129.84	1321.89 / 7.15	1.25 / 0.69	81.12 / 0.89	18.87 / 0.88	0.04 / 0.00
4	1412.85 / 48.42	1312.42 / 16.18	7.50 / 0.25	74.56 / 3.43	25.37 / 3.39	0.05 / 0.00
5	1448.94 / 0.62	1302.30 / 0.14	2.25 / 0.19	81.98 / 0.07	18.01 / 0.07	0.05 / 0.00
6	1489.64 / 0.28	1332.46 / 30.37	2.00 / 0.00	74.03 / 0.60	25.97 / 0.60	0.33 / 0.00
7	1386.84 / 0.36	1271.15 / 14.57	10.33 / 0.89	79.41 / 0.01	20.53 / 0.01	0.04 / 0.00
8	1460.96 / 27.38	1348.32 / 4.42	0.67 / 0.22	86.13 / 0.07	13.87 / 0.07	0.03 / 0.00
9	1467.54 / 23.35	1324.59 / 17.70	5.33 / 0.22	67.77 / 1.69	32.15 / 1.62	0.04 / 0.00
10	1370.28 / 36.85	1335.31 / 14.76	14.33 / 1.56	87.97 / 0.55	11.95 / 0.51	0.03 / 0.00
11	1437.58 / 0.52	1337.77 / 1.40	7.00 / 0.00	82.31 / 0.05	17.67 / 0.05	0.04 / 0.00
12	1321.12 / 37.47	1267.20 / 37.15	13.50 / 3.25	81.57 / 0.84	18.33 / 0.85	0.04 / 0.00
13	1390.61 / 4.04	1334.45 / 6.41	11.25 / 2.19	82.80 / 0.61	17.14 / 0.58	0.04 / 0.00
14	1382.43 / 1.69	1313.13 / 3.94	7.25 / 0.19	83.98 / 0.30	15.97 / 0.30	0.04 / 0.00
15	1520.79 / 43.43	1341.03 / 19.35	0.00 / 0.00	65.55 / 1.39	34.45 / 1.39	0.20 / 0.00
16	1495.91 / 2.10	1347.86 / 14.20	2.00 / 0.50	63.05 / 0.20	36.94 / 0.20	0.24 / 0.00
17	1310.81 / 1.33	1269.01 / 1.20	20.00 / 0.00	84.49 / 0.04	15.45 / 0.03	0.05 / 0.00
18	1469.36 / 24.54	1329.85 / 5.20	3.00 / 0.00	65.82 / 0.36	34.17 / 0.35	0.26 / 0.00
19	1446.43 / 73.30	1354.26 / 32.14	7.50 / 0.75	76.50 / 0.56	23.46 / 0.55	0.03 / 0.00
20	1386.21 / 0.56	1316.14 / 19.09	8.00 / 0.00	86.13 / 0.59	13.80 / 0.61	0.03 / 0.00

Table E.5: Simulation results: TILIA algorithm, 36 nodes, 2 gateways

	TILIA Algorithm					
Topology ID	**Network Failure Time (s) (u/v)**	**Mean Node Lifetime (s) (u/v)**	**Alive Nodes (u/v)**	**Packets Received (%) (u/v)**	**Packets Lost (%) (u/v)**	**Mean Packet Delay (s) (u/v)**
1	1306.55 / 1.15	1251.53 / 0.67	23.00 / 0.00	82.75 / 0.08	17.15 / 0.07	0.04 / 0.00
2	1435.92 / 1.27	1334.79 / 4.70	3.00 / 0.00	86.45 / 0.42	13.54 / 0.43	0.03 / 0.00
3	1448.49 / 1.39	1324.42 / 38.02	2.00 / 0.00	84.57 / 1.00	15.43 / 1.00	0.04 / 0.00
4	1375.18 / 6.20	1283.53 / 11.54	8.00 / 0.00	84.83 / 0.53	15.15 / 0.54	0.04 / 0.00
5	1448.93 / 5.49	1296.05 / 4.01	2.75 / 0.69	82.22 / 0.34	17.78 / 0.34	0.05 / 0.00
6	1489.94 / 0.67	1325.39 / 2.12	2.33 / 0.22	75.79 / 0.33	24.19 / 0.32	0.15 / 0.00
7	1334.70 / 6.57	1239.19 / 19.89	9.33 / 0.22	83.52 / 0.67	16.42 / 0.65	0.05 / 0.00
8	1416.03 / 1.31	1321.55 / 2.59	6.00 / 0.00	87.72 / 0.33	12.27 / 0.33	0.04 / 0.00
9	1460.15 / 2.12	1276.98 / 56.41	5.33 / 0.22	78.35 / 2.05	21.64 / 2.05	0.05 / 0.00
10	1383.26 / 4.24	1324.49 / 0.08	12.00 / 0.00	83.51 / 0.06	16.43 / 0.07	0.03 / 0.00
11	1427.02 / 24.78	1324.25 / 8.21	5.33 / 0.22	82.86 / 0.29	17.13 / 0.28	0.04 / 0.00
12	1302.51 / 0.84	1244.52 / 1.60	14.00 / 0.00	85.51 / 0.07	14.45 / 0.07	0.05 / 0.00
13	1400.76 / 20.73	1343.75 / 0.05	9.00 / 0.00	84.42 / 0.25	15.53 / 0.25	0.04 / 0.00
14	1394.09 / 14.54	1311.94 / 2.99	4.00 / 0.00	85.00 / 0.17	14.98 / 0.16	0.04 / 0.00
15	1469.13 / 2.98	1284.05 / 10.84	2.25 / 0.19	78.43 / 0.03	21.56 / 0.03	0.06 / 0.00
16	1480.69 / 45.74	1313.23 / 21.81	3.50 / 0.25	69.02 / 0.51	30.97 / 0.50	0.28 / 0.00
17	1323.49 / 1.45	1278.54 / 7.64	19.75 / 0.19	82.19 / 0.04	17.73 / 0.04	0.05 / 0.00
18	1471.38 / 19.10	1317.94 / 50.91	4.25 / 1.19	68.79 / 0.23	31.20 / 0.23	0.34 / 0.00
19	1426.32 / 7.52	1322.84 / 3.45	5.75 / 0.19	83.27 / 0.33	16.71 / 0.32	0.03 / 0.00
20	1376.27 / 15.98	1310.57 / 9.00	8.50 / 0.25	85.99 / 0.42	13.97 / 0.40	0.03 / 0.00

Table E.6: Simulation results: Random channel assignment procedure, 36 nodes, 2 gateways

	Random channel assignment procedure					
Topology ID	**Network Failure Time (s) (u/v)**	**Mean Node Lifetime (s) (u/v)**	**Alive Nodes (u/v)**	**Packets Received (%) (u/v)**	**Packets Lost (%) (u/v)**	**Mean Packet Delay (s) (u/v)**
1	1483.17 / 17.31	1333.81 / 28.69	3.00 / 0.67	73.56 / 9.67	26.53 / 10.41	0.31 / 0.02
2	1531.87 / 407.07	1345.92 / 34.58	3.00 / 2.00	75.59 / 2.50	24.39 / 2.50	0.08 / 0.00
3	1523.23 / 261.21	1349.03 / 102.37	0.00 / 0.00	71.41 / 36.52	28.59 / 36.51	0.10 / 0.00
4	1426.82 / 71.25	1300.21 / 52.77	4.33 / 0.22	78.75 / 26.47	21.21 / 26.57	0.22 / 0.02
5	1551.44 / 12.83	1341.02 / 60.96	0.00 / 0.00	69.20 / 1.32	30.80 / 1.32	0.38 / 0.02
6	1532.23 / 34.05	1326.09 / 66.77	2.33 / 0.89	67.40 / 8.15	32.59 / 8.10	0.42 / 0.01
7	1424.99 / 9.86	1298.27 / 15.54	6.00 / 0.67	77.41 / 0.65	22.53 / 0.66	0.23 / 0.02
8	1434.56 / 324.37	1299.65 / 72.92	3.67 / 0.89	68.79 / 46.95	31.18 / 47.17	0.18 / 0.00
9	1522.60 / 30.82	1302.93 / 79.64	0.67 / 0.22	60.69 / 8.67	39.30 / 8.73	0.41 / 0.07
10	1417.28 / 129.73	1327.09 / 81.05	5.33 / 0.22	81.07 / 5.41	18.90 / 5.44	0.08 / 0.00
11	1425.45 / 227.73	1280.62 / 282.60	3.67 / 1.56	71.59 / 23.87	28.37 / 23.64	0.33 / 0.06
12	1373.76 / 2749.97	1267.04 / 834.93	8.67 / 20.22	74.63 / 20.75	25.27 / 20.84	0.11 / 0.00
13	1398.56 / 44.25	1333.70 / 212.37	9.33 / 0.89	85.82 / 2.81	14.14 / 2.77	0.04 / 0.00
14	1398.88 / 570.92	1294.89 / 68.33	5.33 / 1.56	76.11 / 1.04	23.86 / 1.08	0.06 / 0.00
15	1425.29 / 204.01	1269.69 / 8.96	6.00 / 4.67	71.30 / 1.79	28.64 / 1.74	0.25 / 0.01
16	1435.63 / 184.89	1288.18 / 22.32	5.67 / 0.89	72.69 / 0.47	27.28 / 0.50	0.44 / 0.06
17	1333.21 / 93.15	1270.59 / 41.78	14.00 / 12.67	78.94 / 6.06	20.98 / 6.31	0.20 / 0.02
18	1440.62 / 238.55	1322.12 / 190.46	5.33 / 6.89	74.50 / 23.04	25.48 / 23.10	0.08 / 0.00
19	1474.92 / 203.75	1319.07 / 48.00	1.67 / 0.22	71.60 / 1.15	28.38 / 1.13	0.52 / 0.01
20	1393.85 / 67.40	1290.96 / 52.39	8.67 / 0.22	77.87 / 2.55	22.09 / 2.55	0.29 / 0.02

References

[1] Ian F Akyildiz, Tommaso Melodia, and Kaushik R Chowdhury. A survey on wireless multimedia sensor networks. *Computer networks*, 51(4):921–960, 2007.

[2] Tânia Cláudia dos Santos Pinto Calçada. *Topology Aware Channel Assignment in Single-radio Stub Wireless Mesh Networks*. PhD thesis, Universidade do Porto, 2012.

[3] Ian F Akyildiz, Xudong Wang, and Weilin Wang. Wireless mesh networks: a survey. *Computer networks*, 47(4):445–487, 2005.

[4] Anand Prabhu Subramanian, Milind M Buddhikot, and Scott Miller. Interference aware routing in multi-radio wireless mesh networks. In *Wireless Mesh Networks, 2006. WiMesh 2006. 2nd IEEE Workshop on*, pages 55–63. IEEE, 2006.

[5] Jian Tang, Guoliang Xue, and Weiyi Zhang. Interference-aware topology control and qos routing in multi-channel wireless mesh networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 68–77. ACM, 2005.

[6] Bin Ni, Naveen Santhapuri, Zifei Zhong, and Srihari Nelakuditi. Routing with opportunistically coded exchanges in wireless mesh networks. In *Wireless Mesh Networks, 2006. WiMesh 2006. 2nd IEEE Workshop on*, pages 157–159. IEEE, 2006.

[7] Luigi Iannone, Ramin Khalili, Kave Salamatian, and Serge Fdida. Cross-layer routing in wireless mesh networks. In *Wireless Communication Systems, 2004, 1st International Symposium on*, pages 319–323. IEEE, 2004.

[8] Devu Manikantan Shila, Yu Cheng, and Tricha Anjali. Capacity of cooperative wireless networks using multiple channels. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–5. IEEE, 2010.

[9] Hong-Ning Dai, Kam-Wing Ng, RC-W Wong, and Min-You Wu+. On the capacity of multi-channel wireless networks using directional antennas. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE, 2008.

[10] Krishna N Ramachandran, Elizabeth M Belding-Royer, Kevin C Almeroth, and Milind M Buddhikot. Interference-aware channel assignment in multi-radio wireless mesh networks. In *INFOCOM*, volume 6, pages 1–12, 2006.

[11] Murali Kodialam and Thyaga Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 73–87. ACM, 2005.

[12] Paramvir Bahl, Ranveer Chandra, and John Dunagan. Ssch: slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 216–230. ACM, 2004.

[13] Jack H Winters, Jack Salz, and Richard D Gitlin. The impact of antenna diversity on the capacity of wireless communication systems. *Communications, IEEE Transactions on*, 42(234):1740–1751, 1994.

[14] Ruben Merz, Jörg Widmer, Jean-Yves Le Boudec, and Božidar Radunović. A joint phy/mac architecture for low-radiated power th-uwb wireless ad hoc networks. *Wireless Communications and Mobile Computing*, 5(5):567–580, 2005.

[15] Suhas N Diggavi, Matthias Grossglauser, and David N C Tse. Even one-dimensional mobility increases ad hoc wireless capacity. In *Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on*, page 352. IEEE, 2002.

[16] Matthias Grossglauser and David Tse. Mobility increases the capacity of ad-hoc wireless networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1360–1369. IEEE, 2001.

[17] Jungmin So and NF Vaidya. Load-balancing routing in multichannel hybrid wireless networks with single network interface. *Vehicular Technology, IEEE Transactions on*, 56(1):342–348, 2007.

[18] Pradeep Kyasanur and Nitin H Vaidya. Routing in multi-channel multi-interface ad hoc wireless networks. *University of Illinois at Urbana-Champaign, Tech. Rep*, 2004.

[19] IEEE 802 LAN/MAN Standards Committee et al. Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Standard*, 802(11), 1999.

[20] Ashish Raniwala and Tzi-cker Chiueh. Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 2223–2234. IEEE, 2005.

[21] Pradeep Kyasanur and Nitin H Vaidya. Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 10(1):31–43, 2006.

[22] Pradeep Kyasanur and Nitin H Vaidya. Capacity of multi-channel wireless networks: impact of number of channels and interfaces. In *Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 43–57. ACM, 2005.

[23] AA Hagberg, DA Schult, and M Renieris. Pygraphviz a python interface to the graphviz graph layout and visualization package.

[24] Raj Jain, Dah-Ming Chiu, and William Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. 1998.

[25] Tifenn Rault, Abdelmadjid Bouabdallah, and Yacine Challal. Energy efficiency in wireless sensor networks: A top-down survey. *Computer Networks*, 67:104–122, 2014.

[26] Shuguang Cui, Andrea J Goldsmith, and Ahmad Bahai. Energy-constrained modulation optimization. *Wireless Communications, IEEE Transactions on*, 4(5):2349–2360, 2005.

[27] Felipe M Costa and Hideki Ochiai. A comparison of modulations for energy optimization in wireless sensor network links. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5. IEEE, 2010.

[28] Shan Lin, Jingbin Zhang, Gang Zhou, Lin Gu, John A Stankovic, and Tian He. Atpc: adaptive transmission power control for wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 223–236. ACM, 2006.

[29] Evangelos Kranakis, Danny Krizanc, and Eric Williams. Directional versus omnidirectional antennas for energy consumption and k-connectivity of networks of sensors. In *Principles of Distributed Systems*, pages 357–368. Springer, 2005.

[30] Shuguang Cui, Andrea J Goldsmith, and Ahmad Bahai. Energy-efficiency of mimo and cooperative mimo techniques in sensor networks. *Selected Areas in Communications, IEEE Journal on*, 22(6):1089–1098, 2004.

[31] Ramesh Rajagopalan and Pramod K Varshney. Data aggregation techniques in sensor networks: A survey. 2006.

[32] Zhixian Yan, Vigneshwaran Subbaraju, Dipanjan Chakraborty, Archan Misra, and Karl Aberer. Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach. In *Wearable Computers (ISWC), 2012 16th International Symposium on*, pages 17–24. Ieee, 2012.

[33] Shuai Wang, Athanasios Vasilakos, Hongbo Jiang, Xiaoqiang Ma, Wenyu Liu, Kai Peng, Bo Liu, and Yan Dong. Energy efficient broadcasting using network coding aware protocol in wireless ad hoc network. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011.

[34] Naoto Kimura and Shahram Latifi. A survey on data compression in wireless sensor networks. In *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, volume 2, pages 8–13. IEEE, 2005.

[35] He Ba, Ilker Demirkol, and Wendi Heinzelman. Passive wake-up radios: From devices to applications. *Ad Hoc Networks*, 11(8):2605–2621, 2013.

[36] Sudip Misra, Manikonda Pavan Kumar, and Mohammad S Obaidat. Connectivity preserving localized coverage algorithm for area monitoring using wireless sensor networks. *Computer Communications*, 34(12):1484–1496, 2011.

[37] Anfeng Liu, Ju Ren, Xu Li, Zhigang Chen, and Xuemin Sherman Shen. Design principles and improvement of cost function based energy aware routing algorithms for wireless sensor networks. *Computer Networks*, 56(7):1951–1967, 2012.

[38] Ye Ming Lu and Vincent WS Wong. An energy-efficient multipath routing protocol for wireless sensor networks. *International Journal of Communication Systems*, 20(7):747–766, 2007.

[39] Sujesha Sudevalayam and Purushottam Kulkarni. Energy harvesting sensor nodes: Survey and implications. *Communications Surveys & Tutorials, IEEE*, 13(3):443–461, 2011.

[40] Liguang Xie, Yi Shi, Y Thomas Hou, and A Lou. Wireless power transfer and applications to sensor networks. *Wireless Communications, IEEE*, 20(4), 2013.

[41] Chi Ma, Ming Ma, and Yuanyuan Yang. A battery aware scheme for energy efficient coverage and routing in wireless mimo mesh networks. In *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*, pages 3603–3608. IEEE, 2007.

[42] Stefano Avallone. An energy efficient channel assignment and routing algorithm for multi-radio wireless mesh networks. *Ad Hoc Networks*, 10(6):1043–1057, 2012.

[43] HuiJun Miao, Yanliang Jin, Quan Ge, and Yun He. Expected transmission energy routing metric for wireless mesh senor network. In *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*, pages 1–4. IEEE, 2010.

[44] Yinpeng Yu, Yuhuai Peng, Yejun Liu, Lei Guo, and Meng Song. Survivable routing protocol for green wireless mesh networks based on energy efficiency. *Communications, China*, 11(8):117–124, 2014.

[45] Thuy Tran Vinh, Thu Ngo Quynh, and Mai Banh Thi Quynh. Emrp: energy-aware mesh routing protocol for wireless sensor networks. In *Advanced Technologies for Communications (ATC), 2012 International Conference on*, pages 78–82. IEEE, 2012.

[46] Jie Wu. *Handbook on theoretical and algorithmic aspects of sensor, ad hoc wireless, and peer-to-peer networks*. CRC Press, 2004.

[47] Raspberry pi. `https://www.raspberrypi.org/`. Accessed: 2015-06-26.

[48] Wireless usb adapter tp-link tl-wn722n. `http://uk.tp-link.com/products/details/cat-11_TL-WN722N.html`. Accessed: 2015-06-26.

[49] Raspberry pi camera. `https://www.raspberrypi.org/products/camera-module/`. Accessed: 2015-06-26.

[50] Battery dgy12-7.5ev. `http://dynoeurope.com/assets/gel-ev/TDS-DGY12-7.5EV-EN.pdf`. Accessed: 2015-06-26.

[51] Solar panel gsap6. `http://www.germansolarasia.com/product.asp?id=009`. Accessed: 2015-06-26.

[52] Solar charge controller tracer-1210rn. `http://www.epsolarpv.com/en/index.php/Product/pro_content/id/156/am_id/136`. Accessed: 2015-06-26.

[53] Antonio Ortega and Kannan Ramchandran. Rate-distortion methods for image and video compression. *Signal Processing Magazine, IEEE*, 15(6):23–50, 1998.

[54] H ITU-T RECOMMENDATION. 264 “advanced video coding for generic audiovisual services”. *ISO/IEC*, 14496, 2003.

[55] Guido R Hiertz, Dee Denteneer, Sebastian Max, Rakesh Taori, Javier Cardona, Lars Berlemann, and Bernhard Walke. Ieee 802.11 s: the wlan mesh standard. *Wireless Communications, IEEE*, 17(1):104–111, 2010.

[56] Filipe Teixeira, Tânia Calçada, and Manuel Ricardo. Architecture for a channel assignment subsystem. In *Proceedings of the Networking and Electronic Commerce Research Conference (NAEC 2012)*. INESC TEC, Faculdade de Engenharia, Universidade do Porto, 2012.

[57] Filipe Teixeira, Tânia Calçada, and Manuel Ricardo. Protocol for centralized channel assignment in wifix single-radio mesh networks. In *Mobile Networks and Management*, pages 17–30. Springer, 2012.

[58] Network simulator 3. `http://www.nsnam.org/`. Accessed: 2015-01-26.

[59] Kirill Andreev and Pavel Boyko. Ieee 802.11 s mesh networking ns-3 model. In *Workshop on ns3*, page 43, 2010.

[60] He Wu, Sidharth Nabar, and Radha Poovendran. An energy framework for the network simulator 3 (ns-3). In *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, pages 222–230. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011.

[61] Kanishka Lahiri, Sujit Dey, Debashis Panigrahi, and Anand Raghunathan. Battery-driven system design: A new frontier in low power design. In *Proceedings of the 2002 Asia and South Pacific Design Automation Conference*, page 261. IEEE Computer Society, 2002.

[62] Gustavo Carneiro, Pedro Fortuna, and Manuel Ricardo. Flowmonitor: a network monitoring framework for the network simulator 3 (ns-3). In *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, page 1. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[63] Mesh model bug. `https://www.nsnam.org/bugzilla/show_bug.cgi?id=737`. Accessed: 2014-12-10.