Internet on Rails

Juan-Carlos Maureira

UNIVERSITY OF NICE - SOPHIA ANTIPOLIS

DOCTORAL SCHOOL STIC

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

P H D T H E S I S

to obtain the title of

PhD of Science

of the University of Nice - Sophia Antipolis

Speciality : Computer Science

Defended by

Juan-Carlos Maureira

Internet on Rails

Thesis Advisors: **Jean-Claude Bermond** and **Olivier Dalle**

prepared at INRIA Sophia Antipolis, Mascotte Team

defended on January 21st, 2011

Jury :

Reviewers :	Felipe Perrone	-	Associate Professor, Bucknell University
	Marion Berbineau	-	Directeur de Recherche, IFSTTAR-LEOST
	Bertrand Ducourthial	-	Professeur, U. Technologie de Compiègne
Advisor :	Jean-Claude Bermond	-	Directeur de Recherche, CNRS (Mascotte)
Co-Advisor :	Olivier Dalle	-	Maître de Conférences, U. Nice Sophia (Mascotte)
Examinators :	Jorge Amaya	-	Associate Researcher, U. de Chile (CMM)
	Gabriel Wainer	-	Associate Professor, Carleton University
	Frédéric Giroire	-	Chargé de Recherche, CNRS (Mascotte)

To my wife, who stood with me all the way long to get here,

To my friends, who supported me in the hardest times,

To Mascotte, from whom I learned a lot.

Acknowledgments

I would like to express my gratitude to my advisors, Jean-Claude Bermond and Olivier Dalle for guiding me along my education as a young researcher. Also I thank the INRIA Sophia-Antipolis and the MASCCOTE team for receiving me as a Ph.D student, and to my reviewers for the effort to examinate my work. In addition, I thank INRIA and CONICYT for granting me the funds to pursue this Ph.D. Lastly, I would like to thank the *Center for Mathematical Modelling (CMM)* for allowing me to use their cluster to run my simulations. To Fred and Nico, for having the patience to discuss with me and to teach me a lot about research. And to Jean-Claude for teaching and advising me through the mathematical part of my work.

Je voudrais remercier tous les membres de l'équipe MASCOTTE pour tous les barbecues, discussions et fêtes que nous avons partagé, ce seront des beaux souvenirs pour le reste de ma vie. Particulièrement je remercie Patricia pour son aide, pour m'avoir pris le temps de m'écouter et me corriger mon français. Un merci spécial à la gentillesse de Jean-Marc Gambaudo and Elizabeth Pecou pour leur aide à notre arrivée à Nice. Finalement merci à tous les personnes qui nous ont offert leur amitié pendant ces trois années.

Quisiera agradecer a toda la comunidad Chilena en la Costa Azul. A quienes ya partieron y quienes siguen llegando. Por todos esos momentos de felicidad y alegría que compartimos durante estos tres años. Especialmente quiero agradecer a mi amigo Jose Aliste, por ayudarnos a instalarnos en Niza, por todas esas discusiones que tuvimos (y que espero que sigamos teniendo) y por sus consejos en los momentos duros. Tambien quiero agradecer especialmente a la gran persona de Cristian Ruz, por toda su ayuda y compañia durante los ultimos meses de mi tesis. Sus palabras de aliento fueron de mucha ayuda para llegar al fin de este camino de 3 años. A todos los amigos latinos y no latinos que encontramos acá: Vanesa, Elo, Jullie, Sachie, Athena, Stefan y Haidi, con quienes espero de verdad mantener una amistad duradera.

Finalmente, quisiera agradecer a Mara, mi amada esposa, por haber estado siempre a mi lado durante este periodo. Por su apoyo incondicional, por apoyarme en los momentos difíciles y por darme fuerzas cuando ya no me quedaban. Todo esto es por ti mi amor.

Sophia-Antipolis,
January 21, 2010.

Abstract

This thesis proposes a new method for providing network connectivity to vehicles over a predefined trajectory (trains, metros, urban buses, etc.). The communication between the vehicle and the infrastructure network is based only on WiFi technology. The contributions of this work are two-fold: 1) the horizontal handover (between WiFi access points) and 2) the design and analysis of an infrastructure network (backbone network plus WiFi access network) deployed along the trajectory of the vehicle.

In the first contribution, we propose a handover scheme, called Spiderman Handover, which describes the horizontal handover for an in-motion network (on-board the vehicle) considering a procedure to update the routing information of a bridged infrastructure network (OSI layer 2). We evaluate our proposal by means of simulation and we validate our results by experimental measurements.

In the second contribution, we study theoretically the parameters of several chordal like topologies in order to build a backbone network for a linear access network. By comparing these parameters, we propose a backbone network composed by a combination of two chordal topologies. This backbone network provides a good balance between their deployment cost, number of hops to the gateway of the network and a reasonable resilience.

Finally, we evaluate the integration of this infrastructure network and the handover scheme by means of simulations. Results showed that the proposed handover scheme works properly on the proposed infrastructure network, allowing the provision of a continuous network connectivity to passengers on-board trains, metros or urban buses.

Keywords: Train communications, WiFi, horizontal handover, layer 2 routes update, infrastructure network, combined chordal topologies, simulations.

Resumé

Cette thèse propose une nouvelle méthode pour fournir une connexion réseau à des véhicules au cours de trajets prédéterminés (trains, métros, autobus urbains, etc.). La communication entre le véhicule et l'infrastructure réseau est basée uniquement sur la technologie WiFi. Les contributions de ce travail sont d'une part la conception d'une méthode pour réaliser le handover horizontal (entre bornes WiFi), et d'autre part la modélisation et l'analyse de topologies pour le réseau d'infrastructure (réseau backbone plus réseau d'accès WiFi) déployé sur la trajectoire du véhicule.

Dans une première partie, nous proposons une méthode, appelée Spiderman Handover, pour réaliser le handover horizontal d'un réseau en mouvement (embarqué dans le véhicule) et une procédure de mise à jour des informations de routage (couche 2 OSI) lors du handover. Nous évaluons notre proposition par simulation et validons nos résultats par des mesures expérimentales.

Dans une deuxième partie, nous étudions théoriquement les paramétrés de plusieurs familles de topologies du type Chordal pour le réseau backbone construit sur un réseau d'accès linéaire. À partir de la comparaison de ces paramétrés, nous proposons une topologie backbone issue de la combinaison de deux topologies Chordal. Cette topologie fournit un bon compromis entre coût du déploiement, nombre de sauts nécessaires pour atteindre la passerelle du réseau et résilience raisonnable.

Enfin, nous évaluons l'intégration de la topologie proposée pour le réseau d'infrastructure avec le système handover par des simulations. Les résultats présentés suggèrent que l'algorithme de handover proposé fonctionne correctement sur le réseau d'infrastructure proposé. Cela permet la garantie d'une connexion continue aux passagers à bord des trains, métros ou autobus urbains.

Mots Clés: Réseau de communication pour trains, WiFi, horizontal handover, mise à jour des informations de routage, reseau d'infrastructure, topologies combinés corde, simulations.

Contents

1 Introduction **1**
- 1.1 Motivation and Context . . . 1
- 1.2 Challenges . . . 2
- 1.3 Methodology . . . 4
- 1.4 Contributions and Organization . . . 8

2 State of the Art: Internet on Trains **11**
- 2.1 Introduction . . . 11
- 2.2 Current Deployments . . . 12
- 2.3 Technological Review on "Internet on-board" . . . 14
  - 2.3.1 3/3.5G Connectivity . . . 14
  - 2.3.2 Satellite Connectivity . . . 15
  - 2.3.3 WiMax Connectivity . . . 16
  - 2.3.4 WiFi Connectivity . . . 17
- 2.4 Summary and Discussion . . . 18

3 Assumptions **21**
- 3.1 Introduction . . . 21
- 3.2 Speed of the Train and Wireless Coverage Area . . . 22
- 3.3 Traffic Profiles . . . 23
- 3.4 Routing Scheme . . . 24
- 3.5 Failure Probability of a Node . . . 26

I Handover of an In-motion Network along a Predefined Trajectory **27**

4 Horizontal Handover of an In-motion Network **29**
- 4.1 Introduction . . . 29
- 4.2 Problem Definition . . . 31
- 4.3 Assumptions . . . 32
- 4.4 Related Works . . . 32
- 4.5 Proposed Solution . . . 34
  - 4.5.1 System Architecture . . . 35
  - 4.5.2 System Operation . . . 37
  - 4.5.3 Scanning Method . . . 39
  - 4.5.4 Handover Time . . . 41
- 4.6 Discussion . . . 41
- 4.7 Conclusions . . . 43

5 Routes Updating in an Infrastructure Network 45
5.1 Preliminaries . . . 45
5.2 Problem Definition . . . 46
5.3 Proposed Solution . . . 47
5.3.1 GAL Implementation . . . 49
5.3.2 GAL Parameters . . . 50
5.4 Evaluation . . . 55
5.4.1 Simulated Scenario . . . 55
5.4.2 GAL Time - Handover Time . . . 57
5.4.3 Packet Losses . . . 59
5.4.4 Round Trip Time . . . 61
5.5 Discussion . . . 61
5.6 Conclusions . . . 63

II Design of an Infrastructure Network for Railway Scenarios 65

6 Definition of a Backbone Topology for a Linear Access Network 67
6.1 Introduction . . . 67
6.1.1 Requirements . . . 68
6.1.2 Problem Definition . . . 69
6.2 Topology Definitions . . . 70
6.2.1 Chordal-2 Topology . . . 72
6.2.2 Chordal-JC^2 Topology . . . 82

7 Analysis of a Backbone Topology for Linear Access Network 111
7.1 Introduction . . . 111
7.2 Parametric Analysis . . . 111
7.2.1 Number of Gateways . . . 112
7.2.2 Total Length of Links . . . 116
7.2.3 Maximum Distance to the Root . . . 120
7.2.4 Number of Blocked Links . . . 121
7.2.5 Number of Access Nodes between Contiguous Gateways . . . 125
7.2.6 Concluding Remarks . . . 127
7.3 Network Failure Analysis . . . 128
7.3.1 Speed effects on the Disconnection Time . . . 129
7.3.2 Types of Network Failures at the Linear Access Network . . . 130
7.3.3 Probability Analysis of Network Failure . . . 131
7.3.4 Analytical Approximation for large topology sizes . . . 135
7.3.5 Discussion . . . 139
7.3.6 Concluding Remarks . . . 141

8 Selection of a Backbone Topology for a Linear Access Network 143
8.1 Introduction . . . 143
8.2 Methodology . . . 144
8.3 Linear Access Network of 200 Nodes . . . 145
8.3.1 Solutions for a Single Topology . . . 145
8.3.2 Solutions Combining Two Topologies . . . 149
8.3.3 Conclusion . . . 151
8.4 Linear Access Network of 2000 Nodes . . . 151
8.4.1 Solutions for a Single Topology . . . 152
8.4.2 Solutions Combining Two Topologies . . . 155
8.5 Final remarks . . . 156

9 Evaluation of the Infrastructure Network and the Handover Scheme 157
9.1 Metro Scenario - Line 1 in Santiago de Chile . . . 157
9.1.1 Backbone network . . . 157
9.1.2 Simulated Scenario . . . 159
9.1.3 Handover Time - GAL Time . . . 160
9.1.4 Round Trip Time . . . 164
9.1.5 Packet Losses . . . 167
9.1.6 Conclusions . . . 168
9.2 Train Scenario - Nice Marseille Railway . . . 169
9.2.1 Backbone Network . . . 170
9.2.2 Simulated Scenario . . . 171
9.2.3 Synthesis of Results and Conclusions . . . 172
9.3 Summary and Conclusions . . . 174

10 Conclusions 177
10.1 Summary of Contributions . . . 177
10.2 Advantages and Disadvantages of the Proposed Solution . . . 178
10.3 Perspectives . . . 179

A Tables 181
A.1 Tables in Chapter 5 . . . 181
A.2 Tables in Chapter 7 . . . 182

Bibliography 183

List of Figures

1.1 Communication on trains scenario. 2
1.2 Thesis organization. 6

3.1 Effects of the handover frequency on the overlapped coverage distance. . . 23

4.1 Moving switch example. 34
4.2 Spiderman Device architecture. 36
4.3 Wireless Switch Access Point architecture. 37
4.4 Spiderman handover procedure. 38
4.5 Maximum handover time measured for different speeds by simulation. . . . 43

5.1 Path between two consecutive WSAPs. 46
5.2 Gratuitous ARP Loop procedure. 48
5.3 Gratuitous ARP delay for 250 MAC addresses. 51
5.4 ARP Delays for 250 MAC addresses for an in-motion SD at 90 km/h . . . 52
5.5 Theoretical Gratuitous ARP time considering retransmissions. 53
5.6 Theoretical Gratuitous ARP time for different *BurstSize*. 54
5.7 Theoretical Gratuitous ARP time for *InterARPDelay=7* ms, *InterBurstDelay=20* ms and *BurstSize=10*. 55
5.8 Simulated scenario for the GAL evaluation. 56
5.9 GAL delay (in seconds) for different number of hosts. 57
5.10 Maximum GAL time: theoretical and simulated values. 58
5.11 Handover time for 50 MAC addresses at different speeds. 59
5.12 Handover time for 50 MAC addresses at 60 m/s. 60
5.13 ICMP Packet losses for 50 MAC addresses at different speeds. 60
5.14 ICMP Ping Round Trip Time (RTT) for 50 MAC addresses at different speeds. 61

6.1 Backbone and linear access network connectivity. 68
6.2 Linear Topology Network formal representation. 71
6.3 Cases for segments length when $\ell(u, v)$ is odd for Chordal-2. 72
6.4 Chordal-2 for $n = 15, k = 3$, or $C_2(15, 3)$. 73
6.5 Chordal-2 for $n = 12, k = 3$ with the last layer incomplete. 74
6.6 Maximal recursion level $k_{max}(n)$ for Chordal-2. 75
6.7 Number of gateways $m(n, k)$ for Chordal-2 76
6.8 Average distance between contiguous gateways $\mathcal{D}_{avg}(n, k)$ for Chordal-2. . . 76
6.9 Total length of links for Chordal 2 topologies. $\mathcal{L}(n, k)$. 77
6.10 Chordal-2 segment length and maximum distance. 79
6.11 Chordal-2 maximal distance to the root node (d_{max}) for $k = k_{max}$. 80
6.12 Chordal-2 Linear Network Topology for $n = 25, k = 4$ 80

6.13 Chordal-2 disabled links ratios. 82
6.14 Example of a JC^2 topology with parameters $n = 27, s = 3, k = 2$ and a linear kernel. 83
6.15 Balance example for 12 and 13 nodes with root middle-left 84
6.16 Kernel functions for $s = 3, 5$ and 7 85
6.17 Partial deployment of an dtar kernel for $s = 7$ segments. 86
6.18 Linear kernel deployment for $s = 3$ 87
6.19 Maximum level of recursion for $JC^2 - lin$ with $s = 3, 5, 7$. 87
6.20 Number of gateways for $s = 3, 5$ and 7 89
6.21 Average distance between gateways for linear kernels 90
6.22 Total length of links for linear kernels with $s = 3, 5, 7$ segments. 91
6.23 Maximum distance to the root for $s = 3, 5$ and 7. 92
6.24 $JC^2 - lin3$ topology for $n = 19$ and $k = 2$. 93
6.25 Star kernel deployment for $s = 5$ and $s = 7$. 94
6.26 Maximum level of recursion for JC^2-star with $s = 5$ and $s = 7$. 95
6.27 Star kernel for $s = 5$ with $k = 2$ illustrating coincident gateways 96
6.28 Number of Gateways for $s = 5$ and $s = 7$. 97
6.29 Average distance between gateways for star kernels 98
6.30 Total length of links for $s = 5$ and $s = 7$. 99
6.31 Maximum distance to the root for the $JC^2 - star$ kernel with $s = 5$ and $s = 7$. 100
6.32 Star kernel for $s = 5$ with $k = 2$ illustrating loops in the backbone network 101
6.33 Number of blocked links for $JC^2 - star$ kernel with $s = 5$ and $s = 7$. 102
6.34 Binary tree kernel deployment for $s = 7$ 103
6.35 Maximum level of recursion for $JC^2 - bin$ with $s = 7$. 104
6.36 Number of gateways for JC^2 topology with a binary kernel of $s = 7$. 105
6.37 Average distance between gateways for the binary kernel 106
6.38 Total length of links for $JC^2 - bin$ kernel with $s = 7$ 107
6.39 Maximum distance to the root for the $JC^2 - star$ kernel with $s = 7$ 108
6.40 Maximum distance to the root for the $JC^2 - star$ kernel with $k = 4$ and $3 \leq n \leq 2500$ 109

7.1 Gateway density for all topologies and $k = 1$ 112
7.2 Gateways density for Chordal-2 topology. 113
7.3 Gateways density for $lin3$ kernel 114
7.4 Gateways density for $lin5$ kernel 114
7.5 Gateways density for $lin7$ kernel 115
7.6 Gateways density for $star5$ kernel 115
7.7 Gateways density for $star7$ kernel 116
7.8 Gateways density for $bin7$ kernel 116
7.9 Total length of links for Chordal-2 topologies 117
7.10 Total length of links for linear kernels 118
7.11 Total length of links for star kernels 118

7.12 Total length of links for binary kernel . . . 119
7.13 Total length of links for all topologies and $k = k_{max}(n)$ . . . 119
7.14 Maximum distance to the root v/s topology size . . . 120
7.15 Maximum distance to the root for Chordal-2 . . . 121
7.16 Maximum distance to the root for linear kernels . . . 122
7.17 Maximum distance to the root for star kernels . . . 122
7.18 Maximum distance to the root for binary kernel . . . 123
7.19 2 access nodes down and no failure. . . . 123
7.20 Backbone and access network fails together . . . 124
7.21 Average distance between gateways for Chordal-2 . . . 125
7.22 Average distance between gateways for linear kernels . . . 126
7.23 Average distance between gateways for star kernels . . . 127
7.24 Average distance between gateways for binary kernel . . . 127
7.25 Definition of Δ, c_r, and c. . . . 129
7.26 Disconnection time (log-scale) for different speeds and different number of failed nodes. . . . 130
7.27 Probability of observing a least a segment of l access nodes for $n = 40$ and $p = 0.01$ . . . 133
7.28 $\mathbb{P}[L \geq l]$ for $n = 40$, $p = 0.001$ and $p = 0.0001$ . . . 134
7.29 Syndrome dependence: $\mathbb{P}[S^j|S^i] \leq \mathbb{P}[S^j]/(1-p)$. . . . 136
7.30 Sketch of the proof for a syndrome . . . 137
7.31 Length of the contributing syndromes starting in an interval. . . . 138
7.32 $\mathbb{P}[L \geq l]$ for $n = 40$ and $\mathcal{D} = 4$ . . . 139
7.33 $\mathbb{P}[L \geq l]$ $n = 200$ and $p = 0.001$. . . . 140
7.34 $\mathbb{P}[L \geq l]$ $n = 2000$ and $p = 0.001$. . . . 141

9.1 Proposed backbone network for Metro line 1 in Santiago de Chile. . . . 158
9.2 Simulation scenario for Metro line 1 in Santiago de Chile. . . . 159
9.3 GAL delay (in seconds) for two metros **without** traffic. . . . 161
9.4 GAL delay (in seconds) for two metros **with** traffic. . . . 162
9.5 Length of the Spiderman Device's active transmission queue. . . . 162
9.6 ICMP Round Trip Time (in milliseconds) for two trains **with** traffic. . . . 165
9.7 ICMP Round Trip Time (in milliseconds) considering up-to 3 consecutive failures in the access network. . . . 166
9.8 ICMP Round Trip Time (in milliseconds) considering a single failure in the backbone network. . . . 166
9.9 ICMP Packet lost (%) for two trains **with** traffic. . . . 167
9.10 ICMP Packet losses (%) up-to 3 consecutive failures in the access network. 168
9.11 Nice-Marseille backbone network proposal . . . 170
9.12 Nice-Marseille simulated scenario . . . 172

List of Tables

2.1 Summary of train operators providing Internet on-board. 19

8.1 Solutions for $n = 200$ provided by Chordal-2 topology 146
8.2 Solutions for $n = 200$ provided by linear kernel with $s = 3$. 146
8.3 Solutions for $n = 200$ provided by linear kernel with $s = 5$. 147
8.4 Solutions for $n = 200$ provided by linear kernel with $s = 7$. 147
8.5 Solutions for $n = 200$ provided by star kernel with $s = 5$. 148
8.6 Solutions for $n = 200$ provided by star kernel with $s = 7$. 148
8.7 Solutions for $n = 200$ provided by binary kernel with $s = 7$. 149
8.8 Summary of single topology solutions for 200 nodes. 149
8.9 Most attractive feasible solutions for $n = 200$ with two combined topologies. 150
8.10 Solutions for $n = 2000$ provided by Chordal-2 topologies. 152
8.11 Solutions for $n = 2000$ provided by linear kernel with $s = 3$. 152
8.12 Solutions for $n = 2000$ provided by linear kernel with $s = 5$. 153
8.13 Solutions for $n = 2000$ provided by star kernel with $s = 5$. 153
8.14 Solutions for $n = 2000$ provided by star kernel with $s = 7$. 154
8.15 Solutions for $n = 2000$ provided by binary kernel with $s = 7$. 154
8.16 Summary of single topology solutions for 2000 nodes. 154
8.17 Most attractive feasible solutions for $n = 2000$ with two combined topologies. 155

9.1 GAL times (in seconds) for failure scenarios up-to 3 failed access nodes. . . 163
9.2 GAL times (in seconds) considering a single failure in the backbone network. 163

A.1 GAL delay (in seconds) for different number of hosts **without** background traffic. 181
A.2 GAL delay (in seconds) for different number of hosts **with** background traffic. 181
A.3 Disconnection time for different speeds and number of failed nodes 182

Chapter 1

Introduction

This thesis focuses on the problem of providing connectivity (such as Internet) to trains passengers while they travel at high-speed. In this introduction, we present first the motivations, the context and the main challenges entailed by this problem. Then, we identify specific questions raised by the challenges. Finally, we summarize our contributions and answers to these questions, which gives an insight to the general problem.

1.1 Motivation and Context

Internet is almost everywhere. The penetration of mobile devices into the every-day life is fostering new applications in ubiquitous networking, resulting in users being connected all the time, including in mass transit transportation systems. According to a forecast from the International Data Corporation [1] (IDC) [Drake *et al.* 2010], the working force in several regions of the world is giving more importance to mobility in their work. "As mobility continues to play a key role in enabling companies to achieve greater productivity worldwide, IDC expects the global mobile worker population to increase from 919.4 million in 2008 to more than 1.19 billion in 2013, representing nearly 35% of the worldwide workforce", said Sean Ryan, research analyst for IDC's Mobile Enterprise group.

In order to satisfy this increasing demand of nomadic connectivity, 3G networks have been deployed, covering vast geographical areas and allowing users to access the Internet via 3G, when WiFi is not available. Nevertheless, while 3G networks offer a good quality connection, they are expensive and end-user bandwidths still do not reach the levels achieved by fixed broadband networks. Furthermore, when using 3G networks on public transportation systems such as a high-speed train, the continuity of the connection might be affected by the simultaneous handover of all the passengers on-board the train. Therefore, while 3G networks can hold a single connection practically up-to 300 km/h [Jang *et al.* 2009], there are still problems to solve before providing a real broadband Internet to high-speed trains passengers.

We focus our attention on the train networking scenario since every day more people are working on-line when travelling or commuting. Several solutions have already been proposed to provide a continuous connection to passengers on-board high-speed trains. Most of them are based on a blend of wireless technologies which integrates satellites, 3G, WiMax and WiFi networks. While these solutions are operational in several train lines

[1] IDC is a market research and analysis firm specializing in information technology, telecommunications and consumer technology.

around the world, their bandwidth is limited to the satellite or 3G network capacities, which are expensive, and subject to weather conditions in the case of satellites. Therefore, the quality of the service can not be sustained all the time. In this thesis, we raise the question of whether a broadband connectivity is possible for trains passengers with a connectivity based only on an unlicensed wireless technology, such as WiFi technology. We address this question from the point of view of a train company that is willing to fully operate the solution and to provide integrated services with QoS to its passengers.

1.2 Challenges

In Chapter 2 we review the ongoing developments on the problem. This review shows that there are still two main challenging issues to deal with: the handover of all the on-going connections of trains passengers to Internet at high speeds, and the design of an infrastructure network to provide them a continuous connectivity. Hence, the research question addressed in this thesis is: **how to provide a continuous connectivity to trains passengers only relaying on an unlicensed wireless technology, and more specifically, WiFi technology**. We propose to use the WiFi technology since it is a simple, highly available and low cost technology. Furthermore, the bandwidth it is capable to achieve is far beyond the bandwidth offered by other technologies, such as satellites, 3G or even WiMax.

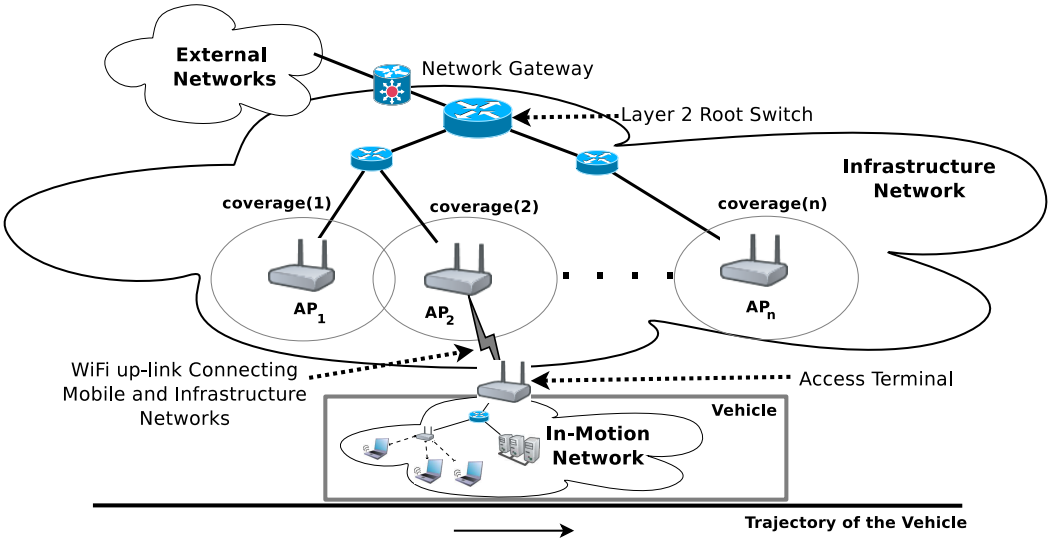


Figure 1.1: Communication on trains scenario.

In this context, as a train follows a well defined trajectory (the railway), our question becomes: how to provide network connectivity to passengers on-board a vehicle (trains, metros, etc.) along a predefined trajectory. As a counterpart, the use of WiFi technology implies to deploy a large number of access points along this trajectory.

Figure 1.1 depicts our reference scenario: At the bottom of the figure we observe a high-speed vehicle following a given trajectory. In this vehicle, passengers get access to the network through an on-board **access terminal**. This access terminal uses a WiFi based link to establish the connection with a set of WiFi Access Points (AP) placed along the trajectory. These APs are interconnected by an **Infrastructure Network**, which carries the traffic between trains passengers and a *Network Gateway* (depicted at the top of the figure). This *Network Gateway* provides the access to **external networks** (such as Internet).

This scenario poses several challenges that need to be tackled, such as the continuous connection of passengers on-board, the provision of QoS to share the access terminal connection, the design of the backbone network to connect the access network, just to mention a few. In this thesis, we study two of them:

1. the **handover** of the networked devices on-board the vehicle in order to provide a continuous connection, regardless of the speed of the vehicle[2] or the number of passengers (or hosts).

2. the design of an **infrastructure network** capable of providing a robust connection to all the vehicles along the same trajectory, assuming there are several vehicles following the same path (Metro/Tramway scenario).

The first challenge is interpreted as the handover of an **in-motion network**, since all the devices on-board the vehicle represent a static network, which follows as a whole the same trajectory as the vehicle. Therefore, from now, we refer to this challenge as how to perform an **horizontal handover** of an in-motion network with a WiFi up-link between the in-motion and infrastructure networks. As defined in the literature [Wiethoelter & Emmelmann 2010], the horizontal handover refers to the handover between access points using the same technology. This challenge entails several problems, for example the scanning of the next AP, the effects of the number of hosts inside the in-motion network, the effects of the speed on the handover, the update of routes when the in-motion network changes its AP. These problems can be classified into two classes:

- **how to perform the handover of an in-motion network on-board a high-speed train while avoiding buffers overflows which causes packet losses**.

- **how to inform the infrastructure network about the new AP by which the in-motion network (passengers) is now reachable**, with all the consequences it implies.

The second challenge opens a wider range of problems, since to design an infrastructure network, we need to find a **backbone topology** to connect all the WiFi AP, which we name from now *access nodes*. All these access nodes constitute the **access network**,

[2] assuming that WiFi operates correctly at that speed

which together with the **backbone network** constitute the **infrastructure network**. The challenge of designing a backbone topology for an access network comes from the **size of the access network**, which might vary from hundreds to thousands of WiFi APs (with small radio coverages). Therefore, the question is how to define a backbone topology capable of connecting all these APs with a reasonable number of hops to reach the *Network Gateway* (which impacts the delay observed in the network). Another question is what **routing protocol** is appropriate to route packets within the network, considering factors such as routes reconfiguration (in case of failure) and the size of the access network. This latter factor reduces considerably the space of possible routing protocols we can use. The final question is how **robust** is the infrastructure network, that is how both networks together should perform in case of failures in order to be considered as a robust network. In other words, how many failures it can tolerate before to be unable of providing the service with an expected QoS.

1.3 Methodology

In the previous section we stated several questions related to our problem. In this section we define the methodology adopted in this thesis, which we summarize as follow:

1. We divide the two main research questions into simpler questions.
2. We define the relation and order between all the simpler questions.
3. We define the studies to answer them.
4. We draw conclusions from our answers.

At the top level, we start this process with the two questions coming from the two challenges identified in the previous section.

1. Q_h : How to perform the **handover** of an in-motion network at high-speed avoiding packet losses?
2. Q_d : How to **design** an infrastructure network providing a robust connectivity to an in-motion network?

With respect to the question Q_h, we assume that any handover scheme should avoid packet losses in order to provide a real continuous connection. When focusing on the horizontal handover of WiFi devices, the main causes of packet losses are: the discarded packets due to the fact that buffer is full, and the late arrival of packets to an Access Point (AP). The first cause is known as the buffer overflow problem. The radio interface needs to buffer the outgoing traffic when scanning for the next AP and negotiating the re-association with it. When this buffer is full, all further packets are discarded. The second cause is known as the misrouting problem, since packets are following a route that is not yet updated on the infrastructure network, leading them to reach the old AP instead the new one. So, we divide Q_h in two sub-questions:

1. Q_{hh} : How to perform the **horizontal handover** of an in-motion network avoiding buffer overflows?

2. Q_{hr} : When performing the **handover**, how to update the **routing** information in the infrastructure network avoiding the misrouting of packets?

With respect to question Q_d, we assume that an infrastructure network should combine two important design elements: 1) a topology to connects all their nodes, and 2) a routing protocol to allow packets to transit through it. In addition, we are interested in assessing how the proposed handover scheme (Q_h) operates on an infrastructure network (proposed by 1) considering a routing protocol (proposed by 2). Thus, we refine Q_d into three sub-questions:

1. Q_{dt} : How the infrastructure network's nodes should be connected to provide a resilient connection? (In other words, the design of the backbone network **topology**).

2. Q_{dr} : What **routing** protocol should be used in the infrastructure network?

3. Q_{de} : Is the proposed handover scheme able to work on the proposed infrastructure network given the selected routing protocol? That is to say an integration **evaluation**.

Questions Q_{dr} and Q_{de} define clear objectives on their own (discussed later on). However, Q_{dt} can be still divided into more simpler questions:

1. Q_{dtd} : What topologies are suitable to define a resilient "backbone network"? We need to **define** the topologies we aim at studying.

2. Q_{dtt} : How is the **trade-off** between cost vs. benefits of the defined topologies?

3. Q_{dtf} : What is the **failure** tolerance of the defined topologies?

4. Q_{dts} : How to **select** from the defined topologies, a good and resilient backbone topology for an access network?

The Diagram 1.2 depicts the relationships between all these questions and provides a notion of the structure of this thesis. In the following, we explain the studies presented in order to answer these questions.

Regarding the last level of questions, we also have the constraint to first answer Q_{dp} before addressing Q_h and Q_d, since the ability of a network to overcome a failure depends on how the routing protocol takes advantage of the topology to provide alternate routes and bypass the failure. In addition, the question Q_{hr} is also related to the selection of a routing scheme, therefore, our first objective is to define a **routing protocol** for the whole network.

Regarding Q_h, we notice the handover scheme should complete as fast as possible all the tasks involved in transferring the data-link layer from one AP to the next one (see

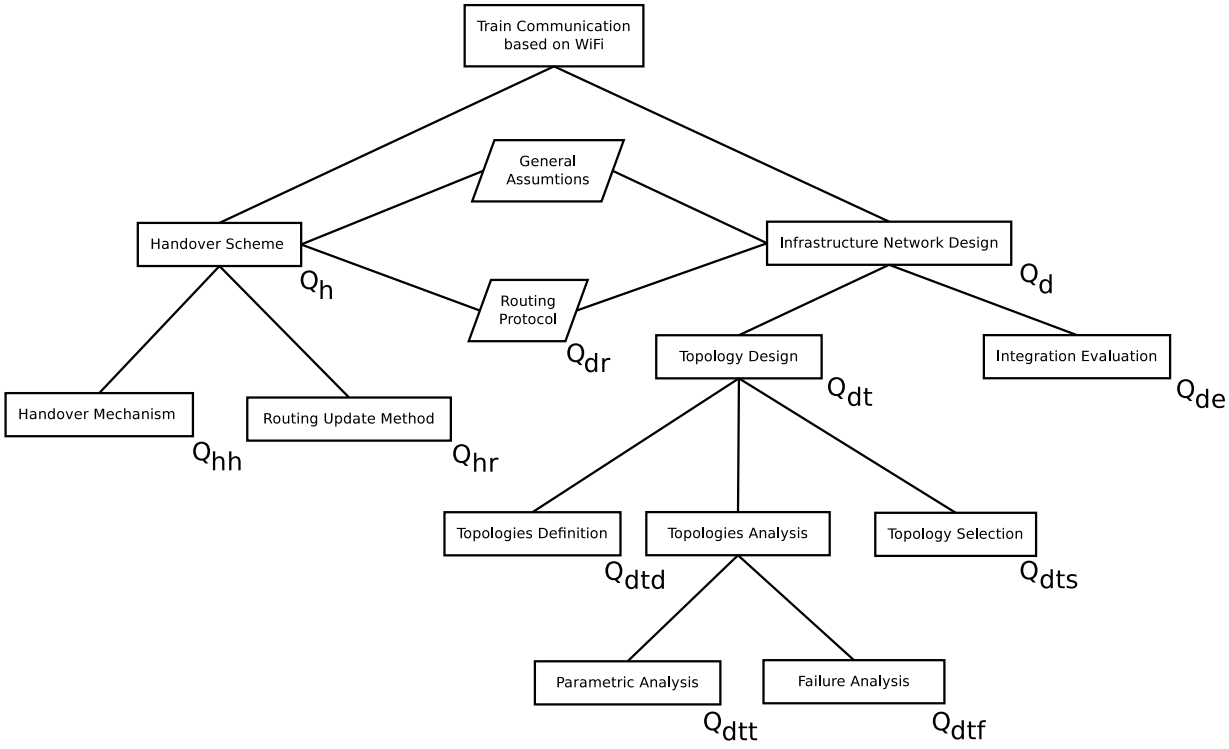


Figure 1.2: Thesis organization.

[Mishra *et al.* 2003] for more details). Otherwise, the probability of observing a buffer overflow is higher. The time to complete these tasks depends mostly on two parameters: the *overlapping distance* of the radio coverages between contiguous APs and the *speed* of the in-motion network (the speed of the vehicle with the in-motion network on-board). Therefore, our next objective is to define the **general assumptions** on the wireless coverage to define the overlapping distance and the maximum speed of the in-motion network. With these assumptions, we study a proposal for an **horizontal handover mechanism** considering no packet losses due to buffer overflows. This handover mechanism also requires a procedure to update the routing information in the infrastructure network. The objective of this updating procedure is to avoid (as much as possible) any packet losses due to the misrouting problem. However, this procedure depends also on two parameters: (i) the number of hosts inside the in-motion network, and (ii) the limited time taken by the in-motion network in crossing the *overlapping distance*. Hence, we aim at studying a **routes updating procedure** considering the routing protocol defined by Q_{dr} and the parameters defined by (i) and (ii). When achieving these two latter objectives, we are in position to propose a **handover scheme** (handover mechanism + routes updating procedure) for an in-motion network (the answer to Q_h).

Regarding Q_d, we focus on the design of a backbone network, since the access network is defined by the length of the trajectory followed by the vehicle. Therefore, it is fixed to a certain number of access nodes. According to this number of access nodes, a **backbone**

topology is defined considering three main requirements: 1) at least two paths from any *access node* to the *Network Gateway*; 2) a good balance between its number of nodes, the total length of links required and the number of hops to the root; and 3) the failure tolerance of the access network. The first requirement is related to the capacity of the routing protocol to find alternative routes to cope with a failure. According to Sterbenz [Sterbenz *et al.* 2010], the survivability of a network is based on the diversity of alternate paths to go from any source to any destination. In our case, we are interested in the diversity of routes between the *access network* and the *Network Gateway*. To provide this diversity within our backbone network, we **define two redundant topologies** coming from the chordal family of topologies. Each one exhibits a variable number of alternative paths, depending on its parameters, so we study how these parameters influence the ability of the routing protocol to exploit the redundancy of the backbone network. The second requirement is about the trade-off between the cost vs. benefits. The relationship between the properties and parameters of each backbone topology yields to the selection criteria, which we study by means of a **parametric analysis**. This criteria is based on the deployment cost of a backbone topology (number of nodes and total length of links), the number of hops between an access node and the root node (the Network Gateway), and the number of alternate paths (disjoint or not disjoint) to the root. The third requirement is related to the perception of a train passenger to experience a disconnection from the network (there are failures cases that might lead or not to an observable error in the network [Sterbenz *et al.* 2010]). Therefore, we are interested in determining the **probability of experiencing a failure case** which might yield to an observable disconnection from the infrastructure network.

Finally, we investigate the question Q_{de}, which raises the question whether the handover scheme works when integrating: 1) the selected infrastructure network, 2) the selected routing protocol, 3) a train with an in-motion network. To answer this question, we perform an **integration evaluation** by studying the delay observed in the infrastructure network (with/without failures), the handover time and packet losses experienced by train passengers when exchanging traffic with an external network. The result of this evaluation are obtained by simulation. They should determine whether all the contributions proposed by this thesis are capable of solving the handover and design problems stated in Section 1.2.

In summary, the 8 objectives of this thesis are enumerated as follows:

1. Define a routing protocol for the whole network.
2. State the assumptions on wireless communications, vehicle's speed and traffic profiles.
3. Study a handover mechanism for a WiFi network capable of performing the handover of an in-motion network at high-speed, without having packet losses.
4. Study a routes updating procedure according to the routing protocol defined in 1.
5. Define a set of backbone topologies for an access network of n nodes and study their network properties.

6. Analyze costs versus benefits of each topology, assessing the trade-off between their properties.

7. Select a backbone topology for an access network of n nodes to build an infrastructure network.

8. Evaluate the integration of the proposed infrastructure network, the handover scheme and the selected routing protocol on a train scenario.

1.4 Contributions and Organization

This thesis is decomposed in several chapters. Each chapter is related to one or more objectives stated above. The contributions of this thesis are two-fold: Chapters 4 and 5 present our contributions to the study of the horizontal handover for high-speed vehicles on a predefined trajectory. Chapters 6 to 8 present our contributions to the design of an infrastructure network for a linear access network (linear since it is deployed along the trajectory of the vehicle). More precisely, the organization of this thesis is the following:

- **Chapter 2 - State of the Art: Internet on trains** In this chapter we present a review on the train communications in a transversal view. While we base our review on the literature, we also review research projects from which the current solutions come, the companies who implement these solutions, and the technologies they use for providing to passengers the service of "WiFi on-board".

- **Chapter 3 - Assumptions** In this chapter we state the assumptions we made to define the scope of this work. We discuss the selection of a routing protocol for the infrastructure network, the wireless communication assumptions and the traffic profiles we expect.

- **Chapter 4 - Horizontal handover of an in-motion network** In this chapter we deal with the problem of performing the horizontal handover of an in-motion network in a high-speed train scenario. The main objective is to propose a handover mechanism capable of transferring the data-link layer from one access point to the next one ensuring low packet losses, or when possible, no packet losses.

- **Chapter 5 - Routing update of an infrastructure network** In this chapter we propose a routes updating procedure to inform the infrastructure network about the handover of the in-motion network. We discuss its scalability according to the number of hosts inside the in-motion network, as well as its operational time, since we consider the time taken by the routes updating procedure the key component of the handover time.

- **Chapter 6 - Definition of a backbone topology for linear access network** In this chapter we define a set of backbone topologies to place on top of the linear

access network. We study their network properties, which are used afterwards for the analysis of their costs versus benefits.

- **Chapter 7 - Analysis of a backbone topology for linear access network** In this chapter we present two studies: 1) a parametric analysis of the network properties of each defined topology; and 2) a failure tolerance analysis to determine the probability of observing a disconnection from the network. Both studies aim at assessing the balance between the cost versus benefit of each proposed backbone topology.

- **Chapter 8 - Selection of a backbone topology for a linear access network** In this chapter we aim at selecting a backbone topology from the proposed backbone topologies. We do the selection for two representative cases: a small topology (200 access nodes/30 km long), representing a small mass transit transportation system (such as a metro/tramway/bus), and a large topology (2000 access nodes/300 km long), representing a inter-urban transportation system, such as a high-speed train.

- **Chapter 9 - Evaluation of the infrastructure network and the handover scheme** In this chapter we present an evaluation by means of simulation of the selected backbone topology. This evaluation aims at determining how the proposed handover scheme (handover mechanism + routes updating procedure) operates on the infrastructure network studied in Chapter 8 for two train scenarios: a metro line (20 km long) and a train line (200 km long).

- **Chapter 10 - Conclusions and Perspectives** In this chapter we draw our conclusions from all the results presented in each chapter. Perspectives of this work are discussed and further directions of work are identified.

Chapter 2

State of the Art: Internet on Trains

In this chapter we present a review of the communication systems for trains passengers. While we base our review on the literature, we also review research projects from which the current solutions come from, how they are implemented by the companies and the technologies they use for providing the "Internet on-board"service.

2.1 Introduction

[Fokum & Frost 2010] published in late 2009 a survey on methods for providing broadband Internet access on trains. They reviewed some of the initial approaches, current technologies, and implementation efforts across the world. While we base our review on this survey, we explore more carefully the implementation efforts, focusing on actual results obtained with the wireless technologies they employed. For definitions and terminology on heterogeneous wireless communications, we invite the reader to check the work presented by [Gondi 2009]. The "learned lessons" pointed out by [Fokum & Frost 2010] are summarized as follows:

- Most of the reviewed deployments provide a **single access terminal** per train, which is **shared** by all passengers on the train. Such an architecture reduces the handover of all passengers' connections into a single terminal handover, which aggregates all the traffic by means of techniques such as Masquerading or Network Address Translation (NAT) [Egevang & Francis 1994, Wing 2010].

- As initially stated by [Rodriguez *et al.* 2004], and then confirmed by several other research and experimentation efforts, the *single access terminal* approach works better when combining a set of wireless technologies instead of relying only on one way to access the Internet. Thus, one technology is used to provide the connectivity, and when this technology is not longer present, a second technology can be used as "gap-filler". This network selection mechanism implies that there will be always a preferred technology to provide the connection.

- Switched Ethernet may be used in a carrier-grade network to support fast moving users, however authors concluded that some extensions are needed to improve Ethernet's recovery from link failures. We add to authors conclusions that several studies have been published [De Sousa 2006, Huynh *et al.* 2009, Padmaraj *et al.* 2005b] with interesting solutions to enhance the QoS and fault tolerance in large Ethernet networks.

- While WiFi networks are able to operate on railways scenarios, this network technology is typically not used as access network because its deployment costs might be too high when covering large scenarios.

In the following, we classify the existing solutions to provide "Internet on-board" of trains. We start reviewing the current developments in order to identify the technologies used by each solution, to furthermore, analyze these technologies from the point of view of their ability to provide a quality connection to trains' passengers.

2.2 Current Deployments

As pointed out by [Fokum & Frost 2010], all the solutions use an "**access terminal**" to provide connectivity to the passengers on-board the train. Usually a hot-spot is deployed inside each carriage, as proposed by [Bianchi *et al.* 2003], and they get access to external networks through the access terminal. A blend of network access technologies are embedded into this terminal. Each technology is used to access an infrastructure network opportunistically. The criteria to select a particular network technology among the available ones are typically the quality of the connection (signal strength), delay, throughput or costs. The point of divergence between solutions lays on the different technologies employed by the access terminal and how it integrates them in order to provide a continuous connection. We identify three categories:

1. Satellite + gap-filler communication.
2. Bonded 3/3.5G communication.
3. Trackside radio communication.

The first category uses a satellite as the main link with external networks (for example Internet). When the satellite is blocked (tunnels, stations, etc.), the access terminal switch to a terrestrial network (the gap-filler) in order keep the access terminal connected. This solution has been introduced by the **ROSIN** project [Fadin *et al.* 1998], studied by the **TRAINCOM** project [Gatti 2003] and the **FIFTH** project [Schena & Ceprani 2004], and applied later by the **InteGRail** project [Shingler *et al.* 2008] within the context of an intelligent integration of train information systems [Billion & Van den Abeele 2007]. With the time, this kind of solution has been adopted by the Industry, and deployed by train operators such as ***TGV*** and ***Thalys***, just to mention two cases. *TGV* uses a satellite as main link and WiFi and 3G as gap-fillers. On the contrary, *Thalys* uses only a 3G network as gap-filler. In both cases, telecommunication companies has been involved in the R&D of the solution: *Orange Labs* in the case of *TGV*, and *21Net* in the case of *Thalys*. The handover between the employed technologies (vertical handover) is handled at layer 3 with a Mobile IP protocol. Thus, all the traffic streams coming from the different access networks (Satellite, 3G or WiFi) are concentrated in a Network Operation Center (NOC), from which the aggregated traffic reaches external networks.

The second category is based on the use of several terrestrial networks to access external networks. We name terrestrial network any 3/3.5G network deployed over landmasses. The access terminal integrates several links (up-to 8 in some cases) with different mobile phone operators in order to balance the traffic among them. Thus, the access terminal can deal with the lack of coverage of one operator by supplying it with another operator with better coverage. In the special case of no coverage, a gap-filler network such as WiMax can be deployed. This solution comes from industry companies like *Icomera* and *Nomad Digital*. Some train operators have adopted this type of solution, for example ***Amtrak*** in the US, and ***GNER*** and ***Virgin*** in the UK. The latter is using WiMax as access network along the track of the train. The case is special with the ***ICE*** trains in Germany. They integrate 3G networks with a FLASH-OFDM based network. The use of this latter technology is new in the context of railways communications. The mobile phone operator *T-Mobile* has deployed such a network has a gap-filler initially, but soon after it was demonstrated as a feasible solution, they extended its coverage. We invite the reader to check the work presented by [Arjona *et al.* 2008] when looking for further details on the operation of a FLASH-OFDM based network. In this work, authors present quantitative measurements of the performance of this technology. The vertical handover in all the mentioned cases is handled at layer 3 by Mobile IP protocols.

The third category is less known, but still very promising. It consists in the deployment of a trackside infrastructure network for providing connectivity to the access terminal. This connectivity is achieved by means of small coverage radio cells, which can use radio bands from 5/6 GHz (ISM Band) up-to 10.0 GHz (licensed band). This solution has been proposed by the research project **WIGWAM** [Fettweis & Irmer 2005] as well as for the **FAMOUS** architecture [Greve *et al.* 2005]. While the latter is still as a research idea, the former has found its way to the Industry. The ***Shanghai Transrapid*** (a MAGLEV train running at 500 km/h) deployed between the Shanghai airport and the city downtown (30 km long) uses a communication system based on a trackside radio communication. The infrastructure network relies on Fiber Optic links and radio base stations are deployed along the track of the train. The company behind the implementation of this solution is *Telefunken RACOMS*, a German company involved in the WIGWAM project. The handover scheme proposed by this project was introduced by [Emmelmann 2005, Emmelmann 2010]) and implemented by *Telefunken RACOMS*. In short, the handover scheme is based on several radio cells controlled by a unique Media Access Control (a centralized MAC scheme). Thus, the handover between cells connected to the same MAC is handled as an inner-handover (transparent to upper layers) and the handover between cells connected to different MACs is handled by a negotiation at MAC level.

In summary, we identify several combinations of technologies that are being used to provide Internet "on-board": Satellites, 3G, WiMax and WiFi. In the following, we review these technologies focusing on their capacity in terms of maximum bandwidth and delay that the access terminal can achieve.

2.3 Technological Review on "Internet on-board"

In this section we review the used technologies on providing Internet to trains passengers. We search for documented cases in the literature, making emphasis on Satellite, 3G, WiMax and WiFi evaluations and experiences.

2.3.1 3/3.5G Connectivity

Based on our review of the literature and documented measurements from field studies, we found the following three cases:

- **O2 UK GPRS/HSDPA Network.** The UK research network operator ja.net has presented the study "Theoretical and Practical Survey of Backhaul Connectivity Options" [Georgopoulos *et al.* 2010]. In this context, they performed several field measurements with mobile phone operator O2 evaluating the TCP performance by using the measurement tool *iperf*. This tool is able to report the bandwidth and packet losses at application level (TCP or UDP) by means of tracking the sequence number of the stream of packets. Their results showed the average throughput of a TCP flow was 30 Kbit/s in GPRS when HSDPA were not available (rural land) and 340 Kbit/s with 20% losses in HSDPA. The worst latency with their campus was 700 ms with GPRS. It is worth to mention that [Georgopoulos *et al.* 2010] pointed out the contrast in respect of the link speed announced by the operator (56 Mbit/s on downstream and 22 Mbit/s on upstream) at the moment of the test. When checking the statement of the operator nowadays (Oct. 2010) in respect of the provided link speed, they only say there is no guarantee on speed.

- **Two Korean EVDO/HSDPA Networks.** The study presented by [Jang *et al.* 2009] compared the performance of two mobile ISP in Korea. Each ISP is operating two networks: a 3G and 3.5G network (CDMA 1xEVDO and HSDPA respectively). The measurements were obtained in two scenarios: 1) a high-speed train running at 300 km/h and 2) car running at 100 km/h. Both scenarios considered TCP and UDP evaluations in mobile and static mobility. Results for mobile scenarios showed, for 3G networks (EVDO), an average on downstream of 500 Kbit/s (UDP) and 1.2 Mbit/s (TCP). In the same scenario, the 3.5G networks (HSDPA) showed an average downstream of 500 Kbit/s (UDP) and 1 Mbit/s (TCP). Contrasting these results against the static scenario, authors highlight the increasing packet losses in mobile scenario. While authors explain the losses arguing Layer 3 issues (TCP mostly), we find more likely the argument of handover effects when looking their results.

In summary, the measurements presented in this section suggest that 3/3.5G network services has an expected real throughput of about 1Mbit/s downstream and 500 Kbit/s upstream. Variability of this throughput may depends on the level of usability of the

backhaul network, quality of the coverage, and of course, any traffic policing or shaping that operators may apply.

2.3.2 Satellite Connectivity

We base our review on the survey of Satellite communications presented by [Chini *et al.* 2010]. In this section we review results when evaluating a TCP flow on a satellite link, as any web connection from a train passenger. A review of the TCP performance on satellite links is presented by [Emmelmann 2000], from which we obtain the base parameters to compare the evaluated services.

Mobile Satellites Networks (MSSs) can be classified in three categories: Geosynchronous Earth Orbit (GEO), Medium Earth Orbit (MEO) and Low Earth Orbit (LEO). The GEO satellites are orbiting the Earth at an altitude of 35 800 km above ground along the equatorial plane. This is the altitude at where the orbit of a satellite matches with the the period at which the Earth rotates. The MEO satellites are orbiting the Earth at between 8000 and 12 000 km of altitude and LEO satellites are orbiting at an altitude between 500 and 2000 km. The last two types of satellites networks (i.e., Iridium and Globalstar) offers normally voice/low-rate data services (9.6-14.4 Kbit/s) mostly targeting personal communications. On contrary, GEO satellites (i.e., Inmarsat, Hispasat, Thuraya and SES-ASTRA) support broadband services targeting coverage over landmasses. We focus our attention on the GEO satellites since they are the only ones capable of providing a broadband connection for moving users. Indeed, all the services offering "Internet on-board" we reviewed use a GEO satellite up-link. A GEO satellite spots the Earth with hundreds of beams, creating on Earth a cellular-like coverage. Hence, the higher the density of beams per region (closer beams), the bigger the interference between satellites which reuses frequencies. When comparing the MSSs provided by Inmarsat and Thuraya based on the results presented by [Chini *et al.* 2010], the BGAN's Inmarsat networks arises as the best provider, but also the most expensive one.

We also reviewed the experimental results presented by [Georgopoulos *et al.* 2010] when evaluating the services provided by SAS ASTRA and Inmarsat. We aim at providing a practical notion on the performance of such a connections.

- **Astra2Connect® (SAS-ASTRA).** Astra2Connect is the name of the service provided by SAS-ASTRA. Their claim is they are able to provide high-speed Internet access (up-to 4 Mbit/s), however it can not be used in motion. The trials were performed on the rural area of Buttermere (Lake District, UK). The test consisted in measuring a TCP stream from the trial site to the Lancaster University (traversing Luxembourg where the terrestrial receiver of the Astra2Connect service is, and then GEANT to reach the campus). Results obtained were in average 990 Kbit/s in downstream and 244 Kbit/s in upstream. Latency about 600 ms. Authors pointed out the contrast between the operator claims (2 Mbit/s downstream, 256 Kbit/s upstream) and the real measurements. No information about the tool used to run the tests.

- **BGAN Network (Inmarsat).** Inmarsat's BGAN network provides Internet, voice and telephony services to end-users in areas where no terrestrial services exists. BGAN terminals can offer speeds up to 492 Kbit/s for both download and upload, although they provide also terminals for lower speeds. The *Terminal* evaluated offers 464 Kbit/s on the downstream and 128 Kbit/s on the upstream. The test evaluated a TCP stream by using the web site `Speedtest.net`, evidencing an average down/upstream of 360 Kbit/s and 120 Kbit/s respectively. Recorded latencies oscillated between 800 ms and 1200 ms.

As of 2010, the cost of a satellite terminal oscillates around 3 000 Euros, plus an annual fee of more-less 1000 Euros per 255 Mbit/s traffic allowance. In summary, based on these experimental measurements we can conclude that upload speeds are more close to the promised service than download speeds. Another interesting fact is the price/allowance, which is around 4.0 - 5.0 Euros per downloaded MByte at the mentioned speeds, without mentioning the inverse relation between weather conditions and channel quality [Durst *et al.* 1996], which makes less available the satellite connection depending on the frequency they use.

2.3.3 WiMax Connectivity

We already mentioned that the train operator Virgin, in UK, is using WiMax to provide WiFi on-board. This solution was implemented by *Nomad Digital*, who is operating similar networks in other parts of the world. Another deployment of WiMax was published in press by [Judge 2005]. The target were the commuters from Brighton to London's Victoria. Nomad Digital was involved in the implementation and T-Mobile was the service provider. The WiMax base stations were connected to T-Mobile network through ADSL up-links at 2 Mbit/s, which was a bottleneck for users since the WiMax access network is capable to run at 48 Mbit/s. These bottleneck become noticeable if no QoS provision is enforced. Therefore, the WiMax implementations in the railway context seems not to be designed to exploit all the benefits of the WiMax technology. The only advantage they obtain from this technology is its 5 km coverage range. We notice there is neither a handover management (within WiMax base stations) nor a need for an appropriate backbone network to support QoS for users.

In the literature there are not many experimental studies published about WiMax applications on railways scenarios. However, [Chow *et al.* 2009] present a novel method to use WiMax to provide an access network to high-speed trains. They propose a distributed antenna system (DAS) based on Radio-Over-Fiber. In this way, WiMax base stations can cover wider areas with less interference. Their results showed this approach might be usable while the received power (from the train) between two contiguous DAS is less than 10dB. In this case the coverage of a single WiMax base station could be extended from 3/4 km to 17 km approximately. [Cho & Pan 2009] present an evaluation methodology for WiMax on mass rapid transit systems. By means of simulations, they showed the benefits

of their model toward the economical evaluation of the deployment of such a communication network. In a different context, but always using WiMax, [Aguado *et al.* 2008] present an architecture for providing broadband wireless communication on trains, focusing on the needs of vehicular public safety systems. They performed field test of WiMax technology focusing on the train mobility, evaluating their proposal by means of simulations. Improvements on mobility managements, specially on the handover between WiMax base stations were presented. Finally, for further discussion on the handover mechanism on WiMax networks, we invite the reader to check the state of the art presented by [Ray *et al.* 2010], which pointed out in its conclusions that while WiMax is a promising technology (in terms of QoS, bandwidth and costs) there are still open issues about the handover management.

2.3.4 WiFi Connectivity

We start this section highlighting the work made by [Hempel *et al.* 2006] and [Zhou *et al.* 2005] with the *Federal Railroad Administration* (FRA) office, in the US. In these works, authors presented results on a real evaluation and testing of the applicability of WiFi to provide connectivity to trains. These works have been followed by the FRA internal research report made by [Tse 2007]. Their conclusions pointed out the 802.11b wireless technology was able to establish a link in a train scenario up-to 144 km/h (90 mph). The only drawback they noticed was with respect to the handover of the train, which pulled the overall throughput of the system down due to the interruptions of the link ("break-before-make" handover of the 802.11b as described by [Ramachandran *et al.* 2006]). Then, analysis on the bit error rate (BER) and others studies presented by [Mahasukhon *et al.* 2007] and [Zhou *et al.* 2009] confirmed these findings, but identifying the same handover issues. [Ott & Kutscher 2004] have made similar measurements, but in the context of the *Drive-thru Internet* project. They proposed an architecture to provide Internet access to mobile users in vehicles along the road - within a city, on a highway, or even on high-speed freeways. The main outcome of this work is the identification of the *entry/exit phases* where the link quality is uncertain, and the *production phase*, where the link quality is good and suitable for a good connection. In parallel, [Gass *et al.* 2006] presented results on measurements of WiFi connections between a in-motion vehicle and an access point located on the side of the road. They concluded that the performance problems on WiFi in high-speed scenarios were related mostly to applications issues rather than link issues. In addition, the authors recommended the use of multiple radios, as suggested by [Brik & Mishra 2005], to exploit the production phases in better way.

The common issue identified by all these works was related to the handover scheme. Results showed that it is not suitable for a train mobility scenario when covered by the 802.11b hot-spots. Furthermore, assuming that the handover problem could be solved, we did not find any study on how to design an infrastructure network for a "line of access points", either WiFi or WiMax, deployed along the train path. The only author mentioning something about it was [Hempel *et al.* 2006]. He pointed out the potentially high cost of deploying a large network of access points along a railway.

Following, we review some documented experiences in the literature about the use of WiFi to provide train communications. We found cases on surface and underground connectivity. In addition we found several companies (for example [WifiRail , MeshDynamics]) claiming they are providing train connectivity based on 802.11 technologies. However, no further details are given.

The first results we reviewed come from the FRA testbed, which consist of a 3.5-miles railway section outdoor Nebraska. It is operated by the Burlington Northern Santa Fe Railway (BNSF), US. The testbed consists in 8 nodes connected to each other sequentially via wireless 802.11a links (the backhaul network) and providing an access point based on 802.11b allowing the train to access the backhaul network. The results presented by [Tse 2007] showed a throughput of 6 Mbit/s, however variations were observed due to handover issues. The average delay observed from the train to the sink point of the backhaul network was 40 ms. All these tests were performed at 40 mph (72 km/h) and 90 mph (144 km/h).

The second results we reviewed are in the context of underground tunnels connectivity. [Zhu *et al.* 2010] discussed a model to evaluate the performance of a 802.11b wireless system in a underground Coal mine scenario. However, their results are not conclusive to decide whether such a the system is able to work properly or not. [Kowal *et al.* 2010] present experimental measurements on throughput, delay and coverage range of 802.11b/g(0.9/2.4 GHz) and WiMax (1.5/3.5 GHz) on an underground mine. Straight tunnels and grid tunnels scenarios were evaluated. In straight tunnels the connectivity was very good, achieving transmission rates up-to 22 Mbit/s, while in grid tunnels the results were variable. Nevertheless, both experimental results yield the same conclusions: WiMax at 1.5 GHz outperforms better than the 802.11b/g at 2.4 GHz. But, the 802.11b/g at 0.9 GHz exhibits results close to WiMax. Another interesting point was the 802.11b/g at 0.9 GHz reported a significant lower delay than WiMax at 1.5 GHz. Indeed, WiMax systems have "significantly" higher delays than WiFi system (35 ms against 25 ms), which relates to their operating principle and transmission protocol. In summary, authors conclude that WiMax is the best way to provide connectivity, however, 802.11b/g at 0.9 GHz and at 2.4 GHz are capable of operating at similar performance level. In a different context, but always on the underground connectivity, [Kassab *et al.* 2010] evaluated the performance of an 802.11a network for the communication infrastructure-to-train in an underground tunnel. They presented an evaluation by means of simulation. The signal attenuation on the tunnel was considered, and their results showed that it is possible to achieve a good transmission in such an environment. However, handover issues were reported, as all the other works discussed previously.

2.4 Summary and Discussion

In this chapter we reviewed some of the current developments of "Internet on-board" services deployed by train operators. From all the solutions we reviewed, we summarize the solutions adopted by each deployment in Table 2.4.

Service	Type	Technology used	Source	Implementer	Performance
TGV	1	Sat + WiFi + HSPA	Research	Orange Labs	2 Mbps/512 Kbps 600-800 ms
Thalys	1	Sat + HSPA/UMTS	Research	21Net	2 Mbps/512 Kbps ?
GNER	1	Sat + 4xHSPA	Research & Industry	Icomera	4Mbps/1Mbps 600 / 30-70 ms
Amtrak	2	8 x HSPA	Industry	Nomad Digital	7 Mbps/3 Mbps 30-70 ms
Virgin	2	WiMax + ADSL uplink	Industry	Nomad Digital	2Mbps 30-70 ms
ICE	2	UMTS + FLASH-OFDM	Research & Industry	T-Mobile	1Mps/512Kbps 20 ms
MAGLEV	3	? + UTMS (backup)	Research	Telefunken RACOMS	38Mbps ?

Type : 1 - Satellite + gap-filler. 2 - 3/3.5G (Bonded). 3 - Radio trackside communication.
? : No information given.
*N*xHSPA : n bonded HSPA modems.
Performance : Throughput and latencies according information provided by operators.

Table 2.1: Summary of train operators providing Internet on-board.

All solutions use a single *access terminal* to share a "connection" in order to reach external networks such as Internet. Solution 1 and 2 use the idea of heterogeneous connectivity (blend of technologies). But, we make the distinction that solution 2 uses a blend of "operators" with (presumably) different technologies. On contrary, solution 3 uses the concept of a radio trackside communication with an infrastructure network deployed along the track. The key point is that the infrastructure network is owned by the train operator, allowing more flexibility to provide a better quality of service. Results seem to be very satisfactory when noticing that the MAGLEV deployment in China is indeed able to communicate an access terminal at 500 km/h.

With respect to each type of solution, we draw the following conclusions:

1. The first solution came from research projects (FIFTH, TRAINCOM and previous projects). It provides up-to a 1 Mbit/s aggregated traffic when using the satellite link. Gap-fillers are used only when the satellite is not available (blocked). While gap-filler links might be faster than the satellite (assuming the up-link is not a DSL link as the case of Virgin), it is used on limited areas. So, on average, the service is characterized by the satellite link. Therefore, the link exhibits large delays ($600 - 1200$ ms) and is subject to weather conditions. This quality of service is not that attractive when user are charged for using an Internet on-board.

2. The second solution comes from the industry. It is better than the satellite solution since it provides better availability. When one operator fails or has poor coverage, the *access terminal* can use another operator to keep the service up and running at the same level. However, they are still bounded to 3G network capacities (1 Mbit/s),

especially when only one operator is available. Hence, in worst case this solution is similar to solution 1 with respect to throughput (at TCP level), but 2 exhibits a better latency than 1.

3. The third solution seems to be the best one. It also came from research (WIGWAM). But, while the FAMOUS architecture still remains in the research world, Emmelmann's work has found its way to the industry. It worth to emphasizing this solution assumes that the train operator owns the infrastructure network and, which means that it controls the radio coverage, traffic handling policies, QoS provision, etc. Hence, the service level depends only on the train operator and not on 3d party operators.

From the solution 3, we have evidenced that WiMax and WiFi can be used as a trackside communication technology. These technologies work as well in surface as in underground train scenarios. So, they can be used to propose an open feasible solution based only on terrestrial networks. However, there are still issues to be solved, especially about the handover mechanism and the provision of a suitable infrastructure network to support a continuous quality communication on trains, which we address in the remaining of this thesis.

Chapter 3

Assumptions

In this chapter we state the assumptions made when defining the scope of this thesis. We discuss the required characteristics of the wireless communication in order to provide the conditions to perform a handover operation at a given speed, the traffic profiles we expect from the in-motion network, the characteristics of a routing protocol used by nodes and their probability to fail.

3.1 Introduction

In the context of the trackside wireless communication of a train, there are several variables we need to fix in order to bound the scope of this thesis. The speed of the train and the overlapping distance between the coverage areas of contiguous Access Point (AP) are important variables to be characterized and given a range of reasonable values to provide an operational context. Also the taxonomy of the traffic is related to this operational context. The communication between the in-motion network and external networks defines a different context of the communication between two in-motion networks. These differences relies on the routing scheme to be used by devices (nodes) forming the infrastructure network. The failure of these devices will cause a reconfiguration of routes in order to keep the network connected (avoiding as much as possible network partitions). Therefore, we also characterize these three variables stating our assumptions to further define the operational context considered for this thesis.

Our main assumptions stated hereafter can be classified in five areas:

1. **Speed of the train**: WiFi is able to operate at high-speed in a railway environment.
2. **Wireless coverage**: Each Access Point has a coverage of 230 meters wide and their location provides the minimal overlapping distance needed to perform a handover operation.
3. **Traffic profiles**: The traffic flows are only between the in-motion network and external networks (through the *Network Gateway*).
4. **Routing scheme**: The routing scheme operates only at OSI layer 2 and it is capable to react in case of failure, such that it always provides the shortest path to the root of the network (*Network Gateway*).
5. **Failure Probability of a node**: Access nodes have a higher probability to fail than backbone nodes.

In the following, we discuss more in detail the rationale behind each one of these assumptions.

3.2 Speed of the Train and Wireless Coverage Area

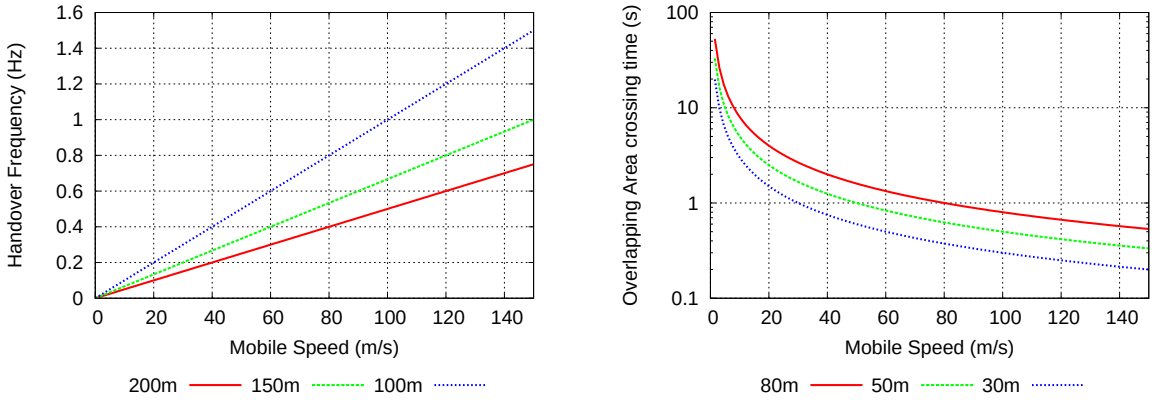
Assumption: 802.11b radio communications (WiFi) are able to operate at high-speed in a railway environment.

Rationale: reviewing the literature about the performance of 802.11 devices on high-speed scenarios, we found empirical evidence supporting the fact that 802.11 devices are able to establish a radio link when one of the devices is in movement at high-speed. In [Ott & Kutscher 2004], the authors reports about their experiment on a WiFi network that demonstrate its ability to operate up-to 180 km/h. The main outcome from this experiment is the identification of three phases of connectivity when crossing the coverage area of an access point at high-speed, namely the entry phase, the production phase and the exit phase. The production phase allows for a high sending rate that is close to the maximum throughput obtained in laboratory under controlled conditions. In the entry and exit phases, the throughput is decreasing due to a higher number of lost packets, link-layer retransmissions, and lower 802.11b sending rates. There is also empirical evidence showing that 802.11b works on a railway environment [Zhou *et al.* 2005, Hempel *et al.* 2006, Mahasukhon *et al.* 2007, Tse 2007] at relatively high-speeds (120 km/h). In addition, we found the studies presented by [Kowal *et al.* 2010] and [Kassab *et al.* 2010] on the 802.11 radio transmissions in underground tunnels. Hence, while wireless transmissions in tunnels (Subway/Metro scenario) requires more field experimentation, we assume that an 802.11 radio link is fully functional under the mobility conditions imposed by the train scenario studied in this thesis. Also we assume the wireless connection at high-speed exhibits the three connectivity phases identified by [Ott & Kutscher 2004]. Therefore, by honoring the fact that only the 802.11b radio link has been studied in the presented literature, we assume a 802.11b WiFi up-link between the in-motion network and the infrastructure network.

Assumption: the coverage area of an Access Point is 230 meters wide and they are placed at regular intervals of 150 meters, which gives a overlapping distance of approximately 80 meters. Therefore, when considering a speed of 350 km/h, the in-motion network will perform a handover each 1.54 seconds. This time is the upper bound for the handover time in order to left time to the production phase identified by [Ott & Kutscher 2004], and by consequent, to allow the exchange of traffic between the in-motion and external networks.

Rationale: as pointed out by [Emmelmann 2005], the handover operation is affected by the speed of the wireless stations (the access terminal on-board the train in our case). Indeed, the handover frequency does not scale well with the overlapping distance between

the radio coverages of the AP participating in the handover operation. The handover frequency is defined as the number of handover operations performed by unit of time. Thereby, when assuming a coverage area of 230 meters wide, and APs are located at 150 meters each other, the overlapping distance is about 80 meters. If the maximum in-motion network speed is 350 km/h, and the handover operation begins at the edge of the coverage of each AP, the in-motion network should perform a handover operation each 1.54 seconds, or 0.64 handovers per second! Figure 3.1 shows how the handover frequency changes with the speed for three different distances between APs.



(a) Handover frequency for fixed distances.

(b) Overlapping area crossing time.

Figure 3.1: Effects of the handover frequency on the overlapped coverage distance.

The overlapping distance plays an important role in two ways: first, for a fixed coverage area, the larger the overlapping distance, the longer the handover. When fixing the handover time, the larger the overlapping distance, the faster the mobile can be and the closer are the APs between each other. However, closer APs implies to require more access nodes to cover the same trajectory. Second, for the same fixed coverage area, the smaller the overlapping distance, the farthest the APs, the slower the mobile can be for a fixed handover time. Nevertheless, fewer APs are required to cover the same trajectory. Each case presents its advantages and disadvantages. Thus, for the train scenario, the overlapping distance is bounded by the maximum speed of the train, from where we compute the handover frequency at maximum speed, and from there, we deduce the distance between access nodes.

3.3 Traffic Profiles

Assumption: the traffic exchange is only between the in-motion network and external networks. Any external network is reachable through the router of the infrastructure network, which is considered as the root of the network for routing proposes.

Rationale: according to our motivation, we aim at proposing a solution to provide network connectivity to passengers on-board a train, tram or metro. However, it is true that such a solution might be applicable to other kinds of applications, such as surveillance, in-route advertising or monitoring (just to mention a few). The common point of all these applications is all of them requires to exchange traffic with **external networks**. Therefore, all the traffic from the access network needs to be routed to the *Network Gateway* of the infrastructure network, from which external networks are reachable. In the other way around, the traffic from external networks needs to be routed to the access network, more precisely, to the access point by which the in-motion network is reachable.

There are many traffic profiles that could exists between the in-motion network and external networks. Bursty traffic profiles, Constant Bit Rate (CBR) traffic profiles, Web traffic profiles, etc. Some of them might require a better QoS than others, depending on the requirements of each application. However, recall the previous assumtion that the wireless link between the in-motion and the infrastructure network is an IEEE 802.11b running at 11 Mbit/s. This introduces a bottleneck on the path to follow by packets towards the *Network Gateway*. Therefore, we consider in this thesis a CBR traffic in such a way to produce a moderate condition of saturation (in average 1 packet in the transmission queue) on the wireless link between the in-motion and infrastructure network. Thus, all the evaluations in respect to the traffic will be bounded by the performance of this wireless link.

In summary, we assume only traffic between the in-motion network and the *Network Gateway* in this thesis, despite the fact that other traffic flows might exist between hosts within the infrastructure network.

3.4 Routing Scheme

Assumption: the routing scheme within the infrastructure network operates only at OSI layer 2.

Rationale: the handover and design problems are related through the routing scheme used within the infrastructure network. The handover scheme requires to trigger a routes updating procedure to keep up-to-date the routing information in the infrastructure network. The handover operation could be at layer 2 (network layer) or layer 3 (IP layer), which constrains the level at which this routing scheme should operate. This selection might have profound implications on the final solution. Indeed, the use of an IP routing scheme implies the consideration of a Mobile IP handover scheme, which might increase the handover times out-of-bounds when considering the speed of the train. In addition, the complexity of handling the layer 2 and layer 3 handover is clearly higher than a solution based only on layer 2. On the contrary, the scalability in layer 3 is by far better than in layer 2.

There are several IP routing protocols for backbone networks (OSPF, IBGP, RIP, etc) and several for Ad-Hoc networks (OSLR, AODV, DYMO, etc). When considering an IP

routing scheme to route packets between the in-motion and the infrastructure networks, we require to solve the layer 2 handover problem anyway in order to allow the routes updating procedure to send updating packets through the new up-link association, since the handover in layer 3 requires to complete first the handover in layer 2 [Malekian 2008]. Therefore, we circumscribe our problem to a routing scheme only at layer 2. This choice has several benefits and drawbacks. As benefits, we highlight the use of virtual networks [IEEE 802.1Q 2006] (VLANs) to provide a wider range of options when providing QoS as described by [De Sousa 2006]. As drawback we point out the size of the infrastructure network. When considering a layer 2 routed network (more formally, a bridged network), the scalability issues are not evident. However, there are some ways to extend a bridged network from a local area network size (tens to hundreds of nodes) to a metropolitan area network size (hundreds or thousands of nodes) [Chiruvolu *et al.* 2004].

Assumption: The routing scheme always finds the shortest path to the root of the network when it exists, and it is capable to converge to another shortest path in case of a node or link fails.

Rationale: As we are interested in the traffic exchange between the in-motion network and external networks, we choose the shortest path to the *Network Gateway* as the more effective path when achieving a minimal delay through the network. In case of a failure, this path must be reconfigured in order to keep the shortest path property. We review the literature about Metropolitan Area Networks (MAN) since the characteristics of MAN are similar to our infrastructure network characteristics (in the size of the network and they are bridged networks). There are several layer 2 routing algorithms applicable to MAN, most of them based on the classical Spanning Tree Protocol (STP) [IEEE 802.1D 2004]. [Huynh & Mohapatra 2007, Huynh *et al.* 2009, Faghani & Mirjalily 2009] claim that the Spanning Tree Protocol and its predecessor, the Rapid Spanning Tree Protocol (RSTP) [IEEE 802.1D 2004] does not perform well in large scale networks [Abuguba & Moldován 2006, Myers *et al.* 2004]. It might suffer from congestion near the root, slow resilience, lack of load balancing, and lack of QoS/OAM support [Huynh & Mohapatra 2007]. However, there are other studies claiming the opposite [Huynh *et al.* 2009, Faghani & Mirjalily 2009, Mirjalily *et al.* 2009]. Most of the drawbacks of RSTP can be solved by tuning the parameters of the algorithm, or by using "regions of spans" [Padmaraj *et al.* 2005a]. But the reconfiguration time due to failures is still a cause of debate. There are many different statements about the recovery time of RSTP from the order of seconds down to the order of ten milliseconds. [Pallos *et al.* 2007] present an evaluation based on experimental measurements showing that the core of RSTP is fast; network recovery can be achieved in the order of ten milliseconds if solely RSTP could detect faster the link-down events. Thus, additional hardware delays might slow its operation about 30 seconds [Myers *et al.* 2004]. In our scenario, this reconfiguration time makes not possible to transport real critical traffic such as signaling, but certainly it is acceptable for end-user traffic or not critical applications, such trains passengers traf-

fic. Therefore, we consider in this thesis a Spanning Tree routing scheme as our default layer 2 routing. In particular, we use for our evaluations the Rapid Spanning Tree Protocol [IEEE 802.1D 2004] due to it presents several improvements when compared with the classical STP (introduced by the ANSI/IEEE 802.1D-2004 standard). Also we assume that there are ways of scaling the network size to thousands of nodes, as suggested by [Ishizu *et al.* 2004], and that it is capable of providing QoS on Ethernet bridged networks, as pointed out by [Padmaraj *et al.* 2005b, Padmaraj *et al.* 2005a].

3.5 Failure Probability of a Node

Assumption: Access nodes have a higher probability to fail than backbone nodes within the same observation period of time.

Rationale: As described in Chapter 1, the infrastructure network is composed of an access network and a backbone network. Access nodes are WiFi access points such as a Linksys WRT54G or any other access point device which allows the modification of its firmware. Backbone nodes are multi-port bridges, or better known as layer 2 switches. The number of backbone nodes should be less than the number of access nodes, otherwise the backbone network would be connected one-to-one with the access network, which is not cost effective. Therefore, we have a large number of access points in the access network and a limited number of switches in the backbone network. As we have a limited number of backbone nodes, it is possible to use better quality devices (redundancy in the power supply, spare ports, etc). On the contrary for access nodes, they should not be expensive (and therefore, less quality devices) when assuming a large number access points covering the path of a train. Hence, it is clear that a better quality switch (such as the Cisco ME 3400 Series Ethernet Access Switches) has a lower probability to fail than a regular access point device, both compared within the same observation window of time τ. In consequence, considering the train scenario, we assume that access points are more likely to fail than layer 2 switches.

Part I

Handover of an In-motion Network along a Predefined Trajectory

Chapter 4

Horizontal Handover of an In-motion Network

In this chapter we propose a mechanism for the horizontal handover within an 802.11 wireless network. Our proposal focuses on the handover of a complete in-motion network, not individual wireless stations travelling at high-speed. We propose a mechanism based on two new 802.11 operational modes: the **Spiderman** mode and the **Wireless Switch Access Point** mode.

4.1 Introduction

In this chapter we aim at proposing a solution to the horizontal handover problem identified in Section 1.2. We consider the handover of an in-motion network instead of the handover of a single end-user station. This in-motion network is composed by a set of hosts (stations) interconnected by a network concentrator such as a switch or an access point. Before presenting our proposed solution for the handover of an in-motion network, we recall briefly the standard handover operation in 802.11 networks.
The handover operation is triggered when a wireless station leaves the coverage area of an Access Point (AP) to enter the coverage area of another one. It describes the sequence of tasks that wireless stations (STAs) need to perform in order to transfer the data-link layer (layer 2) from one AP to another one. The objective of this operation is to provide a continuous connection along their trajectory. For that, APs must cover this trajectory with radio cells (hot-spots) with a minimum overlapping distance to allow STAs to perform the handover operation without interrupting (as much as possible) the connection. The tasks involved in the handover are described by [Wiethoelter & Emmelmann 2010] and we summarize them as follows :

1. *Handover decision.* The STA monitors the current link quality in order to decide when to start the handover operation. This monitoring is mostly based on the received signal strength indicator (RSSI) and periodicity of received beacons.

2. *Neighborhood discovery.* The STA "breaks" the link with the current AP, begins to buffer all the incoming users' dataframes (from upper layers) and starts scanning the neighborhood for the best candidate AP to re-establish the link.

3. *Authentication-Authorization-Accounting (AAA).* The STA has chosen the best AP based on the information provided by the discovery phase. It begins the authentication process, which is normally a two-way or four-way handshake, depending on the security mechanism. If the authentication response is positive, the STA has the authorization to be associated with that AP. The association is a two-way handshake. When the STA receives a positive association response, the AP begins the accountability of this association.

4. *Data-link re-establishment.* When the association is done, the STA re-establish the data-link layer with the new AP. All the buffered dataframes are sent through the new AP and the connection is resumed.

5. *Mobility Management.* The STA performs the required tasks to update the routing information in the infrastructure network. When APs are used as bridges to access an infrastructure network, most of the standard 802.11 devices broadcast a *Gratuitous ARP* through the new association informing the infrastructure network about its new point of attachment. Other mechanism may include tasks related to the mobility management of higher layers, such as Mobile IP [Perkins 2002, Johnson *et al.* 2004].

In the discovery phase, the 802.11 standard defines two ways to scan for the next AP: *Active Scanning* and *Passive Scanning*. The active scanning is based on a probe-response method. First, the NIC broadcasts a "probe" frame. It waits until *MinChannelTime* for a possible "response". If the NIC detects a transmission on the channel, it assumes that it might be an AP, so, it waits until *MaxChannelTime* for a response. When no transmission is detected and no response has arrived before the *MinChannelTime*, the channel is considered free with no APs serving on it. The passive scanning is based on listening a channel in order to capture opportunistically the beacons from the neighboring APs. As beacons are transmitted at regular intervals (normally 100 ms), the wireless NIC needs to listen at least a beacon interval time to ensure the reception of a beacon. This method is much slower and less efficient than the *Active Scanning*, which is the most used one. When any of both procedures is performed on each available channel, the scan operation receives the name of *full scanning*. The AAA phase is defined in the 802.11 standard and depends on the security scheme chosen by the AP (WEP, WPA, etc.). Association (or re-association) is also defined in the standard.
However, many issues are left open to vendors, such as the way to implement the scanning operation, when to decide to perform the handover, how to buffer the dataframes, and how to manage the mobility. This freedom of implementation have lead to significant differences between brands of wireless NIC, making even harder to implement a "seamless" handover operation. In this direction, the standardization bodies have promoted the 802.11f (Inter-Access Point Protocol) and 802.11k (measurements on additional beacons frames) standards towards the improvement of the handover operation. Nevertheless, they were not well adopted by users.
While the 802.11 handover problem for a single STA is still a challenge according to the literature (see section 2.3.4), we address the problem from a different point of view. We

consider several hosts describing together the same mobility along a predefined trajectory. These hosts are accessing the infrastructure network through a single *access terminal*, which manages the handover operation for all of them.
In the following, we discuss the problem we aim at solving in the context of an in-motion network on-board a vehicle (in our case a train). Then, we declare the assumptions on this context and we review the related works. In Section 4.5, we present our proposal. In Section 4.5.2 we describe its operation. In Section 4.6, we discuss the key points to answer the question this chapter aims at solving. In Section 4.7 we summarize these key points providing an answer to whether or not our proposed mechanism is able to handover an in-motion network at high-speed while avoiding buffer overflows.

4.2 Problem Definition

The handover problem is related to the packet losses experienced by the STA when buffering the incoming user's dataframes while the handover operation is in place. Indeed, when the handover operation is not finished before the buffer reaches its capacity, all the incoming packets are discarded, yielding a buffer overflow.
This problem is originated by the nature of the *Distributed Coordination Function* (DCF) used by the MAC protocol of 802.11 based networks. The CSMA/CA algorithm does not allow to know in advance when a STA will transmit a packet, therefore, the handover operation is performed mostly by the STA rather than the AP. The 802.11f and 802.11k were both efforts to tackle this problem, but without much success. The TDMA based MAC protocols have the advantage that STAs and APs know in advance when to perform the handover, reducing considerably the packet losses due to buffer overflows. Nevertheless, the slots' allocation technique of TDMA based networks reduces the system capacity, which makes it less attractive when targeting small coverage areas with only a couple of tens STAs associated to the same AP.
This problem is not yet solved, according to the literature. [Emmelmann *et al.* 2007] has pointed out the lack of the adoption of standardization efforts makes it difficult to cope with this problem. While several solutions have been proposed, they mostly address the problem by trying to reduce the handover time from the STA point of view. In section 4.4 we review these efforts emphasising on the handover decision and the scanning phase, which are the most important phases when reducing the handover time. We do not consider the delay of the authentication phase as an important contributor to the handover time, since we are not studying the delays introduced by complexity of security schemes. On the contrary, we consider a four-way handshake when performing the authentication-authorization to include the delays caused by possible packet collisions during this phase.
Considering the handover of a set of STAs following the same trajectory (from now the in-motion network) the first question that arises is how to transform the handover operation of all the STAs into the handover of a single STA. The answer relies on the use of a single *access terminal*, as proposed by [Rodriguez *et al.* 2004]. This *access terminal* aggregates the in-motion network traffic into a single wireless link. Thus, techniques such as *Natting*

or *IP Masquerading* are possible, which hide the in-motion network addressing behind a single STA address. When this single *access terminal* performs the handover operation (at layer 2) according to the procedure explained in the previous section, the buffer overflow is imminent since it must buffer the aggregated traffic of all the in-motion networks. So, the time required by the standard approach is not sufficient to avoid packet losses. Furthermore, other approaches that try to minimize the handover time do not fully resolve the problem since the level of aggregation of traffic might be large enough to end-up considering 50 ms an excessive time. Thus, when considering that packet losses are initially caused by the first four phases of the handover operation, all the efforts on further reducing them at the last phase become less important since the users' connections are already damaged. Therefore, we focus our attention on reducing, or even eliminating, the buffer overflows in the first four phases of the handover. Furthermore, we discuss the implications of the last phase on the possible buffer overflows in order to have an insight of the next chapter.

4.3 Assumptions

According to the assumptions discussed in Chapter 3, we assume an 802.11b wireless link between the vehicle (the in-motion network) and the trackside infrastructure network (the APs). We also assume that this wireless link is operational in train scenarios and the coverage of each AP is 230 meters wide, overlapped in 80 meters each other. The trackside infrastructure network is a bridged network operating at the level of the OSI Layer 2, and the traffic exchange is between the in-motion network and the *Network Gateway* (see Figure 1.1 on page 2).

4.4 Related Works

In Section 4.1 we described briefly the handover operation for an 802.11 wireless network. In this section we review the efforts to improve the layer 2 handover operation for a single STA. While our approach considers an in-motion network, it can be transformed into a single handover operation by means of the use of a single *access terminal*, but with higher probability of buffer overflows. In Section 4.6 we discuss the implications of this assumption when considering the procedure used to update the routing information of an in-motion network.

An empirical analysis of the factors involved in the handover operation is presented by [Mishra *et al.* 2003], pointing out two important facts: 1) the scanning operation is the most time-consuming operation, and 2) the scanning delays vary from vendor-to-vendor wireless NICs. Later on, [Murray *et al.* 2007] argued that the scanning delays are large due to the overlapping nature of the 802.11b radio band (2.4 GHz). It is well known that 802.11b cards are able to decode packets from the neighboring overlapped channels (co-channel interference). Therefore, when the NIC is probing a channel, it has a higher probability to find it busy, while indeed the channel might be free. When the NIC realizes

that "there is someone in the channel", it waits until the *MaxChannelTime* instead of the *MinChannelTime* before to probe the next channel. This extra wait time (unnecessary when the channel is free) increases the scanning delay even up-to one order of magnitude (from 60 ms to 600 ms).
The handover time has been studied by means of simulations by [Pries & Heck 2004] and experimentally by [Corvaja *et al.* 2004]. Both authors have confirmed the fact that the scanning phase is the most time consuming part of the handover. Thus, several scanning methods have been proposed in order to reduce this delay. [Ramani & Savage 2005, Chen *et al.* 2008] proposed to exploit the regularity of beacons such that the wireless NIC can "synchronize" at the right time to listen a channel; [Ok *et al.* 2008, Shin *et al.* 2004, Kim *et al.* 2004] use the history of the STA, storing in a cache memory the APs addresses (BSSID) that it has visited. So, it can scan selectively for the most probable next AP. [Emmelmann *et al.* 2009] take advantage from the 802.11 power management to perform an opportunistic passive scanning in such a way that the possible packet losses due to buffering are minimized by spreading them along the time. [Rizvi *et al.* 2009] propose a probabilistic approach. The scanning timers and beacon interval are adjusted according to a probabilistic model of collision avoidance, so, the channel probing is done more efficiently, yielding to a lower overall scanning delay. However, we put emphasis on the scanning with neighbors graphs presented by [Shin *et al.* 2004, Kim *et al.* 2004]. This technique is useful on a train scenario since the train mobility allows to exploit the predictability of the trajectory. Results presented by [Pries & Heck 2004] have compared four scanning schemes (normal/fast, active/passive) against a history based scanning method (neighbors graphs). Results pointed out that when using active scanning (probe-response), the neighbors graphs based methods performs better than all the other evaluated scanning schemes. In summary, several scanning schemes have been proposed. All of them pursuing the objective of reducing the handover time under the premise that a seamless handover does not mean a loss-free handover. Hence, while most of them are able to achieve low scanning delays, the overall handover time might still be large enough to cause a buffer overflow in the wireless NIC of the access terminal.
Other handover approaches have been proposed to deal with the frequency of handovers. We understand as frequency of handover the number of handover operations by unit of time. [Emmelmann *et al.* 2008] propose a predictive TDMA micro/macro cell based handover scheme for 802.11 COTS[1] devices. The novelty of this method is that the radio coverage of a base station is extended by having several "radio" transmitters attached to the same MAC. Thus, the inner-cell handover is handled at MAC level and the inter-cell handover is negotiated among base stations. This way, the handover frequency is reduced by enlarging the cell coverage. Authors tested the concept by implementing their proposal in a demonstrator, achieving handover times around 1 ms for a single STA. [Brik & Mishra 2005, Ramachandran *et al.* 2006] propose to eliminate the handover delays by means of using two wireless NICs on the STA, implementing a "Make-before-Break" scheme. While the first NIC is associated, the second is scanning for the next

[1] Commercial off-the-shelf

AP. Once the second get an association with the next AP, all the incoming traffic is forwarded through the second NIC. Then, the roles of the NICs are swapped and the process continues. Both solutions differs mostly in how they handle the addressing of the STA. [Ramachandran *et al.* 2006] swap the IP address and routing tables of both NICs, and [Brik & Mishra 2005] assume the same MAC/IP address in both NICs. Both proposals did not present discussions on the scanning methods, and routes updating procedure is handled by the AP and not by the STA. [Adya *et al.* 2004] present the design of a new protocol that enables scalable multi-hop wireless networks by means of the integration of multiple wireless NIC. While the approach they use is similar to our approach to integrate both wireless NICs, their objective is to take advantage of all the channels to improve the throughput (parallel links). They do not deal with the data-link re-establishment or any handover issues.

4.5 Proposed Solution

In this section we present our proposal to provide an horizontal handover for an in-motion network at high-speed avoiding packet losses. We name it **Spiderman Handover**, since in short, the access terminal uses two wireless NIC to establish the link with the APs along the trajectory in the same way that the superhero Spiderman does when flying among buildings.
The Spiderman handover can be described as a **dual wireless NIC bridge device with handover capabilities**. The Figure 4.1 helps to explain the concept.

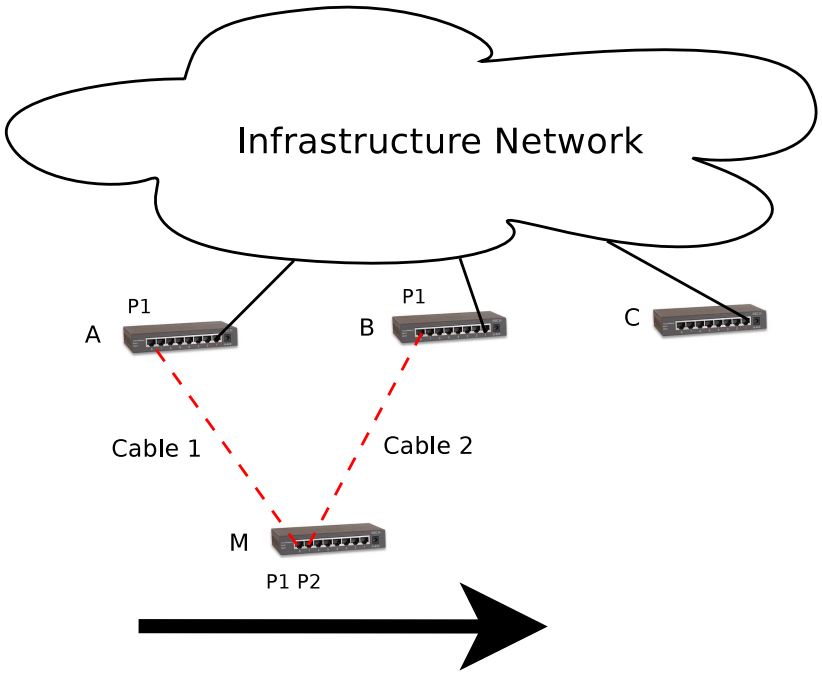


Figure 4.1: Moving switch example.

Let us imagine that the switch M is in motion and we have two cables with a limited length. One cable is used to connect M with the infrastructure network by bridging the port 1 of M with the port 1 of A. In this way, A learns all the MAC addresses behind M through the port 1. When moving M until the cable is extended to its maximum length, we use the second cable to bridge the port 2 of M with the port 1 of B. M is aware of the loop created by adding the second cable, so, it keeps the port 2 inactive, but with a link connected. Before the cable from A to M breaks, M executes on the port 2 a routes updating procedure (explained in the next chapter) to inform B (and all the other bridges) about the new route to reach M (through B). When the route updating packets reaches M through the port 1 of A, the routes updating procedure is complete and the MAC address tables on all bridges between the path between A and B will know M through B. Then, M "unplugs" the cable from A leaving only the bridged link with B active. Thus, this mechanism is repeated between B and C and so on.
To implement the Spiderman handover, we propose two new wireless operational modes: **the Spiderman Mode** and the **Wireless Switch Access Point Mode**, or WSAP mode. The first mode implements a *dual wireless NIC bridge device with handover capabilities* and the second mode can be described as *a layer 2 wireless switch which considers each wireless association as a bridged port.*

4.5.1 System Architecture

In the following, we describe the system architecture to implement both 802.11 operational modes. Note that they must work together to implement the Spiderman handover. While the Spiderman mode is only useful to provide connectivity to an in-motion network (the switch M in the example), the WSAP mode preserves its backward compatibility supporting Spiderman associations as well standard STA associations.

4.5.1.1 Spiderman Mode

The Spiderman mode is defined as a dual wireless NIC *bridge device* with handover capabilities. Figure 4.2 depicts the internal architecture of the **Spiderman device** (SD) which implements the Spiderman handover. This bridge device is placed on-board the train. The word *bridge* means this mode operates only at layer 2. It uses 802.11 4-addresses frames to encapsulate Ethernet packets and exchange them with the associated WSAP. Both wireless NICs have their own wireless stack, controlled by the Spiderman Agent (SA) which implements the layer 2 handover logic. All the primitive operations are implemented in the 802.11 management module (*mgmt*), and the SA commands the *mgmt* module of each NIC to perform the scanning, authentication, association and packet forwarding. The SA defines which NIC is *Active* or *Passive*. Thus, all the packets coming from the *MAC Relay Unit* are forwarded to the active NIC, and all the packets coming from the active or passive NICs are forwarded to the *MAC Relay Unit*. The *MAC Table* keeps the accounting of all MAC addresses known by the *MAC Relay Unit*. Once the passive NIC gets its association, the *routes updating* module uses this *MAC Table* to send routes updating packets through

the passive NIC. These packets should be received through the active NIC in order to finish the handover operation. The *mgmt* module of each wireless NIC should forward the route updating packets with priority in order to ensure a short updating delay. The SA defines the required hooks to allow the the use of different routes updating procedures. In addition, the SA acts as an additional bridged port for the *MAC Relay Unit*, hiding all the internals of how the bridge link is handled. In Figure 4.2, the connections in solid (black) lines show the path followed by packets and the connections in dashed (red) lines show the relationship among components.

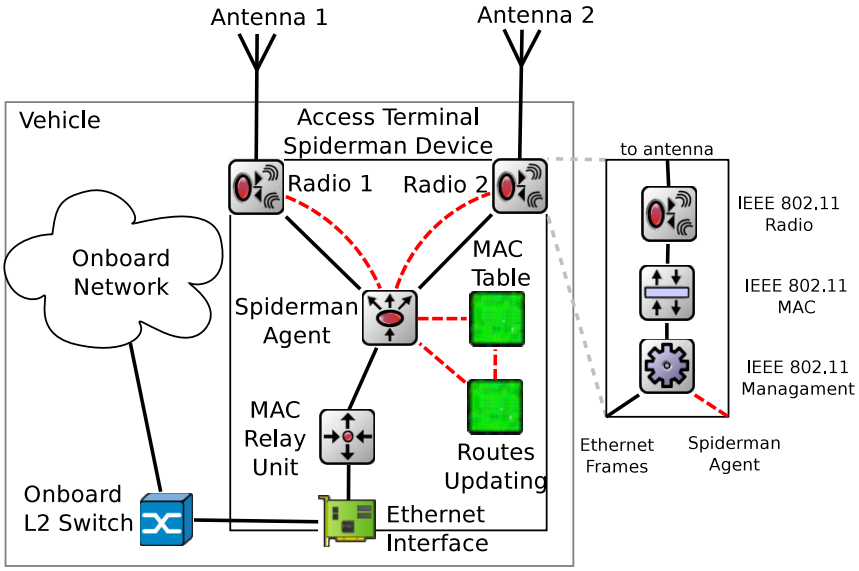


Figure 4.2: Spiderman Device architecture.

The Ethernet device is used to connect the Spiderman device with the in-motion network on-board the vehicle as it was an additional stacked switch. The internals of the in-motion network are "transparent" for the Spiderman device since it handles only Ethernet traffic.

4.5.1.2 Wireless Switch Access Point Mode

The WSAP mode is similar to the *Master* mode of 802.11, which implements the access point functionally. However, the WSAP mode allows the exchange of 802.11 4-addresses frames with the associated Spiderman devices. It keeps a record of the MAC addresses known by each association on a different *MAC Table*. The 802.11 *mgmt* module acts as a secondary *MAC Relay Unit* considering each Spiderman association as a bridged port. So, the *mgmt* is able to forward the traffic to the correct *SD* by matching the association table with its correspondent MAC table. The 802.11 4-addresses format frame allows the encapsulation of the Ethernet traffic between the infrastructure and the in-motion networks in the same way that a Wireless Distribution System (WDS) does. Figure 4.3 shows the internal architecture of a *WSAP device*, depicting in "black" the paths followed by packets and in "red" the interaction between components.

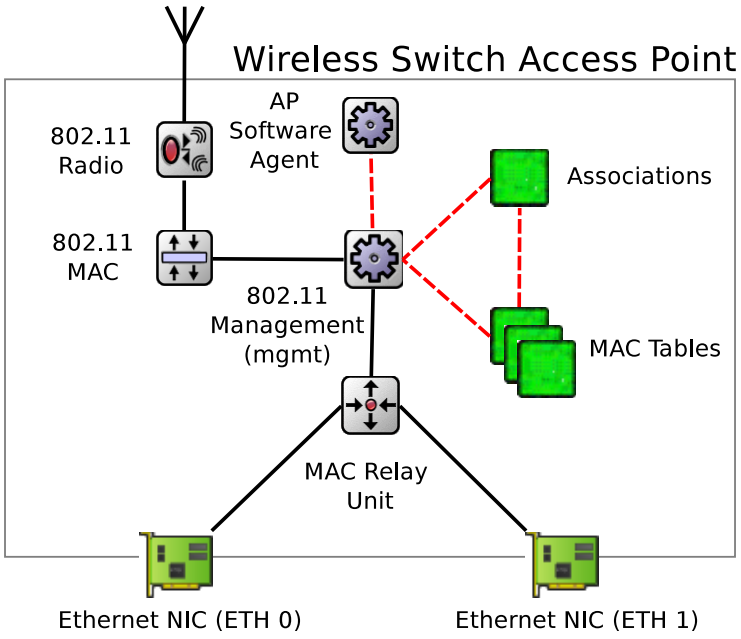


Figure 4.3: Wireless Switch Access Point architecture.

The primary *MAC Relay Unit* makes this device to behave in a way similar to a multi-port bridge device, or better known as a layer 2 switch. Therefore, the WSAP device can have multiple Ethernet NICs and it can implement any layer 2 routing protocol without loss of generality due to the wireless NICs. Within the train communication scenario, the WSAP devices are placed at regular intervals along the track, acting as access nodes to the infrastructure network.

4.5.2 System Operation

In this section we describe how the Spiderman handover works. First, we introduce an example to identify the main aspects of its operation. Then, we discuss the details of the scanning method and how the handover time is measured. As our approach uses a dual wireless bridge device, it requires a different way to measure the handover time.

Figure 4.4 shows the sequence of the basic activities performed by the Spiderman handover mechanism. We assume the *Radio 1* is passive and the *Radio 2* is active. The process begins when the passive radio has lost the connection with its WSAP. At that time the SA commands the passive radio to start the scanning procedure, which we explain in further details later on. The passive radio keeps scanning until it gets the next AP to connect with. Then, the authentication and association is done. When the passive radio gets a positive association, the routes updating procedure begins on the passive radio. This updating procedure (described in the next chapter) uses the *MAC Table* to know which addresses require to be updated. The routes updating packets should be received by the active radio in order to confirm that the path between the two WSAP involved in the

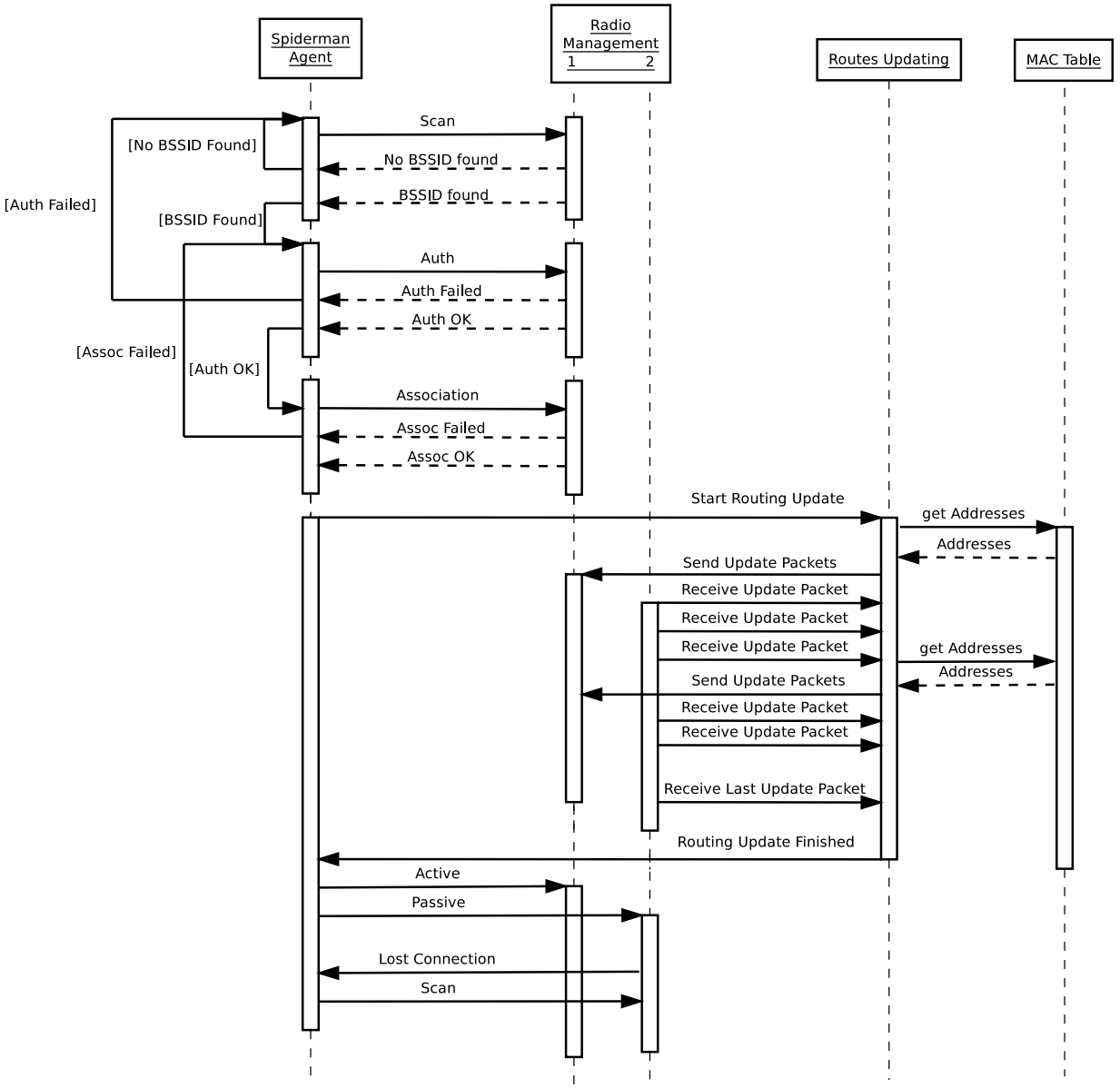


Figure 4.4: Spiderman handover procedure.

handover operation have been fully updated. When the last updating packet has arrived through the active radio, the routes updating module signals the SA that the routing update is complete. The SA notifies the *Radio 1* that now it becomes active and the *Radio 2* becomes passive. Thus, the handover operation is complete. Furthermore, the passive radio holds the link with its WSAP until exiting from its coverage area in order to avoid to lose any misrouted packet. Then, the SA commands the *Radio 2* to start the neighborhood discovery phase, restarting the procedure.

The Spiderman handover time depends mostly on the time required to update the routing information in the infrastructure network. The scanning, authentication and association

times are hidden from the handover operation because they are performed without breaking the active link. Strictly speaking, the update procedure could be also considered as hidden from the active link. However, we did not consider it as hidden since it is in this phase when the real handover takes place. Considering the objective of the handover operation, we observe that it is not sufficient to re-establish the data-link layer with the next WSAP to consider all the users' connections on that link as "transferred". We need to ensure that the traffic from/to the SD has a valid route in the infrastructure network to reach its destination. So, without this update, we can not consider the connection as "transferred". Nevertheless, there is no need to wait until the end of the mobility management phase to start sending traffic through the new association. Therefore, the SA begins to forward the incoming traffic (from the *MAC Relay Unit* through the passive NIC) as soon as it arrives. Thus, the same traffic acts as route updating packets at least for ongoing connections. This fact introduces a risk of extra delays on the route updating packets due to queueing effects. Therefore, the passive wireless NIC transmits the updating packets with top priority to its WSAP.
When the active link loses its connection before the routing update is complete, the passive link becomes automatically active and the former active link start the scanning process. Thus, the route updating procedure is finished "earlier" than it should.

4.5.3 Scanning Method

As mentioned before, all primitive operations of the *Managed* mode of the 802.11 standard are implemented in the *mgmt* module of each wireless NIC. We use these primitives to command the NIC to perform the operations required to perform the neighborhood discovery phase. The Algorithm 1 shows the pseudo-code used to find the next WSAP.
The function *nextChannel()* returns the next channel on the given sequence in relation to the current channel *ch*. The function *previousChannel()* is analogue to the *nextChannel()* function. Both functions operate in a round-robin way. In other words, the next channel of the last channel in a sequence is again the first one. The function *scanChannel()* commands the 802.11 *mgmt* module to perform an active scanning on the given channel. The function *scanAllChannels()* is self explicative. The function *selectBssid()* returns the best BSSID from the list of BSSID found on a channel. The selection criteria is based on the best received signal strength indicator (RSSI).
Basically, the scanning method exploits the predictability of the vehicle's trajectory by means of a predefined sequence of channels configured on the trackside WSAP access network. In other words, the WSAPs should follow a predefined sequence of channels in order to maximize the probability of finding the next WSAP in a very short time. Thus, the scanning method will scan preferably the channels in the sequence according to the current active channel. After certain number of attempts (6 in this case), the scanning method falls back into a full scanning mode to allow the SD to overcome WSAPs failures or misconfiguration, which might break the sequence. After the full scanning, it begins again the selective scanning. This loop is repeated until the next *BSSID* is found.

```
Input: current channel ch
Output: next AP bssid
1  begin
2      channelSequence ← { channel 1, channel 6, channel 11 }
3      count ← 0
4      maxCount ← 6
5      ch2scan ← 0
6      ssid ← "name of the wireless LAN"
7      while bssid not found do
8          if count ≤ maxCount then
9              if ch2scan = nextChannel (ch,channelSequence) then
10                 ch2scan ← previousChannel (ch,channelSequence)
11             end
12             else
13                 ch2scan ← nextChannel (ch,channelSequence)
14             end
15             bssidList ← scanChannel (ch2scan)
16             count ++
17         end
18         else
19             bssidList ← scanAllChannels
20             count ← 0
21         end
22         bssid ← selectBssid (bssidList,ssid)
23     end
24     return bssid
25 end
```

Algorithm 1: Spiderman scanning method.

Notice the sequence of channels corresponds to the orthogonal channels of the 2.4 GHz band. We choose these channels to avoid the scanning primitive to wait until *MaxChannelTime* due to the co-channel interference. Thus, the single channel scan is mostly performed in a *MinChannelTime*, which increases the number of times that a channel is probed, and by consequence, the probability to find the next AP is much faster.

This procedure is followed by the passive NIC when searching for the next WSAP. However, when both NICs are disconnected, the SA begins a cooperative scanning (with both NICs) of all the channels available. This procedure is made asynchronously by commanding each NIC to scan a single channel in parallel. Then, as soon as the NIC reports to the SA the list of BSSIDs found on that channel, the SA commands it to scan the next one. Thus, the list of available channels is covered using half of the time that a single NIC should take. This procedure is also used when the active wireless NIC loses its link before the

passive wireless NIC gets the next one. This procedure also speeds-up the link recovery when there are several consecutive failed WSAPs.

4.5.4 Handover Time

The use of a dual wireless NIC bridge device allows to virtually **hide** the delays caused by the detection, discovery, AAA and link re-establishment phases. However, the mobility management phase (the routes updating) might affect the traffic departing from the passive wireless NIC and arriving to the active wireless NIC. As this handover mechanism keeps both routes available while the handover is in place, the packet losses due to misrouting are minimized. However, the overhead produced by the routes updating procedure should be spread along the time that the SD remains with both routes active. Otherwise the probability of having a buffer overflow might get increased.
As the delays considered as part of the handover time are hidden, we can not account them as the "real handover time". However, the mobility management delay can be accounted as the real handover time since within this phase the data-link layer is really transferred from one AP to the other. Thus, we consider the time taken by the route updating procedure as the real handover time of our mechanism.

4.6 Discussion

In this section we discuss the impact of the scanning method on the handover time in terms of the speed of the vehicle. We aim at describing the bounds of the handover time in order to provide a notion about the time that the routes updating procedure has to perform its task.
As we consider that the handover time equal to the time that the SD remains under the coverage of both WSAP, we need to maximize this time in order to avoid packet losses due to an earlier termination of the routes updating procedure. As the passive NIC begins the neighborhood discovery as soon as it breaks its connection with the leaving WSAP, the probability to begin the handover as soon as the SD enters in the new WSAP's coverage is high. Indeed, the scanning method is probing the next possible channel very often (in the sequence); and as they are orthogonal, this operation is made mostly within the *MinChannelTime*. The only factor that might produce a late handover decision is the full scanning performed by our scanning method. There is a possibility that the passive wireless NIC begins a full scanning just before to ingress into the next WSAP coverage. Under this assumption, two situations may happen: 1) the SD scans the next WSAP channel before to ingress into its coverage, therefore, it has to finish the full scanning in order to begin again with the selective scanning and finds the next WSAP; 2) the next WSAP is in the last channel of the scanning sequence, therefore, the SD will find it after all the previous channels have being scanned. The situation 2 introduces less delay than the situation 1. However, both delays might be large enough to produce an early termination of the route updating procedure.

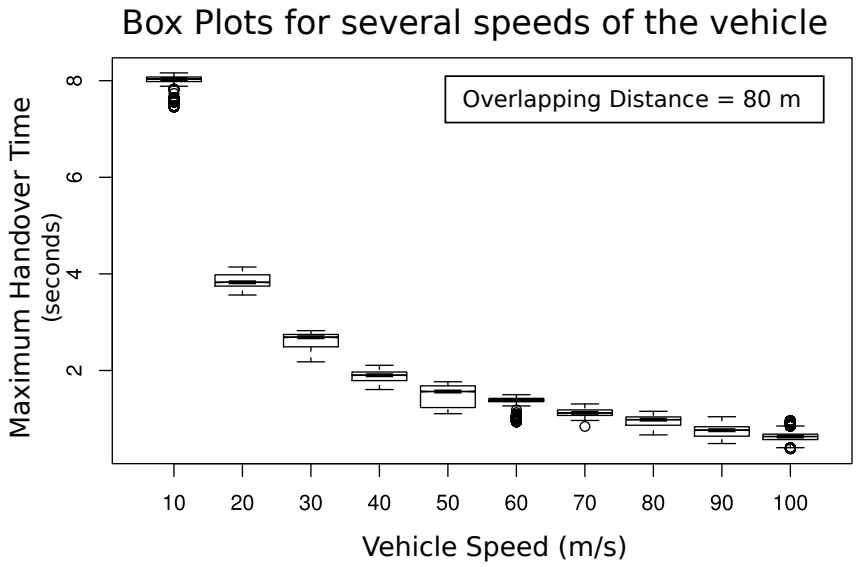
Let us consider an overlapping distance of d_o meters and a vehicle speed of v m/s. Considering the AAA delay ($\approx$ 5 ms) as negligible, the handover time is given by the following expression:

$$t_h = \frac{d_o}{v} - t_{discovery} \tag{4.1}$$

Thus, when considering a vehicle speed of 350 km/h, an overlapping distance of 80 meters, and an optimized scanning method with its maximum discovery delay overestimated at 500 ms, the minimum handover time is approximately 322 ms. Hence, the SD should finish the updating procedure before t_h in order to ensure no packet losses due to misrouting. However, this lower bound might be larger if the next WSAP is detected before the 500 ms (a shorter $t_{discovery}$). Theoretically, the maximum handover time is when $t_{discovery} \rightarrow 0$, then, the upper bound is 822 ms approximately. In order to asses how far is the real handover time from the theoretical bounds, we evaluated the maximum time that the route updating procedure has to complete its task by means of simulation. We simulate the handover of an SD along a trajectory covered by 10 APs at different speeds. The vehicle (containing the SD) travels the trajectory in a round-trip way for 1500 seconds. We assume a wireless coverage of 230 meters overlapped approximately 80 meters ($\pm$ some meters due to the railway is not a straight line). The 802.11 *mgmt* module of each wireless NIC considers the WSAP lost when 2 beacons are missed. We measure the maximum handover time from the moment the SD obtains a positive association until the moment the passive wireless NIC considers its WSAP lost. This means that our estimation is as much 200 ms overestimated. The *MinChannelTime* and *MaxChannelTime* are configured to 1 ms and 10 ms respectively according to the recommendations made by [Velayos & Karlsson 2004]. The Figure 4.5 shows the maximum handover time allowed for speeds from 10 m/s (36 km/h) up-to 100 m/s (360 km/h).

We observe the average of the simulated maximum handover time is closer to the theoretical upper bound than the lower bound. Hence, the scanning algorithm finds in average the next WSAP short time after being entered under its coverage area. However, the outliers evidences the cases when the SD has found the next WSAP very early or very late in the coverage. As these cases were considered outliers, they are not representative of the maximum handover time. Recall the maximum time is overestimated 200 ms due to the WSAP lost detection (2 beacons). In conclusion, the proposed scanning algorithm allows the routes updating procedure to operate almost all the time that the SD is under mutual coverage of two WSAPs.

When considering a heavy traffic condition, we realize it could suffer from buffer overflows due to the prioritization of the updating packets in the transmission queue. Indeed, if the routes update procedure (RUP) is aggressive when transmitting its updating packets, the passive wireless NIC will buffer all the incoming users' dataframes in order to give priority to the RUP packets. Thus, the route updating procedure should not be aggressive when propagating the new route in order to avoid the excessive buffering of traffic and therefore, reduce the probability of buffer overflows. Nevertheless, as both routes are active during



Speed	Theoretical		Simulated			
	Lower	Upper	Min	Max	Mean	Std.dev
10 m/s	7.50	8.00	7.46	8.16	7.98	0.17
20 m/s	3.50	4.00	3.56	4.14	3.85	0.15
30 m/s	2.17	2.67	2.18	2.82	2.59	0.20
40 m/s	1.50	2.00	1.61	2.11	1.88	0.11
50 m/s	1.10	1.60	1.11	1.77	1.48	0.22
60 m/s	0.83	1.33	0.94	1.50	1.37	0.10
70 m/s	0.64	1.14	0.84	1.31	1.13	0.08
80 m/s	0.50	1.00	0.67	1.15	0.95	0.11
90 m/s	0.39	0.89	0.49	1.04	0.74	0.12
100 m/s	0.30	0.80	0.38	0.96	0.64	0.13

(Time in seconds)

Figure 4.5: Maximum handover time measured for different speeds by simulation.

the mobility management phase, the SA might also balance the traffic among both NICs in order to cope the additional overhead caused by the RUP packets.

4.7 Conclusions

The proposed handover mechanism **partially eliminates the packet losses due to buffer overflows from the first four handover phases** (see Section 4.1) since it hides them from the active link. So, there is no need to buffer the users' dataframes in order to discover the neighborhood. However, the last phase of the handover is still threatened by buffer overflows if the routes updating procedure has an aggressive transmission policy. The operation of the Spiderman handover considers this situation when beginning to forward the incoming traffic through the passive wireless NIC as soon as its arrives from the *MAC Relay Unit*. However, this measure is not sufficient when assessing whether our proposed method is able to provide a lossless handover.

The use of a dual wireless NIC bridge device allows us:

- to perform a continuous proactive neighborhood discovery without affecting the active connections. This fact implies that the handover decision phase reduces its importance and the objective of the scanning method changes to find as soon as possible the next WSAP without worrying that much about the scanning delay (opportunistic neighborhood discovery). We showed that the proposed scanning method takes a good advantage from the predictability of the vehicle's trajectory with only a little information about it (the sequences of channels).
- to hide the AAA and the data-link re-establishment phases from the handover operation. Therefore, eliminate the buffering of traffic on these phases.
- to avoid packet losses due misrouting of packets (from the WSAP to the SD) since both routes (active and passive) are available almost all the time the SD remains under the mutual coverage of both WSAP.
- to provide a continuous bridged connection between the in-motion network and the infrastructure network, regardless the layer 3 protocol encapsulated on the Ethernet dataframes.

In summary, regarding the question this chapter aims at answering, we say that our proposed handover mechanism eliminates partially the packet losses due to buffer overflows. We showed that during the first four phases of the handover operation, there is no risk of buffer overflows due to the handover mechanism. However, the success on transferring all the ongoing connections (in both directions) between the *Network Gateway* and the in-motion network relies on how fast and aggressive the route updating procedure will be to update the infrastructure network. Hence our choice of a Layer 2 only mechanism.

Chapter 5

Routes Updating in an Infrastructure Network

In this chapter we describe the routes updating procedure used by the Spiderman handover. We name it the "Gratuitous ARP Loop" since it is based on the propagation of Gratuitous ARPs packets. We study its operational parameters experimentally and analytically, and we evaluate its operation by means of simulation. At the end of this chapter, we draw our conclusions about in terms of its transmission policy and the time to update the routing information in the infrastructure network.

5.1 Preliminaries

The question we aim at answering in this chapter is how to update the routing information in the infrastructure network, avoiding at the same time the misrouting of packets when performing the Spiderman handover.
From the previous chapter we learn that the handover time is bounded by the speed of the vehicle and the overlapping distance between WSAPs' coverage areas. This time correspond to the window of time where both routes (active and passive) are available. Therefore, it is the time budget that the **Routes Updating Procedure** (RUP) has to complete its task. The RUP will consider its task complete when receiving all the updating packets through the active wireless NIC. Thus, the *Spiderman Device* (SD) can ensure that the path between both WSAPs has been fully updated. If the time taken by the RUP is less than the minimum allowed handover time, the update of routes in the infrastructure network is guaranteed. If this time is in between the minimum and the maximum allowed handover time, the success of the RUP operation will depend on how fast the discovery phase is performed. Hence, there is a little risk of a RUP premature ending. If the RUP time is bigger than the maximum allowed handover time, the RUP will be prematurely ended, so, the risk of having misrouted packets is unavoidable.
We also learn from the previous chapter that the RUP might not only cause packet losses due to the misrouting of packets. It also might cause losses due to a buffer overflow at the passive wireless NIC. Indeed, a buffer overflow might occur when the RUP sends aggressively updating packets to the transmission queue of the passive wireless NIC. Recall that the Spiderman Device (SD) is handling the updating packets as priority traffic, so, the passive wireless NIC buffers the incoming users' dataframes while transmitting the RUP packets.

Hence, the basic requirements for a RUP are: 1) not to be aggressive when sending updating packets to the transmission queue in order to avoid buffer overflows; and 2) to receive all the updating packets back from the active wireless NIC before the minimum handover time, ensuring that both routes (active and passive) will be available when the infrastructure network is updated.

5.2 Problem Definition

As we already mentioned, the SD requires to receive all the updating packets through the active wireless NIC to declare the handover operation complete. The time taken by these packets to traverse the path between two consecutive WSAP depends on the topology of the infrastructure network. When considering traffic from external networks to the in-motion network (see Section 3.3), the traffic redirection will take place in some node of the infrastructure network. More precisely in the node where the active and passive routes intersect each other. If we consider a hierarchical topology such as a tree, the worst case is found at the root of that tree. Figure 5.1 represents this case.

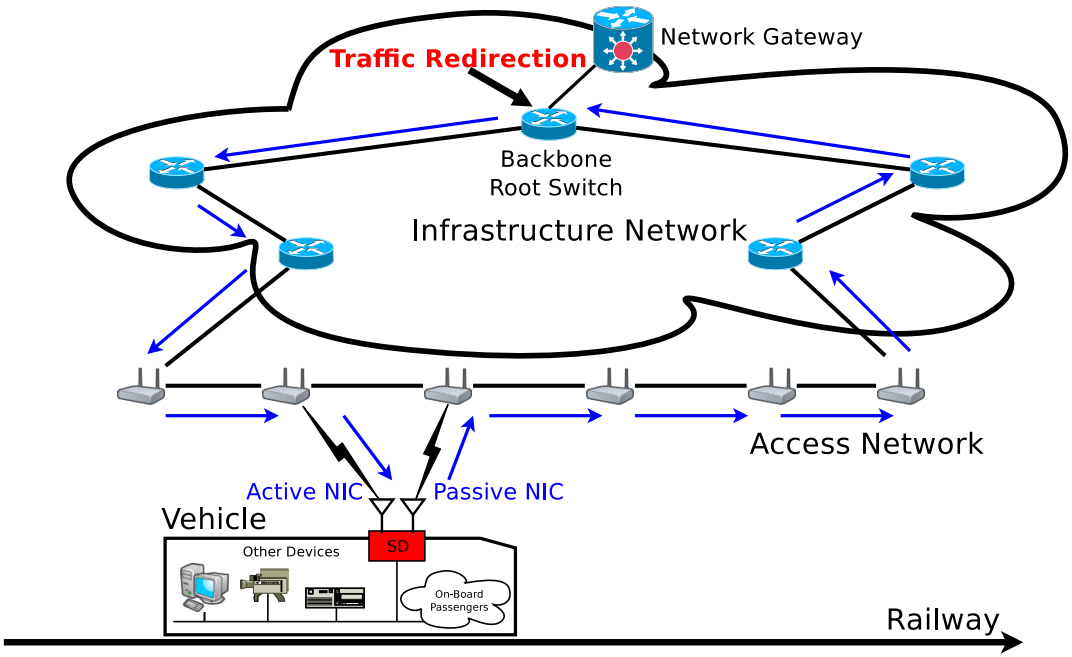


Figure 5.1: Path between two consecutive WSAPs.

Hence, the path between two consecutive WSAP becomes relevant. Therefore, a good infrastructure network should exhibit a low number of hops from the access network to the node (in the backbone network) where the active and passive routes intersect each other. Thus, the length of the path that the updating packets must travel is minimized.
As we are considering a bridged (layer 2) infrastructure network, the question we aim at solving is reformulated as how to traverse the path between two consecutive WSAP subject

to the following constraints:

1. the time in traversing the path must be within a window of time bounded by the minimum handover time allowed by the vehicle's speed and overlapping distance between both WSAP.

2. the transmission of the updating packets should be spread along this window of time in order to avoid the buffering of users' dataframes.

One important consideration when spreading the updating packets along the minimum handover time window is the transition from a wired to a wireless link (SD $\Leftrightarrow$ WSAP). This transition might produce queuing effects which may lead to buffer overflows.

5.3 Proposed Solution

In this section we propose a procedure to perform the updating of routes in a bridged (layer 2) infrastructure network. For this purpose we consider the requirements stated in the previous section and we assume the following: there is always a path between two consecutive WSAP and the traffic flows between the in-motion and external networks.
We name our proposal the **Gratuitous ARP Loop (GAL)** since it is based on the broadcast of *Gratuitous ARP* packets. The Address Resolution Protocol (ARP) is introduced by the RFC 826 [Plummer 1982] to cope with the translation of addresses between layer 2 and layer 3 protocols. In short, it provides the layer 2 address (a *MAC Address* for Ethernet) of a device given a layer 3 address (an *IP Address* for IP) within a layer 2 network segment. Let us consider an example: two devices with IP addresses A and B are connected by the same layer 2 segment. A wants to send a layer 3 dataframe to B. Thereby, A broadcasts an *ARP-Request* asking for the MAC address of B. This broadcast is received by all the devices in the segment, but answered only by B via an unicast *ARP-Response*. When A learns the physical address of B, the layer 2 communication is possible. All the bridged links along the path between A and B have learned by which link (port) A and B are known. This learning process is done by a *MAC Relay Unit* inside each layer 2 switch device, which handles all the bridged links connected to it. A *Gratuitous ARP* is an ARP frame containing the same source and destination addresses (for both layers). When the Gratuitous ARP is directed to a particular layer 2 address, the frame is an *ARP-Response*. When the Gratuitous ARP is a broadcast, it is an *ARP-Request*. For Ethernet and IP protocols, the ARP packet is 28 bytes long.
Our updating procedure is based on the broadcast of *Gratuitous ARP Requests* through the passive wireless NIC. These packets will flood the network and eventually will return to the SD through the active wireless NIC, closing the "loop". The procedure starts when the SD gets a positive association at the passive wireless NIC. The GAL retrieves all the MAC addresses known by the *Spiderman Agent* (SA) from the *MAC Table*. Then, it sends **one Gratuitous ARP (Request) for each MAC address** to the transmission queue of the passive wireless NIC. This delivery of packets is done in **bursts** of *BurstSize* packets. For

this purpose, two timers are defined: the **InterARPTimer** and the **InterBurstTimer**. In summary, our procedure depends on three variables: the **BurstSize**, the **InterARP Delay** and **InterBurstDelay**. The Figure 5.2 depicts the procedure.

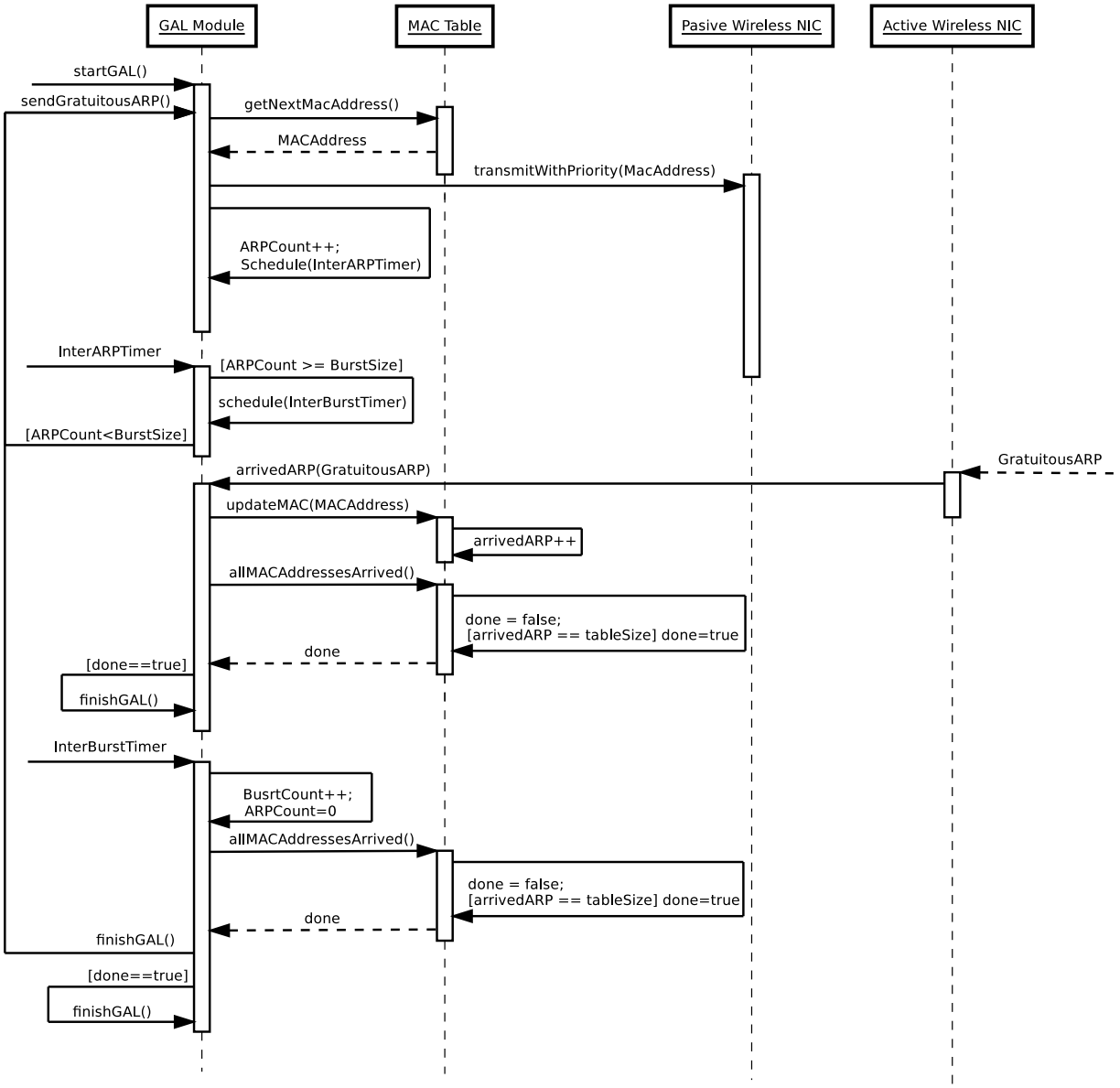


Figure 5.2: Gratuitous ARP Loop procedure.

The method *getNextMACAddress()* returns the next MAC address in the list that has not yet arrived, starting from the last returned address. Thus, the list is covered in a round-robin way. The *MAC Table* accounts the arrived MAC address by means of the *updateMACAddress(MACAddress)* method, and by calling the method *allMACAddressesArrived()*, the GAL determines when to finish the procedure.
Let us consider an example to explain the GAL operation. The *MAC Table* has registered 48 MAC addresses and the *BurstSize* is equal to 10. The passive wireless NIC gets a

positive association with a new WSAP, therefore, the SA commands the RUP to start the GAL procedure. The GAL gets the first MAC from the list and sends it to the passive wireless NIC transmission queue. Then, it schedules the *InterARPTimer* to be triggered at *InterARPDelay*. When the timer expires, the process is repeated as long as the number of ARP sent is equal to or less than *BurstSize* (10 in this case), otherwise, the *InterBurstTimer* is scheduled at *InterBurstDelay*. Meanwhile, the Gratuitous ARP packets might arrive in any order through the active wireless NIC. Thus, not all the transmitted MAC addresses may arrive before to complete the first round of the list. Note that the last burst contains only 8 addresses. Thereby, the *MAC Table* starts from the beginning of list (second round) returning only the addresses that have not yet arrived to complete the burst. Thus, the process is repeated until all the addresses have arrived through the active wireless NIC, when the process is finished. Later on we discuss about the timers delays and the size of the burst.

5.3.1 GAL Implementation

From the Figure 5.2 we identify six methods that the GAL should implement: *startGAL()*, *finishGAL()*, *sendGratuitousARP()*, *arrivedMAC(MACAddress)*, *schedule(Timer)* and the timer handling function, called *handleTimer(Timer)*. The first two methods are in charge of the initialization and termination of the process. The *sendGratuitousARP()* and the *arrivedMACAddress(MACAddress)* methods are asynchronous, therefore, they must be **thread safe** to ensure their correct operation. In addition, the *MAC Table* also must be thread safe in order to provide a mutual-exclusive access to its information. The method *schedule(Timer)* is in charge of scheduling the *InterARPTimer* and *InterBurstTimer* timers at their corresponding triggering times. These timers are defined by the *InterARPDelay* and the *InterBurstDelay* parameters respectively. The *handleTimer(Timer)* method aims at processing each timer, implementing the operations the GAL must do when they are triggered. This latter method should be also thread safe despite the fact that the triggering of each timer should be not concurrent.

According to the architecture of the Spiderman Device, the SA should provide the required hooks to integrate the RUP into the handover operation. The transmission and reception of updating packets (Gratuitous ARP packets for the GAL) is not a direct communication between the RUP and the wireless NIC management module. This communication is made through the SA, which is not depicted in the Figure 5.2 due to space constraints. Thus, when the RUP wants to transmit an updating packet, it relies on the hooks provided by the SA, which immediately should send the packet to the passive wireless NIC. And similarly when receiving an updating packet. The SA detects the packet and calls the method *arrivedMAC(MACAddress)* of the RUP.

5.3.2 GAL Parameters

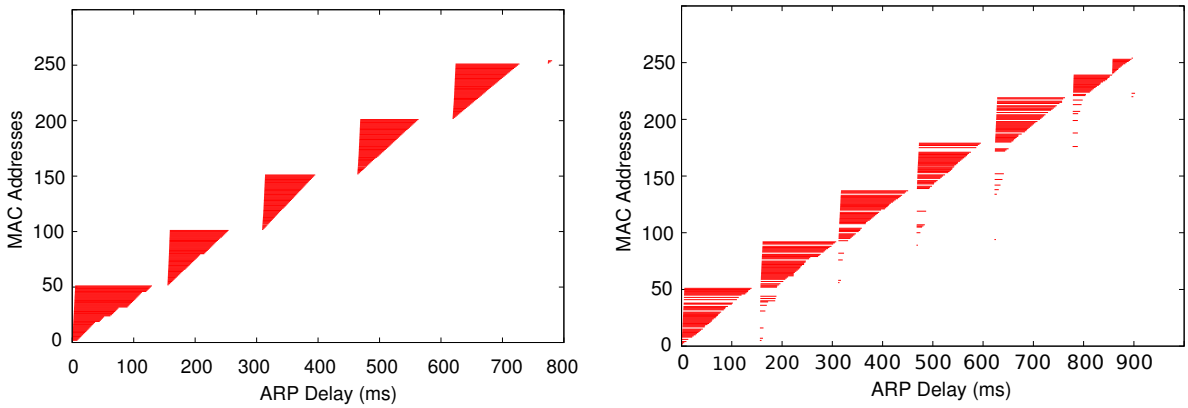
In this section we discuss the key elements of our proposed routes updating procedure: the **Gratuitous ARP Loop**. In the previous section we showed that its operation depends on three parameters: the *InterARPDelay*, *InterBurstDelay* and *BurstSize*.
We start the discussion with the *InterARPDelay*. The objective of this delay is two-fold: 1) to avoid a buffer overflow at the passive wireless NIC's transmission queue (the priority queue) due to the excessive queuing of Gratuitous ARP packets; and 2) to avoid losing a Gratuitous ARP packet due to a collision (recall that broadcast packets are not acknowledged by the receiver). When the passive wireless NIC transmits a broadcast frame, it waits a DIFS+CW[1] before it releases the radio for the next operation (receive or transmit). This delay does not ensure that the passive wireless NIC will not receive its own broadcast when the WSAP retransmits it to the medium. Let us assume that the passive wireless NIC has two broadcast packets to transmit, one after the other. In this case two situations may occur:

1. The WSAP begins the retransmission of the first broadcast before the passive wireless NIC begins the transmission of the second one: in this case, the NIC might find the medium "busy". Therefore, it will perform an exponential back-off, delaying the transmission of the second broadcast frame.

2. When the transmission takes place at the border of the WSAP coverage area, the passive wireless NIC might not detect the first broadcast retransmission due to a low SNIR. Therefore, it will begin the transmission of the second one, producing a collision between both.

Hence, the GAL should wait a longer time before sending the next Gratuitous ARP packet in order to reduce the risk of delaying it (back-off) or losing it (collision). We measure this risk by means of experimentation. We did setup two 802.11b WSAP at 150 meters apart, connected by a backhaul dummy network consisting of a Cisco 2950 switch without background traffic. We put a SD in the middle of both WSAP and we performed the GAL for 250 MAC addresses in bursts of 50 addresses, sending Gratuitous ARPs one after the other, in other words, $InterARPDelay = 0$. We performed 100 trials in two environments: interference-free scenario and interference of neighboring WiFi hot-spots scenario. The Figure 5.3 shows the resulting measured delays by each transmitted MAC Address for both scenarios. Each horizontal line represents the time difference between the depart and arrival of a single Gratuitous ARP (the delay of the loop for each address). The y-axis identifies each transmitted MAC address.
We use an *InterBurstDelay* of 150 ms to isolate each burst transmission in the plot (avoiding overlapped measures), so, bursts start at regular intervals of 150 ms. The triangle shape of the delay measured for the 50 MAC addresses in each burst is due to we put all the ARP packets in the transmission queue in a short time, so, the last ARP must wait the

[1] CW = Contention Window



(a) **without** interference. (b) **with** interference.

Figure 5.3: Gratuitous ARP delay for 250 MAC addresses.

transmission of all the preceding packets in order to be transmitted. We did observe more occurrences of ARPs retransmission along the trials performed in the scenario with interference. Note the single ARPs transmission delays below each burst, starting from the second one. Each one of these lines depict one MAC address that was transmitted in a previous burst that did not arrive before completing the *InterBurstDelay*. Therefore, they were retransmitted in the next burst. The scenario without interference also shown some retransmission cases, but the number of cases were very limited. In order to isolate the causes of these repetitions, we set a *sniffer* device beside each 802.11 radio in order to capture the transmission time of each packet despite the possible collisions (Capture Effect). We determined the causes of losses were mostly collisions, but also we found irregular gaps between consecutive packets. These gaps might be explained by back-off on the transmissions. However, when increasing the *InterARPDelay*, the inter packet time becomes more regular. Therefore, we attribute this irregularity to a back-off on the transmission. Hence, we draw two conclusions: 1) when sending the Gratuitous ARPs one after the other ($InterARPDelay = 0$ ms), we did observe retransmission cases due to both causes identified before, which eventually will yield to larger delays in propagating the new route to reach the SD; and 2) the mean time taken by a burst of 50 MAC addresses to complete the loop is $\approx$ 150 ms. This means that we can use an *InterARPDelay* of 3 ms obtaining the same result, but avoiding the buffering of Gratuitous ARPs in the priority queue of the passive wireless NIC.

We repeat the experiment, but with an in-motion SD. The Figure 5.4 shows the results of the ARP delays when the SD was inside a vehicle at 90 km/h. The coverage area of the WSAP was approximately 300 meters wide. We placed it at the side of a road near our laboratory, in an area without WiFi interference (no nearby hot-spots). The vehicle remained around 12 seconds under the coverage area of the WSAP.

Observe the first burst took about 150 ms to complete the first set of 50 MAC addresses. However, the next bursts were considerably shorter. Indeed, the vehicle was closer to the

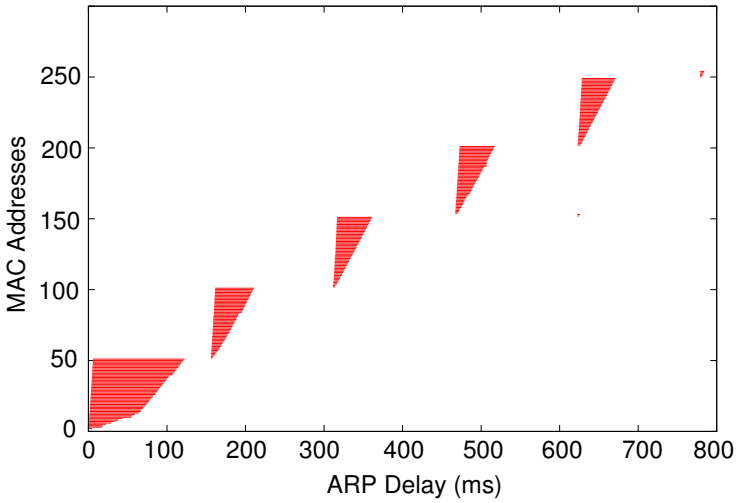


Figure 5.4: ARP Delays for 250 MAC addresses for an in-motion SD at 90 km/h

WSAP and a better radio signal allows a better data transmission rate as pointed out by [Ott & Kutscher 2004] (the entry/production/exit phases).

In summary, by honoring the upper bound we found for a burst transmission delay, we define the constraint of $InterARPDelay > 3$ milliseconds in order to avoid the buffering of Gratuitous ARP packets without considering background traffic. Nevertheless, in a real traffic situation, the *InterARPDelay* should be larger in order to avoid user's dataframes buffering when transmitting a burst.

Regarding the *InterBurstDelay*. Its objective is to avoid an excessive queuing of the users' dataframes due to the transmission of a burst. Note that as soon as the first burst is propagated into the infrastructure network, the traffic starts arriving to the WSAP at which the passive wireless NIC is associated with, therefore, this traffic might get buffered due to further burst transmissions. Hence, we require to wait some time after transmitting a burst in order to allow the WSAP to transmit the queued traffic, and, if some traffic got buffered at the passive wireless NIC, allow its transmission as well. To find a proper value for this delay, we consider the minimum handover time discussed in Section 4.6. We showed that the RUP should be performed before a minimum handover time in order to avoid losing packets due to the misrouting problem. Thus, the time required by the GAL to complete the updating procedure (t_{gal}) should be equal to or less than the minimum handover allowed. In other words: $t_{gal} \leq min(t_h)$. Let be d_{IArp} the *InterARPDelay*, d_{IBurst} the *InterBurstDelay*, B_s the burst size and T_s the number of MAC addresses registered in the *MAC Table*. When considering all the Gratuitous ARP packets are transmitted without retransmissions in further bursts (the minimum t_{gal} we can obtain), the equation 5.1 gives the theoretical time taken by the GAL to complete its task:

$$min(t_{gal}) = d_{IArp} \cdot T_s \left(\frac{B_s - 1}{B_s} \right) + d_{IBurst} \left(\frac{T_s - B_s}{B_s} \right) \tag{5.1}$$

Furthermore, we consider a certain percentage of MAC addresses are lost in the process. It implies that they must be retransmitted in later bursts. This retransmission can be seen as a larger *MAC Table* to be transmitted. Hence, considering a fraction of α addresses to be retransmitted, the expression 5.2 gives us the maximum theoretical time that the GAL should take in transmit $(1+\alpha)T_s$ addresses.

$$max(t_{gal}) = d_{IArp} \cdot (1+\alpha)T_s \left(\frac{B_s - 1}{B_s}\right) + d_{IBurst} \left(\frac{(1+\alpha)T_s - B_s}{B_s}\right) \tag{5.2}$$

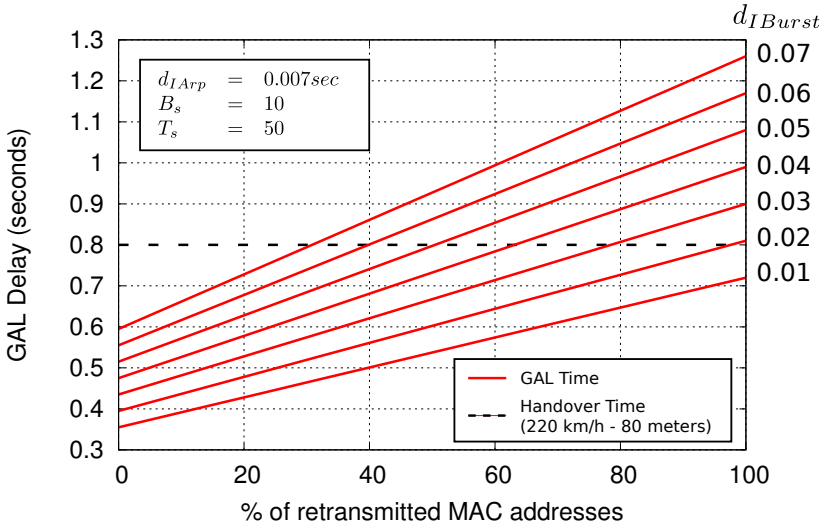


Figure 5.5: Theoretical Gratuitous ARP time considering retransmissions.

Figure 5.5 shows the GAL time (t_{gal}) for different percentages of retransmitted MAC addresses when assuming several *InterBurstDelay* values. Thereby, when assuming a minimum handover time of 0.8 seconds ($\approx$ 220 km/h with an overlapping distance of 80 meters), we observe that an $InterBurstDelay = 0.02$ seconds allows us to retransmit 50 MAC addresses once (100 transmissions).

In summary, we adjust the *InterBurstDelay* according to the minimum handover time allowed for the imposed mobility conditions (maximum speed and maximum overlapped distance of WSAP coverage areas), and the maximum percentage of MAC addresses that might be retransmitted within the routes updating procedure.

Regarding the *BurstSize*. The objective of this parameter is transversal to both parameters discussed above: on the hand hand to avoid an excessive queueing of user's dataframe when transmitting a burst, and on the other hand, to avoid an excessive queueing of Gratuitous ARPs at the passive wireless NIC priority buffer. The basic rule to adjust this parameter comes from the idea of longer bursts means shorter GAL times (and shorter bursts means longer GAL times). Thus, for a low traffic load, a longer burst is reasonable since there is no risk of producing an excessive queueing of users' dataframes, so, the lower the GAL time we can achieve. However, for a high traffic load, a shorter burst is required to avoid

buffer overflows for both Gratuitous ARPs and users' dataframes. Figure 5.6 depicts the GAL time for different *BurstSize* values for the parameters shown within the plot.

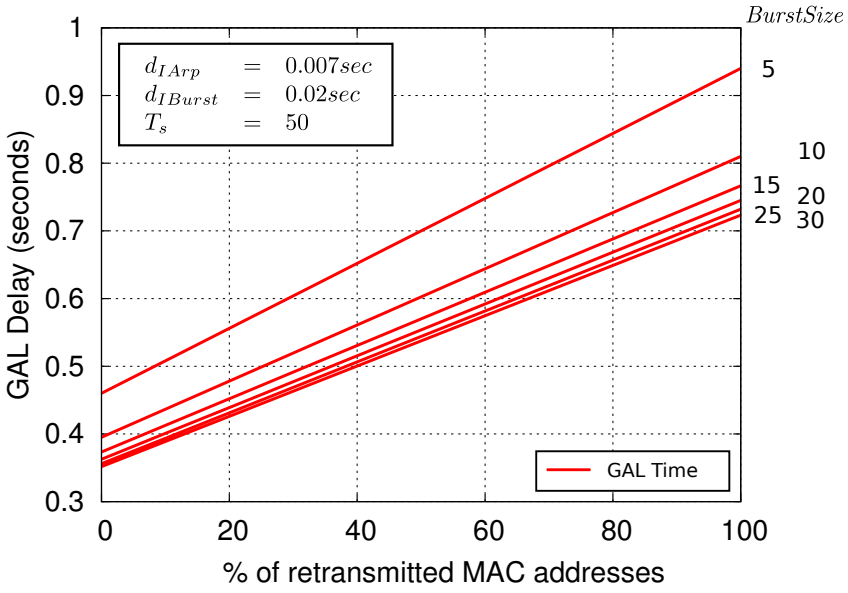


Figure 5.6: Theoretical Gratuitous ARP time for different *BurstSize*.

Thereby, for a minimum handover time of 0.8 seconds, a $BurstSize = 10$ allows us to perform a GAL for 50 MAC addresses, permitting the GAL to transmit each one twice. However, for a minimum handover time of 0.7 seconds, a $BurstSize = 30$ is required to have as much as 94% of MAC addresses lost in the first round of bursts.
Finally, we evaluated different sets of values for the GAL parameters by means of experimentation, considering a background traffic. In the testbed described above, we injected a TCP flow between the SD and an external peer attached to the Cisco switch. We measured the passive wireless NIC transmission queue when performing the GAL of 250 MAC addresses, finding that with an *InterARPDelay* of 7 ms, an *InterBurstDelay* of 20 ms and a *BurstSize* of 10, we did not evidence a significant increase in the transmission queue length. Therefore, the maximum theoretical GAL time, according to these values, are depicted in Figure 5.7 for different number of MAC addresses.
The long dashed horizontal lines (in blue) show the minimum handover time allowed for a given speed, which are indicated beside the plot. Notice when assuming a vehicle speed of 100 m/s (360 km/h) and a 200% of ARP retransmissions (each Gratuitous ARP is transmitted 3 times), the GAL can not operate for more than 10 MAC addresses. For 70 m/s (252 km/h), not more than 20 MAC addresses. And for 60 m/s (216 km/h), the GAL can experience a 100% of retransmission for 50 MAC addresses without an early finish. Therefore, a *BurstSize* of 10 MAC addresses seems to be reasonable when thinking in a high-speed scenario with a security bound of 200% of retransmissions.
Notice we are considering a case with 50 MAC addresses directly connected to the infrastructure network, which does not mean that there are 50 end-users connected to the

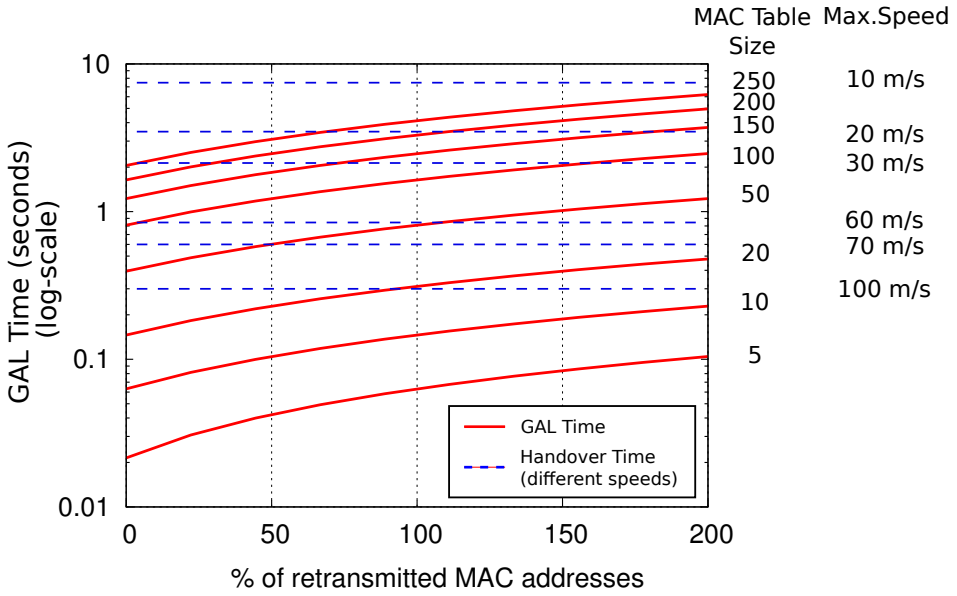


Figure 5.7: Theoretical Gratuitous ARP time for *InterARPDelay=7* ms, *InterBurstDelay=20* ms and *BurstSize=10*.

network. In a more realistic scenario, end-users connections are masqueraded behind a single network address by means of techniques such as *Natting*. Therefore, a system able to operate with a few MAC addresses (or only with one in worst case) is sufficient to provide connectivity to many end-users inside the in-motion network.

5.4 Evaluation

In this section we aim at evaluating the proposed routes updating procedure by means of simulation. We focus on: 1) the effects of speed and number of MAC addresses on the GAL time; 2) the effect of speed on packet losses; and 3) the effect of the handover operation on the observed delay from the in-motion network to an external network.

5.4.1 Simulated Scenario

As we are interested in the effects of the GAL on the handover operation, we consider a small train scenario with a dummy infrastructure network to avoid measuring nuisance factors coming from the network. We consider a small railway of 2 km long covered by 10 WSAP according to the assumptions stated in Section 3.2. All the WSAPs are connected to a layer 2 switch by means of a fiber link. This switch is connected to a layer 3 router (the *Network Gateway*), which provides access to external networks, represented by an *external host*. Each WSAP is following a determined sequence of channels (1,6,11) starting from the first WSAP, according to the requirements of the Spiderman Device (see Section 4.5.3). The Figure 5.8 depicts this scenario.

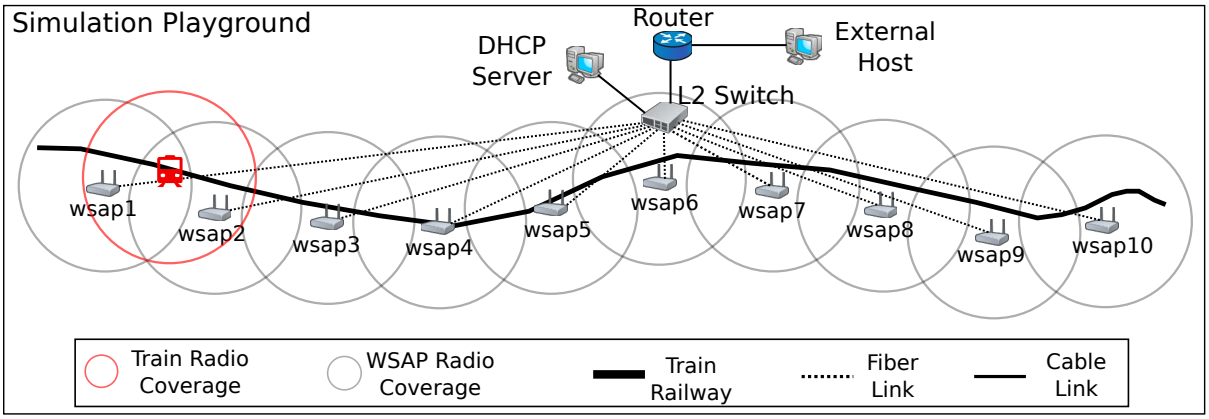


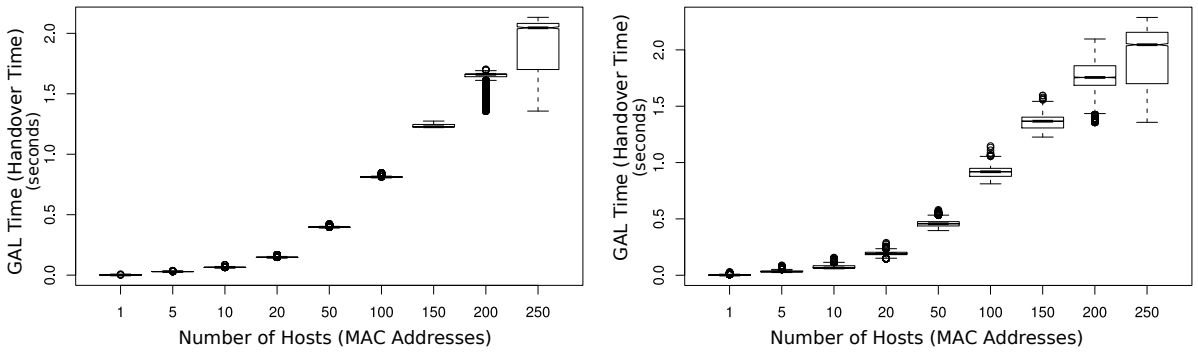
Figure 5.8: Simulated scenario for the GAL evaluation.

For all the comparisons presented in this evaluation, we consider two trains: one equipped with a normal 802.11b access terminal (the **One-Radio Train**) performing *Network Address Translation (Natting)* of all the hosts inside the in-motion network; and one train equipped with a Spiderman access terminal (the **Spiderman Train**), bridging the in-motion network with the infrastructure network. It is worth to mention that the normal access terminal implements an optimized scanning method. It scans only the channels in the WSAP channel's sequence with the same timers delay than the Spiderman access terminal. The scanning timers *MinChannelTime* and *MaxChannelTime* are configured to 1 ms and 10 ms respectively [Velayos & Karlsson 2004], and the WSAP lost detection is set at 10 beacons to avoid a false handover due to the saturation of the WiFi link [Raghavendra *et al.* 2007]. We evaluate each train for speeds from 10 m/s up-to 100 m/s for 1500 seconds of simulation, 10 repetitions for each train. The GAL parameters used for this evaluation are: $BurstSize = 10$, $InterARPDelay = 7$ ms and $InterBurstDelay = 20$ ms, corresponding to values we found by experimentation when avoiding a saturation in an 11 Mbit/s WiFi link.

We use the *OMNeT++ 4.1* discrete event simulator as engine to simulate this scenario, which was modelled by using the *INET Framework*. The version of this model corresponds to the 20100323, branched in march 2010 with further modifications to allow the measurement of the reception power on IEEE 802.11b devices, log of some radio operations, and some improvements of the IEEE 802.11 management stack. The train mobility was modelled according to [Maureira *et al.* 2009] for a fixed train speed. In addition, new models were added to simulate the *DHCP protocol* and the Layer 3 *Network Address Translation* (NAT). The radio interference model is based on an "additive-noise-signal" evaluation among all the airframes "on-the-air". The propagation radio model is the Free Space Path-loss with the path loss coefficient $\alpha = 3.2$ in order to obtain a sensitivity threshold of -86 dBm around the border of a coverage area of 230 meters wide. Thermal noise is set to -110 dBm and radio transmission's power is 100 mW.

5.4.2 GAL Time - Handover Time

In this section we evaluate the GAL time, which correspond to the handover time for the Spiderman handover. We focus our evaluation on the effect of the number of hosts (MAC addresses) inside the in-motion network and the speed effects of the vehicle.



(a) **without** background traffic.

(b) **with** background traffic.

Figure 5.9: GAL delay (in seconds) for different number of hosts.

Evaluating the number of nodes, we set two scenarios with one "Spiderman Train": **without background traffic** and **with background traffic**. Along all the simulation time, all hosts are sending an ICMP ping to the external host at regular intervals of 1 second ($\pm$ 0.01s of random variability) and there is one host exchanging a TCP stream with the external host. This TCP stream generates a constant saturation and overflows at the SD WiFi up-link. We evaluate several number of hosts inside the in-motion network travelling at 60 m/s in order to have the same number of handover operations for all simulation trials. Figure 5.9 shows the distribution of the measured GAL time for the scenario with and without background traffic. In Section A.1, two tables show, for each scenario, the GAL basic statistics of the measured GAL time (the minimum, the maximum, the mean, the standard deviation) and the number of Gratuitous ARPs sent by the GAL (the minimum, the maximum and the percentage of the retransmitted MAC addresses).
The conclusion we draw from these results is that the GAL time increases with the traffic load. The scenario with background traffic has a GAL time in average 10% larger than the scenario without traffic. Also the standard deviation is larger for the scenario with traffic. When observing the minimum observed values for the GAL time, we notice they match with the theoretical minimum value discussed in Section 5.3.2, except for the 1, 200 and 250 hosts. For a single host, it is obvious since the *MAC Table* size is less than the burst size. However for 200 and 250 hosts, the observed minimum is lower than the theoretical minimum and also they tend to converge to 1.357 seconds when evaluating larger number of hosts. Indeed, for 60 m/s, the maximum handover time allowed is 1.33 seconds, which suggest that the GAL has ended prematurely, fact we confirmed by exploring the simulation results. The maximum observed for these two cases is explained by the irregularity of the

vehicle's trajectory. While the WSAP are approximately 150 meters apart, the traveled distance is not necessarily a straight line between contiguous WSAPs. (see Figure 5.8). In some cases the traveled distance is about 130 meters, (between the *wsap9* and *wsap10*). Nevertheless, for the percentages of retransmitted MAC addresses found by simulation, the maximum GAL time agrees with the theoretical maximum bound defined by equation 5.2. Figure 5.10 shows both maximum GAL times.

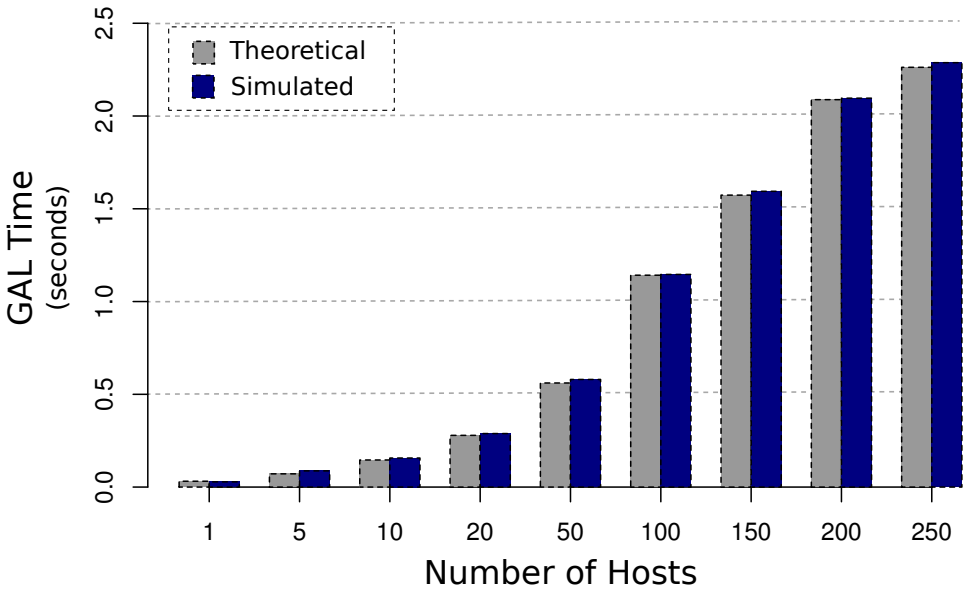


Figure 5.10: Maximum GAL time: theoretical and simulated values.

5.4.2.1 The Spiderman handover and the Standard handover (optimized)

We compare the Spiderman handover with the standard handover at several speeds for a fixed number of hosts. As mentioned at the beginning of this section, we introduced some optimizations to the standard scanning method in order to speed up its operation. We use ideas coming from the literature, specially from neighbors graphs [Shin *et al.* 2004] and tuning of timers [Rizvi *et al.* 2009].

We consider for this evaluation a constant bit rate (CBR) background traffic. It is generated by a modified ICMP Ping application (echo) which generates a bidirectional traffic between the internal hosts (inside the in-motion network) and the external host. We consider 50 internal hosts, each one transmitting a "ping" at intervals between 0.15 and 0.25 seconds (distributed randomly uniform). Each internal host considers a packet size of 1024 bytes to generate a saturation at the wireless up-link. In normal (static) conditions, this up-link has an average of 1 packets in the transmission queue, therefore, the wireless NIC has always a packet to transmit. Figure 5.11 shows the handover time for Spiderman handover (*Spiderman Train*) and the Standard Optimized handover (*One-Radio Train*).

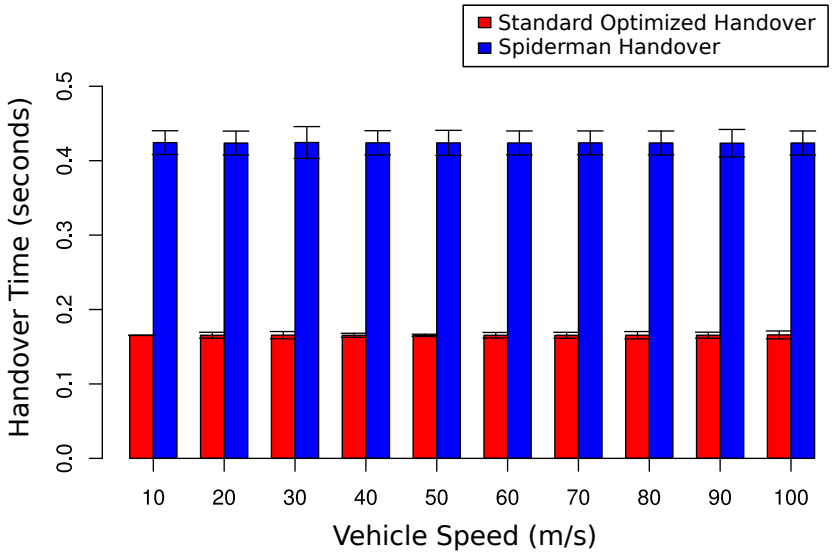


Figure 5.11: Handover time for 50 MAC addresses at different speeds.

Note the handover time for the Spiderman handover is higher than the Standard Optimized handover for the same number of hosts. This result is explained by the fact that the optimized neighborhood discovery finds its worst case when listening the 3 channels in the WSAP channel's sequence for *ProbeDelay+MaxChannelTime* seconds before to find the next WSAP. This upper bound (0.15 sec) is lower than the minimum time taken by the GAL (0.395 sec) to propagate the routing information for 50 hosts. However, the *One Radio Train* is performing *Natting* of all the hosts inside the in-motion network, therefore, the handover is made only for one station (the access terminal). Thus, when comparing the Standard Optimized handover with the Spiderman handover for a single station (the maximum case in Figure 5.9b), the Spiderman handover performs much better than the traditional handover approach, even the optimized version (see Figure 5.9b for one host). Finally, when evaluating the variability of the GAL time (Figure 5.12), we realize the minimum GAL time corresponds clearly to the theoretical bounds discussed in Section 5.3.2. The upper bound at 60 m/s for 50 hosts is 0.58 seconds, which corresponds approximately to a 40% of repeated Gratuitous ARP packets. This means that within the RUP, there were 20 retransmissions of Gratuitous ARP packets before completing the loop.

5.4.3 Packet Losses

We consider for this evaluation the packet losses for both handover schemes at application layer. We use the same ICMP Ping application and configuration as we used for the evaluation in the previous section. We measure the percentage of packet losses along the simulation time (1500 seconds) at different speeds (10 to 100 m/s) for 50 internal hosts. Figure 5.13 shows the percentages of ICMP packet losses for both handover schemes.
Note that the *Spiderman handover* exhibits no apparent packet losses. In fact, the observed

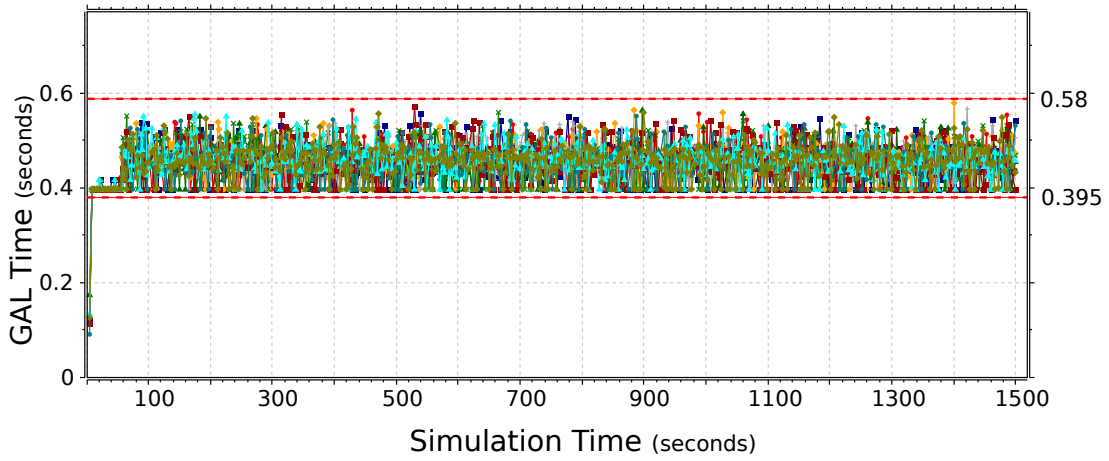


Figure 5.12: Handover time for 50 MAC addresses at 60 m/s.

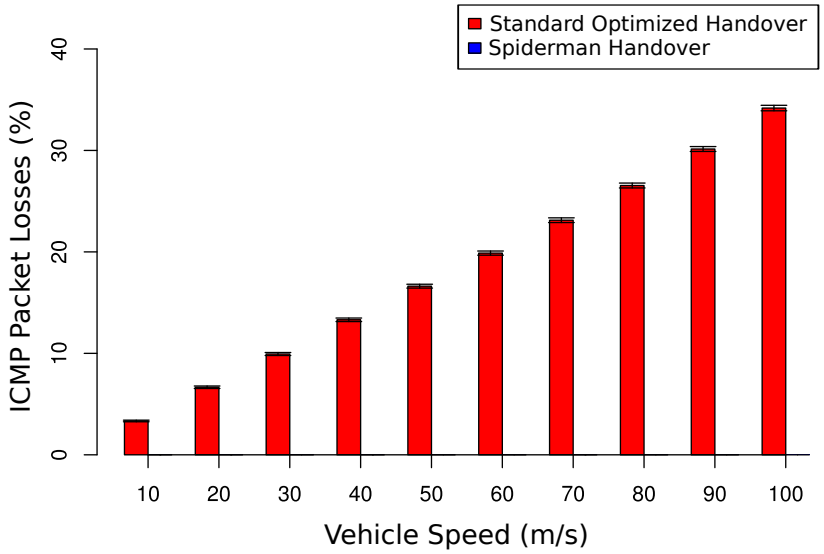


Figure 5.13: ICMP Packet losses for 50 MAC addresses at different speeds.

losses for the Spiderman handover are so small ($\approx 0.02\%$) that they are not significant in comparison with the scale of the plot. This results is self explicative. The Spiderman handover has successfully avoided the packet losses at application layer considering a moderated saturation at the wireless up-link. However, this result corresponds to the traffic load previously mentioned: ICMP packets of 1024 bytes long at intervals between 0.15 and 0.25 seconds (CBR Traffic), generating an average buffer length of 1 packet at the passive wireless NIC (moderated). Under heavier traffic conditions, the buffer overflows might be unavoidable.

5.4.4 Round Trip Time

We use the same modified ICMP application used for both evaluations presented before to compare the Round Trip Time (RTT) between both handover schemes. We measure the RTT from the in-motion network (50 hosts) to the external host (representing an external network). The traffic conditions are similar to the previous ones, saturating the access terminal WiFi up-link up-to an average of 1 packets in the transmission queue. Figure 5.14 depicts the average RTT observed by the 50 hosts for both handover schemes.

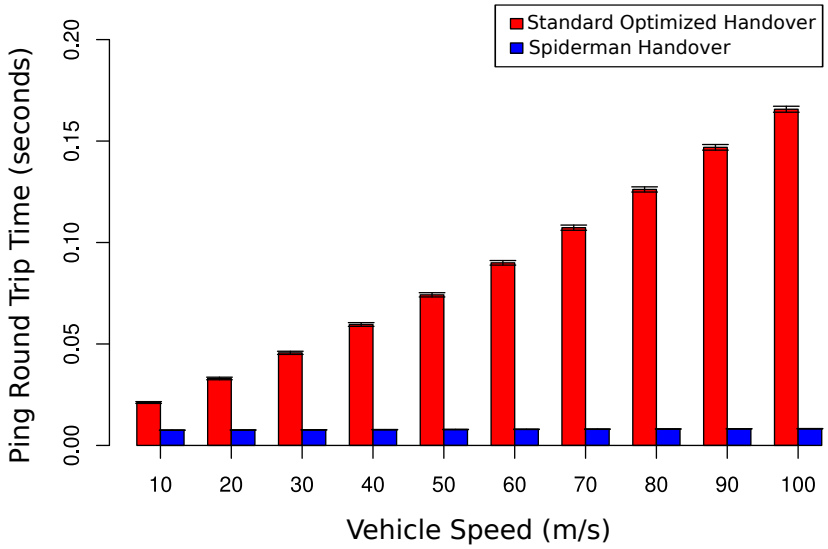


Figure 5.14: ICMP Ping Round Trip Time (RTT) for 50 MAC addresses at different speeds.

The outcome of this evaluation is the RTT for the Standard Optimized handover increases with the speed of the vehicle, while the Spiderman handover remains invariant to it. This result is explained by the fact that the Standard Optimized handover buffers all the incoming users' dataframes while discovering the neighborhood, producing a buffer overflow of the transmission queue. This way, the last queued dataframe suffers the additional delay of dequeuing the whole buffer plus the time when the buffer was active. When combining this extra delay with the frequency of handover (see Section 3.2), the higher the frequency, the larger the average RTT. It does not mean that hosts will experience always a higher delay when increasing the speed. It means the host will experience a larger delay when the access terminal recovers the data-link layer with the next WSAP. Therefore, it will increase the average RTT when performing a larger number of handovers by unit of time.

5.5 Discussion

In the previous section we showed that the Spiderman handover is able to reduce the packet losses (caused by the handover) to "virtually zero" in a condition of moderate saturation

of the WiFi up-link. In addition, we showed the delay perceived by the on-board hosts is invariant to the speed, which indicates no extra buffering of users' dataframes. As a counterpart, the Spiderman handover time is larger than the Standard Optimized handover time for the same number of hosts. However, this comparison is not fair since the only way to allow the Standard Optimized handover to work as an access terminal is by doing *Natting*, which transforms the handover of n MAC addresses into a single MAC address handover. Therefore, when comparing both handover schemes with *Natting*, the Spiderman handover performs much better than the Standard Optimized handover.

The parametrization of the GAL is the hot-topic of this discussion. We showed some bounds for the *InterARPDelay* and the *InterBurstDelay*, however their appropriate values depends on the traffic load on the WiFi up-link: a short *InterARPDelay* might cause a buffer overflow in the passive wireless NIC as well as in the active route WSAP (which transmit the Gratuitous ARP back to the SD); and a large *InterBurstDelay* might lead to an early termination of the GAL. The *BurstSize* helps to balance both parameters by combining the short time between Gratuitous ARP packets and the long time between bursts. We found an appropriate combination of values for a 11 Mbit/s wireless link by means of experimentation, which we validated through the theoretical analysis of the maximum allowed handover time. Hence, for a given number of MAC addresses, it is possible to find a combination of values such that the resulting maximum GAL time is less than the time minimum handover time allowed. Nevertheless, the option of transforming the n MAC addresses handover into a single MAC address handover by using *Natting* allows us to simplify the selection of the GAL parameters. When considering n passenger connections on-board the train, all of them masqueraded behind a single IP address/MAC address (attached to the access terminal), there is only one MAC address to update in the infrastructure network, therefore, the GAL parameters will be a resilient measure to ensure the correct propagation of the route.

As the **Gratuitous ARP Loop** is a RUP based on a layer 2 bridged network, it only copes with the routes updating problem at this level. Any further updating procedure to handle IP or any other layer 3 addressing scheme is avoided. Nevertheless, any layer 3 updating procedure depends on the layer 2 handover delay. Hence, we review some *Mobile IP* protocols to discuss how they consider the layer 2 handover delay. [Vassiliou & Zinonos 2010] analyze the layer 3 (IP) handover delay by experimentation. [Xie *et al.* 2007] present an analytical analysis. Both analysis assume the layer 2 handover delay between 10 ms and 100 ms (considering the disconnection phase when scanning). Their results pointed out the layer 3 handover can be performed in about 1 second. This result shows that a MIP scheme is not adequate for our reference scenario, since the small coverage area increments the handover frequency to such values that a delay of 1 second is unaffordable, as pointed out by [Hernandez & Helal 2001]. The evaluations performed by [Fowler & Zeadally 2006] and [Sethom *et al.* 2004] suggest that by using a micro-mobility handover scheme (HAWAII, Cellular IP, H-MPLS), the overall layer 3 handover time (layer 2 handover and routes updating included) might be reduced to an interval from 30 ms to 180 ms. However, these delays are still large for a 230 meters wide coverage area at 350 km/h with interruptions

caused by the layer 2 handover scheme.

5.6 Conclusions

At the beginning of this chapter, we posed the question of how to update the routing information in a layer 2 infrastructure network when performing the Spiderman handover. Along this chapter we presented our proposal to answer that question. We call this proposal *The Gratuitous ARP Loop (GAL)*, since it is based on the broadcast of Gratuitous ARP packets. We define its operation and propose a set of parameters to make it work in a 802.11b wireless link with a traffic condition of a moderate saturation. We evaluate the maximum time taken by the GAL to perform its task theoretically and by means of simulations, verifying that the simulation results meets the theoretical model. We evaluate the GAL for different number of hosts and vehicle speeds under the same moderate traffic condition. We focus on the packet losses and round trip time observed from the in-motion network.

Our results showed that **the Spiderman handover with the Gratuitous ARP Loop are able to provide a layer 2 handover operation for an in-motion network reducing the packet losses virtually to 0 and keeping the observed delay invariant to the handover frequency**. However, the number of hosts (MAC addresses) allowed for the GAL is bounded by the vehicle speed and the overlapping distance between WSAP coverage areas, which defines at the same time the maximum time of handover. Thereby, when fixing a maximum size of the in-motion network and the maximum speed of the vehicle, it is possible to adjust the GAL parameters in order to provide a handover operation with a very low risk of packet losses within the maximum allowed handover time. We realize that our procedure does not scale with the size of the in-motion network. However, techniques such as *Network Address Translation* and *IP Masquerading* makes our procedure scalable by transforming the handover operation for n hosts into a handover operation for a single host.

In conclusion, by using the Gratuitous ARP Loop together with the Spiderman handover, we are able to provide a routes updating procedure with a very little risk of packet losses. The misrouting problem is successfully coped by our proposal. However the buffer overflow problem depends on the traffic conditions, which are unpredictable. Nevertheless, for a low number of MAC addresses to update, this impact is minimized in such a way that we can state that our handover scheme is a lossless handover operation able to operate at a high mobility speed (up-to 360 km/h and possible more).

Part II

Design of an Infrastructure Network for Railway Scenarios

Chapter 6

Definition of a Backbone Topology for a Linear Access Network

This chapter is the beginning of the second part of this thesis. The aim of this part is to study the design of an *infrastructure network* for railways scenarios. This infrastructure network is formed by an *access network*, which provides the connectivity to the train *access terminal* (for example the Spiderman Device described in Chapter 4), and a *backbone network*, which interconnects this access network with the gateway of the entire network. In this chapter we define the backbone topologies we aim at studying for providing such a connectivity.

6.1 Introduction

In this chapter we aim at studying the design of an infrastructure network (access and backbone networks). The access network is composed by a set of WiFi Access Points (APs) deployed along a railway. The backbone network is connected in "some way" to this access network. Thus, this infrastructure network is used to link the in-motion network (on-board the train) with the *Network Gateway*, which provides to passengers the access to external networks. Due to the size of the access network (from hundreds to thousands of nodes), the infrastructure network should exhibit an appropriate resilience to ensure its correct operation. Otherwise, failures might degrade the quality of service perceived by train passengers. Therefore, the infrastructure network should consider a diversity of paths from the APs to the *Network Gateway* in order to cope with possible failures. In this direction, we assume the access network is **sequentially connected** (in a linear way) and **physically separated** from the backbone network. Thus, on the one hand, any access node has two possible paths to reach the backbone network: backward and forward. And on the other hand, a failure in one network (access or backbone), does not affect the other network. As a consequence, the points where the backbone and linear access networks are connected each other can be seen as "composite" nodes, formed by an access node and a backbone node, with different failure probabilities and called *gateways*. These two nodes are connected by a short "up-link" with 0 cost in terms of the number of hops to the *Network Gateway*. Figure 6.1 depicts this scenario.
As the size of the linear access network is determined by the length of the trajectory of the vehicle (the train in our case), our problem is reduced to the design of a backbone network to interconnect the linear access network with the *Network Gateway*. The points of interconnection (gateways) are defined by the location of the backbone nodes, therefore,

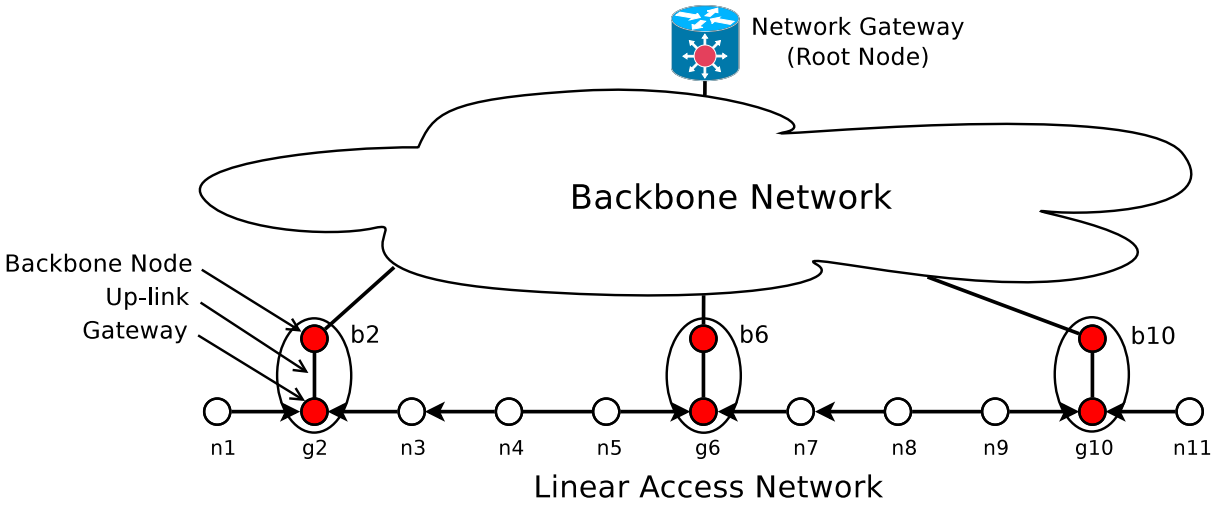


Figure 6.1: Backbone and linear access network connectivity.

we will name a gateway node and its correspondent backbone node with the same sub-index i enabling a better identification of backbone nodes.

6.1.1 Requirements

As mentioned in Section 3.3, we assume a communication between the in-motion network and external networks (i.e., Internet). Therefore, the traffic is flowing in both directions between the in-motion network and the *Network Gateway*, or *root node* in our case. In one direction, packets reach the infrastructure network through access nodes (APs), which forward the traffic according to the shortest path to reach the *root node*. In the other direction, packets depart from the *root node* following the route back to the access node where the in-motion network is reachable.
This way, when assuming the proposed handover scheme described by the first part of this thesis (the *Spiderman Handover*), the worst case of the routes updating procedure (*Gratuitous ARP Loop*) will be when both routes (active and passive) cross each other at the *root node* of the backbone topology. Therefore, the longer the path from the access network to the root node is, the higher is the probability that ARP packets might suffer from delays on each hop. These delays may come from queueing effects, forwarding times, or even propagation delays when the link between two backbone nodes is several kilometers long. Hence, we define our first requirement as **to exhibit a low maximum number of hops to reach the root node** (or equivalently a large number of gateways if they are equally placed).
As discussed in the previous section, the access nodes have two possible paths to reach the backbone network. Thereby, when a backbone node fails, the access nodes near to the failure will reconfigure their routes to reach the infrastructure network through the next closest operative gateway (or backbone node). If the number of access nodes between contiguous gateways is large, this reconfiguration will have a significant impact on the

maximum number of hops to the root. Hence, we define as our second requirement **to exhibit a low number of access nodes between two contiguous gateways**.
In addition, a backbone failure might produce network partitions when no alternative paths to the root are available within the backbone topology. Therefore, we define as our third requirement **to exhibit as many paths to the root as possible within the backbone topology** in order to allow the routing protocol (see Section 3.4) to exploit the redundancy of the backbone network (if any).
Finally, we set a requirement on the deployment cost of the backbone network. This cost is related to the number of backbone nodes and the total length of links required to deploy it. Thus, we aim at having **a number of backbone nodes (or gateways) and a total length of links in such a way that their combined costs would be reasonable when attaining the previous requirements**.
In summary, we define four requirements: 1) a low number of hops from the access nodes to the root node; 2) a low number of access nodes between two contiguous gateways; 3) a resilient backbone topology, and 4) a reasonable deployment cost. We say resilient in terms of the ability of the network to cope with a failure. This ability is expressed by the number of alternative paths to reach the root node of the topology (disjoint or not disjoint) and the number of access nodes between gateways. The deployment cost is expressed in terms of the number of gateways (or backbone nodes) and the total length of links.

6.1.2 Problem Definition

The number of hops from the access network to the root node is composed by two parts: 1) the number of hops from any access node to its closest gateway (following the linear connectivity of the access network); and 2) the number of hops from the backbone node (associated to that closest gateway) to the root node. In case of a failure in a backbone node, the linear part of the path to the root is increased, since the gateway associated to the failed backbone node is "disconnected" from the backbone network. So, the only possible path is to continue through the access network until reaching the next closest gateway. Therefore, the design of the backbone network should find the interconnecting points (gateways) between the access and backbone networks in such a way to minimize the impact of a disconnected gateway. Notice when finding the gateways location, we are implicitly defining the size of the backbone network, but not the way they are interconnected. We are not interested in absolute design rules that might deliver a large number of gateways and/or a large total length of links. So, we discard in advance topologies such as complete graphs and star topologies. On the contrary, we are interested in determining the ability of a recursive design rule to provide a backbone network exhibiting a good trade-off between the stated requirements. For that, we reviewed the literature for designing rules that might be applicable to our context.
The Network Design Problem (NDP) aims at finding a sub-graph of a graph subject to side constraints. The generalized Network Design Problem [Feremans *et al.* 2003] tries to solve the NDP by means of defining a set of clusters of nodes and proposing a way

to connect their root nodes. Inspired by this idea, we apply this criteria to propose a design of a backbone topology by dividing the linear access network into clusters. Each cluster has one (or possible more) designated gateway nodes. We interconnect these gateway nodes in such a way that the resulting topology meet the stated requirements. Hence, when looking for a suitable network topology that could deliver the required redundancy, we realized that chordal like topologies, such as those presented in [Arden & Lee 1981, Beivide *et al.* 2003, Angskun *et al.* 2007] provide a good design rule to meet our expectations. In particular, the Hierarchical Chordal Ring Networks (HCRN) studied by [Kitani *et al.* 2004] and [Bermond *et al.* 2003] propose a design of a hierarchical network topology offering redundancy applicable to our train scenario. Therefore, our problem is reduced to study particular recursive chordal design rules (defined later on) for the backbone network.
In this chapter we aim at describing the resulting network properties of these constructions in terms of the size of the access network and level of recursion. We focus in particular on the maximum number of hops from the access network to the root node, the number of gateways (or backbone nodes), the total length of links required to interconnect these nodes, and the number of access nodes between two consecutive gateways. In addition, we study how a routing protocol based on a spanning tree algorithm (providing the shortest path to the root) operates over the resulting infrastructure network. We are interested in determining the number of blocked links (and their lengths) in order to provide a notion of the number of alternative paths (not disjoint) to the root. In fact, each blocked link defines a potential alternative path (with similar or higher cost) to the root. The larger the number of blocked links is, the larger is the number of options that the routing protocol has to find an alternative path to the root.

6.2 Topology Definitions

The **linear access network** is a directed path of n nodes, corresponding to the WiFi Access Points (APs). The *access nodes* are denoted n_i, $1 \leq i \leq n$, so, n_i is connected to n_{i-1} for $2 \leq i \leq n$ and to n_{i+1} for $1 \leq i \leq n-1$. The length of the path is denoted $\ell = n - 1$ (the real length is $c \cdot \ell$ when the APs are c meters apart). Some of the nodes n_i, $i \in I \subset 1, 2, 3, \ldots, n$ are connected to the backbone network. These nodes are named **gateways** and are denoted g_i (same index as n_i). The **backbone node** connected to g_i is denoted b_i (same index as g_i). The **backbone network** $\mathcal{B} = (B, L)$ has as vertex set the nodes $b_i \in B$. L denotes the set of links interconnecting the nodes in B. A **Linear Topology Network** (LTN) is the infrastructure network resulting from connecting the linear access network with the backbone network. The LTN has a node denoted **root node**, usually at the "middle" of the topology, which connects the LTN to external networks.
Figure 6.2 shows an example of a LTN for $n = 15$. We have 7 gateways nodes: $\{g_1, g_4, g_6, g_8, g_{10}, g_{12}, g_{15}\}$. Here $I = \{1, 4, 6, 8, 10, 12, 15\}$ and $B = \{b_1, b_4, b_6, b_8, b_{10}, b_{12}, b_{15}\}$. The root node is the node b_8. The links are those indicated in the figure in solid blue and dashed red lines between backbone nodes. Active links are

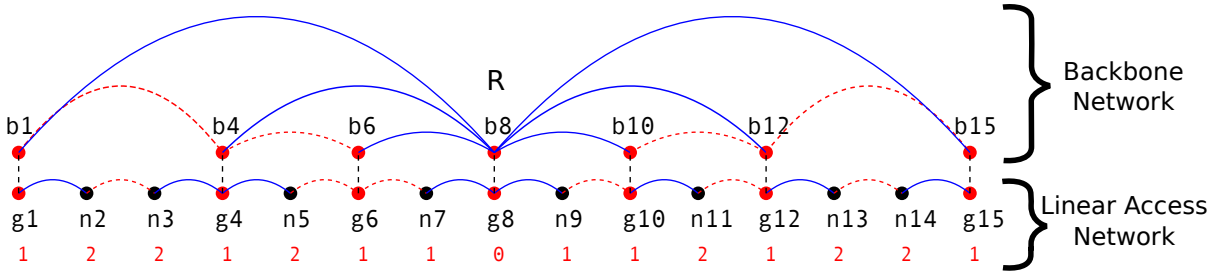


Figure 6.2: Linear Topology Network formal representation.

those who participate to the shortest path to the root (in blue) defined by the routing protocol. In this particular case they are of the form $\{R, b_i\}$, $i = \{1, 4, 6, 10, 12, 15\}$. The disabled links are those blocked by the routing protocol to avoid forming a loop. They are depicted as red dashed lines.

The **length of the link** (b_i, b_j) corresponds to the number of hops in the access network between the gateways g_i and g_j and it is denoted as $\ell(b_i, b_j) = j - i$. In the figure for example, the length of the link between b_1 and b_8 is 7. The distance from a node b_i to the root R is the **distance to the root** (in number of hops) to reach R by using the shortest path defined by the routing protocol. In fact this path only uses active links. A path from an access node n_i to R consists in the path from n_i to one of the two closest gateways g_j (through the linear access network), plus the path from the associated node b_j to R. It corresponds to the minimum number of links between n_i and R in the LTN (where g_j and b_j are considered as forming a unique node). In the figure, the distance to the root from each access node is depicted below each node's label (in red). For example, the distance from n_3 to R is 2 and it consists in the path $n_3 \rightarrow \{g_4 \cup b_4\} \rightarrow R$. The **distance between gateways** is the number of access nodes between two contiguous gateways g_i and g_j. In other words, it is the number of hops in the access network between contiguous gateways. We name this metric as *the number of access nodes between gateways* to avoid any confusion with the distance to the root node of the network. In the figure, the distance between g_1 and g_4 is 3; and the distance between g_6 and g_8 is 2.

In what follows, we consider two specific chordal topologies which are constructed in a recursive manner. We start with the whole path of n access nodes with its root node (usually in the middle) at level 1. The procedure creates new gateways and new links connecting these gateways (in fact, the backbone nodes defined by the gateways), splitting the initial path into sub-paths called **segments**. The length of the segment is the length of the sub-path $[n_i, n_j]$ for $j > i$. Then, at level 2, we apply the same procedure to each created segment and so on until we reach a boundary condition. More generally, the procedure is applied on those segments created at the level $k - 1$, defining new gateways, links and segments at level k when possible.

6.2.1 Chordal-2 Topology

Topologies coming from the Chordal family are mainly based on adding chords (or shortcuts) to a ring of nodes. The objective of adding these chords is two-fold: to reduce the number of hops to the root and, to provide several paths between any pair of nodes. In this section we study a particular chordal topology applied to a path instead of a ring. It is obtained recursively by splitting each segment into two new segments a certain number of times. More precisely, we start at level 0 with the segment $[n_1, n_n]$. We define two gateways g_1 and g_n and we suppose the root R is at the middle of the segment $[g_1, g_n]$. At step 1, we split the segment $[g_1, g_n]$ into the two segments $[g_1, R]$ and $[R, g_n]$. We classify the links (g_1, R) and (R, g_n) as links of layer 1. More generally at the step k, we split a segment $[u, v]$ created at the step $k-1$ into two segments $[u, w]$ and $[w, v]$ where w is at the "middle" of $[u, v]$. We define a new gateway in w and create at layer k the two links (u, w) and (w, v). Let us precise where is located the "middle". If the length of $[u, v]$ is even, the middle w is unique and has the property of $\ell(u, w) = \ell(w, v) = \frac{\ell(u,v)}{2}$. If the length of $[u, v]$ is odd, we have two choices for w:

1. $\ell(u, w) = \lceil \frac{\ell(u,v)}{2} \rceil$ and $\ell(w, v) = \lfloor \frac{\ell(u,v)}{2} \rfloor$ (middle-right)
2. $\ell(u, w) = \lfloor \frac{\ell(u,v)}{2} \rfloor$ and $\ell(w, v) = \lceil \frac{\ell(u,v)}{2} \rceil$ (middle-left)

If $d(u, R) = d(v, R)$ both choices give the same result for the distribution of distances from the access nodes to the root. But if $d(u, R) < d(v, R)$, the first choice gives a better distribution of distances, resulting in some cases in a lower maximum distance to the root. Let us consider an example with $\ell(u, v) = 7$ and $d(u, R) = 1$ and $d(v, R) = 2$. Figure 6.3 shows both cases, indicating the distribution of distances when we choose $\ell(u, w) = 4$ ($\ell(w, v) = 3$), case 6.3a compared to the opposite option $\ell(u, w) = 3$ ($\ell(w, v) = 4$), case 6.3b.

(a) $\lceil \frac{\ell(u,v)}{2} \rceil$, $\lfloor \frac{\ell(u,v)}{2} \rfloor$ segments length.

(b) $\lfloor \frac{\ell(u,v)}{2} \rfloor$, $\lceil \frac{\ell(u,v)}{2} \rceil$ segments length.

Figure 6.3: Cases for segments length when $\ell(u, v)$ is odd for Chordal-2.

At some point if the length of a segment is 1, we can not split it. If the length of a segment is 2, we can apply the procedure, but creating two links in the backbone network of length 1, which duplicates the links in the access network, not producing any gain in terms of distance. Therefore, we will not apply the procedure to segments of length 2. For segments of length 3, the procedure creates a link of length 2 and a link of length 1. Let us consider a segment of four nodes : $\{u, w', w, v\}$ (u and v are connected by a

link). If $d(u,R) = d(v,R)$, the two options of splitting the segment does not change the distances of w' and w. But, if $d(u,R) < d(v,R)$, the links (u,w) and (w,v) reduces the distance of w by 1, as now $d(w,R) = d(u,R) + 1$ to be compared with the distance before $d(w,R) = d(v,R) + 1$. Furthermore, the existence of the links (u,w) and (w,v) increases the redundancy of the network without duplicating (at least not completely) the linear access network connectivity. So, we decide to split segments of length 3 and more. For simplicity, when splitting a segment of length 3 when $d(u,R) = d(v,R)$, we split it in (u,w) and (w,v).
Considering all the elements discussed before, we formally define this topology as the Chordal-2 topology with two parameters: the topology size n and the recursion level k, or in short, $C_2(n,k)$. Its design rule is defined as follows:

Definition 6.2.1. Chordal-2 Design Rule: *We start for $k = 1$ with the segment $[g_1, R]$ and $[R, g_n]$ when $\ell(g_1, R) = \lceil \frac{n-1}{2} \rceil$ and $\ell(R, g_n) = \lfloor \frac{n-1}{2} \rfloor$. Then, at step k for any segment $[u,v]$ defined at at step $k-1$ with u and v two consecutive gateways and of length $\ell(u,v) \geq 3$, we create a new gateway w, two links (u,w) and (w,v) in the backbone network and two segments $[u,w]$ and $[w,v]$ where $\ell(u,w) = \ell(w,v) = \frac{\ell(u,v)}{2}$ if $\ell(u,v)$ is even; and $\ell(u,w) = \lceil \frac{\ell(u,v)}{2} \rceil$, $\ell(w,v) = \lfloor \frac{\ell(u,v)}{2} \rfloor$ if $\ell(u,v)$ is odd and $d(u,R) \leq d(v,R)$.*

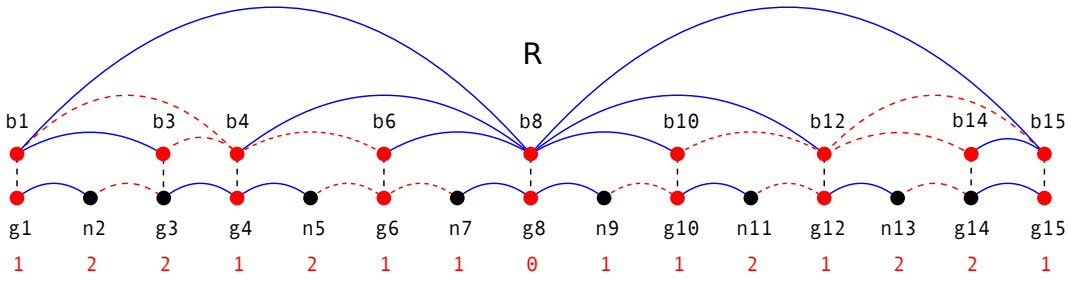


Figure 6.4: Chordal-2 for $n = 15, k = 3$, or $C_2(15,3)$.

Illustrating how this design rule works, let us consider the example of $n = 15$ as Figure 6.4 shows. Observe for $k = 1$, two segments were defined: $[g_1, g_8]$ and $[g_8, g_{15}]$. The root node is g_8. Then, for $k = 2$, we obtain four segments: $[g_1, g_4]$,$[g_4, g_8]$ and $[g_8, g_{12}]$,$[g_{12}, g_{15}]$. Notice the longer segments are placed starting from the lower distance node, in this case g_8. Then, for $k = 3$, we obtain from one side of the root node: $[g_1, g_3]$,$[g_3, g_4]$,$[g_4, g_6]$ and $[g_6, g_8]$. From the other side of the root node: $[g_8, g_{10}]$,$[g_{10}, g_{12}]$,$[g_{12}, g_{14}]$ and $[g_{14}, g_{15}]$. Note that for $[g_1, g_4]$ (resp. $[g_{12}, g_{15}]$) we put w in the middle right of the segment g_3 (resp. g_{14}) as the distances of the end nodes are equal. We can not apply the design rule for $k = 4$, since all segments in $k = 3$ have length less than 3. In this way, there exists a maximum achievable recursion level k_{max} that depends on the topology size n. In addition, it is also possible that the last step k_{max} can not be fully deployed for certain topology sizes, as it is the case when $n = 12$ (depicted in Figure 6.5). Observe that for the last segment $[g_{10}, g_{12}]$, the layer $k = 3$ cannot be deployed.

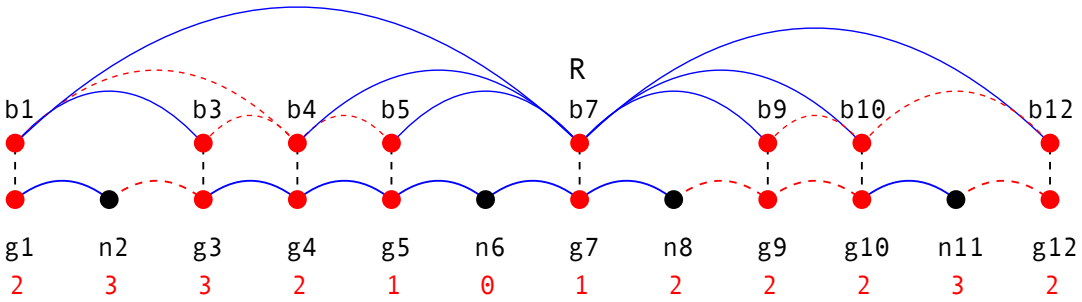


Figure 6.5: Chordal-2 for $n = 12, k = 3$ with the last layer incomplete.

For $n \leq 5$ ($\ell = 4$), only the layer 1 is deployed. For $n = 5$ we have two segments each of length 2. For $n = 6$ we have one segment of length 3, which can be divided at layer 2. For $n = 7$, the two segments of layer 1 can be divided and layer 2 is complete with two segments of length 1 and two of length 2. For $n = 8$ and 9, we have more segments of length 2, but the layer 3 is not yet possible. At $n = 10$, we can split the segment of length 3 and so creates a layer 3 which will be partially deployed until $n = 13$, where the layer 3 is fully deployed with 4 segments of length 1, and 4 segments of length 2. The layer 4 will be deployed only for $n > 17$ and so on. In general we have:

$$\begin{array}{rll} & n \leq 2^k + 1 & \Rightarrow \text{ the layer } k \text{ does not exist.} \\ 2^k + 1 < & n \leq 3 \cdot 2^{k-1} & \Rightarrow \text{ the layer } k \text{ is not fully deployed.} \\ 3 \cdot 2^{k-1} < & n & \Rightarrow \text{ the layer } k \text{ is complete.} \end{array} \tag{6.1}$$

In this way, considering the fact that the last layer k starts its deployment partially at $n = 2^k + 2$, the expression to determine the maximum recursion level allowed for a topology size n is:

$$k_{max}(n) = \lfloor log_2(n - 2) \rfloor \tag{6.2}$$

Figure 6.6 shows the maximum achievable recursion level for topology sizes up-to 5000 nodes. As final remark, as the layer k is not possible for $n \leq 2^k + 1$, there is no sense to provide $k > k_{max}(n)$. Therefore, from now on we bound the value of k to $k_{max}(n)$.

Proposition 6.2.1. *The number of gateways satisfies*

$$m(n,k) = \begin{cases} n - 2^{k-1} & , \quad 2^k + 1 < n \leq 3 \cdot 2^{k-1} \\ 2^k + 1 & , \quad 3 \cdot 2^{k-1} + 1 < n \end{cases} \tag{6.3}$$

Proof. The number of gateways is the number of segments plus 1. When the layer k is fully deployed ($n \geq 3 \cdot 2^{k-1} + 1$) we have exactly 2^k segments and so $2^k + 1$ gateways. The layer k starts to be deployed only for $n > 2^k + 1$. For $n = 2^k + 1$, the layer $(k - 1)$ contains

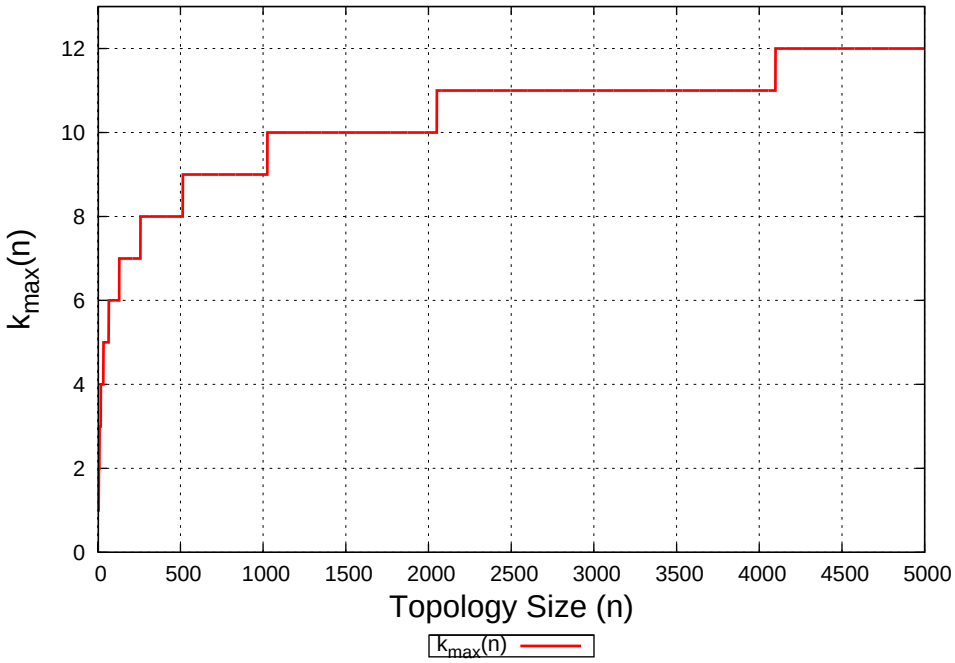


Figure 6.6: Maximal recursion level $k_{max}(n)$ for Chordal-2.

2^k segments each of length 2. Then, when n increases by 1, one segment has length 3 and it can be divided at the step k defining a new gateway, until $n = 3 \cdot 2^{k-1} + 1$, where the layer k is fully deployed. So, for $2^{k+1} < n \leq 3 \cdot 2^{k-1}$ the number of gateways is that of layer $(k-1) : 2^{k-1} + 1$ plus $(n - 2^k + 1)$, that is $n - 2^{k-1}$. □

Figure 6.7 shows the number of required gateways to deploy a topology of size n. Observe a linear growth up-to the value for n where the layer $k-1$ is fully deployed. Then, the number of gateways remain constant until the topology size allows the first segment at layer k, incrementing again the number of gateways according to n until the layer k becomes complete.
In addition, we describe the distance between two contiguous gateways in order to provide a notion of the resilience provided by Chordal-2 backbone networks. If the layer k exists, but is not fully deployed ($2^k + 1 < n \leq 3 \cdot 2^{k-1}$), most of the segments have length 2 with segments of length 1. Then, the layer k is complete and the segments of length 1 become of length 2 until $n = 2^{k+1} + 1$. Then the distance between contiguous gateways increases to 3 and so on. The average distance $\mathcal{D}_{avg}(n, k)$ can be easily computed by dividing the total length $n - 1$ by the number of segments $m(n, k) - 1$, yielding:

$$\mathcal{D}_{avg}(n, k) = \frac{n - 1}{m(n, k) - 1} \tag{6.4}$$

As the length of segments differs by at most 1 when deploying a layer k, the minimum and maximum distance between gateways are given by $\mathcal{D}_{min}(n, k) = \lfloor \mathcal{D}_{avg}(n, k) \rfloor$ and $\mathcal{D}_{max}(n, k) = \lceil \mathcal{D}_{avg}(n, k) \rceil$.

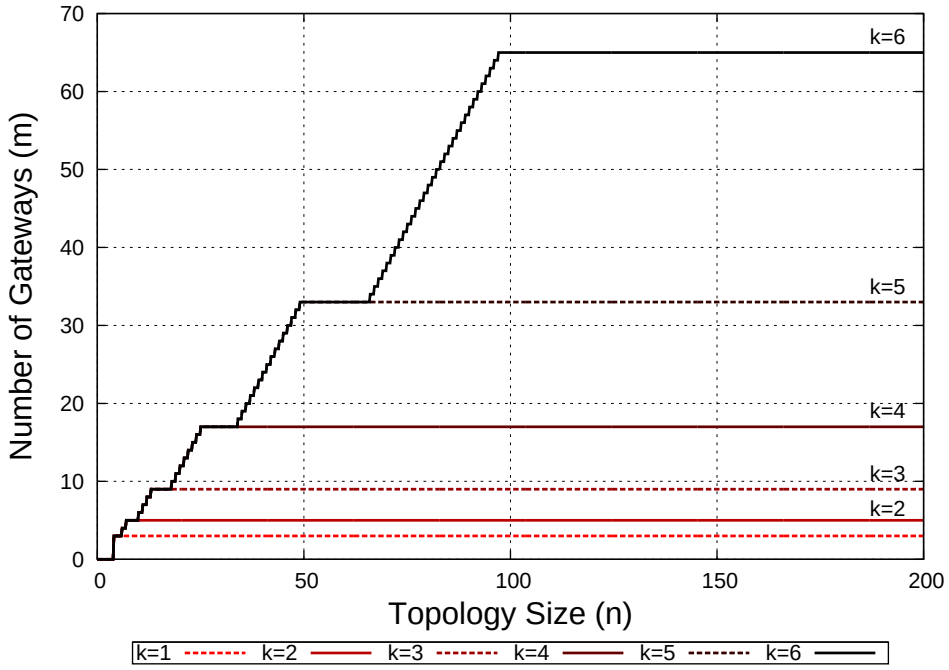


Figure 6.7: Number of gateways $m(n,k)$ for Chordal-2

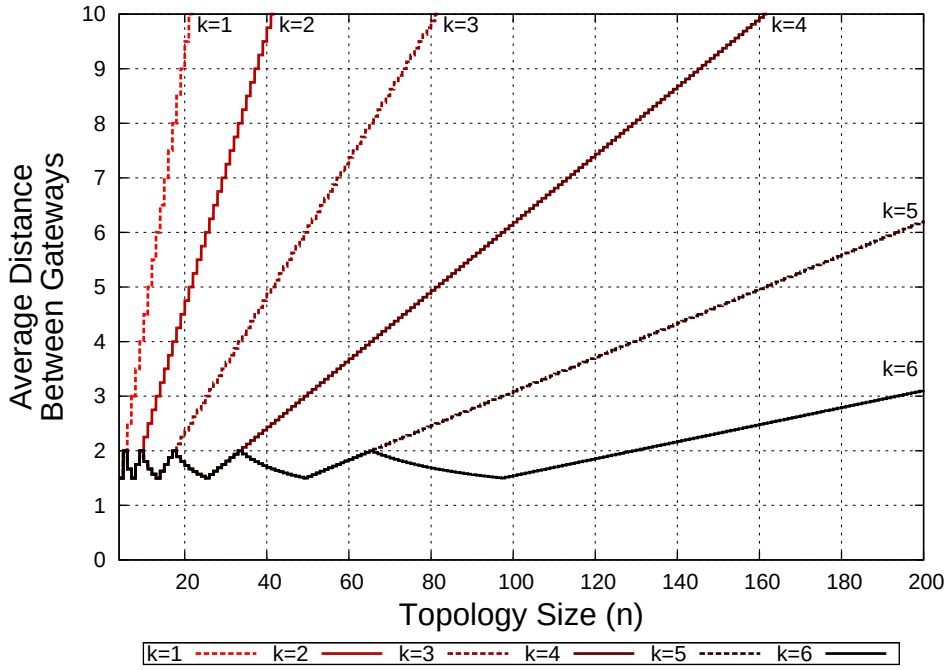


Figure 6.8: Average distance between contiguous gateways $\mathcal{D}_{avg}(n,k)$ for Chordal-2.

Proposition 6.2.2. *The total length of links* $\mathcal{L}(n,k)$ *satisfies*

$$\mathcal{L}(n,k) = \begin{cases} (k-1)(n-1) + 3(n-2^k-1) & , \quad 2^k+1 < n \leq 3\cdot 2^{k-1} \\ k(n-1) & , \quad 3\cdot 2^{k-1} < n \end{cases} \tag{6.5}$$

Proof. The total length of links $\mathcal{L}(n,k)$ is proportional to the number of layers, since each time a new layer k is complete, the total length of added links is $n-1$. Hence, the total length of links for k layers completely deployed is $k(n-1)$. When layer k is not fully deployed, note that when n increases by 1, a segment of length 3 is created at layer $k-1$, giving two segments of length 1 and 2 at layer k, therefore, incrementing the total length by 3. So, for $2^k+1 < n \leq 3\cdot 2^{k-1}$ the total length is $(k-1)(n-1)+3(n-2^k-1)$. □

Figure 6.9 depicts the total length of links for topologies $C_2(n,k)$, for $1 \leq k \leq 6$. For each k, we observe an irregular linear growth up-to the value for n where the layer k is fully deployed. Notice when building a topology $C_2(n,k)$ with $k < k_{max}(n)$, the total length of links increases linearly after the layer k is complete.

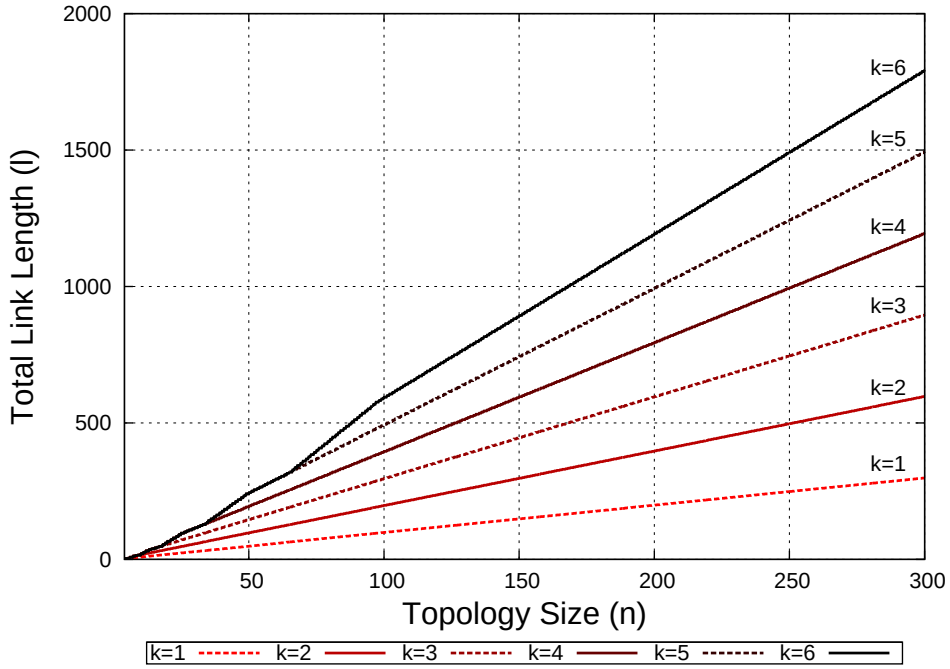


Figure 6.9: Total length of links for Chordal 2 topologies. $\mathcal{L}(n,k)$.

Now we will compute the maximum distance (in number of hops) $d_{max}(n,k)$ of a node to the root node.

Theorem 6.2.1. *The maximum distance from a node to the root is given by:*
for k *odd,* $d_{max}(n,k) = \lfloor \frac{3n-7\cdot 2^k-4}{3\cdot 2^{k+1}} \rfloor + \frac{k+1}{2} + 1$.
for k *even,* $d_{max}(n,k) = \lfloor \frac{3n-5\cdot 2^k-4}{3\cdot 2^{k+1}} \rfloor + \frac{k}{2} + 1$.

Proof. Let us denote $d_{max}^{(i,j)}(\lambda,\kappa)$ the maximum distance to the root of a node in a segment $[A,B]$ of length λ divided recursively κ times and such that $d(A,R)=i$ and $d(B,R)=j$. For a node v within the segment $[A,B]$, we have:

$$d(v,R) = min\{d(A,v)+i, d(v,B)+j\}. \tag{6.6}$$

Let $\ell = n-1$ and $\ell' = \lceil \frac{\ell}{2} \rceil$. With these definitions, we state 5 facts.

- Fact 1: $d_{max}(\ell,k) = d_{max}^{(0,1)}(\ell', k-1)$. Applying at the step 1, we get two segments which will be divided $(k-1)$ times with end nodes at distance 0 and 1. Their lengths are $\lfloor \frac{\ell}{2} \rfloor$ and $\lceil \frac{\ell}{2} \rceil = \ell'$.

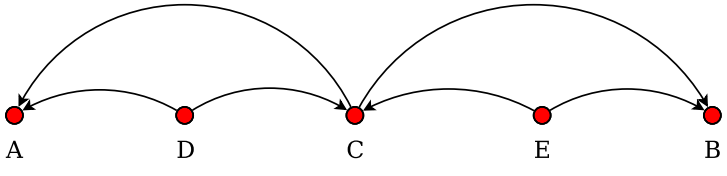
- Fact 2: $d_{max}^{(i,i)}(\lambda,0) = \lfloor \frac{\lambda}{2} \rfloor + i$. The maximum distance to the end node of the segment is $\lfloor \frac{\lambda}{2} \rfloor$. So, by 6.6, the maximum distance is $\leq \lfloor \frac{\lambda}{2} + i \rfloor$; equality being attained for a node at the middle of the segment.

- Fact 3: $d_{max}^{(i,i+1)}(\lambda,0) = \lfloor \frac{\lambda+1}{2} \rfloor + i$. Any node is either at distance $\leq \lfloor \frac{\lambda+1}{2} \rfloor$ from A or at distance $\leq \lfloor \frac{\lambda-1}{2} \rfloor$ from B; so by 6.6 the maximum distance to the root is $\lfloor \frac{\lambda+1}{2} \rfloor + i$, equality being attained for the node at the middle right of the segment.

- Fact 4: $d_{max}^{(i,i+1)}(\lambda,1) = d_{max}^{(i+1,i+1)}\left(\lfloor \frac{\lambda}{2} \rfloor, 0\right) = \lfloor \frac{\lambda}{4} \rfloor + i + 1$. Let be $[A,B]$ a segment of length λ with $d(A) = i$ and $d(B) = i+1$. The division of $[A,B]$ creates two segments $[A,C]$ of length $\lceil \frac{\lambda}{2} \rceil$ and $[C,B]$ of length $\lfloor \frac{\lambda}{2} \rfloor$ and $d(C,R) = i+1$. The distance is therefore $d_{max}^{(i,i+1)}(\lambda,1) = max\left(d_{max}^{(i,i+1)}(\lceil \frac{\lambda}{2} \rceil, 0); d_{max}^{(i+1,i+1)}(\lfloor \frac{\lambda}{2} \rfloor, 0)\right)$. By fact 3 we have: $d_{max}^{(i,i+1)}(\lceil \frac{\lambda}{2} \rceil, 0) = \lfloor \frac{1}{2}\left(\lceil \frac{\lambda}{2} + 1 \rceil\right) \rfloor + i = \lfloor \frac{\lambda+3}{4} \rfloor + i$. And by fact 2 we have: $d_{max}^{(i+1,i+1)}(\lfloor \frac{\lambda}{2} \rfloor, 0) = \lfloor \frac{1}{2}\left(\lfloor \frac{\lambda}{2} \rfloor\right) \rfloor + i + 1 = \lfloor \frac{\lambda}{4} \rfloor + i + 1 = \lfloor \frac{\lambda+4}{4} \rfloor + i$. Note that the second term is bigger or equal than the first one, proving this fact.

- Fact 5: $d_{max}^{(i,i+1)}(\lambda,\kappa) = d_{max}^{(i,i+1)}(\lfloor \frac{\lambda+2}{4} \rfloor, \kappa-2)$ for $\kappa \geq 2$. Dividing the segment $[A,B]$ with two steps of recursion, creates 4 segments $[A,D]$ of length $\lceil \frac{1}{2} \lceil \frac{\lambda}{2} \rceil \rceil = \lfloor \frac{\lambda+3}{4} \rfloor$; $[D,C]$ of length $\lfloor \frac{1}{2} \lceil \frac{\lambda}{2} \rceil \rfloor = \lfloor \frac{\lambda+1}{4} \rfloor$; $[C,E]$ of length $\lceil \frac{1}{2} \lfloor \frac{\lambda}{2} \rfloor \rceil = \lfloor \frac{\lambda+2}{4} \rfloor$ and $[E,B]$ of length $\lfloor \frac{1}{2} \lfloor \frac{\lambda}{2} \rfloor \rfloor = \lfloor \frac{\lambda}{4} \rfloor$. Furthermore, $d(D,R) = d(C,R) = i+1$; $d(E,R) = i+2$. See figure and table 6.10.

  We claim that the maximum distance to the root is attained at least for a node in $[C,E]$. There can be nodes within other segments with distances matching this value, but not greater than that. It is clear that the maximum distance in $[C,E]$ is greater than or equal to that in $[D,C]$ or $[E,B]$ as the length of $[C,E]$ and the distance to the end nodes are greater than or equal for the segment $[C,E]$ compared to the two others. For $[A,D]$, the length can be more than that of $[C,E]$ (case $\lambda = 1 \ mod \ 4$) and so the distance to reach one node of $[A,D]$ can be at most one more than the distance to reach one node of $[C,E]$, but $d(C,R) = d(A,R)+1$ and $d(E,R) = d(D,R)+1$ and so the maximum distance of a node $[A,D]$ to R is at most that of a node of $[C,E]$. Therefore, the fact 5 is true.



$[A,D]$	$[D,C]$	$[C,E]$	$[E,B]$	Total length λ
q	q	q	q	$4q$
$q+1$	q	q	q	$4q+1$
$q+1$	q	$q+1$	q	$4q+2$
$q+1$	$q+1$	$q+1$	q	$4q+3$

Figure 6.10: Chordal-2 segment length and maximum distance.

In the following, we use induction and some computations involving integer parts. In view of facts 1-5, the values of $d_{max}(\ell,k)$ depend on the congruences of ℓ *mod* 2^{k+1}. In fact we note that when the difference between the length of 2 segments is 2^{k+1} after k steps of recursion, we have segments with the same distance at end nodes, but of length differing by 1 and the maximum distance is increased by exactly 1. Furthermore, as by fact 5 every two steps of recursion the distance of the end nodes increases by 1, for $\ell = 2^{k+1}$, the distance to the root is $\lfloor \frac{k}{2} \rfloor$, so we search for a formula of the form $\lfloor \frac{\ell+\varphi(k)}{2^{k+1}} \rfloor + \lfloor \frac{k}{2} \rfloor$.
Case k odd. For $k = 1$, by fact 1 we have $d_{max}^{(0,1)}(\ell,1) = d_{max}^{(0,1)}(\ell',0)$ where $\ell' = \lceil \frac{\ell}{2} \rceil$. By fact 3 we have $d_{max}^{(0,1)}(\ell',0) = \lfloor \frac{\ell'+1}{2} \rfloor$. So, $d_{max}(\ell,1) = \lfloor \frac{1}{2} \left(\lceil \frac{\ell}{2} \rceil + 1 \right) \rfloor = \lfloor \frac{\ell+3}{4} \rfloor$. For $k > 1$, let $\varphi(x) = \lfloor \frac{x+2}{4} \rfloor$ and $\varphi^k(x) = \varphi(\varphi^{k-1}(x))$. By fact 5, $d_{max}^{(0,1)}(\ell',k-1) = d_{max}^{(1,2)}(\varphi(\ell'),k-3) = d_{max}^{(\frac{k-1}{2},\frac{k+1}{2})}(\varphi^{\frac{k-1}{2}}(\ell'),0)$ and by fact 3, $d_{max}^{(0,1)} = \lfloor \frac{\varphi^{\frac{k-1}{2}}(\ell')+1}{2} \rfloor + \frac{k-1}{2}$. Let be λ_{2h} be the integer such that if $\lambda = q \cdot 2^{2h+1} + r$, $\lfloor \frac{\varphi^h(\lambda)+1}{2} \rfloor = q$ for $r < \lambda_{2h}$ and $q+1$ for $r \geq \lambda_{2h}$. We have $\lambda_0 = 1$ and $\lambda_{2h} = 4\lambda_{2h-2} - 2$. So, $\lambda_{2h} = 4^k\lambda_0 - 2(1+4+\ldots+4^{h-1}) = 4^h - \frac{3}{2}(4^{h-1}) = \frac{4^h+2}{3}$. Therefore, applying with $h = \frac{k-1}{2}$, if $\ell' = q \cdot 2^k + r$, $d_{max}^{(0,1)}(\ell',k-1) = q + \frac{k-1}{2}$ for $r < \frac{2^{k-1}+2}{3}$ and $q+1+\frac{k-1}{2}$ for $r \geq \frac{2^{k-1}+2}{3}$. Finally, if $\ell = q_0 \cdot 2^{k+1} + r_0$, $d_{max}(l,k) = q_0 + \frac{k-1}{2}$ for $r_0 < \frac{2^k+1}{3}$ and $q_0 + 1 + \frac{k-1}{2}$ for $r_0 \geq \frac{2^{k+1}}{3}$. That is $d_{max}(\ell,k) = \lceil \frac{\ell - \frac{(2^k+1)}{3}}{2^{k+1}} \rceil + \frac{k-1}{2}$. which yields to a formula for the distance to the root when k odd:

$$d_{max}(\ell,k) = \lfloor \frac{3\ell + 5 \cdot 2^k - 1}{2^{k+1}} \rfloor + \frac{k-1}{2} \tag{6.7}$$

Case k even. For $k = 2$, by fact $d_{max}(\ell,2) = d_{max}^{(0,1)}(\ell',1)$. By fact 4 we have $d_{max}^{(0,1)}(\ell',1) = \lfloor \frac{\ell'}{4} \rfloor + 1$ and so, $d_{max}^{(0,1)}(\ell,2) = \lfloor \frac{\ell+1}{8} \rfloor + 1$. For $k > 2$, we have $d_{max}^{0,1}(\ell',k-1) = d_{max}^{(0,1)}(\varphi(\ell'),k-3) = d_{max}^{(\frac{k}{2}-1,\frac{k}{2})}(\varphi^{\frac{k}{2}-1}(\ell'),1) = \lfloor \frac{\varphi^{\frac{k}{2}-1}(\ell')}{4} \rfloor + \frac{k}{2}$. Let us define λ_{2h-1} as the integer such that if $\lambda = q \cdot 2^{2h} + r$, $\lfloor \frac{\varphi^{h-1}(\ell')}{4} \rfloor = q$ for $r < \lambda_{2h-1}$, and $q+1$ for $r \geq \lambda_{2h-1}$. Thus, $\lambda_1 = 4$ and $\lambda_{2h+1} = 4\lambda_{2h-3} - 2$, and we get $\lambda_{2h+1} = 4^h\lambda_1 - \frac{2}{3}(4^h - 1) = \frac{2^{2h+1}+2}{3}$. Finally, if $\ell = q_0 \cdot 2^{k-1} + r_0$, $d_{max}(\ell,k)$ is equal to $q_0 + \frac{k}{2}$ for $r_0 < \frac{2^k+1}{3}$ and it is equal to $q_0 + 1 + \frac{k}{2}$ for $r_0 \geq \frac{2^k+1}{3}$. That gives the formula for the maximum distance when k even:

$$d_{max}(\ell, k) = \lfloor \frac{3\ell + 2 \cdot 2^k - 1}{3 \cdot 2^{k+1}} \rfloor + \frac{k}{2} \tag{6.8}$$

The theorem 6.2.1 is obtained by replacing ℓ by $n-1$ in 6.7 and 6.8.

□

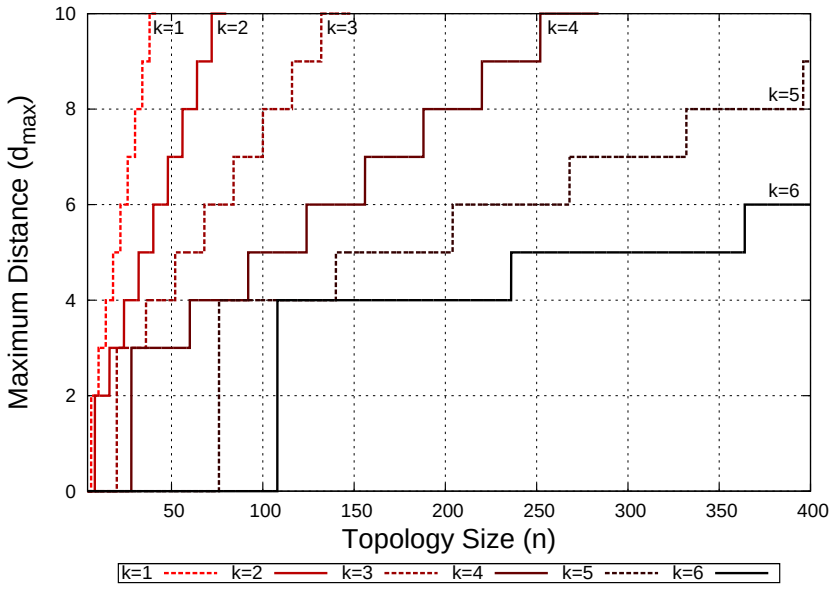


Figure 6.11: Chordal-2 maximal distance to the root node (d_{max}) for $k = k_{max}$.

Figure 6.11 shows $d_{max}(n, k)$ for $5 \leq n \leq 400$, for $1 \leq k \leq 6$. Observe each layer starts existing for $n > 3 \cdot 2^{k-1}$. For $n < 3 \cdot 2^{k-1}$, we only have the linear part of the expression. These results were verified by means of simulations for $3 \leq n \leq 5000$, validating the correctness of the presented expressions.

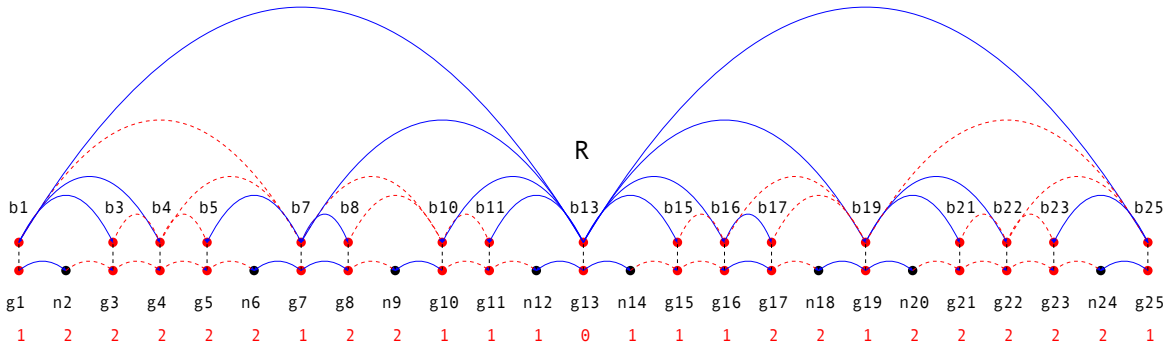


Figure 6.12: Chordal-2 Linear Network Topology for $n = 25, k = 4$

Figure 6.12 depicts an example of a Chordal-2 topology for $n = 25$ and $k = 3$. The distance to the root is specified below each node in red. Links in solid blue lines are the active links (determined by the routing protocol) and the links in dotted red lines are the blocked/disabled links. Regarding blocked links, it is interesting to analyze how many and how long are they in order to determine the proportion of the physical network that is "inactive" waiting to be used in case of failure. This measure can be useful to evaluate the strategy of link redundancy provided by this family in comparison to the canonical strategy of having a backup link for each active link.
Except at layer 1, we have by construction one link over two is blocked when we divide a segment. The total number of links is $|L(n,k)| = 2^{k+1} - 2$ and the total number of blocked links is $|L_{blocked}(n,k)| = \sum_{i=2}^{k} 2^{i-1} = 2^k - 2$. So, the ratio ρ_{number} is:

$$\rho_{number}(n,k) = \frac{|L(n,k)|}{|L_{blocked}|} = \frac{1}{2} - \frac{1}{2^{k+1}-2} \tag{6.9}$$

This ratio tends to $\frac{1}{2}$. Figure 6.13a shows that the ratio converges quickly to $\frac{1}{2}$.

Theorem 6.2.2. *The total length of blocked links $\mathcal{L}_{blocked}(n,k)$ satisfies*

$$(k-1)(\tfrac{n-1}{2}) - (2^{k-1}-1) \;\leq \mathcal{L}_{blocked}(n,k) \leq \;(k-1)(\tfrac{n-1}{2}) + (2^{k-1}-1)) \tag{6.10}$$

Proof. The total length of links is $\mathcal{L}(n,k) = k(n-1)$. If the length of a segment at layer k is λ, the length of the blocked links at layer $k+1$ is either $\lfloor \frac{\lambda}{2} \rfloor$ or $\lceil \frac{\lambda}{2} \rceil$ depending on which side of the segment the routing protocol blocks. If all the lengths of segments are even, which happens if n is a power of two, then at each layer the length of blocked links is the half of the total length, that is $\frac{n-1}{2}$. So, in that case, $\mathcal{L}_{blocked}(n,k) = (k-1)(\frac{n-1}{2})$. If n is not a power of two, the length of a blocked link obtained by the subdivision of a segment of length λ is $\lfloor \frac{\lambda}{2} \rfloor$ or $\lceil \frac{\lambda}{2} \rceil$, that is in the interval $\left[\frac{\lambda-1}{2}, \frac{\lambda+1}{2}\right]$. The sum of the length of segments created at layer $(k-1)$ is $n-1$ and the number of segments at layer $(k-1)$ is 2^{k-1}. Therefore, the total length of blocked links at layer k is between $\frac{n-1-2^{k-1}}{2}, \frac{n-1+2^{k-1}}{2}$. Note that the worst case appears when $\ell = 3 \cdot 2^{k-1}$ where at layer $(k-1)$ we have 2^{k-1} segments of length 3 given at layer k, 2^{k-1} segments of length 1 and 2^{k-1} segments of length 2. Summing up, we get that $(k-1)(\frac{n-1}{2}) - (1+2+\ldots+2^{k-2})) \leq \mathcal{L}_{blocked}(n,k) \leq (k-1)(\frac{n-1}{2}) + (1+2+\ldots+2^{k-2}))$, giving the inequality 6.10. □

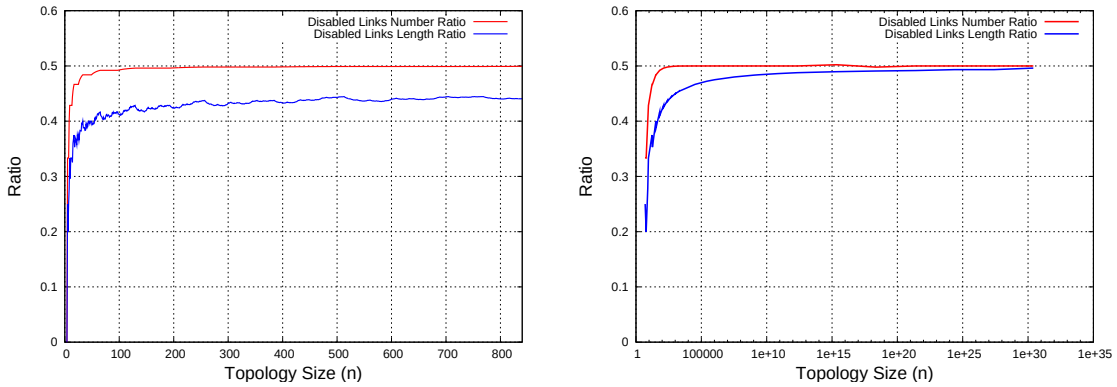
Corollary 6.2.1. *Let $\rho(n,k) = \frac{\mathcal{L}_{blocked}(n,k)}{\mathcal{L}(n,k)}$ the ratio of blocked links tends to $\frac{1}{2}$ when $k \to \infty$.*

Proof. Recall that $n - 1 \geq 3 \cdot 2^{k-1}$, so $\frac{2^{k-1}-1}{k(n-1)} \leq \frac{1}{3k}$ and:

$$\frac{1}{2} - \frac{5}{6k} \leq \rho(n,k) \leq \frac{1}{2} - \frac{1}{6k} \tag{6.11}$$

So, $\rho(n,k) \to \frac{1}{2}$ as $k \to \infty$.

□



(a) Ratios for n up-to 840.

(b) Ratios for $n = 3 \cdot 2^{k-1} + 1, k = 1..1000$

Figure 6.13: Chordal-2 disabled links ratios.

So, the length of blocked links represents almost the half of the total length. In average, the ratio behaves like $\frac{1}{2} - \frac{1}{2k}$. That is confirmed by means of simulations. Figure 6.13a shows the value of the ratio for $5 \leq n \leq 840$, for $n = 512$, $\rho = 0.445$ and $n = 1024$, $\rho = 0.45$. We also compute the ratio for the case $n = 3 \cdot 2^{k-1} + 1$ (which might be a worst case) for $1 \leq k \leq 1000$ and we see the convergence to $\frac{1}{2}$ (notice that the axis is in log-scale for n).

As final words on the Chordal-2 analysis, we realize that blocked links will be active mostly in case of a link failure rather than in case of a backbone node failure, since when a backbone node fails, its associated gateway (in the linear access network) loses its path to the root, learning a new path from its neighboring access nodes. This fail-over situation occurs due to the neighbor gateways "assimilate" the nodes served by the disconnected gateway. In this situation, there is no activation of any disabled link, unless the failed node causes a disconnection of another backbone node, which will activate a blocked link in order to learn a new path to the root node.

6.2.2 Chordal-JC^2 Topology

In this section, we study a particular construction of a chordal topology, which we named the Chordal-JC^2 topology. It defines basically a recursive procedure to select the gateway nodes, leaving their interconnection to an external function, namely the "kernel function". At each step k, we first divide uniformly a segment defined at step $k-1$ into s segments and we put in the "middle" (to be precised after) a gateway. Then, we interconnect the gateways according to a simple structure like paths, stars or binary trees. The resulting backbone topologies have initially the same set of gateways for a given n, but they differ in the way that gateways are interconnected. However, we further apply a restriction on the length of the links which makes vary the final number of gateways. Let us precise first the creation of gateways (see Figure 6.14).

Creation of Gateways

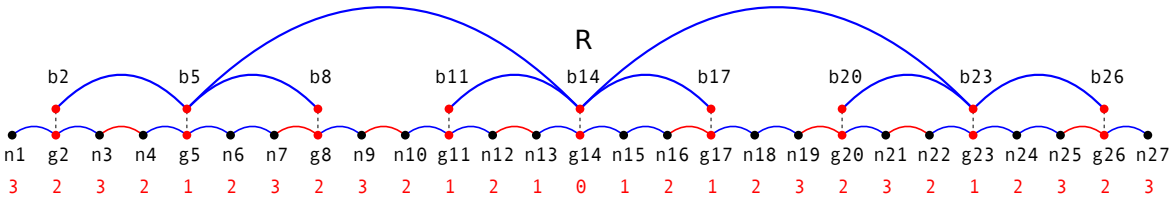


Figure 6.14: Example of a JC^2 topology with parameters $n = 27, s = 3, k = 2$ and a linear kernel.

At the beginning ($k = 0$) we consider the linear access network as a segment with nodes $n_1, n_2, \ldots, n_n$, that is of length $\ell = n - 1$ and with the root in the "middle". When n is odd (ℓ even) the root is located in the middle of the segment that is the node $n_{\frac{n+1}{2}}$. If n is even (ℓ odd), we choose by convention the root in the "middle-left" that is the node $n_{\frac{n}{2}}$. At the step k, we divide uniformly any segment of length λ created at step $k-1$ into s segments with equal (or almost equal) length. We have two cases: the case when $\lambda \geq s$ and the case when $\lambda < s$.

Case $\lambda \geq s$. If the length λ of the segment is congruent to r *modulo* s, that is $\lambda = qs + r$, $0 \leq r < s$ and $q = \lfloor \frac{\lambda}{s} \rfloor$ we create $s - r$ segments of length q and r segments of length $q + 1$. In particular, if $r = 0$ (λ multiple of s) the s segments have equal length $q = \frac{\lambda}{s}$.

Rule of placement of gateways.
We put a gateway in the middle of each new segment if the length of the segment is even and in the middle-left if the length is odd.
When λ is not a multiple of s, we have to decide what are the r segments of length $q + 1$. We want to do that in such a way the gateway of a segment defined at level $k - 1$ is also a gateway of a segment created at level k; that implies that the **number of segments** s **is odd** and its gateway is located in the middle segment $\frac{s+1}{2}$. That implies also that the r segments of length $q + 1$ are equally placed in both "halves" of the segment. Finally, in order to keep the maximum distance to the root as low as possible, we put the r longer segments as middle segments. More precisely, we use the following rule:

Rule of placement of segments obtained by division of a segment of length $\lambda = qs + r$**,** $0 \leq r < s$

1. if r is odd, we choose the first $\frac{s-r}{2}$ segments to be of length q, then the r next segments of length $q + 1$ and the last $\frac{s-r}{2}$ segments of length q.

2. if r is even, we choose

(a) if λ is even, the first $\frac{s-r-1}{2}$ segments of length q, then the r next segments of length $q+1$ and the last $\frac{s-r+1}{2}$ segments of length q.

(b) if λ is odd, the first $\frac{s-r+1}{2}$ segments are of length q, then the r next segments are of length $q+1$ and the last $\frac{s-r-1}{2}$ segments are of length q.

Proposition 6.2.3. *With this rule of placement, the gateway of* $\left(\frac{s+1}{2}\right)$ *th segment (the middle one) of the division of the segment* $[u, v]$ *is exactly the gateway of this segment.*

Proof. Let the nodes of the segment be labelled as j, $1 \leq j \leq \lambda+1$ with $\lambda = qs+r$, with $0 \leq r < s-1$. The gateway of the segment is at node $\frac{\lambda+1}{2}$ (λ odd) or at node $\frac{\lambda+2}{2}$ (λ even).

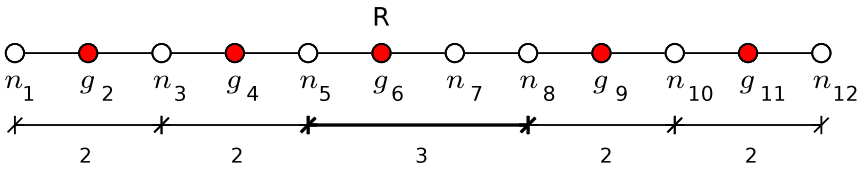
1. if r is odd, the gateway of the $\left(\frac{s+1}{2}\right)$th segment of the division is at node $1+q\left(\frac{s-r}{2}\right)+\lfloor\frac{(q+1)r}{2}\rfloor = \lfloor\frac{qs+r+2}{2}\rfloor = \lfloor\frac{\lambda+2}{2}\rfloor$ exactly the same node as the original gateway.

2. if r even and λ is even, the gateway of the $\left(\frac{s+1}{2}\right)$th segment is at the node $1+q\left(\frac{s-r-1}{2}\right)+\lfloor\frac{(q+1)(r+1)}{2}\rfloor = \lfloor\frac{qs+r+3}{2}\rfloor = \lfloor\frac{\lambda+2}{2}\rfloor$.

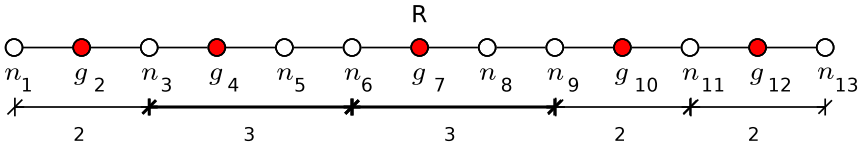
3. if r even and λ is odd, the gateway is at the node $1+q\left(\frac{s-r+1}{2}\right)+\frac{(q+1)(r-1)}{2} = \lfloor\frac{qs+r+1}{2}\rfloor = \lfloor\frac{\lambda+1}{2}\rfloor$.

□

Figure 6.15 illustrates the placement of the divided segment according to the rules previously described.



(a) Topology size $n = 12$ nodes. $s = 5$.



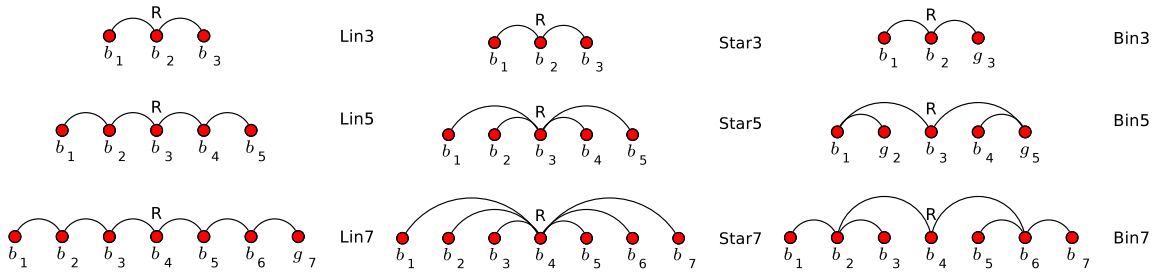
(b) Topology size $n = 13$ nodes. $s = 5$.

Figure 6.15: Balance example for 12 and 13 nodes with root middle-left

Case $\lambda < s$. We create exactly λ gateways, one in each node of the segment. The difference between the case for $\lambda \geq s$ is that we consider the last node of the segment as a gateway in order to decrease its distance to the root.

Kernel topologies

At each level we will interconnect the backbone nodes associated to each gateway created by the division of a segment $[u, v]$ according to a "kernel" function where the specific node r corresponds to the gateway associated to $[u, v]$. Figure 6.16 describes the types of kernels we will use: They are paths (6.16a), stars (6.16b) and binary trees (6.16c), for the values of $s = 3, 5, 7$.



(a) Linear Kernel (b) Star Kernels (c) Binary Tree Kernels

Figure 6.16: Kernel functions for $s = 3, 5$ and 7

Link length restriction

In general, we keep the topology as it is obtained. But, it might happen for λ small, that some of the links created are of length 1. In that case, they are duplicating the links in the linear access network. These duplicated links might lead to the backbone network to be connected one-to-one with the access network. Therefore, we delete these links and also the gateways which are not connected to the backbone network; in other words, we remove all the gateways and their associated backbone nodes with degree equal to 0. Figure 6.17 shows the partial topology for $s = 7$ and a star kernel. For example, if the length of the segment is 7, we have at the beginning the gateways $g_1, g_2, \ldots, g_6, g_7$ and the links (g_1, g_4),(g_2, g_4), (g_3, g_4), (g_4, g_5), (g_4, g_6) and (g_4, g_7). But the links (g_3, g_4) and (g_4, g_5) are length 1 and so deleted as well as their gateways g_3 and g_5. After the application of this restriction, if the number of gateways is less than s, we say that the kernel is **partially deployed**.

Hereafter, we describe the resulting topologies by applying each kernel function for $s = 3, 5, 7$. We focus on the the number of gateways (or backbone nodes), the maximum distance (in number of hops) from any access node to the root of the topology (the *Network Gateway*), the number of access nodes between contiguous gateways, the total length of links.

6.2.2.1 Linear Kernel

Let us consider a segment of length λ with its gateway in the "middle". This segment is divided into s sub-segments, defining a set B of backbone nodes b_j, $1 \leq j \leq s$ associated to the gateway of each sub-segment. The linear kernel creates the links (b_j, b_{j+1}), interconnecting all the nodes in B forming a path (or a line). Then, we apply the link

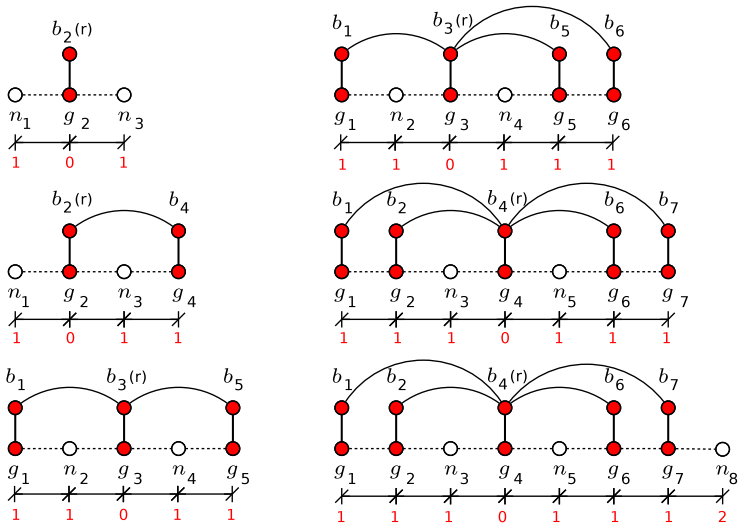


Figure 6.17: Partial deployment of an dtar kernel for $s = 7$ segments.

restriction, removing all the links of length 1 and the backbone nodes (and gateways) with degree 0. Figure 6.18 shows the linear kernel connectivity rule applied for a linear access network from 4 nodes up-to 11 nodes. For less than 4 nodes there is no links (and gateways) deployed due to the restriction on the length of links. Therefore, we define the partial deployment of this kernel for $\lambda < s + 2$. In the figure the access nodes considered as gateways (with their respective backbone nodes) are shown in red and the resulting sub-segments are depicted by extension lines under the access nodes.

From Figure 6.18, note that the last depicted topology (11 nodes) is showing the creation of a link at the step $k = 2$. Indeed, for $n \geq 5$, the middle segment (created at the step $k = 1$) allows the application of the kernel function for $k = 2$. However, the resulting links connecting the new backbone nodes do not meet the restriction of a length. Therefore, the step $k = 2$ is not deployed until $n \geq 11$, where the links created at the step $k = 2$ have the required minimum length.

Proposition 6.2.4. *The maximum recursion level for the linear kernel topologies satisfies*

$$k_{max}(n) = \lfloor log_s(n-2) \rfloor \tag{6.12}$$

Proof. For a segment of length $\lambda = qs + r$ created at the step $k - 1$ of recursion, the first link created at the step k appears when $q > s$. That means the level k begins when $\ell_0 = s \cdot s^{k-1} + 1$. For $\ell_0 = n - 1$, we obtain the proposition 6.12. □

Figure 6.19 shows the maximum recursion level for $3 \leq n \leq 5000$. Notice each layer is defined within $s^k + 1 < n \leq s^{k+1}$. However, the partial deployment produces segments progressively until $s - 1$ links are placed. Hence, from $s^k + 1 < n \leq (2s - 1)s^{k-1}$ the layer is not fully deployed. Therefore, we define for each layer k three intervals : the first one when the layer does not exist. The second one where the layer exists and the kernel is "not fully deployed" (meaning new gateways appear according to new links are added);

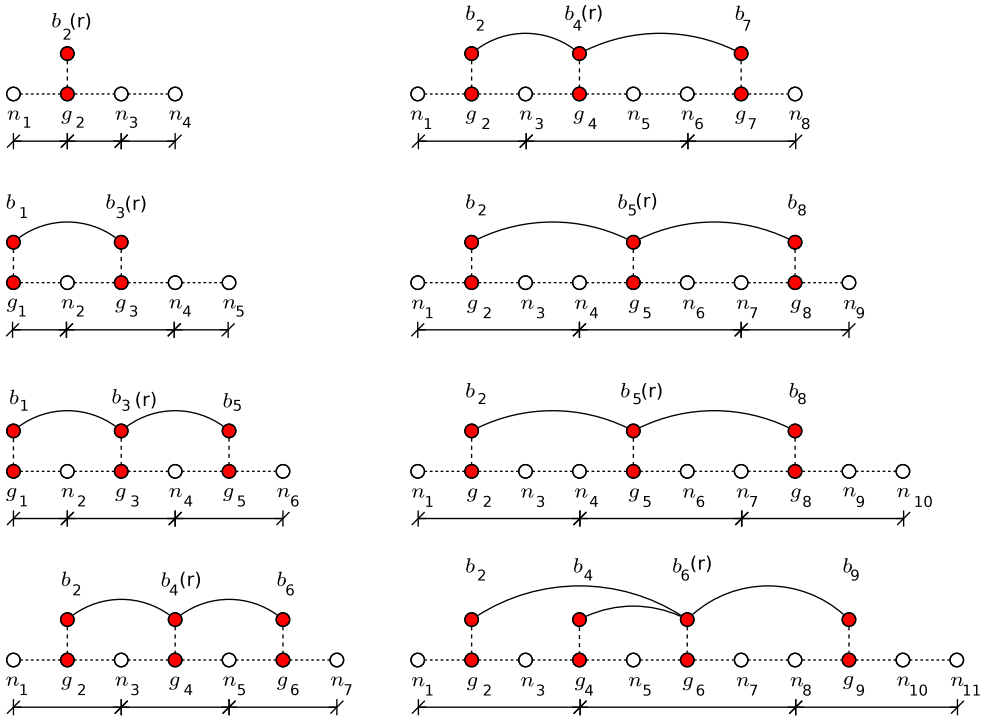


Figure 6.18: Linear kernel deployment for $s = 3$

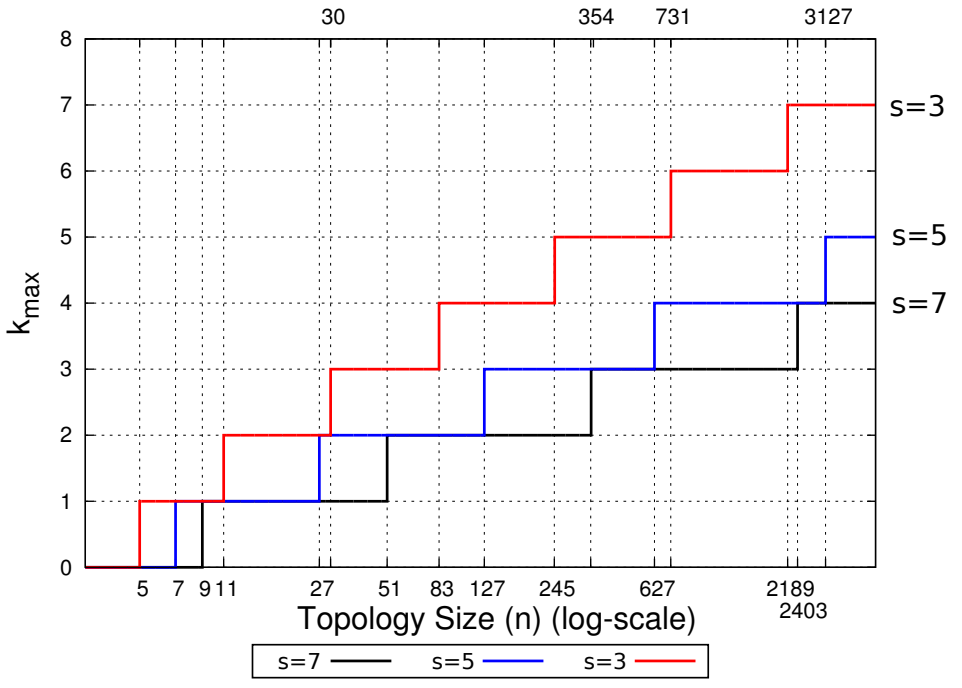


Figure 6.19: Maximum level of recursion for JC^2 – *lin* with $s = 3, 5, 7$.

and the third one where the kernel is "complete", meaning s gateways are defined within

each segment.

$$\begin{array}{rll} & n \leq s^k + 1 & \text{layer } k \text{ does not exist} \\ s^k + 1 < & n \leq (2s-1)s^{k-1} & \text{layer } k \text{ is not complete} \\ (2s-1)s^{k-1} < & n & \text{layer } k \text{ is complete} \end{array} \tag{6.13}$$

Proposition 6.2.5. *The number of gateways of a linear kernel topology satisfies*

$$m(n,k) = \begin{cases} n - (s-1)s^{k-1} - 1 & , \quad s^k + 1 < n \leq (2s-1)s^{k-1} \\ s^k & , \quad (2s-1)s^{k-1} < n \end{cases} \tag{6.14}$$

Proof. When a layer k is not complete, that is for $s^k + 1 < n \leq (2s-1)s^{k-1}$), the number of gateways is increased by 1 when n increases by 1. As for $n = s^k + 1$ the number of gateways is s^{k-1}, its value is $s^{k-1} + n - (s^k + 1) = n = (s-1)s^{k-1} - 1$. □

Figure 6.20 shows the number of gateways defined for the linear kernel with $s = 3, 5, 7$. Notice when $k < k_{max}(n)$, the number of gateways increases until the step k is complete. Then, it remains constant due to no further layers are deployed.

Describing the number of access nodes between contiguous gateways, notice that we can not use exactly the same approach that we used for Chordal-2 topologies, since the access nodes at the ends (n_1 and n_n) are not considered as gateways. So, we subtract the nodes "left" at both ends of the linear access network when dividing the length of the access network over the number of gateways. For this propose, we define a function $left(n,k)$ which gives the number of nodes left at both ends.

Proposition 6.2.6. *The average number of nodes between contiguous gateways of a linear kernel is given by*

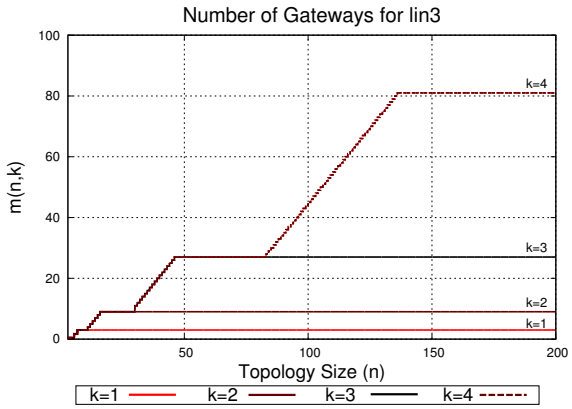
$$\mathcal{D}_{avg}(n,k) = \frac{(n-1) - left(n,k)}{m(n,k) - 1} \tag{6.15}$$

with the function left(n,k) defined as follows:

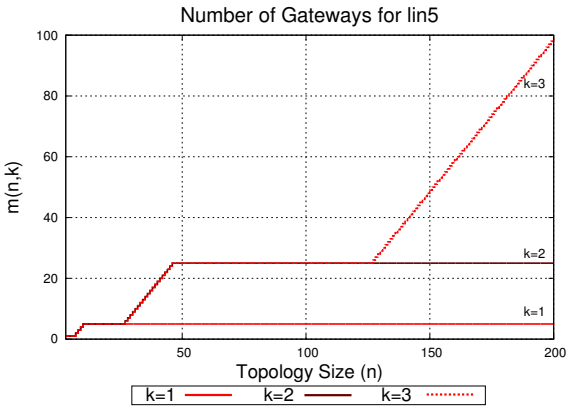
$$left(n,k) = \begin{cases} \lfloor \frac{2s^k - n + 1}{s^{k-1}} \rfloor & , \quad s^k + 1 \leq n < 2s^k + 1 \\ \lfloor \frac{n-1}{s^k} \rfloor & , \quad 2s^k + 1 \leq n \end{cases} \tag{6.16}$$

Proof. Let $\lambda = qs + r$ a segment created at the first step of recursion ($k = 1$). When $q = 1$ ($s \leq \lambda < 2s$), $left(\lambda + 1, 1) = 2s - \lambda$. When $q > 1$ ($\lambda \geq 2s$), $left(\lambda + 1, 1) = \lambda$. For a segment $\lambda_k = qs + r$ created by dividing k times an initial segment of length λ_0, $q = 1$ is obtained for $s^k \leq \lambda_0 < 2s^k$, so $left(\lambda_0 + 1, k) = 2s - \lambda_k$. For $q > 1$, $left(\lambda_0 + 1, k) = \lambda_k$. Note that $\lambda_k = \lfloor \frac{\lambda_0}{s^k} \rfloor$ and $\lambda_0 = n - 1$. Then, expressing $left(\lambda_0 + 1, k)$ in terms of n we obtain the proposition 6.2.6. □

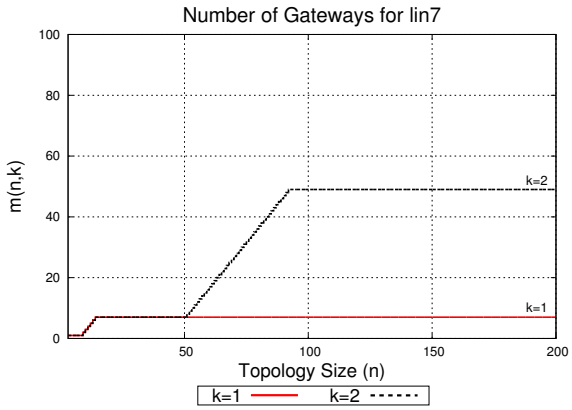
Figure 6.21 depicts the average number of access nodes between contiguous gateways for $s = 3, 5$ and 7 and $1 \leq k \leq 4$. The minimum and maximum number of nodes for a linear kernel topology of n nodes is well described by the *floor* and *ceil* of the average number



(a) linear kernel with $s = 3$



(b) linear kernel with $s = 5$



(c) linear kernel with $s = 7$

Figure 6.20: Number of gateways for $s = 3, 5$ and 7

of access nodes between gateways. However, on the contrary to Chordal-2 topologies, the minimum and maximum when deploying a layer $k_{max}(n)$ vary from 1 to s access nodes. This behaviour is explained by the effect of the restriction on the length of links when deploying the last layer $k_{max}(n)$. For $k < k_{max}(n)$, the number of access nodes between gateways is incremented linearly when $n \geq 2s^k + 1$

Noe let us compute the total length of links.

Proposition 6.2.7. *Let* $n - 1 = \ell_0 = q_0 s + r_0$, $0 \leq r_0 < s$.
If $q_0 \leq 2$, *then*

$$\mathcal{L}(n,k) = (s-1)q_0 + r_0 + (s - r_0)\mathcal{L}(q_0 + 1, k-1) + r_0 \cdot \mathcal{L}(q_0 + 2, k-1) \tag{6.17}$$

If $q_0 = 1$, *then*

$$\mathcal{L}(n,1) = 2(n - 1 - s)$$

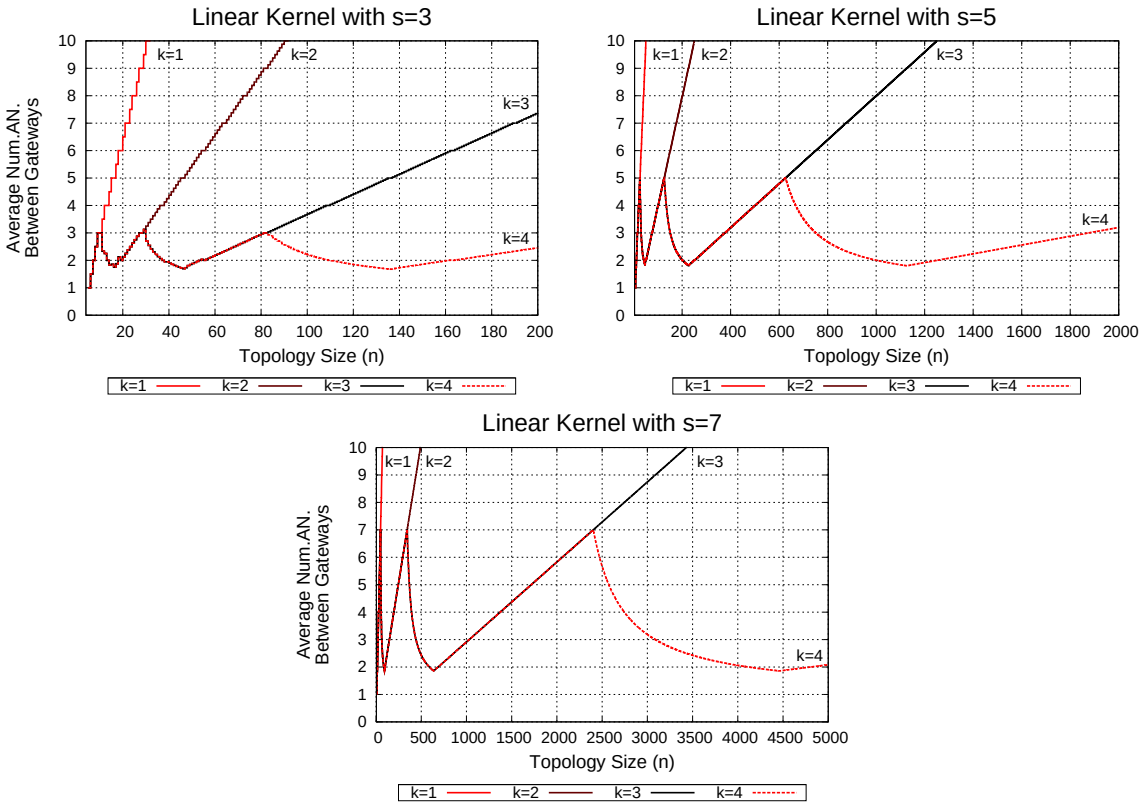


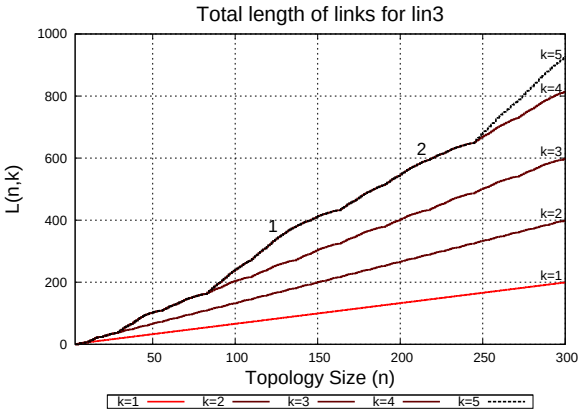
Figure 6.21: Average distance between gateways for linear kernels

Proof. When dividing completely a segment of length $\lambda = qs + r$, we create $(s-1)$ links interconnecting the gateways and the total length of links created is $(s-1)q + r$. If we apply the formula to the initial segment length $n-1 = \ell_0 = q_0 s + r_0$, we get for $\ell_0 > 2s-1$ $\mathcal{L}(n,1) = (s-1)q_0 + r_0$. Then, we apply recursively the formula to the $s - r_0$ segments of length q_0 and to the r_0 segments of length q_0+1, which have each to be divided $k-1$ times, getting the recurrence relation 6.17. For $n = s+1$, we have no links deployed. Then, for $s+2 \leq n \leq 2s-1$, when n is increased by 1, we create a new link of length 2, which give us the expression for $k = 1$. □

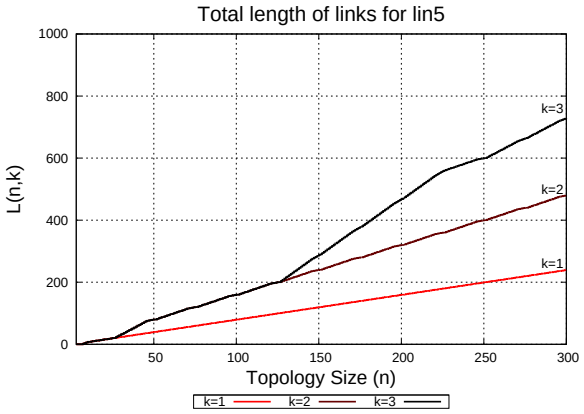
Figure 6.22 shows the total length of links to deploy a linear kernel topology for $s = 3, 5, 7$ considering $1 \leq k \leq k_{max}(n)$. Observe the irregular linear growth with two evident rates: the first one when the kernel is being deployed, and the second one when the kernel is fully deployed, but the sub-segments length do not allow the deployment of a new layer. In addition, when $k < k_{max}(n)$, the total length of links remains in the second growth rate.

Corollary 6.2.2. *If $n-1$ is multiple of s^k, the total length of links for a linear kernel topology is:*

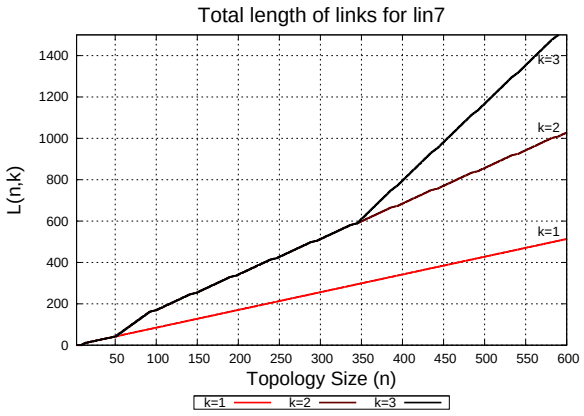
$$\mathcal{L}(n,k) = k\left(\frac{s-1}{s}\right)(n-1) \tag{6.18}$$



(a) s=3



(b) s=5



(c) s=7

Figure 6.22: Total length of links for linear kernels with $s = 3, 5, 7$ segments.

Proof. By induction on k. Let $n-1 = \ell_0 = ps^k$, $p \geq 2$. By proposition 6.2.7, $\mathcal{L}(n,1) = (s-1)ps^{k-1} = \left(\frac{s-1}{s}\right)(n-1)$. Suppose $\mathcal{L}(n,k) = k\left(\frac{s-1}{s}\right)(n-1)$. Then, by proposition 6.2.7 with $q_0 = ps^{k-1}$ and $r_0 = 0$, we get $\mathcal{L}(n,k+1) = (s-1)ps^{k-1} + sk\left(\frac{s-1}{s}\right)ps^{k-1} = \left(\frac{s-1}{s}\right)(k+1)ps^k$. □

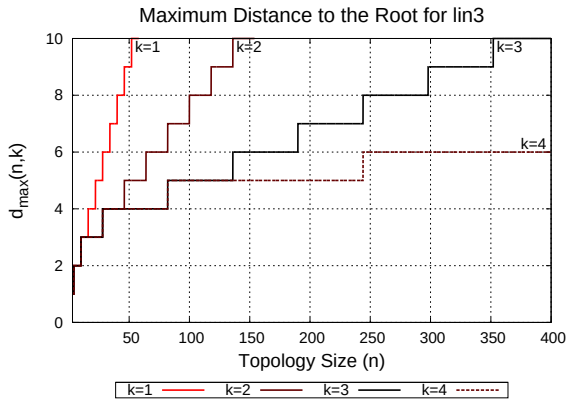
Theorem 6.2.3. *The maximum distance $d_{max}(n,k)$ from an access node to the root for a linear kernel topology satisfies*

$$d_{max}(n,k) = \left(\frac{s-1}{2}\right)k + \left\lceil \frac{n-s^k}{2s^k} \right\rceil \tag{6.19}$$

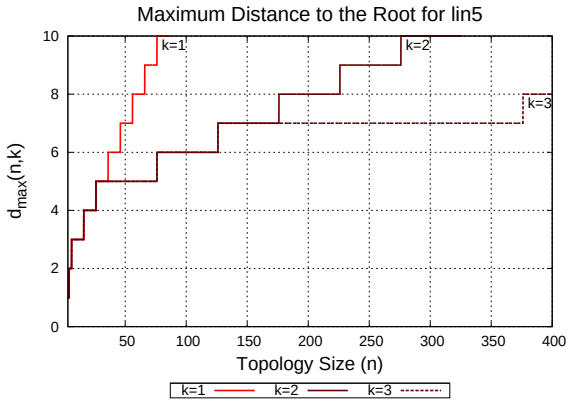
Proof. The maximum distance to the gateway associated to a segment of length λ is $\lceil \frac{\lambda-1}{2} \rceil$. Let us prove by induction that at level k the distance from a gateway to the root is at most $\left(\frac{s-1}{s}\right)k$, that is true for $k = 1$, as in a path with s nodes the distance to the middle

node is at most $\frac{s-1}{2}$ (with equality only for the end nodes). Then, gateways created at level k (from a segment defined at level $k-1$) are also at distance at most $\frac{s-1}{2}$ from the root of this segment and therefore, by induction hypothesis, the gateways are at most $\frac{s-1}{2} + \left(\frac{s-1}{2}\right)(k-1) = \left(\frac{s-1}{2}\right)(k)$ hops to the root, with equality only for the end segments. Let $n-1 = \ell_0 = qs^k + r$ with $r > 2$. At level k we have created segments of length q and $q+1$. The maximum distance is at most $\lceil \frac{q}{2} \rceil + \left(\frac{s-1}{2}\right)k$ except if q is even and we have a segment of length $q+1$ at the end of the path. If we have a segment of length $q+1$ in the middle, the distance from its gateway to the root is $< \left(\frac{s-1}{2}\right)k$. Furthermore, by the rule of placement of segments, the segments of length $q+1$ do not appear at the end of the path until $r = s-1$, so we always have $d_{max}(n,k) = \left(\frac{s-1}{2}\right)k + \lceil \frac{q}{2} \rceil$. As $q = \lfloor \frac{\ell_0}{s^k} \rfloor$, we get $\lceil \frac{q}{2} \rceil = \lceil \frac{\ell_0 - s^k - 1}{2s^k} \rceil = \lceil \frac{n - s^k}{2s^k} \rceil$.

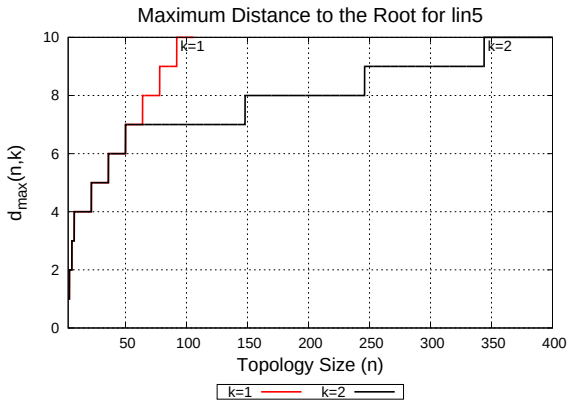
□



(a) Linear kernel with $s = 3$



(b) Linear kernel with $s = 5$



(c) Linear kernel with $s = 7$

Figure 6.23: Maximum distance to the root for $s = 3, 5$ and 7.

Figures 6.23 show the maximum distance for linear kernels for $s = 3, 5$ and 7. Observe for $k < k_{max}(n)$ the maximum distance to the root follows a different linear growth rate at steps of s^{k-1} nodes. These results has been validated by simulation for $3 \leq n \leq 5000$.

Finalizing the study of the linear kernels, we analyze the number and length of blocked links produced by routing protocol. We evaluate this by means of simulations for access network sizes up-to 2000 access nodes. We computed the blocked links in the same way that a BFS spanning tree algorithm does. Results pointed out that there is no blocked link within the backbone topology. Indeed, the only loop existing in these topologies are caused by the connectivity of the linear access network. Figure 6.24 shows an example for $n = 19, k = 2$ and $s = 3$.

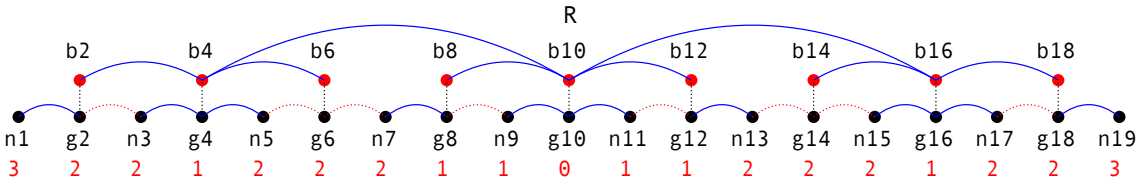


Figure 6.24: $\mathrm{JC}^2 - lin3$ topology for $n = 19$ and $k = 2$.

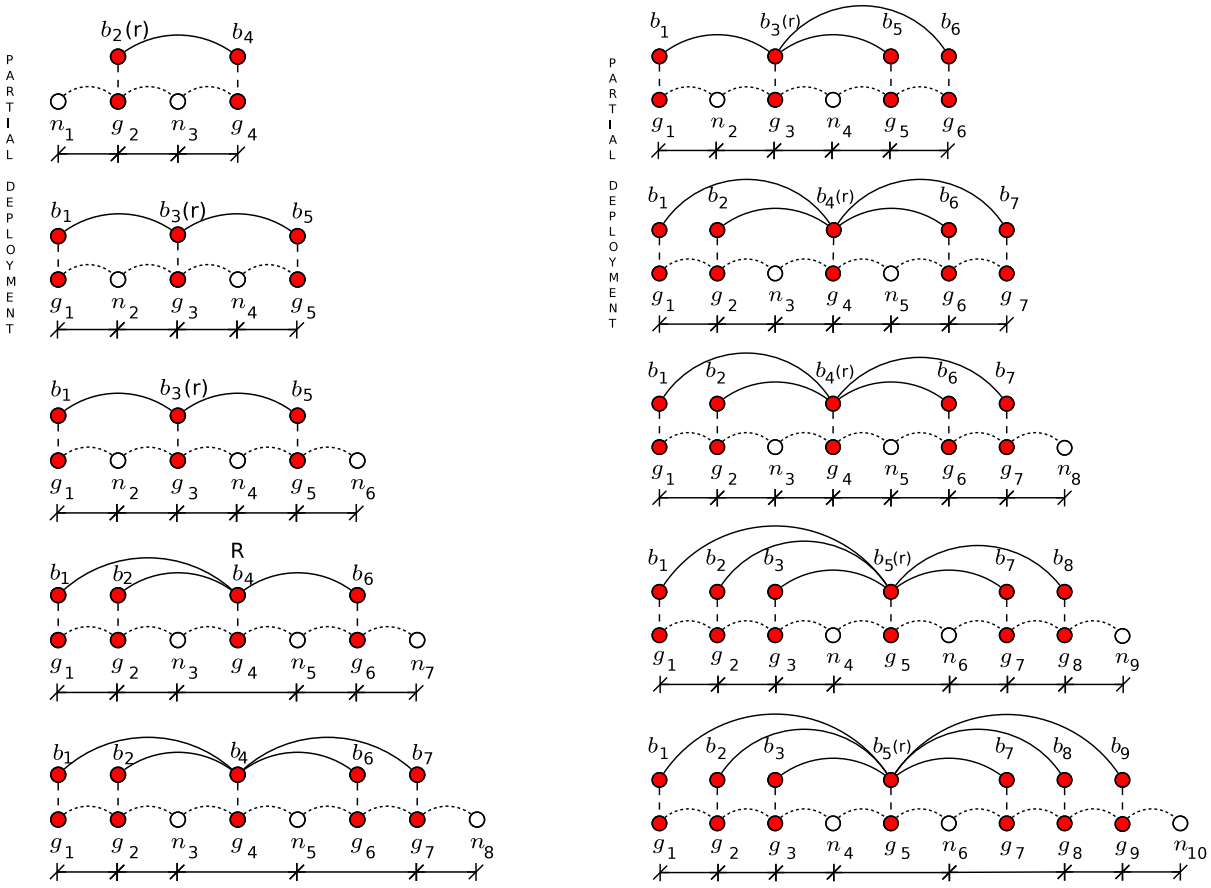
6.2.2.2 Star Kernel

Let us consider a segment of length $\lambda = qs + r$ created at layer k which defines a set $B = \{b_j\}$, $1 \leq j \leq s$ of backbone nodes associated to the new created gateways. The star kernel creates the links (b_j, b_r), with $j \neq r$. The node b_r corresponds to the node at the "middle" of B. When B has s nodes, $r = \frac{s+1}{2}$. When B has λ nodes (partial deployment) $r = \frac{\lambda+1}{2}$ for λ odd and $\frac{\lambda+2}{2}$ for λ even. Then, we apply the restriction on the length of links, removing all the links of length 1, and furthermore we remove all the backbone nodes (and their gateways) with degree 0. Note that for $s = 3$, as the star and a path are the same, the star kernel is the same as the linear kernel. Therefore, we analyze only the star kernel for $s = 5$ and $s = 7$. Figure 6.25 shows the deployment of the star kernel function. Both cases are depicted until $n = s + 3$ in order to illustrate the case when $q = 1, r = 2$ and s gateways (and backbone nodes) are created. The resulting layout mimics the connectivity of a star topology, but with their points arranged as a line.

Proposition 6.2.8. *The maximum recursion level for the star kernel topologies with $s = 5$ or $s = 7$ is given by*

$$k_{max}(n) = \lfloor log_s(\frac{n-2}{2}) \rfloor + 1 \tag{6.20}$$

Proof. A segment is divided only if its length $\lambda \geq 3$ and in that case by the rule of construction, we have a link to at least one gateway of length ≥ 2. So, we create a link at level k if $n - 1 = \ell_0 \geq 2s^{k-1} + 1$ (first value for which is created a segment of length 3). Therefore, the layer k exists if $k - 1 \leq log_s(\frac{n-2}{2})$, that gives the proposition 6.2.8. □



(a) Star kernel for $s = 5$

(b) Star kernel for $s = 7$

Figure 6.25: Star kernel deployment for $s = 5$ and $s = 7$.

In summary, the interval where a layer is defined are the following:

$$\begin{array}{rrll} & n & \leq 2s^k + 1 & \text{layer } k \text{ does not exist} \\ 2s^k + 1 < & n & \leq (s+2)s^{k-1} & \text{layer } k \text{ is not complete} \\ (s+2)s^{k-1} < & n & & \text{layer } k \text{ is complete} \end{array} \tag{6.21}$$

Figure 6.26 shows the maximum recursion level for $3 \leq n \leq 7000$. The access network size n is shown in logarithmic scale in order to improve the readability of the points where layers begin.

Theorem 6.2.4. *The number of gateways $m(n,k)$ for a star kernel topology is given by intervals as follows:*

$$m(n,k) = \begin{cases} n - (s^{k-1}+1) & , \quad 2s^{k-1} < n \leq 3s^{k-1} \\ 2s^{k-1} & , \quad 3s^{k-1} < n \leq 4s^{k-1} \\ n - 2s^{k-1} & , \quad 4s^{k-1} < n \leq s \cdot s^{k-1} \\ n - (2s^{k-1}+1) & , \quad s \cdot s^{k-1} < n \leq (s+2)s^{k-1} \\ s^k & , \quad (s+2)s^{k-1} < n \end{cases} \tag{6.22}$$

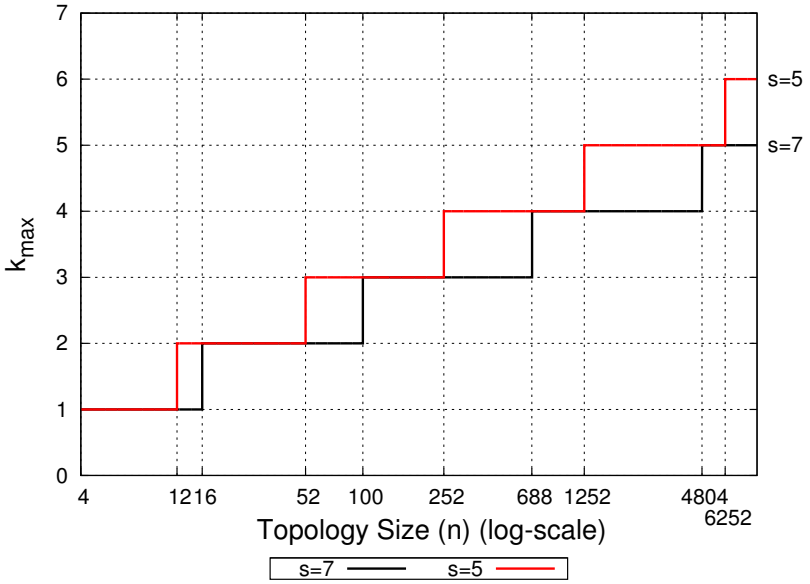


Figure 6.26: Maximum level of recursion for JC^2-star with $s = 5$ and $s = 7$.

We state 3 facts that can be verified by observing Figure 6.25. Let us consider a segment created at some step of recursion:

1. **Fact 1**: For $\lambda < s$ (partial deployment), there are $\lambda - 1$ gateways after deleting the links of length 1 and the isolated gateways. One of the defined gateways is at the right end of the segment and for $\lambda \geq 4$ there is another gateway at the left end.

2. **Fact 2**: For $\lambda = s$, there are $s - 2$ gateways and for $\lambda = s + 1$ there are $s - 1$ gateways. No gateway is defined at the right end of the segment.

3. **Fact 3**: For $\lambda \geq s + 2$, there are s gateways (and no gateway is defined at the right end).

With these facts, we construct a proof for the theorem 6.2.4.

Proof. Let us consider a segment of length λ created at the step $k - 1$ of recursion.

- For $2s^{k-1} \leq n < s^{k-1}$, segments created at layer $k - 1$ have length $\lambda \leq s$ and by Fact 1 we have $\lambda - 1$ gateways. Furthermore, those segments of length $3 < \lambda < s$ have their rightest gateway coincident with the lefter of the next segment. Thus, for $2s^{k-1} \leq n < 3s^{k-1}$, when n is increased by 1, a segment of length is divided creating a new gateway. As for $n - 1 = 2s^{k-1}$ we have s^{k-1} gateways at level $k - 1$, we get $n - (s^{k-1} + 1)$ gateways. For $3s^{k-1} \leq n < 4s^{k-1}$, when n is increased by 1, a "new" gateway is created by dividing a segment of length 4, but at the left end of this segment, therefore, coinciding with the right gateway of the preceding segment

(except of the first segment). So, for $n = 3s^{k-1}$, the number of gateways is $2s^{k-1}$ and remains constant until $4s^{k-1}$. Figure 6.27 depicts the case. For $4s^{k-1} \leq n < s \cdot s^{k-1}$, when n is increased by 1, we create a new gateway and so, we get $n - 2s^{k-1}$ gateways.

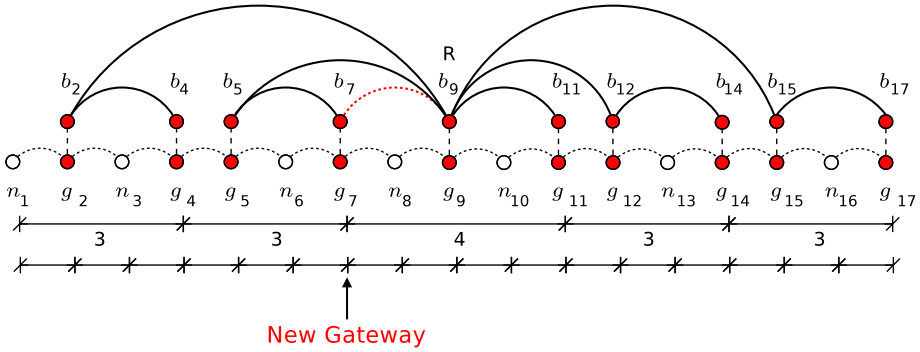


Figure 6.27: Star kernel for $s = 5$ with $k = 2$ illustrating coincident gateways

- For $s \cdot s^{k-1} \leq n < (s+2)s^{k-1}$, segments have length λ between s and $s+2$, and by Fact 2, we have $\lambda = 2$ gateways and further new gateways are not coinciding with gateways from neighboring segments. For $n = s^k + 1$ we have $(s-2)s^{k-1}$ gateways and when n is increased by 1, the number of gateways increases by 1, so we have $n - 1 - 2s^{k-1}$ gateways.

- Finally, if $n > (s+2)s^{k-1}$, the layer is complete. All segments at layer $k-1$ have $\lambda \geq s+2$ and when divided, we create $s-1$ gateways by Fact 1, and so the total number of gateways is s^k.

□

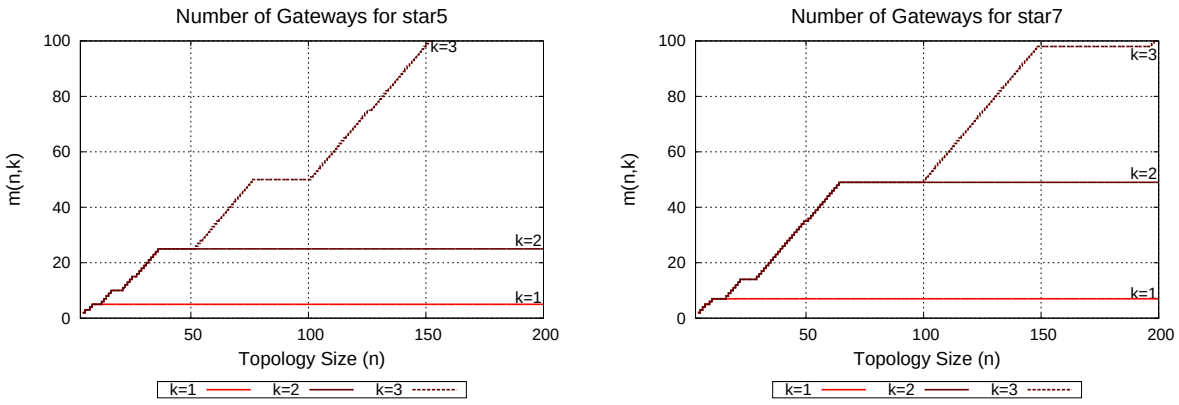
Figure 6.28 shows the total number of gateways for $s = 5$ and $s = 7$. This result has been verified by simulations for $3 \leq n \leq 5000$ access nodes.

Regarding the number of access nodes between contiguous gateways. We use the same approach used for linear kernels because this kernel also have nodes left at both ends of the access network for certain intervals of n.

Proposition 6.2.9. *The number of nodes left at the borders of the access network for a star kernel topology is given by*

$$left(n,k) = \begin{cases} \lfloor \frac{5s^{k-1}-n+1}{s^{k-1}} \rfloor & , \quad 2s^k + 1 < n \leq 5s^{k-1} \\ \lfloor \frac{n-1}{s^{k-1}} \rfloor & , \quad 5s^{k-1} < n \end{cases} \tag{6.23}$$

Proof. By simple inspection of Figure 6.25 and considering each depicted segment as a segment of length $n - 1 = \lambda_0 = q_0 s + r_0$ created at the first step of recursion ($k = 1$). For $\lambda_0 = 3$, $left(n,1) = 1$. For $\lambda_0 = 4$, $left(n,1) = 0$. For $\lambda_0 \geq 5$, $left(n,1) = 1$. For $\lambda_0 = 2s$ ($q_0 = 2, r_0 = 0$), $left(n,1) = 2$. And for For $\lambda_0 > 2s$ ($q_0 = 2, r_0 \geq 1$), this cycle



(a) Star Kernel with $s = 5$ (b) Star Kernel with $s = 7$

Figure 6.28: Number of Gateways for $s = 5$ and $s = 7$.

is repeated for $k = 2$. Let us consider an initial segment of length λ_0 divided k times. At layer k we have s^k sub-segments of length $\lambda_k = \lfloor \frac{\lambda_0}{s^k} \rfloor = qs + r$. At the layer $k + 1$ we have sub-segments of length q or $q + 1$.

- For $q = 1$, $left(\lambda_0, k) = 1$. Note that $q = 1$ for $s^k \leq \lambda_0 < 2s^k$.
- For $q = 2$, $left(\lambda_0, k) = 2$ since all the segments in $k = 1$ have their gateways at the middle of a segment of length 2, leaving one left node at each border of the access network. $q = 2$ for $2s^k \leq \lambda_0 < 3s^k$.
- For $q = 3$, $left(\lambda_0, k) = 1$ since a segment of length 3 can be divided into 3 sub-segments of length 1, allowing the creation of a new gateway at the right border of the topology (in Figure 6.25, $n = 4, s = 5$). $q = 3$ for $3s^k \leq \lambda_0 < 4s^k$.
- For $q = 4$, $left(\lambda_0, k) = 0$ since both ends nodes of the access network are considered as gateways. $q = 4$ for $4s^k \leq \lambda_0 < 5s^k$.
- For $5 \leq q < s$, $left(\lambda_0, k) = 1$, repeating the cycle for $k + 1$.

Note that for $2s^k \leq \lambda_0 < 5s^k$, the left nodes can be computed by $\lfloor 5 - \lambda_k \rfloor$, and for $\lambda_0 \leq 5s^k$, the number of nodes left is λ_k. Therefore, replacing λ_0 by $n - 1$, we have $\lambda_k = \lfloor \frac{n-1}{s^k} \rfloor$. Then, $left(n, k) = \lfloor 5 - \frac{n-1}{s^k} \rfloor$, which gives us the first interval of the proposition. The second interval is obtained in a similar way, completing the proposition. □

Furthermore, we use this new $left(n, k)$ formula in the equation 6.15 in order to compute the average number of access nodes between two gateways. Results are shown in Figure 6.29 for $1 \leq k \leq 4$. The minimum and maximum number of nodes for a star kernel topology of n nodes is well described by the *floor* and *ceil* of the average number of access nodes between gateways.

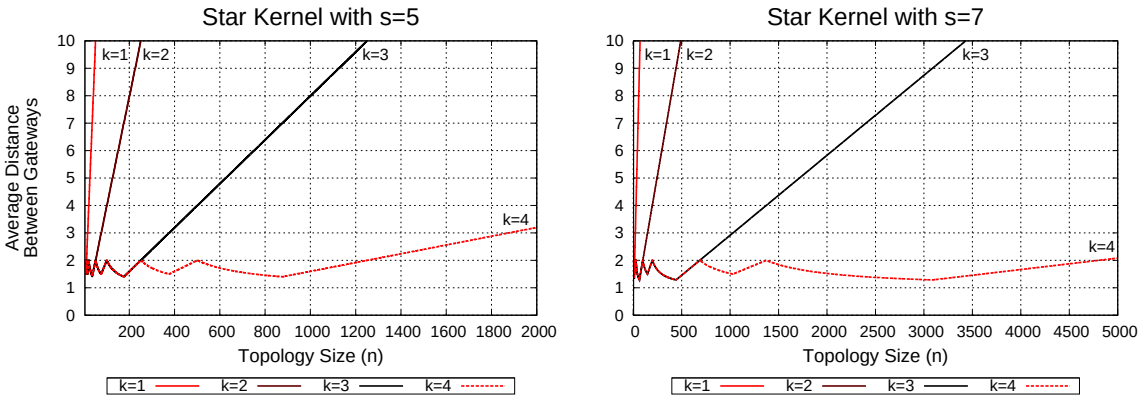


Figure 6.29: Average distance between gateways for star kernels

Note that, on the contrary to the linear kernel topologies, we obtain as much as 2 access nodes between contiguous gateways when deploying $k_{max}(n)$ layers.

Regarding the total length of links, we use the same procedure for linear kernels, but we change the way to compute the length of links for the layer k before to proceed with the recursive part of the equation.

Proposition 6.2.10. *Let* $n-1 = \ell_0 = q_0 s + r_0,\ 0 \leq r_0 < s$.

If $n \geq s+2$, *then*

$$\mathcal{L}(n,k) = \left(\frac{s^2-1}{s}\right) q_0 + \ell_s(r_0) + (s-r_0)\mathcal{L}(q_0+1, k-1) + r_0 \cdot \mathcal{L}(q_0+2, k-1) \quad (6.24)$$

where: for $s=5$ *we have* $\ell_5(0)=0,\ \ell_5(1)=2,\ \ell_5(2)=4,\ \ell_5(3)=5, \ell_5(4)=6$. *For* $s=7$ *we have* $\ell_7(0)=0,\ \ell_7(1)=3,\ \ell_7(2)=6,\ \ell_7(3)=8,\ \ell_7(4)=10,\ \ell_7(5)=11, \ell_7(6)=12$.

The initial values for the recursion are obtained by inspecting Figure 6.25 for $n < s+2$ and $n < s+3$ respectively as follows:

For $s=5$:

$$\begin{aligned} \mathcal{L}(n,1) &= 0 \quad \text{, for } n<4 \\ \mathcal{L}(4,1) &= 2 \\ \mathcal{L}(5,1) &= 4 \\ \mathcal{L}(6,1) &= 4 \\ \mathcal{L}(7,1) &= 7 \end{aligned} \quad (6.25)$$

For $s=7$:

$$\begin{aligned}
\mathcal{L}(n,1) &= 0 \quad \text{, for } n < 4 \\
\mathcal{L}(4,1) &= 2 \\
\mathcal{L}(5,1) &= 4 \\
\mathcal{L}(6,1) &= 7 \\
\mathcal{L}(7,1) &= 10 \\
\mathcal{L}(8,1) &= 10 \\
\mathcal{L}(9,1) &= 14
\end{aligned} \tag{6.26}$$

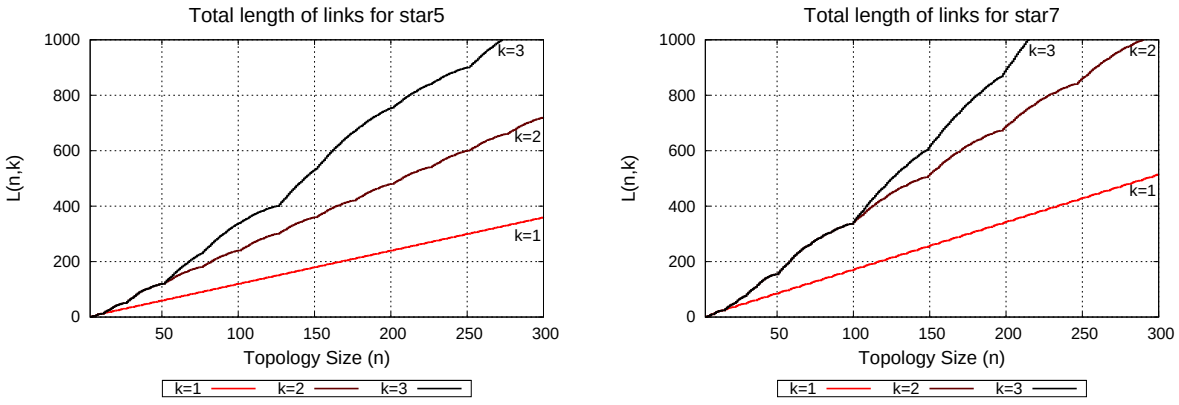
Proof. Similar to that of 6.2.7 except the length of the new links is not constant. The values are obtained for small n by Figure 6.25 and otherwise by looking at the number of segments with length $q+1$. For $r=0$, all the segments are of equal length q and the length of the links are for $s=5$ twice q and twice $2q$, so, altogether $6q$. For $s=7$, the length of the links are twice q, twice $2q$, twice $4q$, summing up $12q$. For $r>0$, as the rule of placement of segments assigns the length $q+1$ from the middle to the end of the segment, incrementing the total length of links deployed at layer k according to $\ell_s(r)$. □

Corollary 6.2.3. *If $n-1$ is multiple of s^k, the total length of links is given by:*

$$\mathcal{L}(n,k) = k\left(\frac{s^2-1}{4s}\right)(n-1) \tag{6.27}$$

Note that the this expression agrees with the linear kernel for $s=3$ since $\frac{s^2-1}{4s} = \frac{2}{3} = \frac{s-1}{s}$.

Figure 6.30 shows the total length of links for $3 \leq n \leq 300$. In addition, we verified these results by means of simulations for $3 \leq n \leq 5000$.



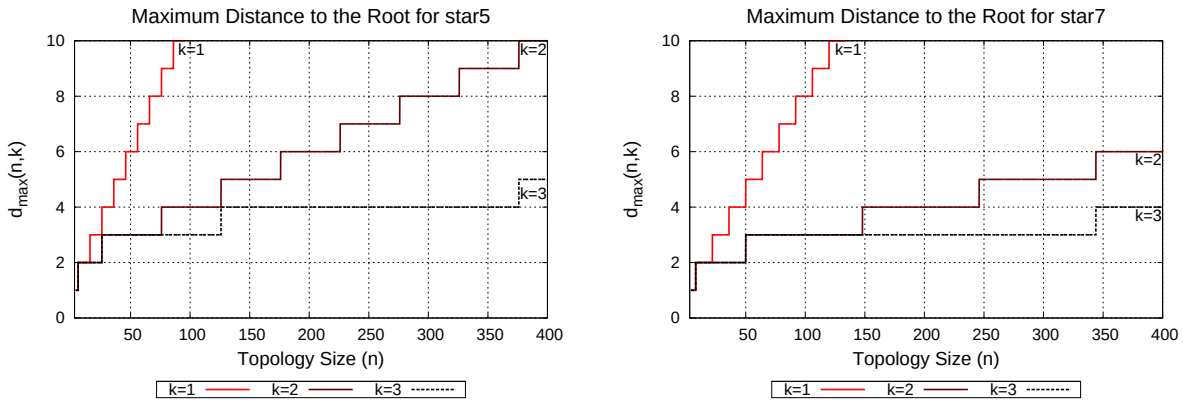
(a) Star kernel with $s=5$ (b) Star kernel with $s=7$

Figure 6.30: Total length of links for $s=5$ and $s=7$.

Theorem 6.2.5. *The maximum distance to the root for the star kernel with $s = 5$ or $s = 7$ is:*

$$d_{max}(n, s) = k + \lfloor \frac{n - s^k}{2s^k} \rfloor \tag{6.28}$$

Proof. Like in Theorem 6.2.3, the maximum distance of a node to the gateway is associated to a segment of length λ is $\lceil \frac{\lambda-1}{2} \rceil$. But now, due to the star structure, the distance of a gateway at layer k to the root of the topology is k. So, like in the proof of Theorem 6.2.3, we get the expression 6.28. □



(a) Star Kernel with $s = 5$ (b) Star Kernel with $s = 7$

Figure 6.31: Maximum distance to the root for the $JC^2 - star$ kernel with $s = 5$ and $s = 7$.

Figure 6.31 depicts how the maximum distance to the root is increased according to the topology size for $3 \leq n \leq 400$ nodes. This result was validated by simulations for $3 \leq n \leq 5000$.

Finally, we study the number and length of blocked links produced by the routing protocol. It is worth to mention that this kernel does produce loops in the backbone network (when both ends of a segment are defined as gateways). Therefore, there are blocked links when deploying it. Describing how many and how long they are, we simulate star kernel topologies computing the number of blocked links when applying a BFS spanning tree algorithm up-to 2000 nodes. We considered only the number of blocked links in the backbone network. Results showed that up-to $k_{max}(n)$ layers, the number of blocked links increases linearly for $3s^{k-1} < n \leq 4s^{k-1}$, For $n > 4s^{k-1}$, new links do not produce any new loops since they are placed "under" those producing them. Figure 6.32 depicts this situation for a topology with $s = 7$ (link (g_{11}, g_{13})). Thus, for $n > 4s^{k-1}$ the number of blocked links remains constant until shared gateways start getting apart ($n > (s-1)s^{k-1}$), decreasing linearly until $n = s \cdot s^{k-1}$, where there is no shared gateways among contiguous segments.

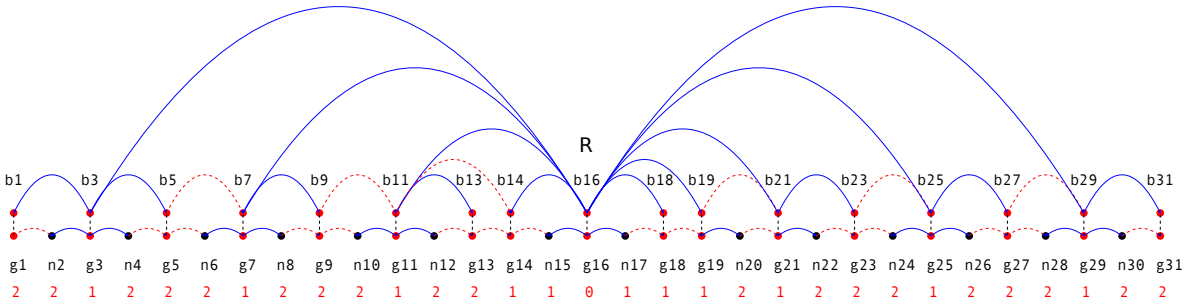


Figure 6.32: Star kernel for $s = 5$ with $k = 2$ illustrating loops in the backbone network

Proposition 6.2.11. *The number of blocked links for the star kernel topologies is obtained by*

$$L_{blocked}(n,k) = \begin{cases} n - 3s^{k-1} - 1 & , 3s^{k-1} < n \leq 4s^{k-1} \\ s^{k-1} - 1 & , 4s^{k-1} < n \leq (s-1)s^{k-1} \\ s \cdot s^{k-1} - n & , (s-1)s^{k-1} < n \leq s \cdot s^{k-1} \end{cases} \tag{6.29}$$

Proof. For $k = 1$ there is no blocked links since the loops are solved by blocking the links in the access network. For $k = 2$, we have $(s-1)$ blocked links, as Figure 6.32 shows. For $k = 3$, there are $(s-1)+s(s-1)$ blocked links. For $k = 4$, there are $(s-1)+s(s-1)+s^2(s-1)$ blocked links, and so on. Let us consider only a segment of length $\lambda_0 = 4s^{k-1}$, where the layer k reaches the maximum number of blocked links. Recall that when $n = 4s^{k-1} + 1$ each segment in k have a shared gateway with its preceding segment. The maximum total of blocked links is obtained by:

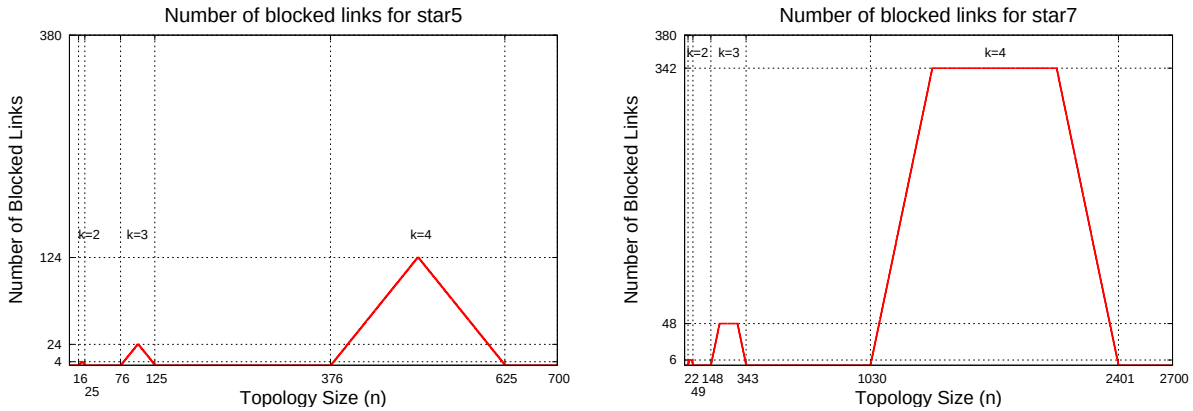
$$L_{blocked}(4s^{k-1} + 1, k) = (s-1)\sum_{i=0}^{k-2} s^i = s^{k-1} - 1 \tag{6.30}$$

Thus, for $3s^{k-1} \leq \lambda_0 < 4s^{k-1}$, loops are created progressively according to λ_0 increases. Therefore, we compute the total number of blocked links within this interval by subtracting $3s^{k-1}$ to λ_0, giving $\lambda_0 - 3s^{k-1}$. For $(s-1)s^{k-1} \leq \lambda_0 < s \cdot s^{k-1}$, we do the opposite, having $s^k - 1 - \lambda_0$. Replacing λ_0 by $n - 1$, we obtain the proposition.

□

Notice there is a "flat" interval where the total number of blocked links is not increased. As mentioned before, this interval it exists when new links are placed within segments with gateways at both ends. However, for $s = 5$, this flat interval does not exist since the initial topology size for the decreasing interval starts in $s - 1$. Figure 6.33 shows the number of blocked links according to the presented intervals. Be aware that the scales of the x-axis are not the equal in order to depict properly the blocked links boundaries.

Regarding the total length of blocked links, we observe different cases for $s = 5$ and $s = 7$. For $s = 5$, all the blocked links have a length 2, since the loop is formed at the layer



(a) Number of blocked links for $s = 5$

(b) Number of blocked links for $s = 7$

Figure 6.33: Number of blocked links for $\mathrm{JC}^2 - star$ kernel with $s = 5$ and $s = 7$.

$k - 1$ when segments have a length 4. Therefore, there is one blocked link with of length 2 at the layer k. Thus, for $s = 5$, the total length of blocked links is:

$$\mathcal{L}_{blocked}(n, k) = 2L_{blocked}(n, k) \tag{6.31}$$

However, for $s = 7$, segments start being blocked for when segments in $k - 1$ have a length 4, and they remains blocked until segments become length 5. Thus, blocked links can be either of length 2 or 3. Therefore, the total length of blocked links for $s = 7$ is bigger than $2L_{blocked}(n, k)$ and less than $3L_{blocked}(n, k)$.

$$2L_{blocked}(n, k) < \mathcal{L}_{blocked}(n, k) < 3L_{blocked}(n, k) \tag{6.32}$$

In conclusion, when comparing the total length of blocked links with the total length of deployed links, we observe this kernel blocks about 10% of the total length of links. But, this situation happens only for small values of n.

6.2.2.3 Binary Kernel

Let us consider a segment of length λ created at the step k of recursion. The division of this segment into s sub-segments defines a set of backbone nodes $B = \{b_j\}$, $1 \leq j \leq s$ associated to the gateways of each sub-segment. The binary kernel creates the links (b_4, b_2), (b_4, b_6), (b_2, b_1), (b_2, b_3), (b_6, b_5), (b_6, b_7). The resulting structure mimics a 2-level binary tree. In the case when $3 \leq \lambda < s$, we place first the links of the first level of the tree, and then, the second level. Note that for $s = 3$, this kernel creates the same structure than the linear and star kernel for $s = 3$ segments; and for $s = 5$, it creates the same structure than the linear kernel (a path) with $s = 5$ segments. Therefore, consider only the binary kernel for $s = 7$ where we can define a complete binary tree. Figure 6.34 shows the deployment of this kernel for $3 \leq n \leq 14$.

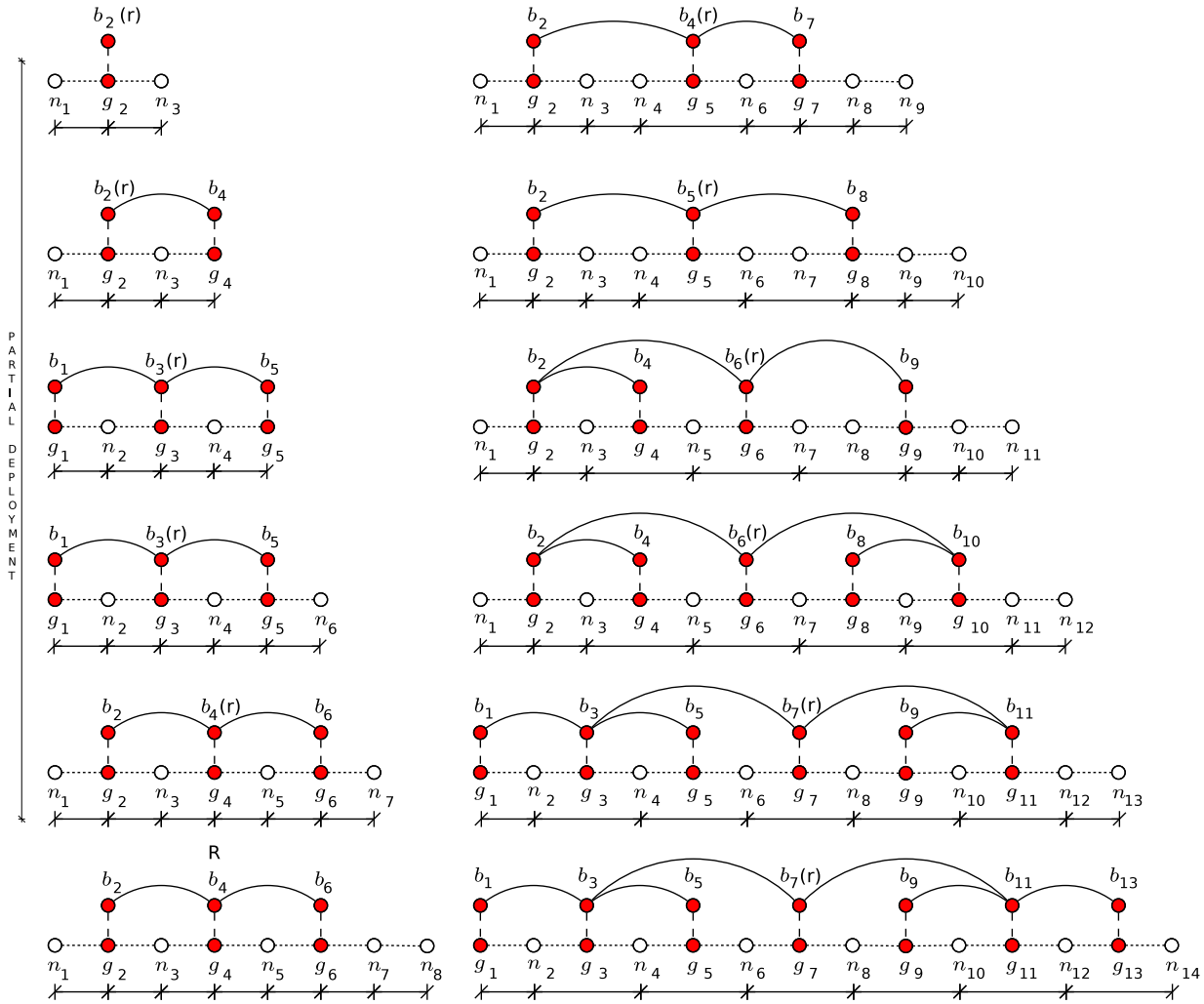


Figure 6.34: Binary tree kernel deployment for $s = 7$

Proposition 6.2.12. *The maximum recursion level $k_{max}(n)$ for a binary kernel satisfies*

$$k_{max}(n) = \lfloor log_s(\frac{n-2}{2}) \rfloor + 1 \tag{6.33}$$

Proof. In a similar way than the star kernel, this one creates its first link within a segment for $\lambda = 3$. Therefore, the proof is the same than the proof presented for the star kernel. □

Figure 6.35 shows the maximum recursion level for topology sizes from $n = 4$ up-to $n = 7000$. The topology size (x-axis) is shown in logarithmic scale in order allow a better comparison with previous presented kernel functions.

Likewise the two previous kernels, the binary kernel exhibits two behaviours – a linear one and constant one – when describing the number of gateways. They are described by intervals according as the kernel is partially or completely deployed. For a given level of

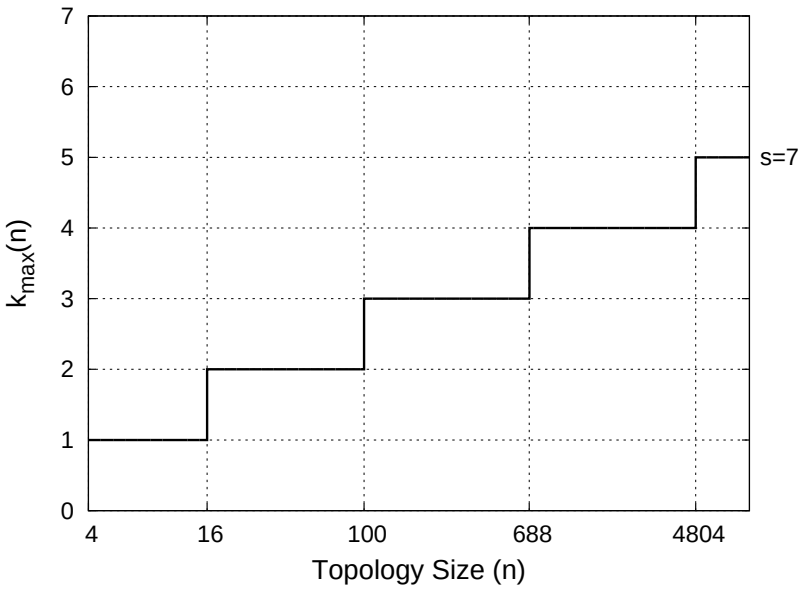


Figure 6.35: Maximum level of recursion for $\mathrm{JC}^2 - bin$ with $s = 7$.

recursion k, the partial deployment begins at $n > s^{k-1}$ and ends when $n \leq (2s-1)s^{k-1}$. The kernel is fully deployed for $(2s-1)s^{k-1} < n \leq 2s^k$. Nevertheless, the partial deployment also shows a linear and a constant part, since within the partial deployment of a segment $[u, v]$, u and v are defined as gateways ($\ell(u, v) = 5$).

Proposition 6.2.13. *The number of gateways created for the binary kernel is given by*

$$m(n,k) = \begin{cases} n - (s^{k-1}+1) & , \ 2 \cdot s^{k-1} & < n \leq & 3 \cdot s^{k-1} \\ 2 \cdot s^{k-1} & , \ 3 \cdot s^{k-1} & < n \leq & 4 \cdot s^{k-1} \\ n - 2 \cdot s^{k-1} & , \ 4 \cdot s^{k-1} & < n \leq & 5 \cdot s^{k-1} \\ 3 \cdot s^{k-1} & , \ 5 \cdot s^{k-1} & < n \leq & (s+2) \cdot s^{k-1} \\ n - ((s-1) \cdot s^{k-1} + 1) & , \ (s+2) \cdot s^{k-1} & < n \leq & (2s-1) \cdot s^{k-1} \\ s^k & , \ (2s-1) \cdot s^{k-1} & < n & \end{cases} \tag{6.34}$$

Proof. Let us consider a segment of length λ created at the first step of recursion ($k = 1$). We have two cases: **Case** $\lambda < s$: For $\lambda = 3$, the first gateway is defined (additionally to the initial gateway of the segment). For $\lambda = 4$ the new gateway correspond with the gateway of the preceding segment, therefore the number of gateways remains equal at 2. For $\lambda = 5$ the third gateway is created since the right end of the segment is not longer considered a gateway. For $5 < \lambda < s$ the number of gateways remains equal due to the restriction on length of links. **Case** $\lambda \geq s$: The segment defines s gateways, but from $s \leq \lambda < s+2$ only 3 gateways are created due to the restriction on the length of links. For $s+2 \leq \lambda < s+6$ ($q = 1, r \geq 3$), it is possible to create new links defining new gateways progressively until to have s gateways for $q = 1, r = s-1$.

Now, let us consider an initial segment of length λ_0 divided k times, creating s^{k-1} segments of length $\lambda_k = qs + r$ at layer k. When $\lambda_k = 2$ there are s^{k-1} gateways. For $\lambda = 3$ obtained when $2s^{k-1} \leq \lambda_0 < 3s^{k-1}$ there are $(\lambda_k - 1)s^{k-1}$ gateways. For $\lambda_k = 4$ when $3s^{k-1} \leq \lambda_0 < 4s^{k-1}$ there are $2s^{k-1}$ gateways. For $\lambda_k = 5$ when $4s^{k-1} \leq \lambda_0 < 5s^{k-1}$ there are $(\lambda_k - 2)s^{k-1}$ gateways. For $5 \leq \lambda_k < s + 2$, when $5s^{k-1} \leq \lambda_0 < (s + 2)s^{k-1}$ the number of gateways remains equal at $3s^{k-1}$. For $s + 2 \leq \lambda_k < 2s + 1$, when $(s + 2)s^{k-1} \leq \lambda_0 < (2s-1)s^{k-1}$ we have $(\lambda_k - (s-1))$ gateways. Finally, for $\lambda_k > 2s-1$, we complete s^k gateways. Considering $\lambda_k = \lfloor \frac{\lambda_0}{s^{k-1}} \rfloor$ and $\lambda_0 = n - 1$ on the previous intervals and number of gateways, we obtain the proposition. □

Figure 6.36 shows the number of gateways according to the topology size n. This result has been verified by simulations for $3 \leq n \leq 5000$ access nodes.

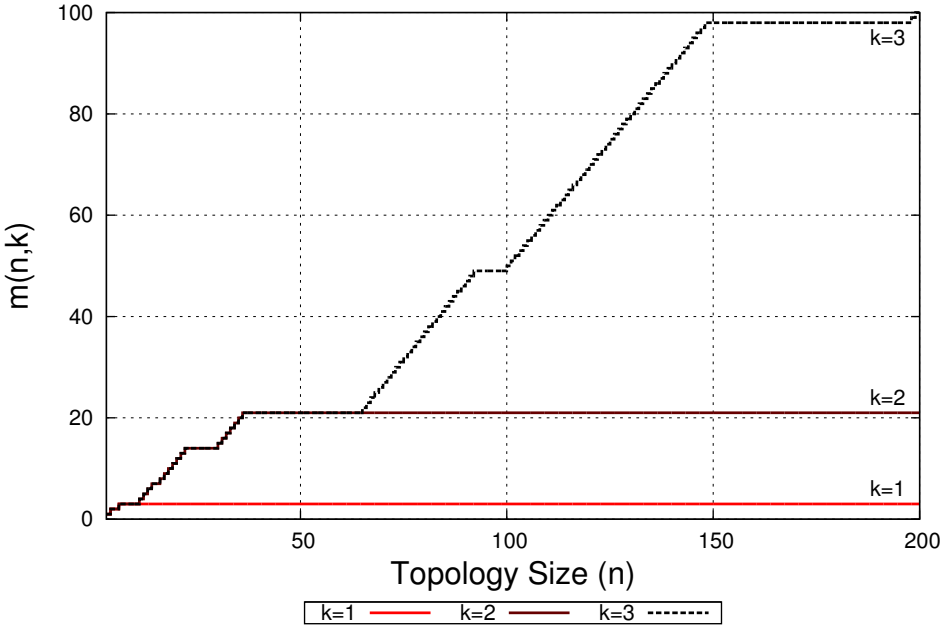


Figure 6.36: Number of gateways for JC^2 topology with a binary kernel of $s = 7$.

Regarding the number of access nodes between contiguous gateways, we use the same approach used for linear and star kernels.

Proposition 6.2.14. *The number of nodes left at the borders of the access network for a binary kernel topology is obtained by*

$$left(n,k) = \begin{cases} \left| \left(\frac{s+1}{2}\right) - \lfloor \frac{n-1}{s^{k-1}} \rfloor \right| & , \quad 2s^k + 1 < n \leq 7s^{k-1} \\ 3 & , \quad 7s^{k-1} < n \leq 11s^{k-1} \\ 2s - \lfloor \frac{n-1}{s^{k-1}} \rfloor & , \quad 11s^{k-1} < n \leq 2s^k \\ \lfloor \frac{n-1}{s^k} \rfloor & , \quad 2s^k < n \end{cases} \tag{6.35}$$

Proof. For a segment of length $\lambda_k = qs + r$ created at the k step of recursion from an initial segment of length λ_0. The number of nodes left changes in steps of s^{k-1} nodes for $2 \leq q < s$, starting from 2 down to 0 (for $q = 4$) and then going up up-to 2 (for $q = s - 1$). The number of left nodes is given by $left(\lambda_0 + 1, k) = |\left(\frac{s+1}{2}\right) - \lambda_k|$. From $s < q \leq 2s - 3$, the number of left nodes remains constant at 3. For $q = 2s - 2$, it descend to 2, to finish the cycle for $q = 2s - 1$ when the number of left nodes is 1. We write this last interval as $left(\lambda_0 + 1, k) = \lambda_k$ in order to consider the case $q > 2s - 1$ (when $k < k_{max}(n)$). Considering $\lambda_k = \lfloor \frac{\lambda_0}{s^{k-1}} \rfloor$ and $\lambda_0 = n - 1$ we obtain the proposition. □

Figure 6.37 shows the average number of access nodes between contiguous gateways for $1 \leq k \leq 4$. Observe for the last deployed layer $k = 4$, The minimum and maximum number of access nodes between gateways are also well described by the $\mathcal{D}_{min}(n, k) = \lfloor \mathcal{D}_{avg}(n, k) \rfloor$ and $\mathcal{D}_{max}(n, k) = \lceil \mathcal{D}_{avg}(n, k) \rceil$.

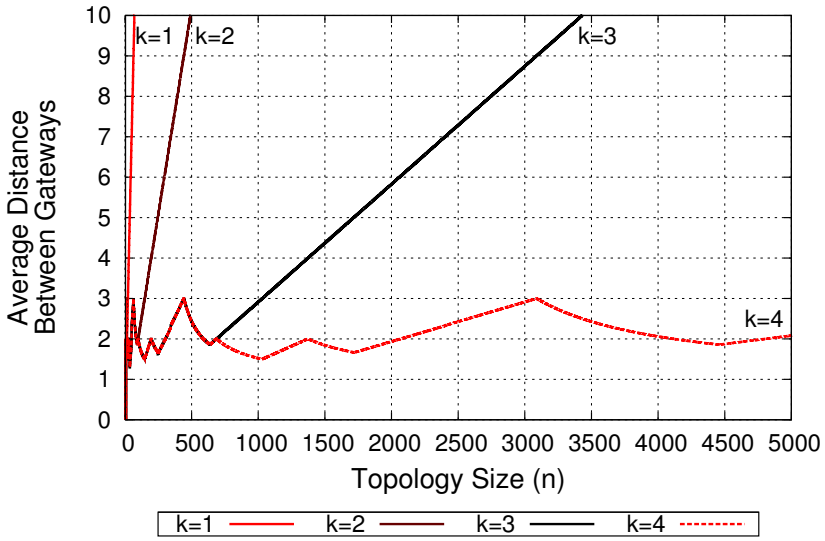


Figure 6.37: Average distance between gateways for the binary kernel

The total length of links is described by the same recursive equation defined for the linear and star kernels, but changing the expression to compute the length of links for the layer k.

Proposition 6.2.15. *Let* $n - 1 = \ell_0 = q_0 s + r_0$, $0 \leq r_0 < s$, *and* $s = 7$.
If $n \geq 2s + 1$, *then*

$$\mathcal{L}(n, k) = (s + 1)q_0 + \ell_7(r_0) + (s - r_0)\mathcal{L}(q_0 + 1, k - 1) + r_0 \cdot \mathcal{L}(q_0 + 2, k - 1) \tag{6.36}$$

with $\ell_7(0) = 0$, $\ell_7(1) = 1$, $\ell_7(2) = 2$, $\ell_7(3) = 4$, $\ell_7(4) = 5$, $\ell_7(5) = 7$,$\ell_7(6) = 8$.

Thereby, in a similar way than the other JC^2 topologies, the initial values for the recurrence 6.36 are obtained for $k = 1$ and $n < 14$.

$$\mathcal{L}(n,1) \quad = 0 \quad \text{(for } n < 4) \qquad \begin{array}{llll} \mathcal{L}(4,1) & = 2 & \mathcal{L}(9,1) & = 5 \\ \mathcal{L}(5,1) & = 4 & \mathcal{L}(10,1) & = 6 \\ \mathcal{L}(6,1) & = 4 & \mathcal{L}(11,1) & = 9 \\ \mathcal{L}(7,1) & = 4 & \mathcal{L}(12,1) & = 12 \\ \mathcal{L}(8,1) & = 4 & \mathcal{L}(13,1) & = 14 \end{array} \tag{6.37}$$

Figure 6.38 shows the total length of links for topology sizes $3 \leq n \leq 300$. In addition, we have verified these results by means of simulations for $3 \leq n \leq 5000$. Notice the growth rate is linear for $k = 1$. However, in conformity with new layers are deployed, the total length of links becomes more irregular.

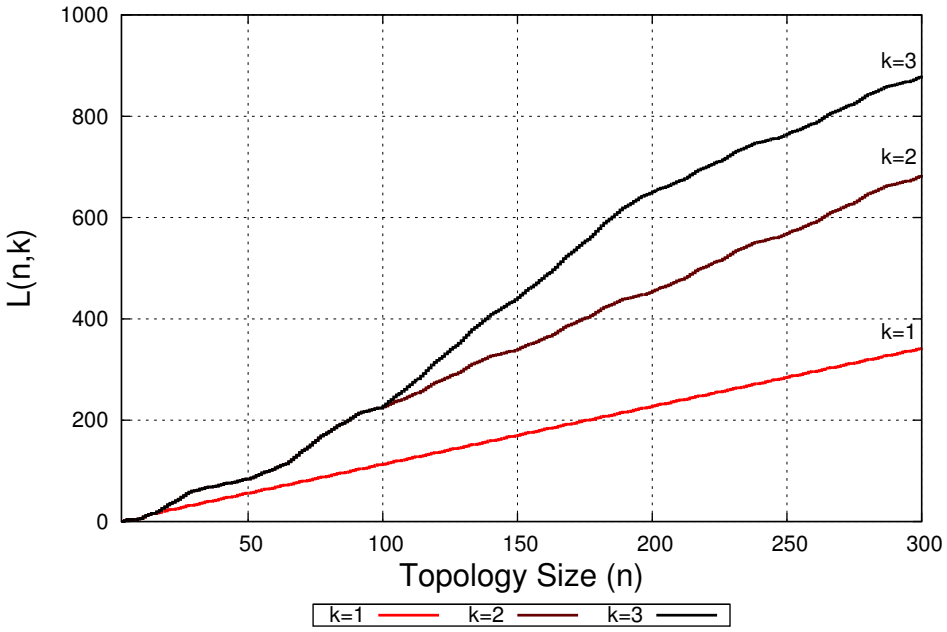


Figure 6.38: Total length of links for JC^2 – *bin* kernel with $s = 7$

Proposition 6.2.16. *The maximum distance from an access node to the root for a binary kernel topology satisfies*

$$d_{max}(n,k) = \begin{cases} 2k-1 & 2s^{k-1}+1 < n \leq 5s^{k-1} \\ 2k & 5s^{k-1} < n \leq s \cdot s^{k-1} \\ 2k + \lceil \frac{n-s^k}{2s^k} \rceil & s \cdot s^{k-1} < n \end{cases} \tag{6.38}$$

Proof. The maximum distance (in number of hops) to the root is obtained in two parts: 1) the maximum number of hops from any access node to its closest gateway; and 2) the

maximum number of hops from each gateway (in fact from each backbone node) to the root node. By proposition 6.2.12, we know that a layer k exists for $n > 2s^{k-1}$, defining segments of length less than 3 up-to $n < 5s^{k-1}$, were the first segment of length 3 appears. Note that the maximum distance to the root for $2s^{k-1} + 1 < n \leq 5s^{k-1}$ is well described by $2k - 1$ since only one level of the binary tree is deployed. For $5s^{k-1} < n \leq s \cdot s^{k-1}$, the second level of the binary tree is deployed, therefore, the maximum distance is $2k$. For $n > s \cdot s^{k-1}$, the maximum distance is incremented in steps of s^k nodes until the next layer is deployed ($n > 2s^k$). □

Figure 6.39 shows the maximum number of hops to the root for $1 \leq k \leq 3$. In addition, we present a plot for $k = 4$ for larger values of n.

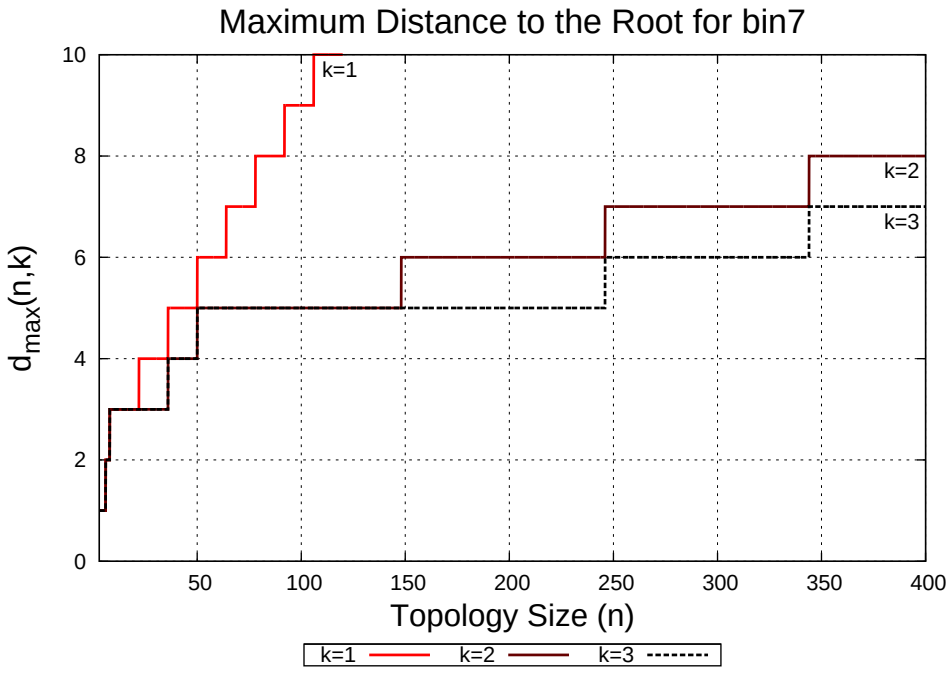


Figure 6.39: Maximum distance to the root for the $JC^2 - star$ kernel with $s = 7$

Concluding this section, we evaluate the impact of the routing protocol on the this topology. By simple observation of the Figure 6.34, we observe there is an interval for the topology size n where this kernel produces loops within the backbone network. Indeed, for $3s^{k-1} < n \leq 4s^{k-1}$, each segment has gateways at both edges. However, on the contrary to star kernels with $s = 7$, these loops are broken when adding s^{k-1} access nodes to the access network ($4s^{k-1} < n \leq 5s^{k-1}$). Therefore, the number of blocked links exhibits a triangle distribution within $3s^{k-1} < n \leq 5s^{k-1}$ with its peak value at $s^{k-1} - 1$. Furthermore, when evaluating the total length of blocked links, we realize all blocked links are placed between the layer $k - 1$ and $k - 2$ having lengths 2 and 3 respectively. Hence, the total length of blocked links is in between two and three times the number of blocked links.

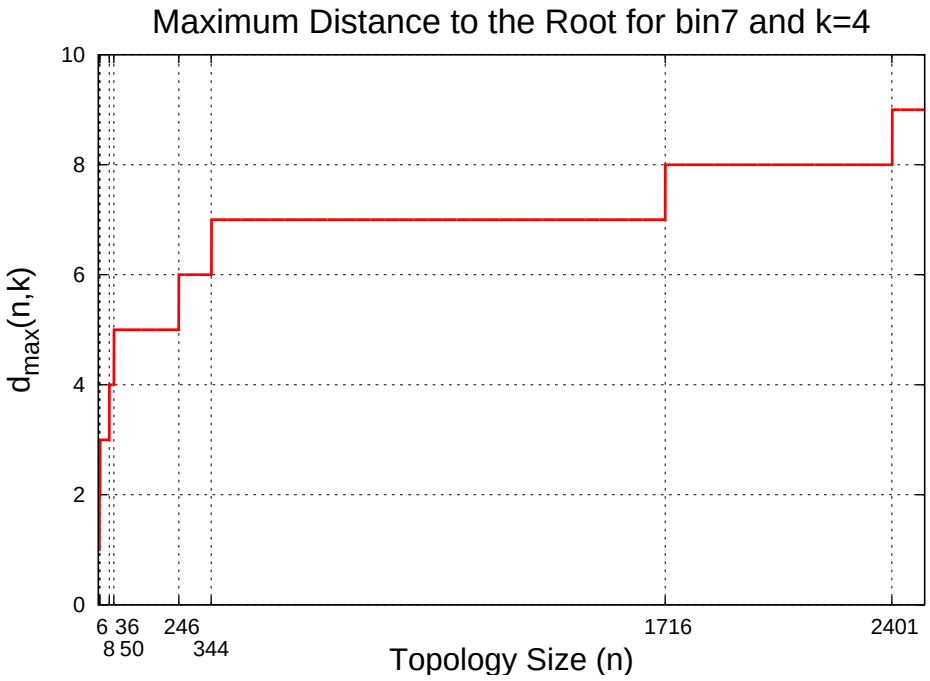


Figure 6.40: Maximum distance to the root for the $JC^2 - star$ kernel with $k = 4$ and $3 \leq n \leq 2500$

$$2L_{blocked}(n,k) < \mathcal{L}_{blocked}(n,k) < 3L_{blocked}(n,k) \tag{6.39}$$

In summary, we finish this section stating that the binary kernel disables the same number of links than star kernels, but the interval when these links exist is much smaller than star kernels with $s = 7$. In consequence, binary kernel topologies have in overall, an inferior rate of blocked links than star kernels.

Chapter 7

Analysis of a Backbone Topology for Linear Access Network

In this chapter, we analyze the topologies defined in the previous chapter. We aim at finding the relationship between their parameters and their resulting network properties. We drive our analysis in two directions: 1) the trade-off between the network properties of each studied backbone topology; and 2) the impact of failures in the access network on the time during which the in-motion network remains disconnected from the infrastructure network.

7.1 Introduction

The topologies defined in the previous chapter exhibit different network properties. These properties depend on the selection of the parameters defining each one of them (the size of the access network n and the level of recursion k). Depending on these parameters, each resulting backbone network will respond to our requirements in a different way. In this chapter we aim at analyzing the relationship between their network properties and their parameters. For that, we present a **Parametric Analysis** of each network property for all the studied topologies. We aim at providing a notion of the variation of the maximum distance to the root, the number of gateways, the total length of links, the number of blocked links and the number of access nodes between contiguous gateways within a wide interval for the linear access network size and the possible levels of recursion. Furthermore, we present a **Network Failure Analysis**, in which we describe the possible cases of failures within the access network and their impact on the probability of observing a disconnected segment of l contiguous access nodes. This probability is related to the number of access nodes between contiguous gateways and it provides a notion of the severity of a failure. We express the severity of a failure in terms of the time during which the in-motion network remains disconnected from the infrastructure network.

7.2 Parametric Analysis

In this section we analyze how the network properties of each topology vary according to the size of the linear access network (n) and the level of recursion (k). We consider access network sizes from 3 up-to 4000, corresponding approximately to a linear access network of 600 km long. For each n, we apply each design rule up-to $k_{max}(n)$ times. However,

the plots presented are depicted up-to $k = 11$ despite the fact that $k > k_{max}(n)$ does not exist for certain topologies. Note that for $k > k_{max}$, the resulting backbone network is the same. We fix $k = 11$ to allow a better comparison among topologies. At the end of this section, we discuss the trade-offs of each topology in terms of their properties, drawing our conclusions.

7.2.1 Number of Gateways

We analyze the number of gateways (or backbone nodes) by means of the **density of gateways**. The density of gateways is defined as the ratio between the number of gateways and the topology size n for a given level of recursion k, in other words, $\frac{m(n,k)}{n}$. Figure 7.1 shows the density of gateways for all the studied topologies for $k = 1$. Note that once the first layer is fully deployed, all topologies exhibit a density that drops asymptotically to 0 with four falling rates: $\frac{2^k+1}{n}$, $\frac{3^k}{n}$, $\frac{5^k}{n}$ and $\frac{7^k}{n}$. In the figure, we observe only 3 rates since Chordal-2 topologies defines 3 gateways at the beginning, the same as JC^2 topologies with $s = 3$. However, for $k > 1$, the Chordal-2 topology shows a drop rate of $\frac{2^k+1}{n}$.

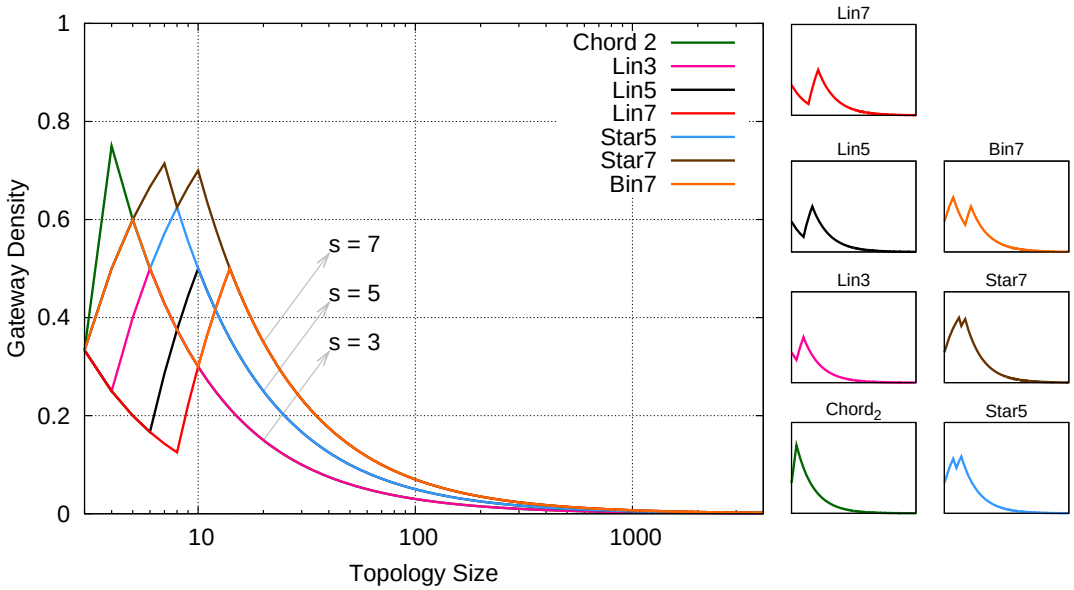


Figure 7.1: Gateway density for all topologies and $k = 1$

Next to Figure 7.1, the shape of the density of gateways for each topology is represented. The behaviour of these shapes is repeated over each layer, showing peaks at the end of each interval where the number of gateways is increasing. More generally, topologies exhibit as much peaks as the number of intervals where $m(n, k)$ is increasing. In the following, we analyze each of the studied topologies for recursion levels up-to $k_{max}(n)$. We use a **color palette** illustrating how the density of gateways changes for each layer within the given interval for the topology size.

Figure 7.2 shows the density of gateways for the Chordal-2 topology. As this topology requires to add 2^k nodes to complete each layer, we observe for $n > 3 \cdot 2^{k_{max}(n)-1}$ the density of gateways decreases asymptotically to 0 since the number of gateways remains constant when the last possible layer is completely deployed. However, for $n < 3 \cdot 2^{k_{max}(n)-1}$, the density exhibits a "wavy" behaviour, since within each layer, the density of gateways increases until 0.75 while the layer is being deployed. Then, it tails-off to 0.5 until $n = 2^k + 1$ when the layer is complete. In addition to the density plot, we present some basics descriptive statistics in order to improve the plot readability when $k = k_{max}(n)$. For this topology, $k_{max}(4000) = 11$.

Basic Statistics for $k = 11$	
Average Density	0.587529
Std.Deviation	0.047893
Minimum ($n > 10$)	0.500244
Maximum	0.750000

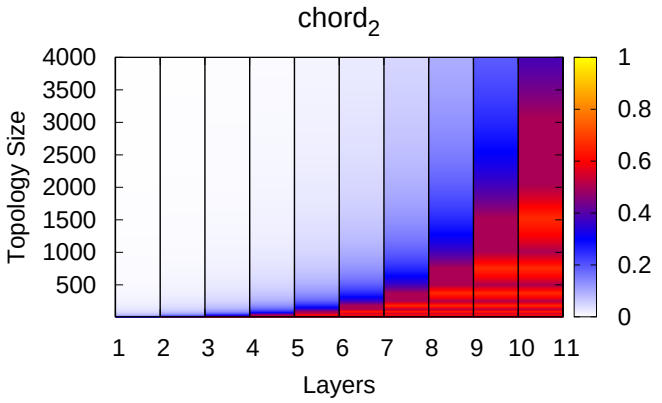


Figure 7.2: Gateways density for Chordal-2 topology.

Figures 7.3, 7.4 and 7.5 show the density of gateways for linear kernels for $1 \leq k \leq 11$. Note that $k_{max}(4000)$ for $s = 3, 5, 7$ are 7,5,4 respectively. Therefore, for $k > k_{max}(4000)$, the density of gateways remains constant since the topology is the same. Also some basics descriptive statistics for $k = k_{max}(4000)$ are shown to improve the readability of each plot. The first obvious difference we notice is depending on the value of s, the resulting topologies cover the topology size n with less levels of recursion, which is expected since the bigger s is, more slowly the Expression 6.12 increases. Also observe the $lin3$ topology shows a higher density of gateways than $lin5$ and $lin7$ for each depicted layer. Indeed, the effects of the restriction on the length of links (bigger than 1) over linear kernels with $s = 5$ or $s = 7$ avoid to define gateways for segment lengths smaller than s.

The minimum density of gateways for $lin5$ and $lin7$ kernels are quite low (0.16 and 0.125) when deploying $k < k_{max}(n)$ (large the blue sections). It means that the required number of gateways to deploy completely a layer is no more than one sixth of the access network size. It is not the same for $lin3$, which exhibits a lower bound of $1/4$ of the topology size. Also notice the maximum density becomes significant (≈ 0.6) when $k > 3$ for $s = 5$, and $k > 2$ for $s = 7$ (the red sections). For $s = 3$, the intervals for n where more than the 60% of the access network is considered as gateways is larger.

Basic Statistics for $k = 7$	
Average Density	0.478080
Std.Deviation	0.081278
Minimum ($n > 10$)	0.250000
Maximum	0.599835

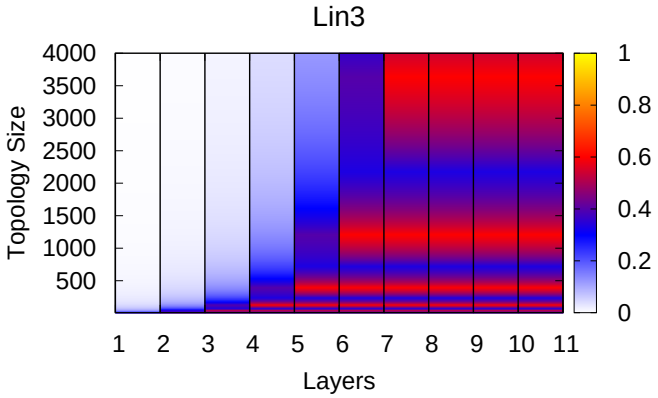


Figure 7.3: Gateways density for *lin*3 kernel

Basic Statistics for $k = 5$	
Average Density	0.327734
Std.Deviation	0.095904
Minimum ($n > 10$)	0.166667
Maximum	0.555062

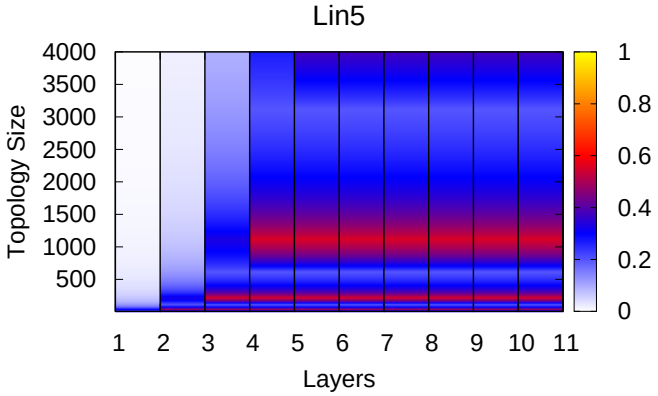


Figure 7.4: Gateways density for *lin*5 kernel

Star kernels deliver the highest number of gateways among the studied topologies, as showed by Figures 7.6 and 7.7. Indeed, the kernel connectivity is not much affected by the restriction on length of links. Therefore, the intervals where the number of gateways remain constant are shorter compared to linear kernels. This fact yields a higher density within each layer (red and orange areas on the plot). In addition, observe in the plots there is no "blue" areas between peaks, fact that is supported by the shape of the density curve depicted by Figure 7.1.

Examining numerically the density for both star kernel topologies, the minimum density for $s = 5$ and $s = 7$ are about half of the topology size, and the maximum is almost $^3/_4$ of the topology size. In summary, star kernels requires in average about 60% of the access network size to be considered as gateways when deploying $k < k_{max}(n)$.

Finally, we analyze the density of gateways for the binary kernel. Recall this kernel

Basic Statistics for $k = 4$	
Average Density	0.301955
Std.Deviation	0.111010
Minimum ($n > 10$)	0.125000
Maximum	0.537618

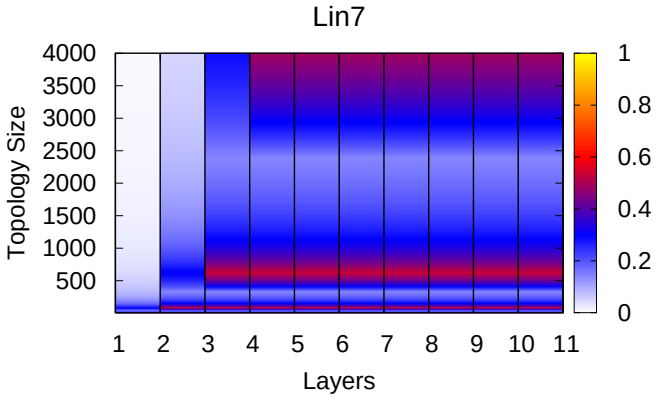


Figure 7.5: Gateways density for *lin*7 kernel

Basic Statistics for $k = 5$	
Average Density	0.598896
Std.Deviation	0.055368
Minimum ($n > 10$)	0.454545
Maximum	0.713470

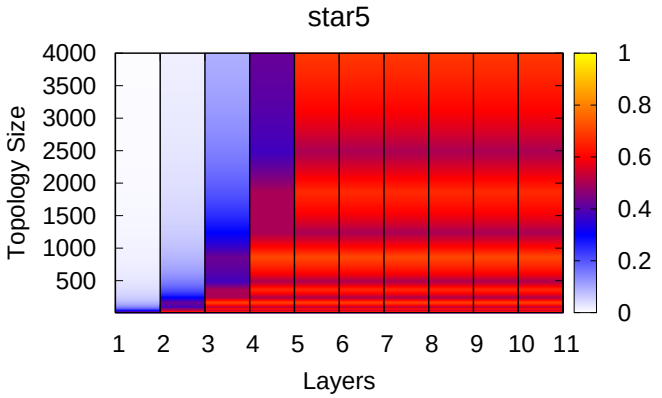


Figure 7.6: Gateways density for *star*5 kernel

presents the best trade-off between the topology size and the number of gateways for $k = k_{max}(n)$. For 4000 nodes, the last possible layer is $k = 4$, defining in average almost half of the access nodes as gateways. However, notice for $k = 3$, the average density of gateways drops to 0.23, allowing an important reduction in the number of gateways only by not deploying the last layer. This reduction is not evident for linear kernels and star kernels, since they require at least not to deploy the last two layers in order to have a similar reduction. Notice the impact of not deploying the last layer have an effect in the maximum distance to the root and the number of access nodes between gateways (analyzed later on).

In conclusion, when analyzing the number of required gateways to deploy a backbone network, we notice that linear kernels deliver the minimum number of gateways when deploying $k = k_{max}(n)$. However, when deploying $k < k_{max}(n)$ layers, this situation might change due to the fact that each topology require a different number of layers to cover the

Basic Statistics for $k = 4$	
Average Density	0.654105
Std.Deviation	0.077098
Minimum ($n > 10$)	0.466667
Maximum	0.777526

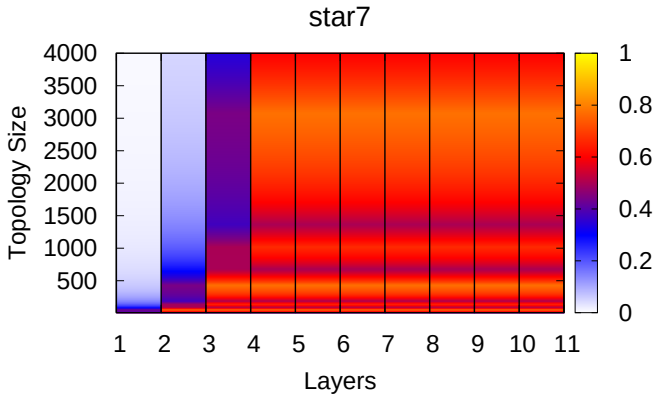


Figure 7.7: Gateways density for *star*7 kernel

Basic Statistics for $k = 4$	
Average Density	0.47611
Std.Deviation	0.08827
Minimum ($n > 10$)	0.30000
Maximum	0.66602

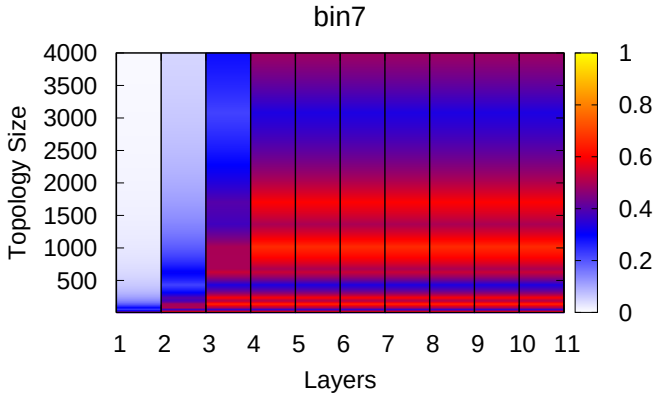


Figure 7.8: Gateways density for *bin*7 kernel

same access network size, fact that might yield to have different local optimum topologies for different combinations of the parameters n, k.

7.2.2 Total Length of Links

We use the same type of plots employed to analyze the density of gateways, but now the color palette represents the total length of links. It starts from white (0), passing by blue (10k), yellow (30k) and finishing with green, representing a total length of 45 000, or 45k. Recall the basic unit for this property is the distance (in meters) between two contiguous access nodes (c), which is assumed to 150 meters according to Section 3.2. Hence, 10k means 1500 km.

Firstly, the Chordal-2 topology is one that requires the larger total length of links to be deployed due to its inherent redundancy. In addition, the short number of nodes required

to deploy each layer (2^k nodes) causes the design rule to be applied more times than other topologies. Figure 7.9 shows the required total length of links to deploy for $1 \leq k \leq 11$.

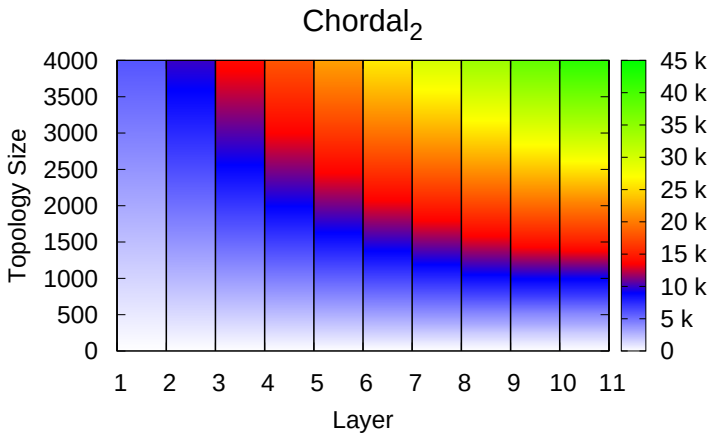


Figure 7.9: Total length of links for Chordal-2 topologies

Note that the difference in the total length of links between deploying k and $k+1$ layers is not large, since each layer adds $(n-1)$ to the total length of links when fully deployed (in total $k(n-1)$ for k deployed layers). In conclusion, Chordal-2 topologies requires k times the length of the linear access network to deploy k layers.

Secondly, the JC^2 linear kernel topologies exhibit a lower total length of links than Chordal-2 topologies (larger blue sections in the plot). Note that for linear kernels, the required total length of links to fully deploy a layer is as much $\frac{(n-1)}{s}(s-1)+\delta$ with $\delta \in [0, s^{k-1}-1)$, which of the same order as $(n-1)$, required by the Chordal-2 topology. Nevertheless, the lower number of times that the linear kernel is applied to an access network size n is enough to produce resulting backbone topologies with a lower total length of links. Figure 7.10 shows the three linear kernels applied for $3 \leq n \leq 4000$. Observe the $lin3$ topology reaches a total length of links of 19k for $n = 4000$, 14k for $lin5$ and 13k for $lin7$, each one reaching its maximum for $k = 7, 4$ and 3 respectively.

Thirdly, the star kernels topologies require roughly $\frac{(s^2-1)}{4s}(n-1)$ to deploy a single layer. Thus, the resulting topologies requires more length of links to be deployed. Indeed, when comparing linear kernels topologies with the star kernels for the same value of s, we obtain a relation of $\frac{s^2-1}{4(s-1)}$, leading to concrete ratios of $\frac{3}{2}$ and 2 for $s = 5$ and $s = 7$ respectively, both verified by simulations. Comparing star kernels topologies with the Chordal-2 topology, this ratio is not that direct to compute because the difference in the number of layers they use to cover the same topology size. Figure 7.11 shows the total length of links required by star kernel topologies for $1 \leq k \leq 11$. Observe the difference between deploying k and $k+1$, which is large for k small, but when k approaches to $k_{max}(n)$, the difference becomes smaller.

Fourthly, the binary kernel topology adds about $8\left(\frac{n-1}{7}\right)$ to the total length of links

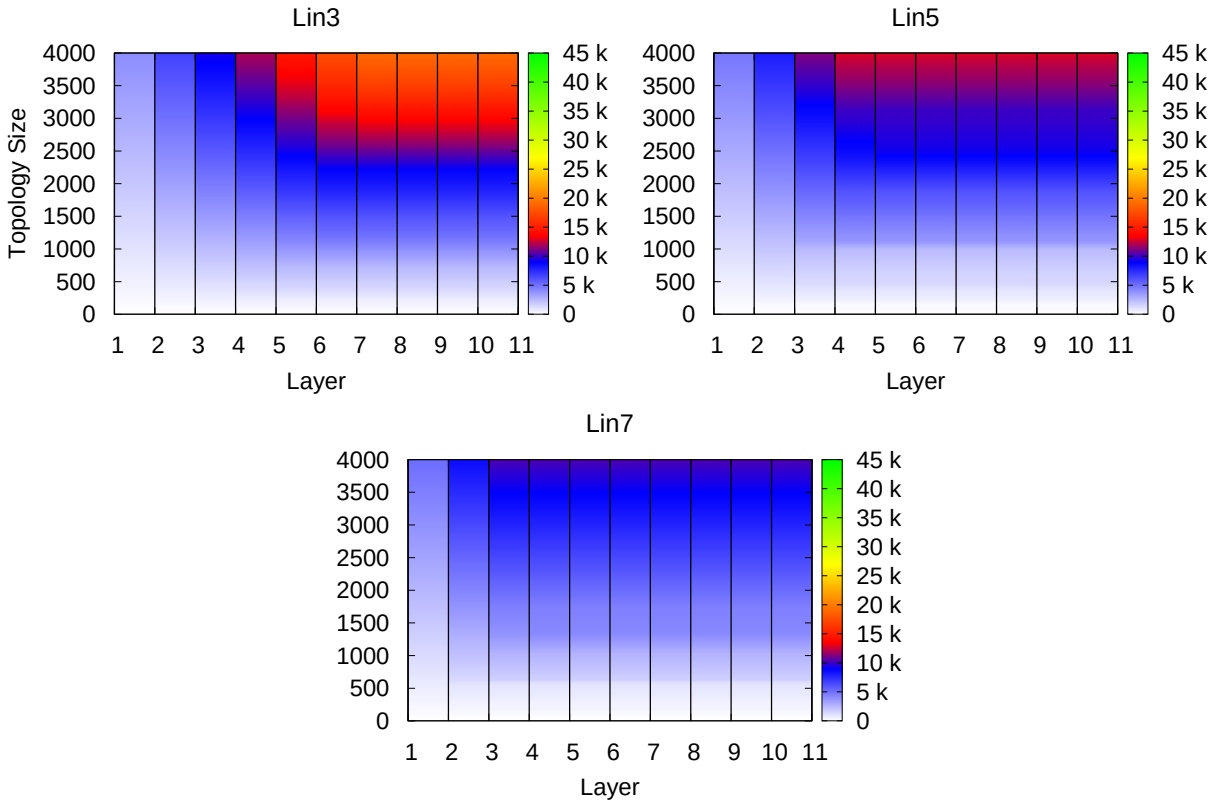


Figure 7.10: Total length of links for linear kernels

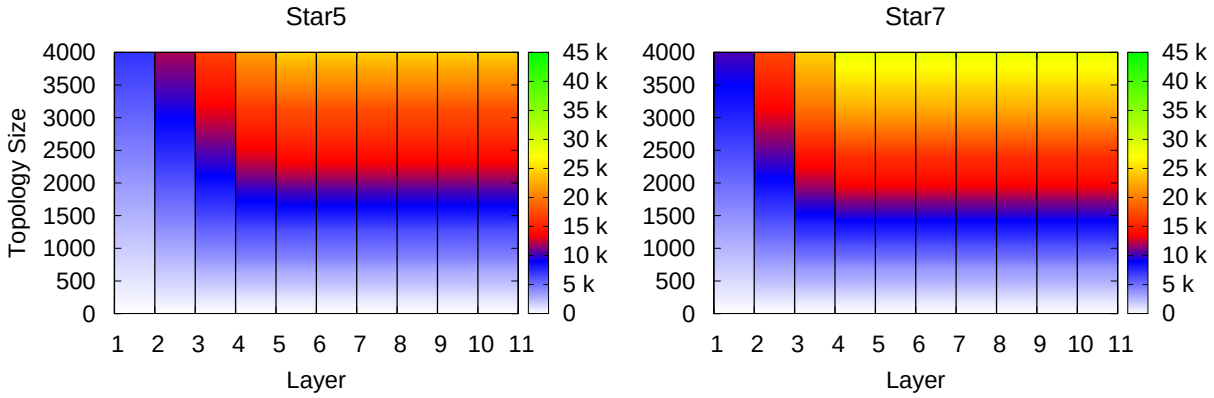


Figure 7.11: Total length of links for star kernels

when deploying each layer. Thus, when comparing it against the $lin7$ and $star7$ kernels, we obtain ratios of $\frac{s+1}{s-1}$ and $\frac{4(s+1)}{s^2-1}$ respectively, yielding a relation of $\frac{4}{3}$ with $lin7$ and $\frac{2}{3}$ with $star7$. Figure 7.12 shows the total length of links from $1 \leq k \leq 11$.

Lastly, we introduce a comparison between all the studied topologies for $k = k_{max}(n)$

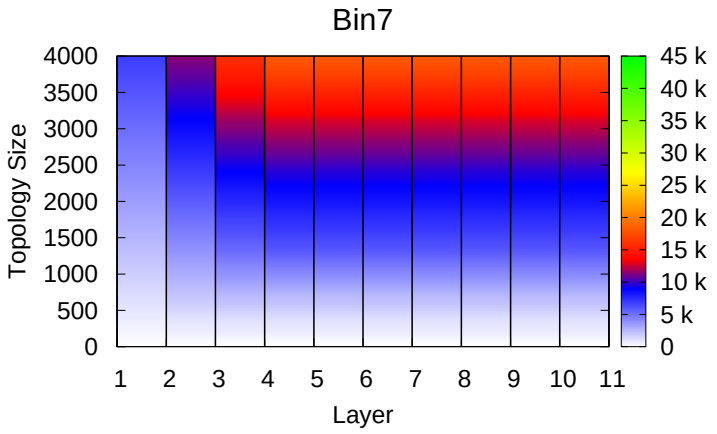


Figure 7.12: Total length of links for binary kernel

in order see how their total length of links vary according to the topology size. Observe the $lin3$ and $bin7$ topologies are very close each other, presenting both a good trade-off between the total length of links versus the topology size. Topologies $lin5$ and $lin7$ are the ones that present the lower length of links required to deploy a backbone network, while star kernels and Chordal-2 topologies are the more expensive ones in terms of the length of links.

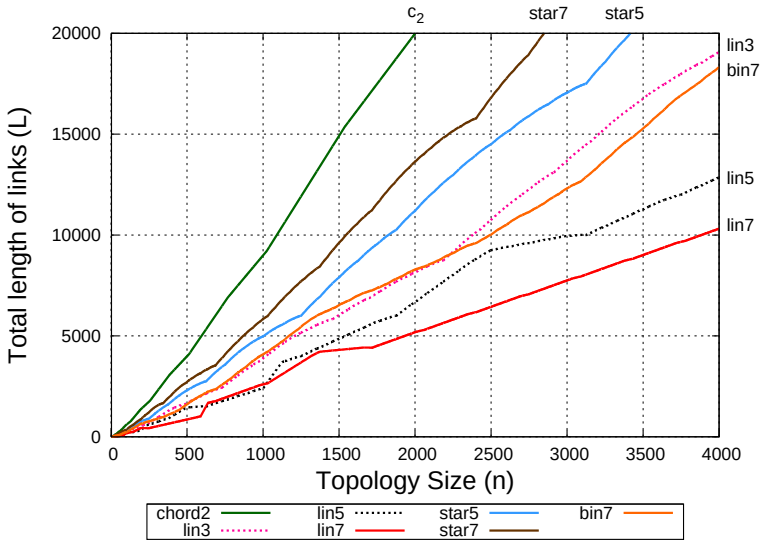


Figure 7.13: Total length of links for all topologies and $k = k_{max}(n)$

7.2.3 Maximum Distance to the Root

Figure 7.14 shows the maximum distance for all the studied topologies when deploying all the possible layers. Note that the topologies that are better "connected" exhibit lower distances to the root. Star kernels topologies are the best (lower distance) when deploying all possible layers, being followed by the Chordal-2 topologies, then the linear kernels with the worst distance. Notice that the binary kernel is "at the middle" of all of them. Note also there are some intervals for n where some topologies exhibit the same distance, for instance, Chordal-2 and star kernels and $lin3$ with $bin7$. It means that when selecting one of them, the decision must be based on another network property. In summary, the lower distances to the root are achieved by $star7$, $star5$ and C_2 (in order of precedence) when deploying all the possible layers. In the following, we analyze each topology.

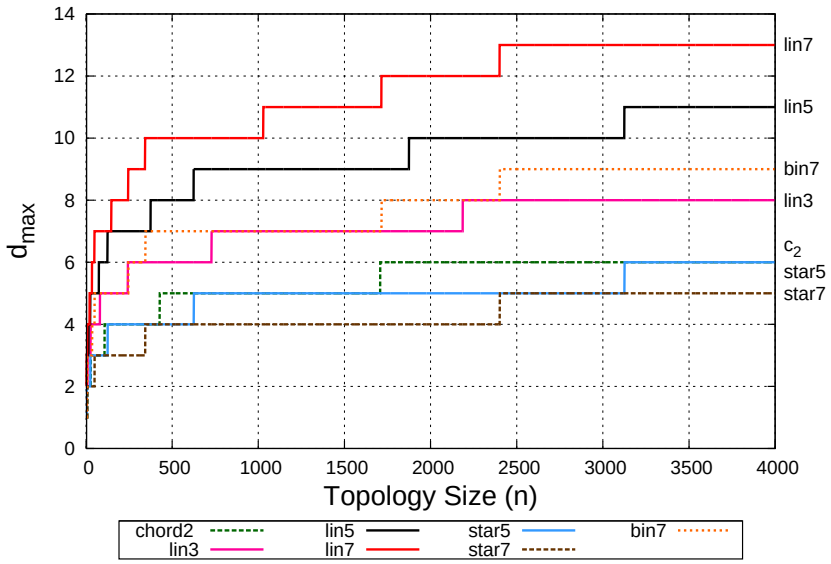


Figure 7.14: Maximum distance to the root v/s topology size

Firstly, the Chordal-2 topology. Figure 7.15 shows the maximum distance to the root (in log-scale) for $1 \leq k \leq 11$. Note that after the last layer $k_{max}(n)$ is deployed, the maximum distance increases rapidly, reaching unpractical distances when thinking in a real deployment. In addition, the gain in distance between deploying k and $k+1$ layers is about the half. However, while k approaches to the maximum possible layer $k_{max}(n)$, the gain becomes smaller. Thus, Chordal-2 topologies only delivers a good distance to the root only when deploying all the possible layers.

Secondly, linear kernels topologies. Observe in the Figure 7.16, the minimum distance is reached far before the Chordal-2 topologies, being $lin7$ the topology which exhibits the better convergence rate. However, the minimum distance delivered by $lin7$ is higher than distances achieved by $lin5$ and $lin3$ topologies. Indeed, the bigger is the segment length for linear kernels, the bigger is the linear part to travel within the segment to reach the closest

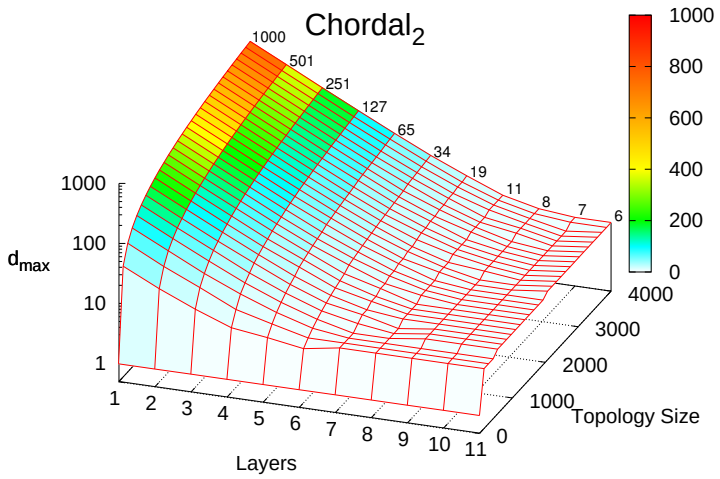


Figure 7.15: Maximum distance to the root for Chordal-2

gateway. However, larger segments show a better gain when deploying $k+1$ instead of k layers. In addition, not deploying all the possible layers also leads to maximum distances that are not practical on real deployments.

Thirdly, star kernels topologies are the absolute winner in terms of the maximum distance to the root. Observe in Figure 7.17 the convergence rate to achieve the minimum distance. They converge very quick to the minimum possible distance, which is, at the same time, a very good lower bound considering the access network size. Observe also the proportion of the network that remains at the same distance, which is quite large. This fact makes the star kernels topologies attractive for a large linear access network.

Lastly, the binary kernel topology (Figure 7.18) also shows a rapid convergence to the minimum distance when deploying layers. However, its minimum is not as good as the minimum achieved by star kernels. Nevertheless, the attained minimum is a good compromise between access network size and the distance to the root when comparing it with linear kernels and Chordal-2 topologies. The range of the topology size that remains constant when deploying $k_{max}(n)$ layers is not as long as the range provided by star kernels, however, it is not small as the range of linear kernels.

7.2.4 Number of Blocked Links

We analyze the number of blocked links in order to have a notion of redundancy and failure tolerance. In Chapter 6, we mentioned the number of blocked links might be used as a measure of redundancy, since each blocked link gives an alternative path to the root, and a network is as much redundant as the number of alternative paths it provides. Thus, when considering the shortest path between any pair of nodes u, R (u in the access network and R the root of the topology) each blocked link represents a valid next hop that might be used by the routing protocol in case of failure.

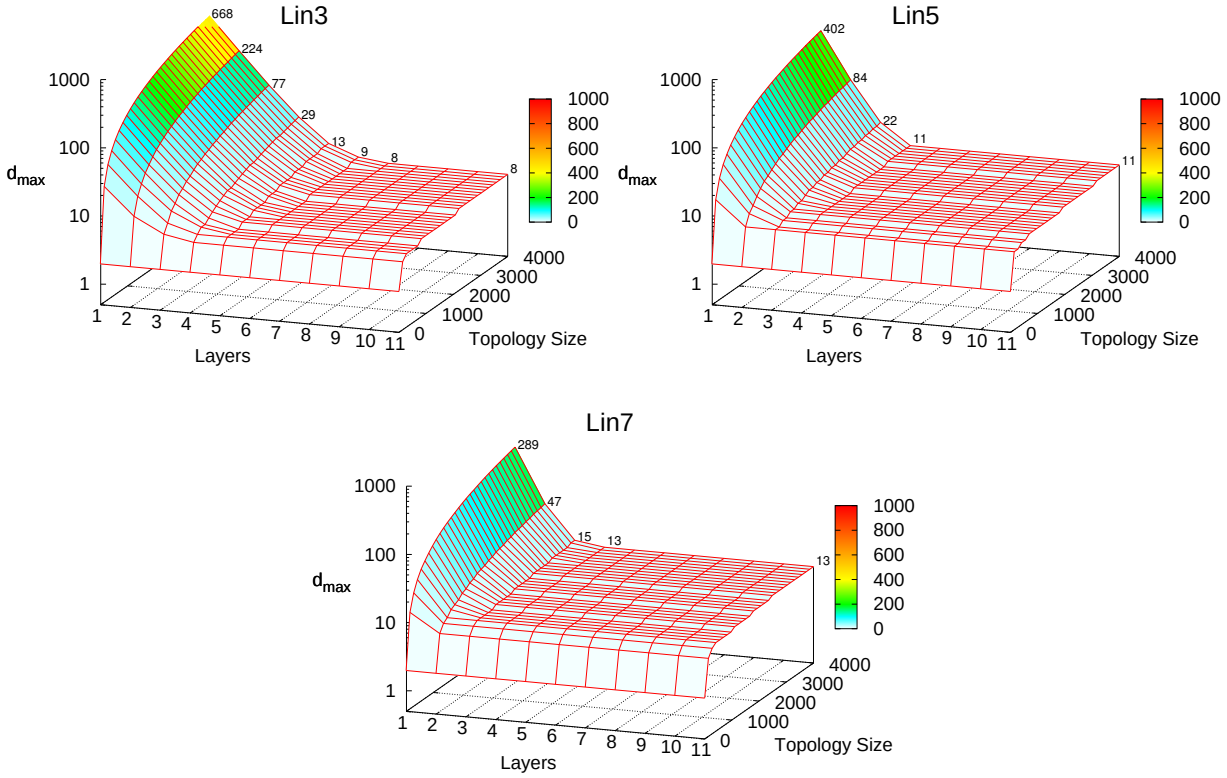


Figure 7.16: Maximum distance to the root for linear kernels

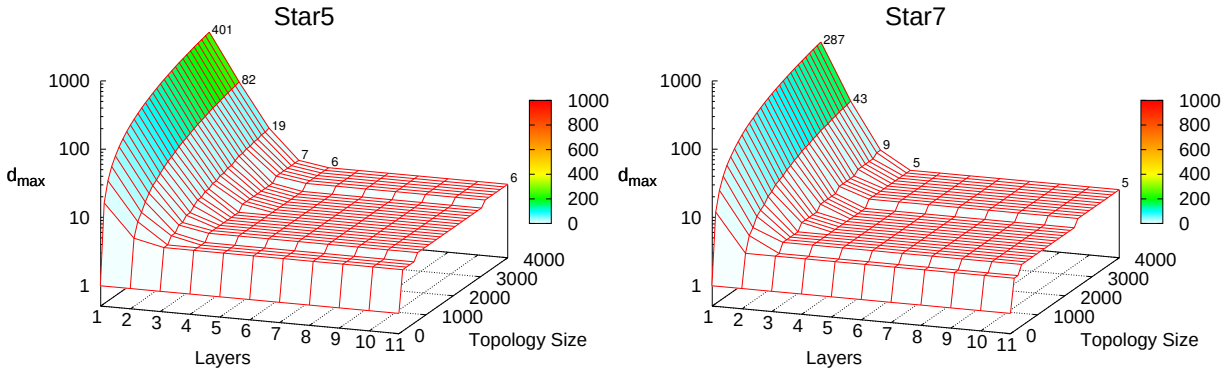


Figure 7.17: Maximum distance to the root for star kernels

7.2.4.1 Failure Tolerance in the context of a Linear Access Network

A failure, or service failure, occurs when a system is not behaving as expected due to the presence of an error. This error is originated by a flaw (design flaw, hardware flaw, human flaw, etc). Thus, as pointed out by [Sterbenz *et al.* 2010]: "a fault may be triggered to

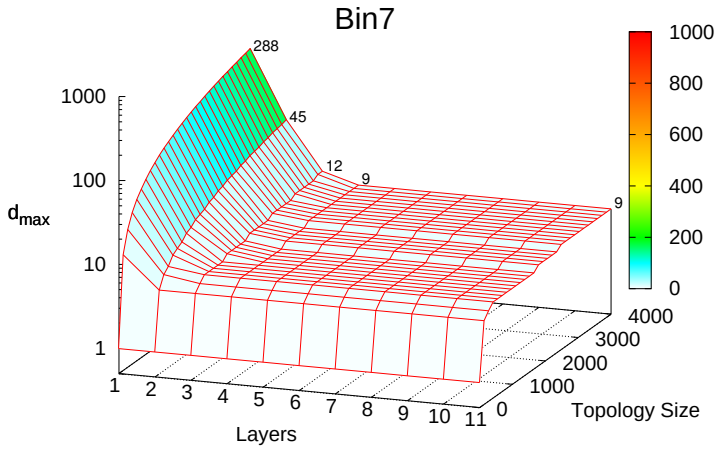


Figure 7.18: Maximum distance to the root for binary kernel

cause an observable error, which may result in a failure if the error is manifested in such a way that causes the system not to meet its service specification". In our scenario, failures can occur either in nodes or links in the form of node unavailability due to hardware failure, misconfiguration, power supply outage, etc.; or links unavailability, due to accidents, theft, etc. Furthermore, failures can happen either in the linear access network, in the backbone network, or in both networks at the same time. The first class of failures, namely **linear access network failures**, are defined as the failure of two or more access nodes in such a location that their failure isolate all access nodes enclosed by them, and by consequent, causing their disconnection from the rest of the network (a partition). When two nodes fail and there is a gateway in between, no failure occurs, since all the other nodes are still connected, or in other words, two node failures does not produce a service failure when there is an operative gateway in between them. Figure 7.19 depicts the scenario, where two segments of length 1 are disconnected from the network.

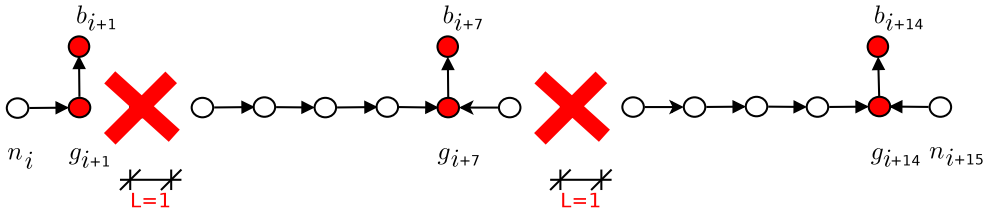
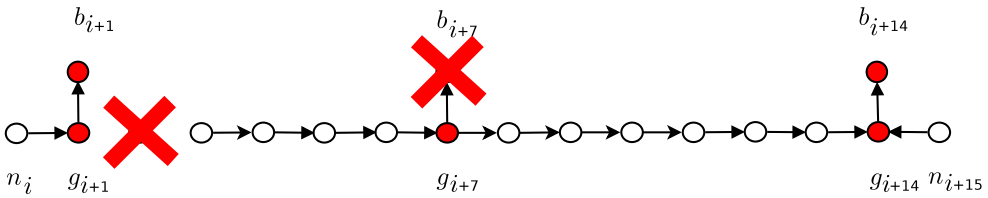


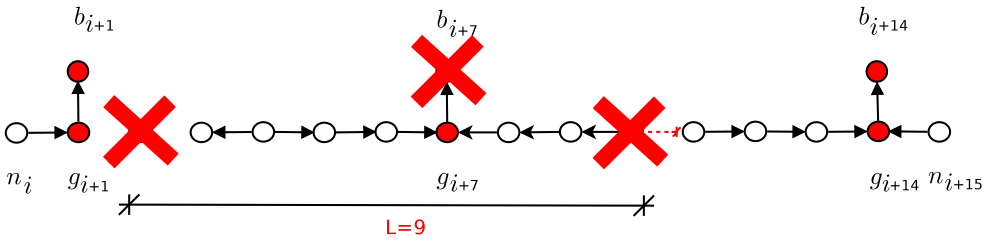
Figure 7.19: 2 access nodes down and no failure.

A failure in the backbone network occurs when either a backbone node or link fails in such a way that it causes a network partition. In our case, as the access network is also (linearly) connected, when a backbone node is unavailable, its gateway loses the connection with the backbone network, incrementing the number of access nodes between the consecutive (operative) gateways. Thus, the maximum distance to the root is increased

and raising the probability of having a larger network partition when further gateways fail. Therefore, when assuming no failure in the linear access network, the only impact of a backbone failure is the increment of the maximum distance to the root. Nevertheless, when gateways and backbone network fails together, the final impact on the service might be significantly due to high number of hops or the isolation of large block of access nodes. Figure 7.20 depicts both cases, when backbone nodes fail or gateways and backbone nodes fail.



(a) Backbone and access network failure degrading the distance to the root



(b) Backbone and access network failure isolating a segment of $L = 9$

Figure 7.20: Backbone and access network fails together

As we consider access nodes more likely to fail than backbone nodes (see Section 3.5), we focus our attention in the linear access network failures, since they might produce a larger impact on the service level of the network (a disconnected segment of access nodes). We assume from here that backbone network failures only leads eventually to a degradation in the network delay, and the failure scenarios such as the one depicted by Figure 7.20b are subjected to failures in the linear access network.

7.2.4.2 Redundancy and the Number of Blocked Links

Analyzing each topology within the failure context described above, the Chordal-2 topologies are the most redundant ones among the studied topologies. Its tolerance comes from the fact that each backbone node is at least connected with two other nodes. These redundant connections are blocked by the the routing protocol in order to avoid loops within the network. Thus, in case of failure, each blocked link represents a possible next hop to reach the root node of the network. JC^2 topologies are less redundant since they do not present many loops within the backbone network. In particular, the star kernels provide redundancy for $3s^{k-1} < n \leq s^k$. Within this interval segments define gateways at the ends nodes, making them to coincide. Thus, a loop is created within the backbone network and a new alternate path to the root is provided (see Figure 6.27). Next, we have the binary

kernel topology, which exhibits a smaller interval for n where redundancy is provided. ($3s^{k-1} < n \leq 4s^{k-1}$). At the end, the linear kernels do not provide any redundancy except one alternative path to the root through the linear access network.

In summary, Chordal-2 topologies are the most redundant one since they present the larger number of blocked links. Then, the star kernel topologies follows in redundancy. Following, the binary kernel topology and finally the linear kernel topologies.

7.2.5 Number of Access Nodes between Contiguous Gateways

We analyze the number of access nodes between gateways by studying its average value. Recall that all the studied topologies exhibit a quasi regular number of access nodes between gateways when deploying all possible layers. However, when deploying $k < k_{max}(n)$, gateways are placed at different intervals of access nodes, depending on how the connectivity rule of each topology is affected by the restriction on the length of links.

In general the number of access nodes between gateways is reduced according to new layers are deployed, and its value depend on the regularity of the interval they are placed. For the Chordal-2 topology, Figure 7.21 shows the average number of access nodes between contiguous gateways (in log-scale). When deploying $k_{max}(n)$ layers, the segments at the last layer have only two possible lengths: 1 and 2. When $k < k_{max}(n)$, the average number of access nodes between gateways increases linearly with a growth rate of $\frac{1}{2^k}$. In other words, when not deploying all possible layers for a given topology size n, the distance between contiguous gateways is roughly $\frac{n}{2^k}$.

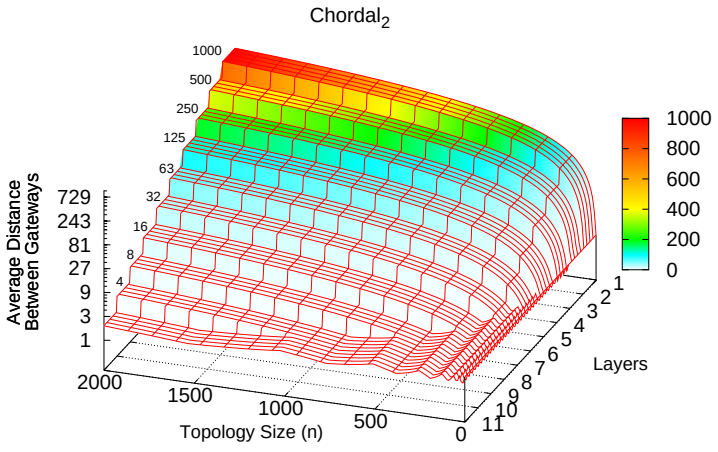


Figure 7.21: Average distance between gateways for Chordal-2

Linear kernels on the contrary, exhibit a larger number of nodes between contiguous gateways, as depicted by Figure 7.22 and discussed in Section 6.2.2.1. They exhibit maximum distances of 3, 5 and 7, corresponding to the peaks in the figure when deploying all possible layers. This maximum is due to the restriction on the length of links. However,

within some intervals ($\lambda > s$), the maximum number of access nodes is lower than s, since new links appears in when deploying partially the kernel. For $k < k_{max}(n)$, linear kernel topologies increase their average number of access nodes between gateways with a ratio of $\frac{1}{s^k}$.

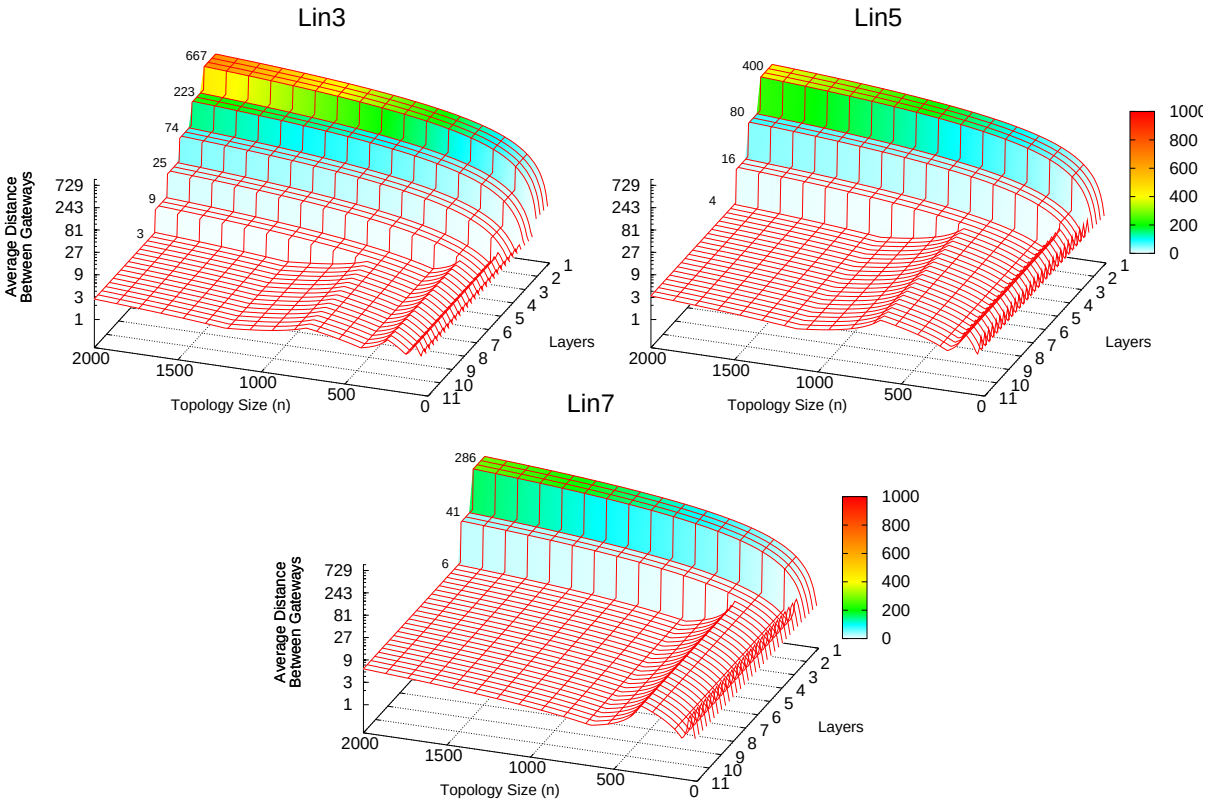


Figure 7.22: Average distance between gateways for linear kernels

The star kernel topologies have two possible number of access nodes between gateways when deploying $k_{max}(n)$ layers: 1 and 2. For $k < k_{max}(n)$, the average distance is increased at the rate of $\frac{1}{s^k}$, in the same way than linear kernels.

The binary kernel shows a number of access nodes between gateways between 1 and 3, with intervals when it is equal to 2 when deploying all possible layers. However, when deploying less layers, the number of access nodes between contiguous segments grows at the rate of $\frac{1}{7^k}$, in a similar way than $lin7$ and $star7$. Figure 7.24 shows the average number of access nodes between gateways. The observed peak for $k > 3$ is 3, reached for $7s^{k-1} < n \leq 11s^{k-1}$.

Concluding the analysis of the number of access nodes between contiguous gateways, we have two cases: 1) $k = k_{max}(n)$ and 2) $k < k_{max}(n)$. The first case (when deploying all possible layers), we see that Chordal-2 and Star kernel topologies have gateways 1 or 2 nodes apart. The binary kernel topology have gateways between 1 and 3 nodes apart, and finally the linear kernels, having gateways between 1 and s nodes apart. For the second

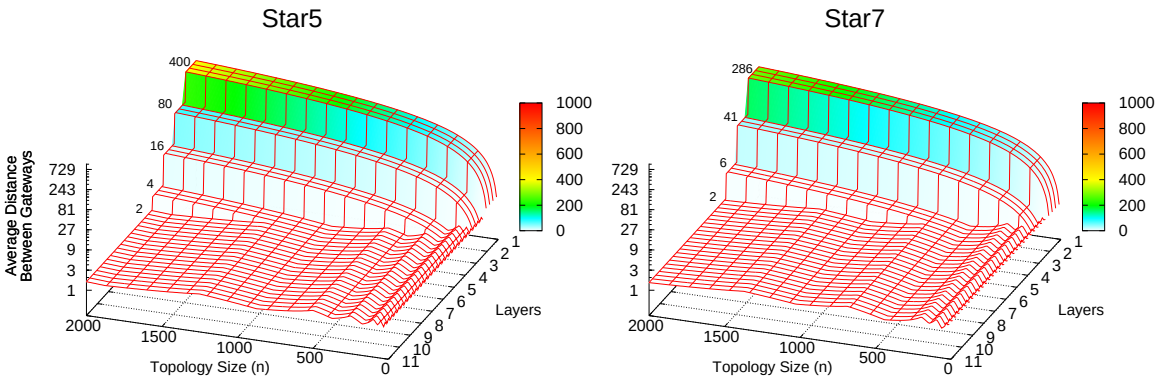


Figure 7.23: Average distance between gateways for star kernels

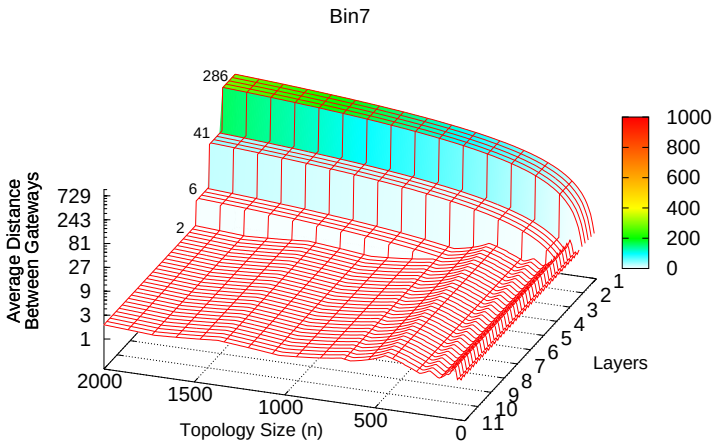


Figure 7.24: Average distance between gateways for binary kernel

case (when deploying less layers), we see that the Chordal-2 topology has their gateways about $\frac{n}{2^k}$ nodes apart; and JC^2 topologies have their gateways roughly $\frac{n}{s^k}$ nodes apart.

7.2.6 Concluding Remarks

For Chordal-2 topologies, we can not achieve a low distance to the root without paying in a large number of gateways and total length of links. Each deployed layer divides the number of access nodes between gateways ($\mathcal{D}$) by 2. So, the only way to get a low $\mathcal{D}$ is by deploying $k_{max}(n)$ layers. The redundancy of Chordal-2 topologies increases with each deployed layer, since each one provides a new disjoint path to the root, and when combined with the links created by upper layers, a large combination of not disjoint paths are possible.

For linear kernels topologies, a larger size of segments (s) gives less gateways and total

length of links, but in return of a larger maximum distance to the root. With each layer deployed, $\mathcal{D}$ is divided by s, thus, a large s gives lower values of $\mathcal{D}$ and a lower number of gateways and total length of links. The resilience provided by the linear kernel is low for two reasons: 1) a poor redundancy since only one alternative path to the root is provided (through the linear access network); and 2) a large $\mathcal{D}$ within each layer.

For star kernel topologies we see that they provide the best distance to the root, but in return of a large number of gateways and total length of links. Conversely, to get a low number of gateways (and links), we pay in distance to the root and in $\mathcal{D}$. A better resilience is attained only for an interval of n within each deployed layer. For those values of n,k such that $3s^{k-1} < n \leq s^k$, loops are created within the backbone network, providing a new alternative path to the root. Thus, star kernels provides better resilience than linear kernel topologies.

Finally, for the binary kernel topology, we see that it gives a trade-off in between linear and star kernels. Similarly to star kernel topologies, the binary kernel provides a better redundancy for an interval of n within each layer, but the length of this interval is shorter ($3s^{k-1} < n \leq 5s^{k-1}$). Hence, we say that this kernel is less redundant than the star kernel.

In general, we see that among the studied topology, **we can not achieve the best values for a single property without paying with another property (or properties)**. To achieve a good distance to the root, we require to deploy a large number of layers, increasing the number of gateways and total length of links. On the contrary, when reducing costs (in terms of the number of gateways and total length of links), we pay in a larger distance to the root and a larger number of access nodes between gateways, which at the same time, reduces the resilience of the network.

7.3 Network Failure Analysis

As discussed in Section 7.2.4, there are three possible combinations of failures on the network: 1) failure in the linear access network, 2) failure in the backbone network, and 3) failure in both networks. We study only the errors produced by failures in the access network as they are more likely than failures in the backbone network (assumption stated in Section 3.5). When the vehicle (containing the in-motion network) is passing by a segment of several failed access nodes, the in-motion network is disconnected from the infrastructure network. The duration of this disconnection depends on the speed of the vehicle and it is perceptible when this duration is larger than $\mathcal{T}$ seconds. For example, an on-going data stream is broken if it suffers a disruption of 4 seconds (or more). In this section, we study the probability of observing a segment of l^* contiguous access nodes in a failure state, which causes a disconnection time larger than $\mathcal{T}$ wihtin an observation period of time τ.

We begin by analysing the effect of speed on the disconnection time. Then, we describe the failures in the access network according to the number and location of failed nodes. Furthermore, we analyze the probability of having each of these cases in terms of the length of a segment of contiguous failed nodes. Finally, we draw a conclusion about the

probability of experiencing a network failure producing an observable disconnection from the infrastructure network.

7.3.1 Speed effects on the Disconnection Time

In this section we aim at determining the effect of the speed on the disconnection time of the in-motion network produced by a segment of l^* contiguous failed access nodes. We consider the coverage conditions assumed in Section 3.2.

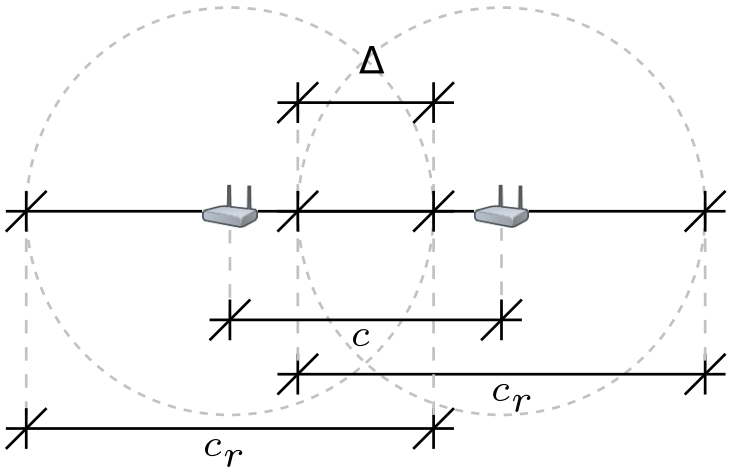


Figure 7.25: Definition of Δ, c_r, and c.

Each access node represents a *Wireless Access Point* device with a radio coverage of c_r meters. The access nodes are placed at c meters apart. Their radio coverages thus overlap by $\Delta = c_r - c$ meters, as it is depicted in Figure 7.25. Thereby, when calculating the distance covered by l^* access nodes under the mentioned conditions, we have:

$$c_r(l) = l^*(2c - c_r) + (l^* - 1)\Delta \tag{7.1}$$

When considering l^* contiguous failed nodes, the disconnection time when moving at v m/s is described by the following expression:

$$t_{disc}(l^*, v) = \frac{c_r(l^*)}{v} = \frac{l^*(2c - c_r) + (l^* - 1)(c_r - c)}{v} = \frac{c \cdot l^* + (c - c_r)}{v} \tag{7.2}$$

Figure 7.26 shows the disconnection time for different number of failed nodes l^* and in terms of the speed of the in-motion network v. Thus, by defining a network failure as $t_{disc} > \mathcal{T}$, we have an upper bound for t_{disc} which leads to an upper bound for l^* and a lower bound for v in order to observe a disconnection time less than $\mathcal{T}$ seconds. For example, letting $\mathcal{T} = 25$ seconds, the network could support one failed node for any speed over 10 km/h; two failed nodes for speeds bigger than 30 km/h; three failed nodes for $v > 60$ km/h, and so on. Table A.3 (in the annexe A) shows in detail the disconnection

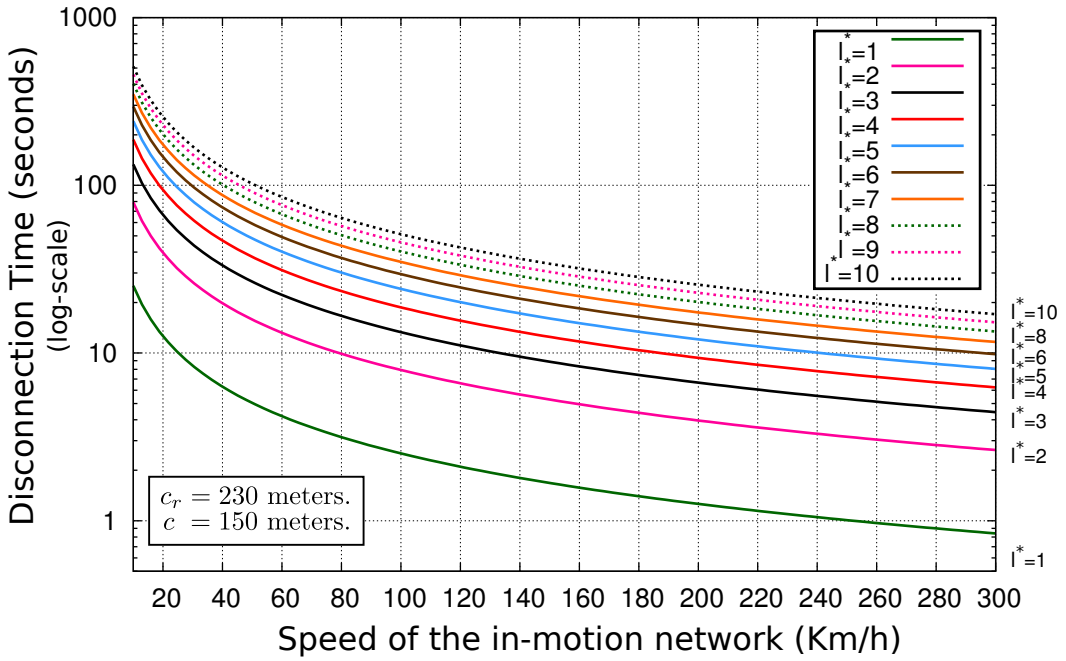


Figure 7.26: Disconnection time (log-scale) for different speeds and different number of failed nodes.

times for $1 \leq l \leq 10$ and $10 \leq v \leq 300$ km/h considering access nodes each $c = 150$ meters, a radio coverage range of $c_r = 230$ meters, and an overlapping distance of 80 meters.

Notice the faster the in-motion network moves, the larger is the number of failed nodes the topology can support. When the in-motion network moves slower than the minimum speed determined by the table A.3, no guarantee can be given that the users will not perceive a disconnection time smaller than $\mathcal{T}$ seconds. In addition, observe that in Expression 7.2, the disconnection time is 0 when the half of the radio coverage range is equal to the distance between nodes. Indeed, when nodes are placed at $\frac{c_r}{2}$ meters each other, actually, they lay at the cell borders, producing a double coverage over the path followed by the in-motion network (on-board the train).

7.3.2 Types of Network Failures at the Linear Access Network

In this section we analyze the possible causes of error that might lead to a network failure. We assume a linear access network of n nodes with m gateways placed every $\mathcal{D}$ (counting one of them) access nodes. We classify a **node failure** as follows:

1. **Error state 1**: when a node is experiencing hardware, power supply, or any other problem that make it not operate as it should.

2. **Error state 2**: when a node is isolated as there is no way to reach the root of the backbone network from it.

The first type of error requires an external intervention to be recovered. The second type of error is automatically recovered when the defective nodes (causing the failure) are recovered. Nevertheless, we emphasize that two nodes in the error state 1 not always causes nodes in the error state 2. From here, we name "defective node" any node that is experiencing an error type 1, and we name "isolated node" any node experiencing an error type 2. Therefore, x defective nodes might produce i isolated nodes, resulting in $l^* = x + i$ failed nodes.

The length of the disconnected segment when there is a single defective node (error state 1) is 1, except when the defective node is near the end of the linear access network and ends nodes are not considered as gateways. For two defective nodes, we identify 4 sub-cases:

1. Two contiguous defective gateways: the length of the disconnected segment is $\mathcal{D} + 1$ access nodes.

2. Two non-contiguous defective gateways: there is at least one operative gateway in between both defective gateways. So, all the access nodes affected by this failure reconfigure their path to reach the root through the operative gateways in between. We have 2 disconnected segments of length 1.

3. One gateway and one access node are defective: when there is at least one operative gateway in between both defective nodes, we have two disconnected segments of length 1. Otherwise, the length of the disconnected segment is at most $\mathcal{D}$.

4. Two access nodes are defective: Similar to the previous case, but when there is no operative gateway in between both defective nodes, the length of the disconnected segment is at most $\mathcal{D} - 1$.

For more defective nodes in the access network, we see that all cases are conditioned to the existence of an operative gateway in between two defective nodes. Thus, when that happen, we have multiple occurrences of a disconnected segment of length 1. In summary, the most failure-prone error cases are when contiguous gateways are defective. These cases lead to a larger number of failed nodes. These failed nodes may cause an observable error (a network failure) when the disconnection time (caused by the failed nodes) is greater than the upper bound $\mathcal{T}$. In the following, we study the probability of observing such a disconnection in order to assess how likely it is to observe a network failure from inside the in-motion network.

7.3.3 Probability Analysis of Network Failure

In this section we describe the probability of having l^* consecutive failed nodes (defective or isolated), causing an observable disconnection of the in-motion network. The event of having l^* failed nodes is triggered by x nodes in a type 1 error state (defective), which may or may not lead to a larger number of failed nodes (together error states 1 and 2). Let

be p the probability of observing a node in an error state of type 1 within an observation period of τ. The probability p is obtained from the mean-time-to-failure (MTTF) when observing a sample of devices working under certain operational conditions. Considering a MTTF of one year and an observation period of one day, the probability of observing a failed node is $p = \frac{1}{365} \approx 0.0027$. As far as we know, there are no published studies on the failure probability of an Access Point device. Therefore, we assume for this study values of p of 0.01, 0.001 and 0.0001, which are reasonable when considering the depicted MTTF and an observation period for a railway scenario.

First, we compute the exact probability of observing l^* consecutive failed nodes for small topology sizes ($n \leq 40$). Second, we present an approximation formula for p small in comparison to n (i.e. $np << 1$) valid for n smaller than few hundreds of access nodes. Third, we present an approximation which is valid for larger values of n and $np^x << 1$. Thus, we sketch a new decision criteria that can be used to design a backbone network in terms of the cost of the network's maintenance operation (observation window τ and hence the probability of failure p).

We consider only regular topologies with a constant number of access nodes between contiguous gateways, let us say $\mathcal{D} = \frac{n-1}{m-1}$, and there is always a gateway at the beginning of the linear access network. Also we define the function $x(l^*)$ as the minimum number of defective nodes (type 1 error) required to have a disconnected interval of length l^*. Thus, we have a $i(l^*) = x(l^*) - 2$ isolated gateways. Let F be the number of failures (type 1 error) during the time of observation τ. We define the event $\{L = l\}$ as to observe **at least** one disconnected interval of length l^*. The goal in this section is to estimate $\mathbb{P}[L = l]$. By the law of total probability, we have:

$$\mathbb{P}[L = l] = \sum_{x=0}^{n} \mathbb{P}[L = l | F = x]\mathbb{P}[F = x]. \tag{7.3}$$

The distribution of the number of failures during the observation window is given by :

$$\mathbb{P}[F = x] = \binom{n}{x} p^x (1-p)^{n-x}.$$

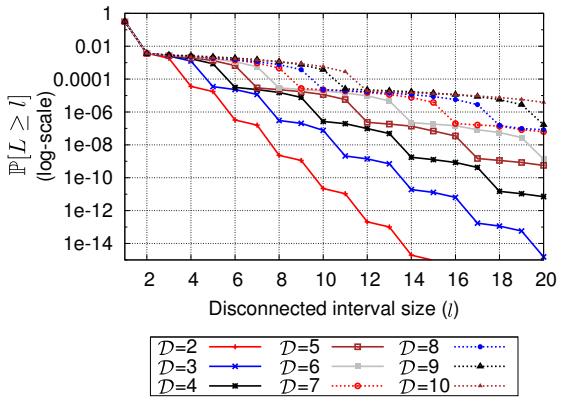
Thus, considering x nodes in the type 1 error state, the probability of observing the event $\{F = x\}$ is ruled by $\mathbb{P}[F = x]$. Therefore, the probability that such event leads to a disconnected interval of l^* nodes is given by $\mathbb{P}[L = l|F = x]$ for $l = l^*$ and $0 \leq x \leq n$. When calculating this latter probability, the size of the combinatorial space to explore is exponential with the topology size, therefore, it is hard to compute. Hence, we start our analysis by studying $\mathbb{P}[L = l|F = x]$ by computing all the possible failure cases for small topology sizes ($n = 40$).

7.3.3.1 Exact Computation for Small Networks.

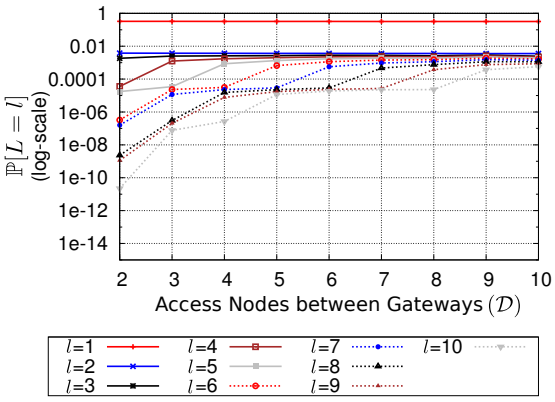
We simulate the failure of x nodes on a network of $n = 40$ nodes with gateways at regular intervals (from 2 to 10 nodes). For each computed failure scenario, we account the length

of the disconnected segments of contiguous failed access nodes in order to compute the probability of observing at least one disconnected interval of size l. More in details, at each step of simulation, we generate each possible scenario of having x defective nodes in the network, giving $\binom{n}{x}$ cases, where we compute the frequency of observing a disconnected interval of size $L = l$. Notice the x defective nodes are included in l. In this way, given a certain l, we compute the individual probabilities of $\mathbb{P}[L = l|F = x]$ for $1 \leq x \leq n$, and we iterate this process for $1 \leq l \leq 20$ (half of the topology size). As we stated before, we study the resulting network failure probability for three node's individual type 1 error probabilities: $p = 0.01$, 0.001 and 0.0001.

Figure 7.27 shows for $p = 0.01$ the resulting probabilities of observing **at least** one disconnected interval of size **equal to or larger than** l for topologies with gateways placed at regular intervals of $2 \leq \mathcal{D} \leq 10$ nodes. Indeed, by letting $l = 2$, we are stating that the in-motion network will suffer from a disconnection when it observes at least one disconnected interval of length equal to or larger than 2. On the contrary, when the in-motion network crosses at least one disconnected segment of length 1, it will not notice the disconnection due to speed of the train. Figure 7.27a shows the evolution of $\mathbb{P}[L \geq l]$ according to the disconnected interval length l. The length of the disconnected segment is plotted for $2 \leq \mathcal{D} \leq 10$. In addition, we present the same probability $\mathbb{P}[L \geq l]$ at the Figure 7.27b, but how depicting its evolution according to $\mathcal{D}$ for a fixed value of l.



(a) $\mathbb{P}[L \geq l]$ for $l = 1..20$



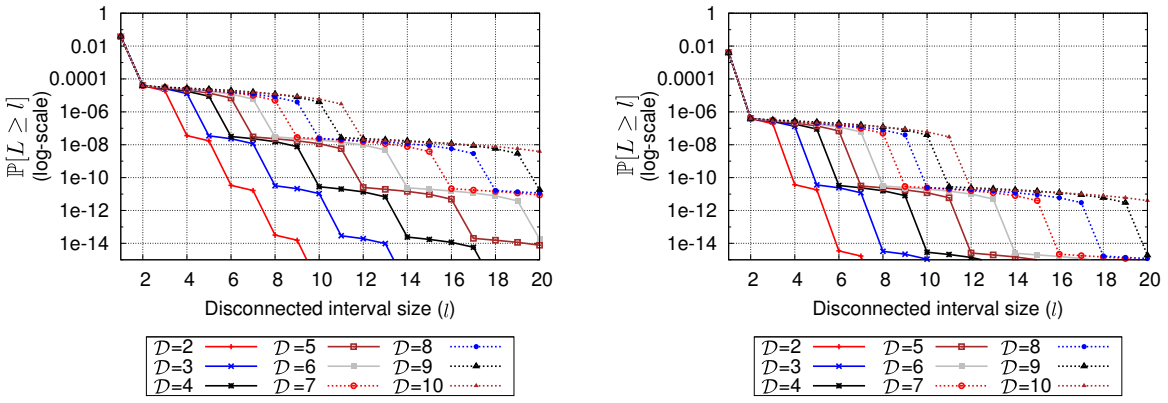
(b) $\mathbb{P}[L = l]$ for $\mathcal{D} = 2..10$

Figure 7.27: Probability of observing a least a segment of l access nodes for $n = 40$ and $p = 0.01$

The Ladder Effect.

Note that the probability of observing a disconnected interval of $L \geq 1$, or in simpler words, at least one failed node, is approximately the same for all the evaluated values of $\mathcal{D}$. Thus, $\mathbb{P}[L \geq 1] \approx 0.32$. This probability is the resulting addition of $\mathbb{P}[L = 1] + \mathbb{P}[L = 2] + \ldots + \mathbb{P}[L = n]$. For larger values of l, the $\mathbb{P}[L \geq l]$ suffers a drop in steps of $\mathcal{D}+2$ nodes. We explain this behaviour by two facts: 1) the probability of observing exactly $\mathbb{P}[L = l]$ is

dominated by first terms of the summation 7.3; and 2) the number of the defective nodes required to obtain a disconnected segment of l access nodes. Indeed, in the summation 7.3, note that $\mathbb{P}[F = x] = \binom{n}{x} p^x (1-p)^{n-x} \approx O(p^x)$, and for p small, $\mathbb{P}[F = x]$ tends to 0 according to x, or in other words, $\mathbb{P}[F = x] >> \mathbb{P}[F = x+1]$, for $x \geq 1$. Therefore, each term added to the summation is each time smaller. Furthermore, to obtain $L = 1$, we require at least one defective node ($x \geq 1$); to obtain $L = 2$ or $L = 3$, we require at lest 2 defective nodes ($x \geq 2$); for $L = \mathcal{D} + 2$, we require at least 3 defective nodes; and each time that L is multiple of $\mathcal{D} + 2$, we require an additional defective node. Therefore, as for $x < x(l^*)$, $\mathbb{P}[L = l | F = x] = 0$, the summation 7.3 begins from the minimum number of defective nodes required to obtain a segment of l^* failed nodes. Hence, as this number increases according to l^*, the summation becomes smaller in steps of $\mathcal{D} + 2$. We call this effect the "ladder effect". Figure 7.28, shows this effect for $p = 0.001$ and $= 0.0001$, which becomes more evident, since the dominant term of the summation 7.3 becomes smaller, generating a larger drop in steps of $\mathcal{D} + 2$.



(a) $p = 0.001$ (b) $p = 0.0001$

Figure 7.28: $\mathbb{P}[L \geq l]$ for $n = 40$, $p = 0.001$ and $p = 0.0001$

7.3.3.2 Analytical Approximation for a hundred of nodes

When the probability of the error type 1 is small ($p \to 0$), the expression 7.3 can easily be approximated. The dominant term is the one with the minimum number of failures causing a disconnected interval of length l, as expressed in Theorem 7.3.1 (as a matter of fact, $\mathbb{P}[L = k | F = x] = O(1)$ and $\mathbb{P}[F = x] = O(p^x)$). We first introduce some notations:

Definition 7.3.1. *Minimum number of failures:*

- *$x(l)$ is the minimum number of failures necessary to have a disconnected interval of size l, i.e., $x(l) = 2 + \lfloor \frac{l-2}{\mathcal{D}} \rfloor$.*
- *We also use the notation $i(l) = x(l) - 2$ to express the isolated nodes by an interval of size l.*

Proposition 7.3.1. (Approximation of $\mathbb{P}[\mathbf{L = l}]$) *The probability to observe a disconnected interval of length l is well approximated by*

$$\mathbb{P}[L = l] \underset{p\to 0}{=} \mathbb{P}[L = l|F = x(l)]\mathbb{P}[F = x(l)] + o(p^{x(l)}),$$

which can be expressed as

$$\mathbb{P}[L = l] \approx (m - 1 - i(l))(\mathcal{D} + 2 - (l - i(l)\mathcal{D}))p^{x(l)}.$$

Proof. The approximation is given by $\mathbb{P}[L = l|F = x] = O(1)$ and $\mathbb{P}[F = x] = O(p^x)$.

The expression is given by Proposition 7.3.1 and

$$\mathbb{P}[F = x] = \binom{n}{x} p^x (1 - p)^{n-x} \approx \binom{n}{x} p^x.$$

□

Lemma 7.3.1. *For $2 \leq l \leq \mathcal{D} + 1$,*

$$\mathbb{P}[L = l|F = 2] = \frac{(m - 1)(\mathcal{D} + 2 - l)}{\binom{n}{2}}.$$

Proof. When the system experiences two errors of type 1, we have a disconnected interval if the two failures happen in the same segment, or in other words, in between the same pair of gateways. In this segment, there are $\mathcal{D} + 2 - l$ cases of failures that form a disconnected interval of length l. As there are $m - 1$ intervals, we get the result. Note that to have a disconnected interval of length $> \mathcal{D} + 1$, more than two failures are necessary. □

Proposition 7.3.2. *More generally, we have for $2 \leq l \leq n$,*

$$\mathbb{P}[L = l|F = x(l)] = \frac{(m - 1 - i(l))(\mathcal{D} + 2 - (l - i(l)\mathcal{D}))}{\binom{n}{x(l)}}.$$

Proof. We adapt the proof of the previous lemma. Note that when the first failure is chosen, there no choice for the other failures. The $i(l)$ next failures have to be at the next following gateways. The last failure has to be at distance $l - 1$ of the first one. For the first failure, we have $\mathcal{D} + 2 - (l - (x - 2)\mathcal{D})$ possibilities in an interval. We have $m - 1 - (x - 2)$ possible segments, giving the formula. □

7.3.4 Analytical Approximation for large topology sizes

We call *syndrome*, or shortly S, to observe a disconnection of length l. We define the event S^i as "There exists a syndrome (of length $\geq l^*$) starting at node i". We note $\mathcal{S}$ the event "There exists **at least** one syndrome in the network". We have $\mathcal{S} = \cup_{i \in N} S^i$.

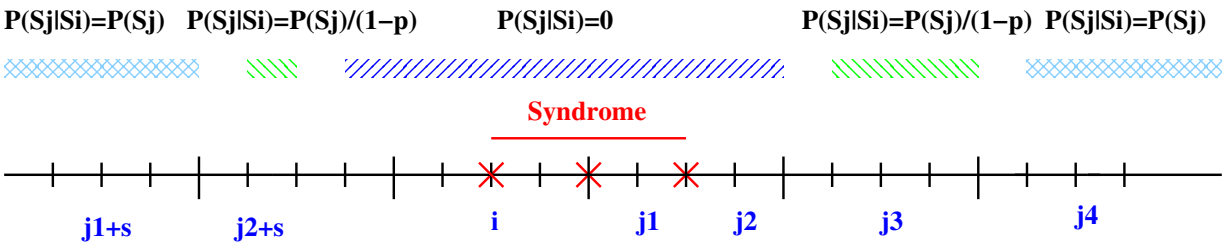


Figure 7.29: Syndrome dependence: $\mathbb{P}[S^j|S^i] \leq \mathbb{P}[S^j]/(1-p)$.

It gives

$$\mathbb{P}[\mathcal{S}] = \sum_{i \in N} \mathbb{P}[S^i] - \sum_{i,j \in N} \mathbb{P}[S^i \cap S^j] + \dots$$

By definition, we have $\mathbb{P}[S^i \cap S^j] = \mathbb{P}[S^i|S^j]\mathbb{P}[S^j]$. Note now that it is impossible to have two overlapping syndromes since the occurrence of such case implies to observe a larger syndrome of length $s_i + s_j - \delta$, being δ the number of nodes they overlap (see Figure 7.3.4) and s_i the length of the syndrome S_i. Therefore, we have have:

$$\begin{array}{ll} \mathbb{P}[S^i|S^j] = 0 & \text{if } j \text{ and } i \text{ "overlaps"} \\ \mathbb{P}[S^i|S^j] \leq \mathbb{P}[S^j]/(1-p) & \text{if } j < i \text{ or if } j \text{ and } i \text{ "close"} \\ \mathbb{P}[S^i|S^j] = \mathbb{P}[S^i] & \text{otherwise} \end{array}$$

where overlaps means $i - (i \mod \mathcal{D}) - 1 \leq j \leq (i + s_i - (i + s_i \mod \mathcal{D}) + \mathcal{D} + 1$. Hence,

$$\mathbb{P}[S^i \cap S^j] \leq \frac{\mathbb{P}[S^i]\mathbb{P}[S^j]}{1-p}.$$

Similarly, when intersecting all possible syndromes in the topology, we have

$$\mathbb{P}[\cap_{l^*} S^{i_l^*}] \leq \frac{\prod_{l^*} \mathbb{P}[S^{i_l^*}]}{(1-p)^{l^*-1}}.$$

Hence, $\mathbb{P}[\cap_z S^{i_z}] \leq N^z \max_i P[S^i]^z/(1-p)^z$. As $\forall i \in N\ \mathbb{P}[S^i]/(1-p) << 1$ (anticipating from Equation 7.6, we get

$$\mathbb{P}[\mathcal{S}] \approx \sum_{i \in N} \mathbb{P}[S^i].$$

Therefore, we compute the probability to have a syndrome of length l starting at node i, denoted by $\mathbb{P}[S_l^i]$. We define $\mu = l - 2 \mod \mathcal{D}$.

There exist two classes of nodes.

- First case ($0 \leq i \mod n_1 < n_1 - \mu$): $x(l^*)$ failures are necessary to disconnect the interval. The probability of a syndrome is

$$\mathbb{P}_1[S] = p^{x(l^*)}(1-p)^{(i(l^*)+1)\mathcal{D}-l+1}.$$

We have ν_1 such cases with

$$\nu_1 = (\mathcal{D} - \mu)\lfloor \frac{n - 1 - l^*}{\mathcal{D}} + \mathbb{1}_{\mu \neq n-1} \rfloor.$$

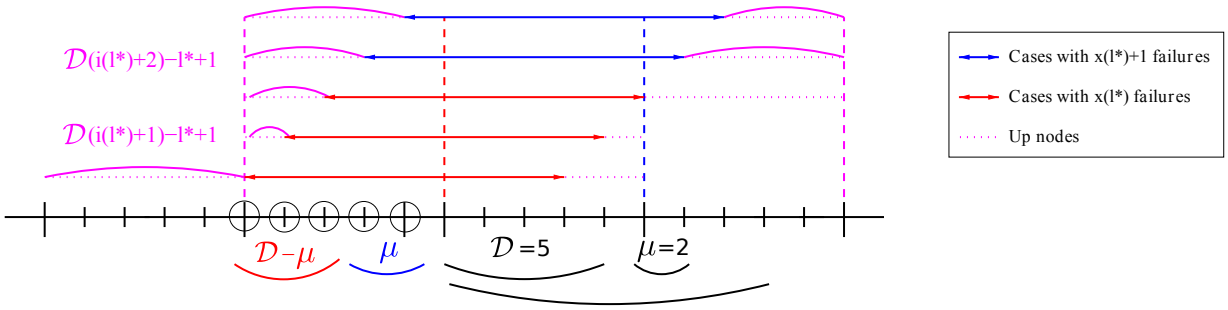


Figure 7.30: Sketch of the proof for a syndrome

- Second case: $x(l^*)+1$ failures are necessary to disconnect the interval. The probability of a syndrome is

$$\mathbb{P}_2[S] = p^{x(l^*)+1}(1-p)^{(i(l^*)+2)\mathcal{D}-l^*+1}.$$

  We have ν_2 such cases with

$$\nu_2 = \mu\lfloor\frac{n-1-l^*}{\mathcal{D}}\rfloor + 1 + \mathbb{1}_{\mu\neq n-1}.$$

We verify that we have $\nu_1+\nu_2 = n-l+1$ possible syndromes.

Proof. In this way, given the disconnected interval length l^*, the first and last nodes of the interval, plus the intermediate gateways have to fail to obtain a syndrome. It happens with probability $p^{x(l^*)}$ (first case) and $p^{x(l^*)+1}$ (second case). The nodes inside the interval can have any state. The nodes outside the interval between the two[1] neighboring gateways have to be up and running. There are $(i(l^*)+1)\mathcal{D}-l^*+1$ of them (first case) and $(i(l^*)+2)\mathcal{D}-l^*+1$ of them (second case), giving the probabilities of failure. To obtain ν_1 and ν_2, consider the nodes between two gateways and the first gateway (the $\mathcal{D}$ circled nodes in Figure 7.3.4). If the first node of the disconnected interval is one of the $\mathcal{D}-\mu$ first such nodes, only $x(l^*)$ failures are necessary (first case). For the next μ, we are in the second case. There are $\lfloor\frac{n-1-l^*}{\mathcal{D}}\rfloor$ such intervals. The remaining terms of the formulas deal with the border of the network (first node and last non complete interval). □

Therefore, when p is small,

$$\mathbb{P}_1[S] = p^{x(l^*)}(1-p)^{(i(l^*)+1)\mathcal{D}-l+1} \approx p^{x(l^*)}, \tag{7.4}$$

$$\mathbb{P}_2[S] = p^{x(l^*)+1}(1-p)^{(i(l^*)+2)\mathcal{D}-l+1} \approx p^{x(l^*)+1}. \tag{7.5}$$

Hence $\mathbb{P}_1[S] >> \mathbb{P}_2[S]$.

[1] There are two special cases on the border of the network that we not consider for this analysis

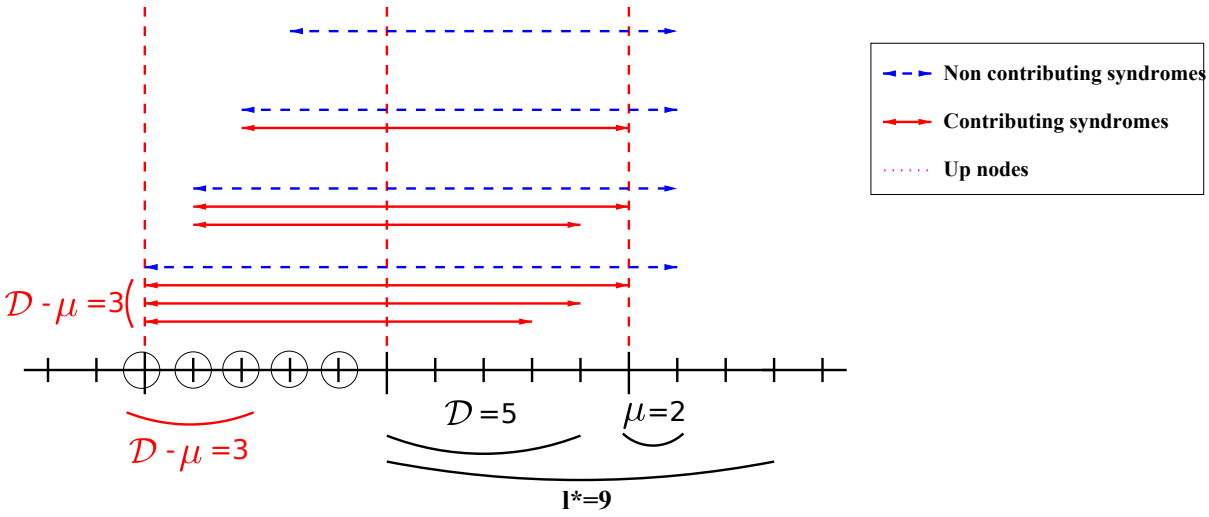


Figure 7.31: Length of the contributing syndromes starting in an interval.

Approximation of $\mathbb{P}[\mathcal{S}]$ when p is small. We have seen that

$$\mathbb{P}[\mathcal{S}] \approx \sum_{i \in N} \mathbb{P}[S^i] \quad \text{and} \quad \mathbb{P}[S^i] = \sum_{z \geq l} P[S^i_z],$$

giving

$$\mathbb{P}[\mathcal{S}] \approx \sum_{i \in N} \sum_{z \geq l} P[S^i_z].$$

Equations (7.4) and (7.5) say that $P[S^i_z] \approx p^{x(z)}$ or $P[S^i_z] \approx p^{x(z)+1}$. As $x(z)$ is a non decreasing function, only the z such that $x(z) = x(l^*)$ are contributing to the double sum. Furthermore, only the nodes of the first class are contributing to $\mathbb{P}[\mathcal{S}]$ (basically, only the terms in $p^{x(l^*)}$ contribute).

We thus only express $\mathbb{P}[S^i]$ for the nodes of the first class. In an interval (syndromes starting in $\mathcal{D} \leq i + j$ with $0 \leq j < \mathcal{D}$), we have $\mathcal{D} - \mu$ nodes of class 1 (nodes of index $0 \leq i \mod \mathcal{D} < \mathcal{D} - \mu$), see Figure 7.3.4. For the last node, only a syndrome of size l^* contributes. For the one before last node, two syndromes of sizes l^* and $l^* + 1$ contribute. More in general, $\mathcal{D} - \mu - j$ syndromes of size $l^* \leq j \leq l^* + \mathcal{D} - \mu - 1$ contribute. We get

$$\mathbb{P}[S^i] \underset{p \approx 0}{=} (\mathcal{D} - \mu - 1 - (i \mod \mathcal{D}))p^{x(l^*)}. \tag{7.6}$$

It gives

$$\mathbb{P}[\mathcal{S}] \approx \lfloor \frac{n-1-l}{\mathcal{D}} + \mathbb{1}_{\mu \neq n-1} \rfloor \sum_{k=1}^{\mathcal{D}-\mu} k p^{x(l^*)},$$

yielding the following result

Theorem 7.3.1. *When $np^{l^*} << 1$, the probability to experience* **at least** *one syndrome in the network is well estimated by*

$$\mathbb{P}[\mathcal{S}] \approx \lfloor \frac{n-1-l}{\mathcal{D}} + \mathbb{1}_{\mu \neq n-1} \rfloor \frac{1}{2}(\mathcal{D}-\mu)(\mathcal{D}-\mu+1)p^{x(l^*)}, \tag{7.7}$$

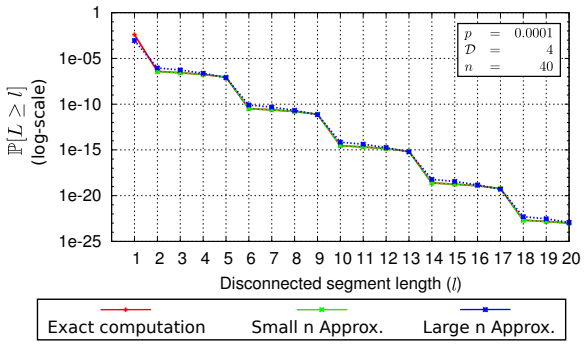
where $\mu = l^* - 2 \mod \mathcal{D}$ *and* $x(l^*)$ *is the minimum number of failures necessary to have a disconnected interval of size* l^**, i.e.,* $x(l^*) = 2 + \lfloor \frac{l^*-2}{\mathcal{D}} \rfloor$.

7.3.5 Discussion

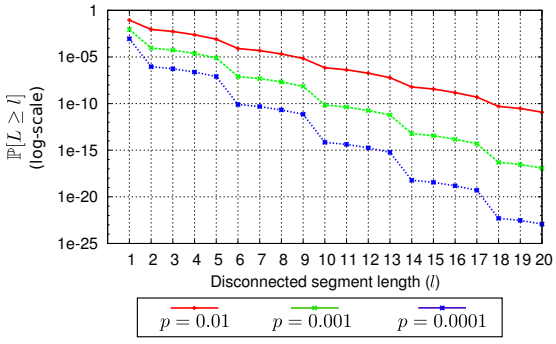
In this section we compare for a small n both analytical approximations with the exact computation of the probability of observing a disconnected segment of l^* nodes. We aim at assessing how good these approximations are. Then, we approximate $\mathbb{P}[L \geq l]$ for a larger topology size in order to sketch a selection criteria based in the relation of the number of access nodes between contiguous gateways and the probability of observing a disconnected segment of at least l^* nodes.

7.3.5.1 Validation of the Analysis

When comparing for small values of n, $\mathbb{P}[L \geq l]$ obtained by the approximations and the exact computation (for $1 \leq l^* \leq 20$ and for $2 \leq \mathcal{D} \leq 10$), we found that the analytical approximations deliver a good estimation of $\mathbb{P}[L \geq l]$. Figure 7.32a shows the two approximations and the exact probability computation for $n = 40$, $p = 0.0001$ and $\mathcal{D} = 4$ as an example of the obtained results when comparing the three evaluated failure probabilities for different values for $\mathcal{D}$.



(a) $\mathbb{P}[L \geq l]$ for $p = 0.0001$.



(b) $\mathbb{P}[L \geq l]$ for $p = 0.01$, $p = 0.001$ and $p = 0.0001$.

Figure 7.32: $\mathbb{P}[L \geq l]$ for $n = 40$ and $\mathcal{D} = 4$

7.3.5.2 Effect of the Number of Nodes Between Gateways

The results confirm that we have a ladder effect as defined in Section 7.3.3.1. There is a drop in the probability when l^* goes from $i \cdot \mathcal{D}+2$ to $i \cdot \mathcal{D}+3$ for $i = 1, 2, 3, \ldots$. Observe that the ladder effect is smoother for larger values of p (see Figure 7.32b). This fact suggests

that for a large p, and $l^* = \mathcal{D} + 2$, $\mathbb{P}[L \geq l]$ is not that different than $\mathbb{P}[L \geq l + 1]$. On the contrary (for p small), $\mathbb{P}[L \geq l] >> \mathbb{P}[L \geq l + 1]$, and this effect is repeated for l^* in "steps" of $\mathcal{D} + 2$.

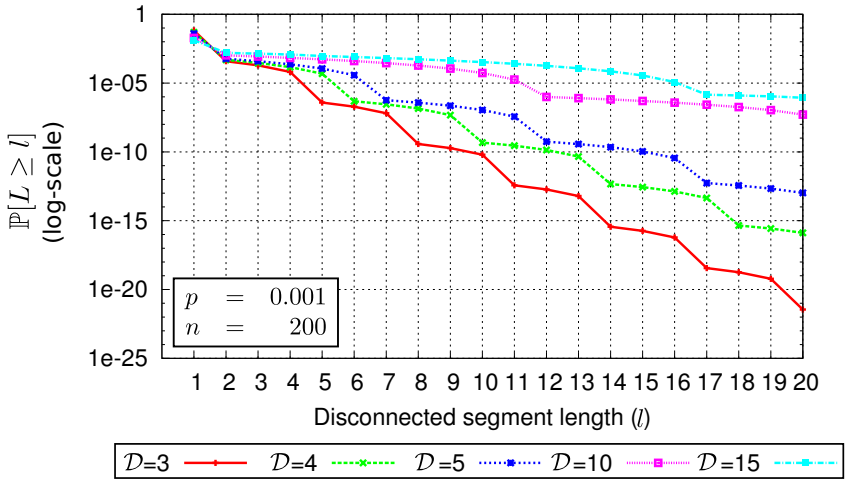


Figure 7.33: $\mathbb{P}[L \geq l]$ $n = 200$ and $p = 0.001$.

Furthermore, when estimating $\mathbb{P}[L \geq l]$ for a larger topology size, let us say $n = 200$ and $n = 2000$, we obtain interesting results respecting the number of nodes between contiguous gateways. Figure 7.33 shows the probability of observing a disconnected interval of l^* nodes for an access network of 200 nodes considering different values for $\mathcal{D}$. Note in the figure that the probability of observing an interval of 2 nodes is not much different than observing 5 or 6 access nodes for $\mathcal{D} = 10$ (or 15) (as much 20% of difference). In other words, for $l \leq \mathcal{D}$, $\mathbb{P}[L \geq l + 1] \approx 0.8\mathbb{P}[L \geq l]$. This result suggests that it is more probable to observe a disconnected interval of l^* nodes in a network exhibiting a large $\mathcal{D}$ than in a network with a small $\mathcal{D}$. Hence, larger values of $\mathcal{D}$ yield to a larger $\mathbb{P}[L \geq l]$ (for the same value of l^*).

Repeating the analysis for a topology of 2000 access nodes, we observe a similar result, but evidencing a larger probability of $L \geq l^*$. For example, for $\mathcal{D} = 5$ and $p = 0.001$, $\mathbb{P}[L \geq 2] = 0.00059$ for $n = 200$ and $\mathbb{P}[L \geq 2] = 0.00599$ for $n = 2000$ (one order of magnitude in difference). This probability might be not negligible when considering an access network of 2000 access nodes (about 600 km long).

In summary, the number of access nodes between gateways does affect the probability of observing a disconnected interval of l^* access nodes. And when considering a relatively small probability of failure of an access point device ($p = 0.001$), the probability of observing more than 2 disconnected access nodes is about 0.6% for a topology of 200 access nodes with 5 access nodes between gateways. On the contrary, for a topology of 2000 nodes with the 5 access nodes between gateways, the probability of observing more than 2 disconnected nodes is about 6%, which is not negligible.

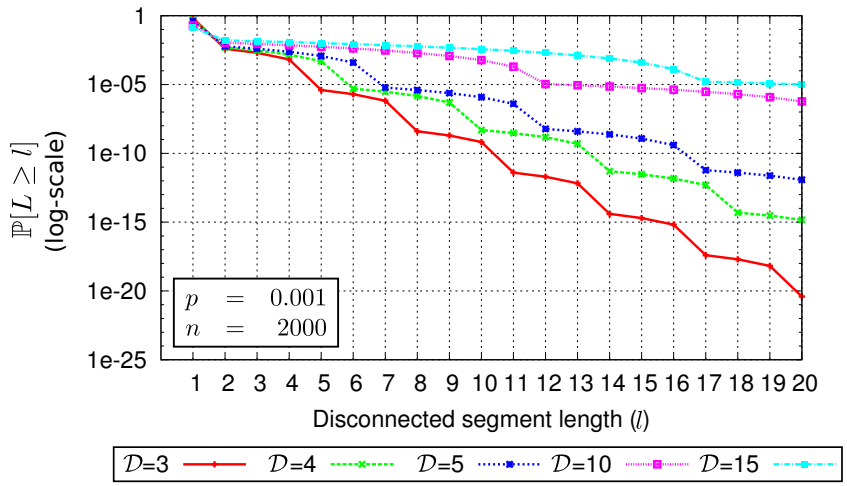


Figure 7.34: $\mathbb{P}[L \geq l]$ $n = 2000$ and $p = 0.001$.

7.3.6 Concluding Remarks

In this section we studied the probability of observing a disconnected interval of l^* contiguous nodes in a linear access network. Such a disconnection causes an observable error for the end-user as the in-motion network loses its connection with the infrastructure network for a long period of time. Note that the number of contiguous disconnected nodes l^* depends on the speed of the vehicle containing the in-motion network.

We have seen that the probability of observing **at least** one disconnected interval of l^* access nodes ($\mathbb{P}[L \geq l]$) is well approximated by the formulas presented in this section and it decreases when l^* increases. We showed that $\mathbb{P}[L \geq l]$ is in the order of $p^{x(l^*)}$, with $x(l^*)$ the minimum number of defective nodes required to have a disconnected segment of l^* access nodes. As $x(l^*)$ depends on $\mathcal{D}$, a small number of access nodes between gateways gives a larger $x(l^*)$, which yields a lower probability of observing a disconnected segment of l^* nodes. Furthermore, the ladder effect is also explained by $x(l^*)$, since $\mathbb{P}[L \geq l]$ is of the same order of $p^{x(l^*)}$ in steps of $\mathcal{D}+2$ disconnected access nodes. Thus, when passing from $l^* = \mathcal{D}+2$ to $l^* = \mathcal{D}+3$ we see that $\mathbb{P}[L \geq l] >> \mathbb{P}[L \geq l+1]$ for a first step, and in a second step between $2\mathcal{D}+2 >> 2\mathcal{D}+3$, an so on. However, this effect is minimized when p is large (for example ≈ 0.01). Hence, the number of access nodes between gateways has an important impact on the probability of observing a disconnection from the network. However, for small values of l^* (1 or 2), the impact of $\mathcal{D}$ is not important. These cases are possible when the train moves at low speed. Thus, in order to cope with these cases, the train might use two *Spiderman Devices* (in the first carriage and in the last carriage) in order to exploit the length of the train to bypass the disconnected nodes.

Nevertheless, this analysis only addresses the event $\{L = l\}$ (to observe **at least** one disconnected interval of length l^*). No information about how many times the in-motion network might observe this event is given. The probability to observe t times the event $\{L = l\}$ in network of n access nodes can be modelled as a *negative binomial* distribution

with parameters n, t (number of events), and the probability $\hat{p}$ of observing **exactly** one event $\{L \geq l\}$ on the network. This latter probability can be estimated with the same method presented in this section. This analysis will be addressed as further work.

Chapter 8

Selection of a Backbone Topology for a Linear Access Network

In this chapter we aim at selecting a backbone topology for a linear access network of n nodes. Our selection is based on the topologies studied in Chapter 6 and our criteria is the following: firstly, we want **to have the smallest distance to the root** (in number of hops); secondly, we want **to have the best resilience properties** as possible; and finally, we wish to have a reasonable deployment cost. We focus on the selection of a backbone topology for two linear access networks: the first one with 200 nodes (30 km long) and the second one with 2000 nodes (300 km long). For each topology size, we evaluate backbone networks formed by a **single topology** as well as from a **combination of two topologies**. At the end of this chapter we summarize our findings proposing a backbone topology design for a linear access network.

8.1 Introduction

From the parametric analysis presented in the previous chapter, we learn that we cannot determine a single topology exhibiting good values in all their properties. We loose in one property in return to gain in another. This inherent trade-off suggest that we need to choose which property we want to optimize first, then evaluate the topologies according to this property, and then we need to choose among the others which property (or properties) are we willing to lose. Evaluating which property we can loose, we present the following arguments:

- **We need to have a small maximum distance to the root** due to the following reasons: 1) a low number of hops to the root minimizes the delay caused by possible queuing effects at each hop; 2) the convergence of the routing protocol, which is not assured for more than 7 hops to the root node; 3) the handover scheme proposed in chapters 4 and 5 requires a low number of hops to the root of the network when considering a routing scheme that configures the network as a tree (as a spanning tree routing like protocol does).
- **We can increase the number of access nodes between contiguous gateways** ($\mathcal{D}$). From Section 7.3, we learn that the probability of observing a disconnected interval of at least l access nodes ($\mathbb{P}[L \geq l]$) depends on $\mathcal{D}$ when $l > 2$. The smaller is the value of $\mathcal{D}$, the lower is $\mathbb{P}[L \geq l]$. However, for $l = 1$ or $l = 2$, $\mathbb{P}[L \geq l]$ is not

affected by $\mathcal{D}$. Therefore, 1) there is no reason to have 1 or 2 access nodes between gateways; and 2) when access nodes have a low probability of failure ($p < 0.001$), there is no risk in increasing $\mathcal{D}$ over 3. Hence, when considering that a spanning tree based routing protocol converges quickly for $d_{max} \leq 7$, a number of 3 or 4 access nodes between contiguous gateways seems to be reasonable.

- The resilience of the studied networks depends mostly on the type of the topology since the larger the number of blocked links within the backbone network is, the better are the chances that the routing protocol has to find an alternative path to the root in case of failure. So, the consequences of reducing the resilience are a lower deployment cost (in terms of the number of gateways and total length of links) and a higher maximum distance to the root (as it is the case when using linear kernel topologies). On the contrary, a higher resilience will be accompanied by an increment in cost and a reduction on the maximum distance to the root.

Considering these arguments, the selection criteria should prioritize the maximum number of hops to the root. As this property is directly related to the resilience of the network, we assign also priority to the number of blocked links. Then, we explore the number of access nodes between contiguous gateways and the deployment cost in order to find a good trade-off between them. Thus, when considering a single topology, the only parameter we use to adjust their properties is the recursion level k. However, as the difference in deploying k or $k+1$ layers might be large for some topologies, we combine two topologies in order to have two variables to adjust: the type of each topology and their recursion level. Therefore, we use the proposed criteria to select a solution for a fixed size of the access network by exploring solutions using a single topology as well as a combination of two of them.

8.2 Methodology

We aim at proposing a solution for $n = 200$ and $n = 2000$ access nodes. We consider as feasible solutions only those topologies exhibiting a **maximum distance to the root less than or equal to 7 hops**. This restriction comes from the convergence time of the routing protocol (see Section 3.4). We evaluate all the topologies studied in chapters 6 and 7, having as main criteria to have **the smallest possible maximum distance to the root**. Then, we search among the candidate solutions for the best resilience properties according to their deployment cost. The resilience of the network is expressed in terms of two properties: 1) the number of alternative paths to the root within the backbone network; and 2) the number of access nodes between contiguous gateways. The number of alternative paths to the root is related to the number of links blocked by the routing protocol since they cause a loop within the backbone network. Thus, the larger is the number of blocked links, the larger is the number of alternatives the routing protocol has to cope with a failure. Therefore, **we aim to have the larger number of blocked links**. The number of access

nodes between contiguous gateways is related to the probability of observing a disconnected interval of l^* access nodes (failed or isolated). Thus, as discussed in the previous section, we search for topologies exhibiting around **3 or 4 access nodes between contiguous gateways**. Furthermore, we express the **deployment cost** by the linear combination of the number of gateways and the total length of links: $\alpha m(n,k) + \beta\mathcal{L}(n,k)$, where α is the cost of to acquire and deploy one backbone node and β is the cost of to acquire and deploy c meters of link[1]. We define an equivalence δ between the cost of one backbone node and c meters of link. This equivalence allows us to compute a relative cost when comparing topologies.

We evaluate backbone networks formed of a **single topology** as well as formed from a **combination of two of topologies**. We contrast both solutions assessing whether a single topology or a combined topology delivers a better solution in terms of the balance of their properties. We restrict the number of layers for all the topologies to such values of k allowed by the size of the access network. In other words, for both access networks' sizes we evaluate up-to $k_{max}(200)$ and $k_{max}(2000)$ respectively. In addition, we assume a ratio between $m(n,k)$ and $\mathcal{L}(n,k)$ of $\delta = 1:5$ (the cost of deploying $5c$ of links is similar to acquire 1 backbone node).

8.3 Linear Access Network of 200 Nodes

A linear access network of $n = 200$ nodes covers a trajectory of about 30 km when considering a distance between access nodes of $c = 150$ meters and an overlapping distance of 80 meters. This distance is comparable to trajectories followed by urban buses, a metro lines, or any mass transit system. We divide this section into two parts: in the first part, we propose a backbone network only by using a single topology; and in the second part, we propose a backbone network by using a combination of two topologies. Our goal is to find a solution exhibiting a good trade-off between their network properties while keeping the maximum distance to the root as low as possible and the resilience as high as possible (high in terms of a large number of blocked links and 3 or 4 access nodes between contiguous gateways).

8.3.1 Solutions for a Single Topology

Firstly, we evaluate the **Chordal-2** topology. Table 8.1 depicts the resulting properties until the maximum step of recursion allowed by the access network size ($k_{max} = 7$). We discard the topologies for $k \leq 4$ since they exhibit $d_{max} > 7$. Therefore, the feasible solutions are $5 \leq k \leq 7$. The solutions for $k = 6$ and $k = 7$ meet our first requirement (smallest maximum distance to the root). When comparing them, we note that $k = 7$ provides the largest number of blocked links, but we loose the property of 3 or 4 access nodes between gateways. On the contrary, the solution for $k = 6$ meets the criteria of number of

[1] c is the distance between two contiguous access nodes

access nodes between gateways, but reduces by 50% the blocked links. According to the conclusions about the number of access nodes between gateways, 1 or 2 access nodes report not much gain in terms of resilience. Therefore, we prefer to reduce number of blocked links in order to have the minimum deployment cost. Hence, we choose $k = 6$.

Properties	Recursion Level (k)						
	1	2	3	4	5	6	7
Number of gateways	3	5	9	17	33	65	129
Total length of links	199	398	597	796	995	1194	1393
Cost of the topology ($\delta = 1:5$)	214	423	642	881	1160	1519	2038
Maximum distance to the root	50	26	14	8	5	4	4
Number of blocked links	0	2	6	14	30	62	126
Length of blocked links	0	99	199	297	393	489	582
Min. dist. between gateways	99	49	24	12	6	3	1
Max. dist. between gateways	100	50	25	13	7	4	2

Table 8.1: Solutions for $n = 200$ provided by Chordal-2 topology

Secondly, we evaluate the **$JC^2 - lin3$ topology**. Their resulting network properties are shown by Table 8.2. We discard the solutions for $k \leq 2$, since they have $d_{max} > 7$. The remaining two feasible solutions do not provide blocked links, so, the resilience provided by them is based only on the number of access nodes between gateways. The solution for $k = 4$ gives the lowest maximum distance to the root, and gateways placed every 2 or 3 access nodes seems to be reasonable when no blocked links are provided. Note that the other solution reduces the deployment cost by the half, but it increases the number of access nodes between gateways. Hence, we prefer $k = 4$.

Properties	Recursion Level (k)			
	1	2	3	4
Number of gateways	3	9	27	81
Total length of links	133	266	402	547
Cost of the topology ($\delta = 1:5$)	148	311	537	952
Maximum distance to the root	34	13	7	5
Number of blocked links	0	0	0	0
Length of blocked links	0	0	0	0
Min. dist. between gateways	66	22	7	2
Max. dist. between gateways	67	23	8	3

Table 8.2: Solutions for $n = 200$ provided by linear kernel with $s = 3$.

Thirdly, the **$JC^2 - lin5$ topology**. Table 8.3 shows its network properties up-to 3 levels of recursion ($k_{max} = 3$). The only feasible solution for this topology is $k = 3$.

Properties	Recursion Level (k)		
	1	2	3
Number of gateways	5	25	99
Total length of links	160	320	468
Cost of the topology ($\delta = 1:5$)	185	445	963
Maximum distance to the root	22	8	7
Number of blocked links	0	0	0
Length of blocked links	0	0	0
Min. dist. between gateways	40	8	2
Max. dist. between gateways	40	8	2

Table 8.3: Solutions for $n = 200$ provided by linear kernel with $s = 5$.

Fourthly, the **JC² – *lin7* topology**. Note that in spite of it can be deployed for $k \leq 3$, there is no new gateways and links when deploying $k = 3$ due to the restriction on the length of links (all segments created in $k = 3$ have a length less than 3). Hence, the topology for $k = 3$ is the same as the topology for $k = 2$. Therefore, we show in Table 8.4 only $k = 1$ and $k = 2$. The two solutions provided by this kernel are not feasible, since both violate the solution feasibility criteria of $d_{max} \leq 7$. Hence, we discard this kernel.

Properties	Recursion Level (k)	
	1	2
Number of gateways	7	49
Total length of links	171	342
Cost of the topology ($\delta = 1:5$)	206	587
Maximum distance to the root	17	8
Number of blocked links	0	0
Length of blocked links	0	0
Min. dist. between gateways	28	4
Max. dist. between gateways	29	5

Table 8.4: Solutions for $n = 200$ provided by linear kernel with $s = 7$.

Fifthly, the **JC² – *star5* topology**. Table 8.5 shows the resulting network properties. The solutions for $k = 2$ and $k = 3$ are the only feasible solutions and both do not provide blocked links. Therefore, the resilience of these solutions is based only on the number of access nodes between contiguous gateways. We choose $k = 3$ arguing the same argument that we used for linear kernels with $s = 3$.

Sixthly, the **JC² – *star7* topology**. Table 8.6 shows the resulting network properties. The feasible solutions are $k = 2$ and $k = 3$. The solution for $k = 3$ meets the criteria for the maximum distance and blocked links, but it misses (or over-provision) the number of access nodes between gateways. Thus, when comparing $k = 3$ with $k = 2$, we note that a

Properties	Recursion Level (k)		
	1	2	3
Number of gateways	5	25	125
Total length of links	240	480	754
Cost of the topology ($\delta = 1:5$)	265	605	1379
Maximum distance to the root	21	6	4
Number of blocked links	0	0	0
Length of blocked links	0	0	0
Min. dist. between gateways	40	8	1
Max. dist. between gateways	40	8	2

Table 8.5: Solutions for $n = 200$ provided by star kernel with $s = 5$.

reduction of $1/3$ in the deployment cost is traded-off by no blocked links, a larger number of nodes between gateways, and a higher distance to the root. Therefore, it is reasonable to increase in the deployment cost in return to a gain in 2 hops to the root and better a resilience. Hence, we choose $k = 3$.

Properties	Recursion Level (k)		
	1	2	3
Number of gateways	7	49	102
Total length of links	344	689	894
Cost of the topology ($\delta = 1:5$)	379	934	1404
Maximum distance to the root	21	6	4
Number of blocked links	0	0	48
Length of blocked links	0	0	96
Min. dist. between gateways	28	4	1
Max. dist. between gateways	29	5	2

Table 8.6: Solutions for $n = 200$ provided by star kernel with $s = 7$.

Lastly, the **$JC^2 - bin7$ topology**. The feasible solutions are $k = 2$ and $k = 3$. We choose $k = 3$ based on the same arguments that we used for the star kernel with $s = 7$.

8.3.1.1 Selection for a Single Topology

After the evaluation of a backbone network only by using a single topology, we noticed that always we have a property that unbalance the solution. Table 8.8 summarize the selected solutions for each studied topology.

When choosing among the selected solutions according to our selection criteria, we choose the Chordal-2 with $k = 6$, the $JC^2 - star7$ with $k = 3$, and the $JC^2 - bin7$ with $k = 3$, in order of preference. The three topologies propose three different trade-offs between the maximum distance to the root, the number of blocked links, the number of

Properties	Recursion Level (k)		
	1	2	3
Number of gateways	7	49	102
Total length of links	228	455	651
Cost of the topology ($\delta = 1:5$)	263	700	1161
Maximum distance to the root	16	6	5
Number of blocked links	0	0	48
Length of blocked links	0	0	106
Min. dist. between gateways	28	4	1
Max. dist. between gateways	29	5	2

Table 8.7: Solutions for $n = 200$ provided by binary kernel with $s = 7$.

Num.	Topology	k	d_{max}	$\lvert L_{bl}\rvert / L_{bl}$	$\mathcal{D}$	Cost
1	Chordal-2	6	4	62/498	3-4	1519
2	$JC^2 - lin3$	4	5	0/0	2-3	952
3	$JC^2 - lin5$	3	7	0/0	2	963
4	$JC^2 - star5$	4	4	0/0	1-2	1379
5	$JC^2 - star7$	3	4	48/96	1-2	1404
6	$JC^2 - bin7$	3	5	48/96	1-2	1161

Table 8.8: Summary of single topology solutions for 200 nodes.

access nodes between gateways and the deployment cost. We prefer the solution 1 over solutions 5 and 6 since they are indulgent in terms of the resilience of the network. We base our choice in the fact that 1 or 2 access nodes between gateways increase unnecessarily the deployment cost when considering that the most probable failure in the access network is up-to 2 contiguous access nodes. In addition, it seems to be excessive to have gateways 1 or 2 nodes apart when the solution provides alternative paths to the root.

8.3.2 Solutions Combining Two Topologies

In this section we combine two topologies in order to asses whether this type of solution gives a better trade-off than a single topology solution. We combine only 2 types of topologies since the access network size does not left any room for more combinations. Thus, we deploy a **first topology** for values of $1 \leq k \leq 3$, and the **second topology** up-to the layers allowed by each resulting segment (from the first topology). We use the same selection criteria as before: we filter the resulting topologies for all feasible solutions ($d_{max} \leq 7$) and then we select a set of candidate solutions by prioritizing the maximum distance to the root and the resilience of the network. Then, we look for a good trade-off between their network properties in terms of their deployment cost.

Table 8.9 shows the most attractive solutions in terms of the trade-off between their

properties. These solutions were selected from a set of 462 feasible solutions, where we balance the deployment cost against the gain in distance and resilience. At first sight, we observe the combined solutions with a **Chordal-2** as second topology are the more attractive since they deliver a good redundancy, reasonable deployment cost (to be compared with the single topology solutions) and a good distance to the root. Note also that less expensive solutions (at the beginning of the table) exhibit larger maximum distances to the root than the more expensive solutions (at the end of the table).

Topology Combination					Network Properties					
No.	First	k	Second	k	m	$\mathcal{L}$	Cost	d_{max}	$\|L_{bl}\|/L_{bl}$	$\mathcal{D}$
1	*Lin*7	1	C_2	2	29	569	714	7	20/186	7-8
2	*Star*5	1	C_2	2	21	638	743	7	14/179	9-10
3	*Bin*7	1	C_2	2	29	626	771	6	20/201	7-8
4	*Lin*5	2	C_2	1	51	519	774	6	24/96	3-4
5	*Lin*3	2	C_2	2	37	664	849	5	26/184	5-6
6	*Lin*3	1	C_2	3	25	730	855	6	20/262	8-9
7	*Lin*3	3	C_2	1	55	601	876	5	26/92	3-4
8	*Star*7	1	C_2	2	29	742	887	6	20/186	7-8
9	*Star*5	2	C_2	1	51	679	934	4	24/96	3-4
10	*Lin*5	1	C_2	3	41	757	962	6	34/279	4-5
11	*Star*5	1	C_2	3	41	837	1042	5	34/279	4-5
12	*Lin*7	1	C_2	3	57	768	1053	6	48/284	3-4
13	*Bin*7	1	C_2	3	57	825	1110	5	48/301	3-4
14	*Lin*3	1	C_2	4	49	929	1174	5	44/360	4-5
15	*Star*7	1	C_2	3	57	941	1226	4	48/288	3-4
16	*Lin*5	1	C_2	4	81	956	1361	5	74/371	2-3
17	*Star*7	2	C_2	1	99	888	1383	3	48/96	2-3
18	*Star*7	2	*bin*7	1	102	885	1395	3	48/96	1-2
19	*Star*5	1	C_2	4	81	1036	1441	4	74/370	2-3
20	*Star*5	1	*star*7	3	130	911	1561	3	39/103	1-2
21	*Bin*7	1	C_2	4	113	1024	1589	5	108/410	1-2

Table 8.9: Most attractive feasible solutions for $n = 200$ with two combined topologies.

When selecting candidate solutions in order of priority, we have for $d_{max} = 3$, the solution 17 is the closest one to our criteria. For $d_{max} = 4$, solutions 9 and 15, and for $d_{max} = 5$, solutions 7 and 13. Note that solution 17 provides a lower distance than all the single topology solutions with a lower deployment cost, but with less blocked links. However, this reduction is compensated by reducing by 1 the number of access nodes between gateways, which is reasonable. Also the combined solutions exhibiting the same distance to the root as single topology solutions (4 and 5) are less expensive. They show the ideal number of access nodes between gateway and they have the same number of blocked links for solutions 13 and 15. In addition they exhibit the half of the blocked links than

solutions 7 and 9. In summary, a combined solutions is able to deliver a better trade-off by less deployment cost than simple topology solutions. We note that all the selected solutions provide a good distance to the root and number of access nodes between gateways while reducing the cost (by the half in the case of solution 7). However, the number of blocked links is not as good as the Chordal-2 simple topology solution. Nevertheless, solution 15 reduces the deployment cost in about the 20% in return of losing the 22% of the blocked links, which is a fair trade-off.

Several solutions meet our criteria, but when assigning them an order of preference, we propose the solution 17 ($JC^2 - star7$ with $k = 2$ combined with a Chordal-2 with $k = 1$) as the best solution among the evaluated ones. Then, the solution 15 ($JC^2 - star7$ with $k = 1$ combined with a Chordal-2 with $k = 3$) as the second best, and finally, the solution 13 ($JC^2 - bin7$ with $k = 1$ combined with a Chordal-2 with $k = 3$) as the third best. Solutions 7 and 9 are also good when reducing the deployment cost, but they also reduces the resilience of the network according to our criteria.

8.3.3 Conclusion

When comparing the two presented solutions, the combined one deliver a better trade-off of network properties than the best single topology: less deployment cost due to a better combination of the number of gateways versus the total length of links; the same (or less) distance to the root; and a balanced resilience in terms of alternative paths to the root and a the number of access nodes between gateways. Indeed, we are combining the benefits from the chosen topologies: the low distance to the root from star kernel and the high redundancy of Chordal-2 topologies. The latter one acts as "a resilient layer" for the linear access network and the former one provides to each Chordal-2 segment a good distance to the root.

Hence, after the comparison of all the possible solutions presented in this section, we propose **the combination of a Star7 with 2 layers, and a Chordal-2 with 1 layer** to design the backbone network for a linear access network of 200 nodes. The main outcome from this selection is the fact of selecting a combination of a redundant topology (such as the Chordal-2 topology) to provide a better resilience to the infrastructure network, and a low distance topology to ensure a low delay when traversing the network and a quick convergence of the routing protocol.

8.4 Linear Access Network of 2000 Nodes

In this section we aim at evaluating a linear access network of 2000 nodes, covering a distance of about 300 km according to the assumptions we made in Section 3.2. This access network size is applicable to public transportation systems such as high-speed trains. We address this evaluation in the same way as before, initially evaluating possible solutions by using a single topology, and then evaluating a combination of two topologies. Only

solutions with $d_{max} \leq 7$ are considered as feasible and the selection criteria is the same one as the used in the previous section.

8.4.1 Solutions for a Single Topology

Firstly the **Chordal-2** topology. We present the resulting network properties for recursion levels from 5 to 10. We omit $k \leq 4$ due to space constraints and also there is no sense in evaluating small values of k for a large access network size. The feasible solutions are $k = 9$ and $k = 10$. Recall that according to the conclusions about the number of access nodes between gateways, 1 or 2 access nodes report not much gain in terms of the resilience when there are blocked links within the backbone network. So, we prefer to reduce number of blocked links in order to have the minimum deployment cost. Hence, we choose $k = 9$.

Properties	Recursion Level (k)					
	5	6	7	8	9	10
Number of gateways	33	65	129	257	513	1025
Total length of links	9995	11994	13993	15992	17991	19990
Cost of topology ($\delta = 1:5$)	10160	12319	14638	17277	20556	25115
Maximum dist. to the root	34	18	11	8	6	6
Number of blocked links	30	62	126	254	510	1022
Length of blocked links	3995	4987	5984	6986	7983	8985
Min. dist. between gateways	62	31	15	7	3	1
Max. dist. between gateways	63	32	16	8	4	2

Table 8.10: Solutions for $n = 2000$ provided by Chordal-2 topologies.

Secondly, the JC^2 linear kernels topology with $s = 3$ or $\mathbf{JC^2 - lin3}$ **topology**. Table 8.11 depicts its resulting backbone network properties when deploying all the possible layers. Notice the only feasible solution is $k = 6$.

Properties	Recursion Level (k)					
	1	2	3	4	5	6
Number of gateways	3	9	27	81	243	729
Total length of links	1333	2666	3999	5349	6700	8158
Cost of the topology ($\delta = 1:5$)	1348	2711	4134	5754	7915	11803
Maximum distance to the root	334	113	40	16	9	7
Number of blocked links	0	0	0	0	0	0
Length of blocked links	0	0	0	0	0	0
Min.dist.between gateways	666	222	74	24	8	2
Max.dist.between gateways	667	223	75	25	9	3

Table 8.11: Solutions for $n = 2000$ provided by linear kernel with $s = 3$.

Thirdly, we evaluate the **JC² — *lin*5 topology**. At first sight when observing Table 8.12, we realize there is no feasible solution. In addition, there is also no feasible solution for **JC² — *lin*7 topology**. Hence, we continue with star kernels.

Properties	Recursion Level (k)			
	1	2	3	4
Number of gateways	5	25	125	625
Total length of links	1600	3200	4800	6424
Cost of the topology ($\delta = 1:5$)	1625	3325	5425	9549
Maximum distance to the root	202	44	14	10
Number of blocked links	0	0	0	0
Length of blocked links	0	0	0	0
Min. dist. between gateways	400	80	16	3
Max. dist. between gateways	400	80	16	4

Table 8.12: Solutions for $n = 2000$ provided by linear kernel with $s = 5$.

Fourthly, the **JC² — *star*5 topology**. Table 8.4.1 shows that there are two feasible solutions: $k = 4$ and $k = 5$. As our criteria prioritize the maximum distance to the root and the number of blocked links, we choose $k = 5$, but we highlight this solution does not meet the requirement of the number of access nodes between gateways.

Properties	Recursion Level (k)				
	1	2	3	4	5
Number of gateways	5	25	125	625	1250
Total length of links	2400	4800	7200	9698	11196
Cost of the topology ($\delta = 1:5$)	2425	4925	7825	12823	17446
Maximum distance to the root	201	42	11	6	5
Number of blocked links	0	0	0	0	124
Length of blocked links	0	0	0	0	248
Min. dist. between gateways	400	80	16	3	1
Max. dist. between gateways	400	80	16	4	2

Table 8.13: Solutions for $n = 2000$ provided by star kernel with $s = 5$.

Fifthly, the **JC² — *star*7 topology**. Table 8.4.1 shows its resulting network properties. Similarly to previous topology, we choose $k = 4$ since we achieve the lowest maximum distance to the root and we have blocked links, even if we over provision the number of access nodes between gateways.

Lastly, the **JC² — *bin*7 topology**. Table 8.4.1 shows its resulting network properties. Notice there is no feasible solution Hence, we discard it.

Properties	Recursion Level (k)			
	1	2	3	4
Number of gateways	7	49	343	1314
Total length of links	3430	6871	10389	13642
Cost of the topology ($\delta = 1:5$)	3465	7116	12104	20212
Maximum distance to the root	144	22	6	4
Number of blocked links	0	0	0	342
Length of blocked links	0	0	0	1074
Min. dist. between gateways	285	40	5	1
Max. dist. between gateways	286	41	6	2

Table 8.14: Solutions for $n = 2000$ provided by star kernel with $s = 7$.

Properties	Recursion Level (k)			
	1	2	3	4
Number of gateways	7	49	343	1029
Total length of links	2285	4578	6920	8292
Cost of the topology ($\rho = 1:5$)	2320	4823	8635	13437
Maximum distance to the root	145	24	9	8
Number of blocked links	0	0	0	0
Length of blocked links	0	0	0	0
Min. dist. between gateways	285	40	5	1
Max. dist. between gateways	286	41	6	2

Table 8.15: Solutions for $n = 2000$ provided by binary kernel with $s = 7$.

8.4.1.1 Selection for a Single Topology

Table 8.16 shows the summary of the selected backbone topologies by using a single topology solution.

Num.	Topology	k	d_{max}	$\lvert L_{bl}\rvert / L_{bl}$	$\mathcal{D}$	Cost
1	Chordal-2	9	6	510/7983	3-4	20556
2	$JC^2 - lin3$	6	7	0/0	2-3	11803
3	$JC^2 - star5$	5	5	124/248	1-2	17446
4	$JC^2 - star7$	4	4	342/1074	1-2	20212

Table 8.16: Summary of single topology solutions for 2000 nodes.

Solution 4 is the best in terms of the maximum distance to the root, but it does not provide the best number of blocked links and over provisions the number of access nodes between gateways. However, note that its deployment cost is similar to the Chordal-2, which gives the maximum number of blocked links and the ideal number of access nodes

between gateways. But, its distance to the root is 6. As our first priority is the maximum distance to the root, we choose solution 4 as the best among them. Then, solution 3, which gives less blocked links and gateways 1 or 2 nodes apart (meaning more than the half of the access network is considered a gateway). As third, we select the solution 1 only because its distance to the root, since its resilience is far better than the solutions 3 and 4. We discard solution 2 due to evident reasons (max. distance to the root and no blocked links).

8.4.2 Solutions Combining Two Topologies

In this section we evaluate the solutions resulting from the combination of two topologies. We aim at assessing whether this approach is able to improve the solution we already found by using a single topology. We evaluate the first and the second topology for recursion levels up-to 5 when possible. Chordal-2 topologies were evaluated for further values of k without evidencing much gain. Therefore, we focus on values for $1 \leq k \leq 5$. Table 8.17 shows a selection of the best combinations among 151 feasible solutions. The first filter is to exhibit $d_{max} \leq 7$ and the selection criteria is the same one we used before. We include some extra combinations of topologies with an attractive cost to give a base of comparison.

Topology Combination					Network Properties					
No.	First	k	Second	k	m	$\mathcal{L}$	Cost	d_{max}	$\lvert L_{bl}\rvert/L_{bl}$	$\mathcal{D}$
1	$Star5$	3	C_2	1	251	9199	10454	7	124/992	7-8
2	$Star7$	2	$Lin3$	2	441	9644	11849	6	0/0	4-5
3	$Lin3$	2	$Star7$	2	441	9734	11939	6	0/0	4-5
4	$Star5$	3	C_2	2	501	11198	13703	5	374/1991	3-4
5	$Star7$	2	C_2	3	393	12868	14833	5	342/2973	5-6
6	$Star5$	3	$Star7$	1	875	10947	15322	5	0/0	2-3
7	$Star7$	3	C_2	1	687	12388	15823	5	342/976	2-3
8	$Star5$	3	$Star7$	3	1124	11445	17065	5	124/248	1-2
9	$Star7$	3	C_2	2	1314	14269	20839	5	969/1709	1-2

Table 8.17: Most attractive feasible solutions for $n = 2000$ with two combined topologies.

From the presented solutions, we note that the solution 4 meets almost all the selection criteria. The lowest maximum distance to the root, the ideal number of access nodes between gateways and a lower deployment cost. However, note that the number of blocked links is not the largest one we can achieve. Solution 9 reports the maximum number of blocked links, outperforming the Chordal-2 topology single solution for a similar deployment cost. But, the over provision of the number of access nodes between gateways make us prefer solution 4 since we reduce the cost in a 34% approximately in return to reduce the number of blocked links in about 60%, which is not dramatic since solution 4 uses two layers of a Chordal-2 to improve the resilience of the network. In summary, we see that the combined solution provides a better trade-off between network properties than the single solution.

Similarly to the conclusion we draw in the previous section, we see that **the combination of a Star5 with with 3 layers, and a Chordal-2 with 2 layers** seems to deliver a good backbone network for a linear access network of 2000 nodes. The difference between both solutions is the number of segments of the star topology and the number of deployed layers, which makes sense when addressing different linear access network sizes.

8.5 Final remarks

In this chapter we have seen that the combination of topologies might be the best way to build a backbone network for a linear access network. By combining two types of topologies, we also combine their resulting properties, achieving better trade-offs between the deployment cost versus distance to the root, number of access nodes between contiguous gateways and number of alternative paths to the root. For the two access network sizes we evaluated, the **combination of Star kernels with Chordal-2 topologies** seems to be the best combination. The Chordal-2 topology acts as a "resilient layer" between the access network and the *Network Gateway*, and the star kernel topology reduces the maximum distance to the root on each chordal segment. Also, this implicit separation of roles when combining two topologies allows the use of different link types to implement the star topology in order to reduce deployment costs. For example, radio links might be used to deploy the star kernel topology while fiber/cable links can be used to deploy the Chordal-2 topology. Thus, redundancy relies on a wired network and radio links are used to lower the overall distance to the root. This option also opens the possibility to interconnect the first layer root's nodes by another topology (a linear topology for instance) in order to provide an additional layer of resilience (1+1 link protection) in case of failure of a radio link. These alternatives, together with others are not evaluated in this thesis, however, they will be addressed as further work.

Chapter 9

Evaluation of the Infrastructure Network and the Handover Scheme

In this chapter we evaluate the proposed infrastructure network by means of simulation. We consider two scenarios: a **metro line** of 20 km, and a **train line** of 200 km. The aim of this evaluation is two-fold: 1) to evaluate the impact of the backbone topology on the handover time; and 2) to evaluate two QoS parameters (delay and losses) perceived by passengers on-board the train. Finally, we draw our conclusions based on the simulated results.

9.1 Metro Scenario - Line 1 in Santiago de Chile

In the previous chapter we proposed a backbone topology for a linear access network of 200 nodes, covering a distance about 30 km long. In this section, we aim at simulating an in-motion network describing a trajectory according to the mobility of a metro train along a comparable distance. For this purpose, we choose the *Metro Line 1* in the city of *Santiago de Chile*. This line is **20 km long** with 27 stations and trains can travel at the maximum **speed of 70 km/h**.

9.1.1 Backbone network

The *Metro Line 1* scenario requires **134 access nodes** to cover its railway according to the assumptions made in Section 3.2. All these nodes are connected sequentially to each other, forming the *linear access network*. The backbone network is built by applying the same strategy used for $n = 200$ access nodes showed in Section 8.3, but now considering $n = 134$ nodes. Thus, after evaluating the possible solutions, we found a good trade-off by combining a $JC^2 - star5$ topology for 1 layer and a Chordal-2 topology with 3 layers. This construction delivers a **backbone network of 41 nodes**. Figure 9.1 illustrates the proposed backbone topology for this scenario.

The properties for this backbone network are the following:

- Number of gateways: 41 nodes (backbone nodes).
- Total length of links: $560c = 84.0$ km.
- Maximum number of hops to the root: 4 hops.

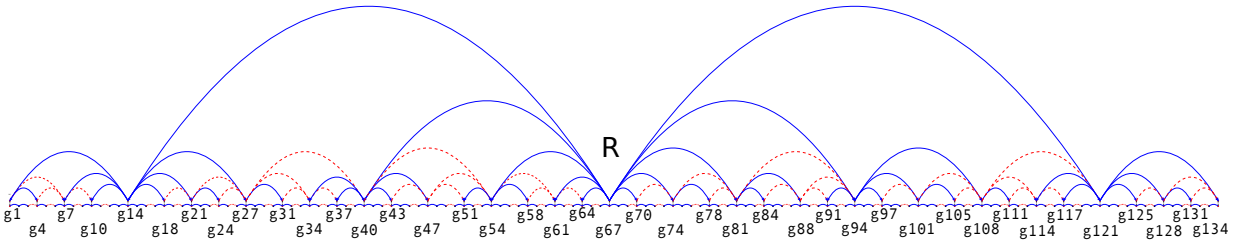


Figure 9.1: Proposed backbone network for Metro line 1 in Santiago de Chile.

- Number of blocked links: 34 links.
- Length of the blocked links: $183c = 27.450$ km (32.7% of the total length).
- Min. num. of access nodes between gateways: 3 access nodes.
- Max. num. of access nodes between gateways: 4 access nodes.

In addition, we compute other properties that are useful to further characterize this backbone network. All these new properties were calculated empirically by simulating the deployment of the infrastructure network.

- Average number of hops to the root: 2.60448 hops.
- Maximum backbone node's degree : 12.
- Average backbone node's degree : 3.08955
- Number of backbone nodes with degree equal or less than 4: 34 nodes.
- Length of links for the longest path to the root : $71c = 10.65$ km.

These empirical properties allow us to have a better notion of the backbone network characteristics, such as 1) the deployment cost, 2) the potential delay due to queueing effects on each hop, and 3) the length of the longest path to the root. For the first, it is clear that the cost of nodes differs according to the degree (number of ports) they can support. So, a large number of low degree nodes means a significant economy in the cost of backbone nodes. For the second, a low average number of hops to the root means that "in average" packets have less opportunities to be delayed on transmission queues. For the third, the length of the maximum path to the root gives a notion of the larger propagation delay packets could expect. For this case, the propagation delay is $\approx 35\mu$sec.

9.1.2 Simulated Scenario

In this section we describe the simulated scenario used for this evaluation. We consider three variables: 1) the **GAL time** as a measure of the handover time (see Section5.3); the **Round Trip Time** (ICMP Ping) from the in-motion network to an *external host*; and 3) the **packet losses** perceived at the application level. The train mobility is based on the model presented by [Maureira *et al.* 2009] for a fixed speed of 20 m/s (72 km/h). Inside each simulated train, a *Spiderman Device* (SD) is configured to provide connectivity to the in-motion network. The GAL parameters are the same used for the evaluation in Section 5.4. ($InterARPDelay$ = 7 ms, $InterBurstDelay$ = 20 ms and $BurstSize$ = 10). The in-motion network consists in a layer 2 switch with **50** *internal hosts* wired to it at 100 Mbit/s. The **wireless up-link bandwidth is set to 11 Mbit/s** (IEEE 802.11b) and the traffic profile (see Section 3.2) of the internal hosts generates a **moderate condition of saturation** on the SD's WiFi link (in average 1 packet in the transmission queue). For that purpose, we use a Constant Bit Rate (CBR) traffic generated by a modified ICMP Ping application, which sends an ICMP echo request packet of 1024 bytes at intervals between 0.15s and 0.25s (randomly uniform). The *Wireless Switch Access Points* (WSAP) use the sequence of channels 1,6,11 according to the specifications of the Spiderman Handover. Figure 9.2 depicts the networking scenario represented in the simulator.

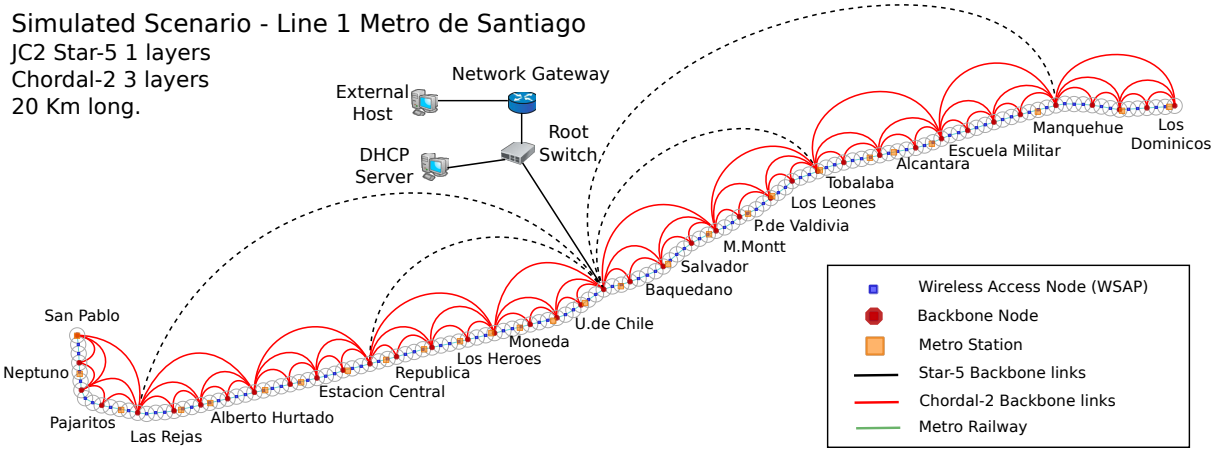


Figure 9.2: Simulation scenario for Metro line 1 in Santiago de Chile.

The deployment of the infrastructure network (access and backbone networks) is following the real metro line topology in order to obtain the correct propagation delay on links. In the figure, we represent the links as *arcs* only to improve its readability. The links in dashed (black) lines represent the *star*5 topology and the links in solid (red) lines the Chordal-2 topology. We assume a propagation delay on wired links of 3.3 μs per km (Optic Fiber). Also, we use layer 2 switches as backbone nodes with a forwarding delay of 5 μsec (5 Mpps[1]). An average performance switch (for example a Cisco 2950) is able to forward

[1]pps = packets per second.

4.9 Mpps. These nodes are connected to gateways (on the access network) by a 0 cost link in terms of the number of hops to the root. The layer 2 routing protocol used within the infrastructure network is the Rapid Spanning Tree Protocol (RSTP), which agrees with the assumptions made in Section 3.4. The root node of the RSTP tree is located at the root of the infrastructure network. This root node is connected to a layer 3 *Network Gateway*, which provides access to an external host representing an external network. The DHCP server is used to manage the layer 3 (IP) address requests from the *internal hosts* inside the in-motion network.

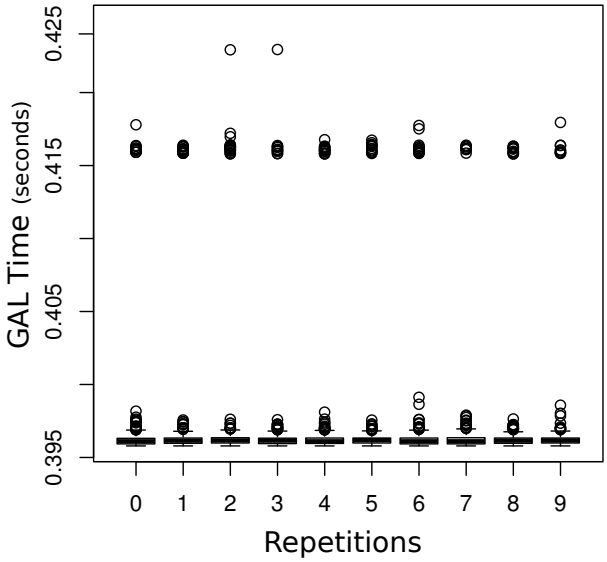
We use the *OMNeT++ 4.1* [Varga 2001] discrete event simulator as engine to simulate the depicted network, which was modelled by using the *INET Framework*. The version of this model corresponds to the 20100323, branched in march 2010 with further modifications to allow the measurement of the reception power in IEEE 802.11 devices, log radio operations and support our failure model of nodes. In addition, new models were added to simulate the *Rapid Spanning Tree Protocol* (RSTP) and *DHCP protocol*. These models were verified according to their standard definitions (IEEE 802.1w for the RSTP and the RFC 2131 for the DHCP protocol). The radio interference model is based on an "additive-noise-signal" evaluation among all the airframes "on-the-air". The propagation radio model is the Free Space Path-loss with a path loss coefficient $\alpha = 3.1$ in order to obtain a sensitivity threshold of -84 dBm at the border of a coverage area of 230 meters wide. Thermal noise is set to -110 dBm and radio transmission's power is 100 mW.

9.1.3 Handover Time - GAL Time

In this section we evaluate the *Spiderman* handover time for 50 MAC addresses (hosts) inside an in-motion network. These hosts exchange traffic with the external host through the *Spiderman Device* (SD) located on-board each train.

We simulate **two trains**, one departing from each end of the track at the same time in order to measure the effects when they cross each other in the middle of the track. We set two scenarios: the first scenario **without background traffic** and the second scenario **with background traffic**. Furthermore, we evaluate the **effect of failures** in the access and backbone networks, but only with **one train** travelling end-to-end. We use the *Gratuitous ARP Loop* time (GAL time) to measure the handover time, as discussed in Section 5.3.

Figure 9.3 shows the dispersion of the measured GAL time for 10 trials (repetitions) for the scenario **without traffic**. Each trial consists in 133 handover operations under different experimental conditions (different random seeds). At the bottom of the plot, the GAL time of each train is shown, including the maximum percentage of retransmitted Gratuitous ARP packets. Note that a large proportion of the sample is clustered near to the theoretical minimum of the GAL time. Nevertheless, some handover operations took more time (around 0.415 seconds). These maximum GAL times agree with the 4% of retransmitted ARPs, according to the equation 5.2. However, minimum and maximum are slightly above of the theoretical values. We explain this increase by the number of



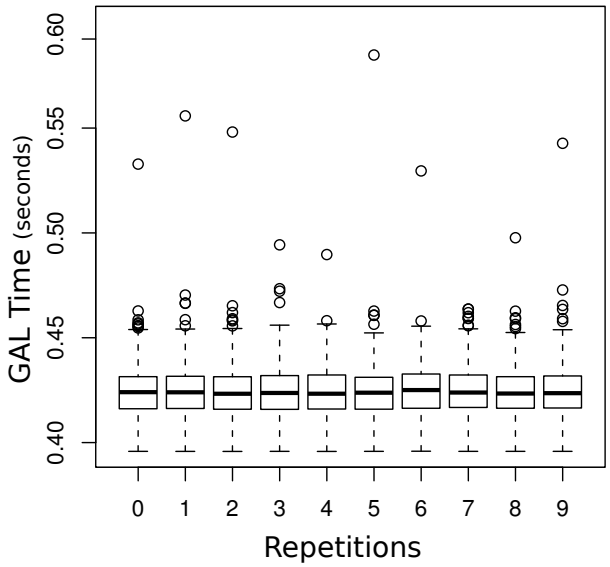
Train	GAL Time				G.ARPs sent		
	Min	Max	Mean	Std.dev	Min	Max	% Retrans.
1	0.3958	0.4230	0.3971	0.0042	50	52	4%
2	0.3958	0.4229	0.3969	0.0039	50	52	4%

Figure 9.3: GAL delay (in seconds) for two metros **without** traffic.

hops and links' lengths (topology effects) when completing the GAL. In summary, when comparing this result with the result obtained when using a fixed 2-hop infrastructure network (see Section 5.4) without traffic, we conclude that for this scenario the backbone network topology effect on the GAL time is negligible.

Figure 9.4 shows the dispersion of the GAL time for the scenario **with traffic**. Notice the minimum is still near the theoretical minimum and the variability of the GAL time is larger than the scenario without traffic. These results agree with the results obtained in Section 5.4 for the same traffic profile, except for two differences: 1) the maximum GAL time is larger for this scenario (there is a single handover operation that last significantly more than the others for each repetition) and 2) the ARPs retransmission are different among both trains. When exploring the simulation results, we realize that when both trains cross each other, a larger saturation of the WiFi channel occurs, since both SD are associated and exchanging traffic with the same WSAP (access node). Figure 9.5 shows the transmission's queue of each train for the replication 0 showed on Figure 9.4.

This plot emphasises the moment when both trains cross each other. First, Train 2 enters into the WSAP's coverage area, and few seconds after, Train 1 enters into it. Train 2 performs its GAL without causing a large increase in its transmission queue, since it is the only one contending for the channel. When Train 1 enters and starts its GAL, both SDs are contending for the channel. This contention race leads to an increase in both transmission queues (Train 1 and 2). For this specific case, this increase does not reach an overflow of the transmission queue when the Train 1 performs its GAL. Nevertheless, soon after



Train	GAL Time				G.ARPs sent		
	Min	Max	Mean	Std.dev.	Min	Max	% Retrans.
1	0.3958	0.5427	0.4228	0.0151	50	67	34%
2	0.3958	0.5948	0.4236	0.0166	50	71	42%

Figure 9.4: GAL delay (in seconds) for two metros **with** traffic.

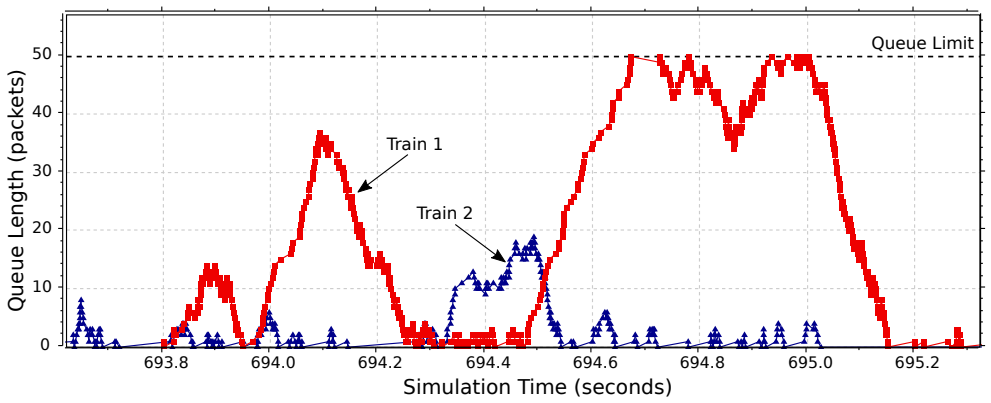


Figure 9.5: Length of the Spiderman Device's active transmission queue.

it is complete (just when the queue's length drops), a new increase is observed, but now produced by the traffic of both SDs contenting the channel. This situation generates two buffer overflows in the transmission queue of Train 1. Also notice the small increase in the length of the transmission queue of Train 2. By observing more in detail the simulation results, we can see that Train 1 is more affected due to a starvation effect (both trains configure a hidden node scenario in respect to the WSAP). However, this condition might affect either to both trains, depending who wins the contention race for accessing the channel. In the depicted example, Train 2 have won the contention race for the channel

the most of the times when transmitting a packet, being Train 1 the unfortunate loser. Recall the size of the ICMP packet is 1024 bytes, which facilitate a condition of starvation of the Train 1.

In summary, when both trains cross each other, the GAL time suffers a significant increase due to the saturation of the WiFi channel. These increase is evidenced by the highest values registered in Figure 9.4 (0.5948 seconds). Nevertheless, this situation might affect equally Train 1 or Train 2, depending which one enters second into the same WSAP coverage area.

Failure Scenario	Min	Max	Mean	Std.dev.	% Retrans.
0 Failure	0.3958	0.4831	0.4235	0.01446	20%
1 Failure	0.3958	0.4890	0.4234	0.01450	20%
2 Failures	0.3958	0.4861	0.4231	0.01464	20%
3 Failures	0.3958	0.4874	0.4231	0.01461	20%

Table 9.1: GAL times (in seconds) for failure scenarios up-to 3 failed access nodes.

Regarding the GAL time when considering failures in the access network, we consider 3 failure scenarios. Each one with 10 random failures. The first scenario considers the failure of a single WSAP node. The second scenario considers the failure of 2 consecutive WSAPs, and the third scenario considers the failure of 3 consecutive WSAP. For this evaluation, we consider only one train with 50 hosts and the traffic conditions previously described. Table 9.1 shows the obtained results. Note that the minimum and maximum GAL times are similar to the times exhibited by the control scenario (0 Failures). Their variability is also similar and they registered the same percentage of maximum retransmissions of Gratuitous ARPs packets. Therefore, this result suggest that the reconfiguration of routes caused by a segment up-to 3 failed WSAP does not affect the GAL time, and if it does, the effect is confounded within its variance.

Failure Scenario	Min	Max	Mean	Std.dev.	% Retrans.
No Failure	0.3958	0.4831	0.4235	0.0145	20%
b_{18}	0.3958	0.4771	0.4235	0.0146	18%
b_{40}	0.3958	0.4691	0.4235	0.0138	16%
b_{61}	0.3958	0.4856	0.4232	0.0145	20%
b_{97}	0.3958	0.4853	0.4238	0.0143	20%
b_{121}	0.3958	0.4830	0.4226	0.0142	20%

Table 9.2: GAL times (in seconds) considering a single failure in the backbone network.

Finally, when evaluating the failures in the backbone network, we consider the failure of a single predefined backbone node in order to generate particular failure scenarios. We evaluate 5 scenarios under the same predefined conditions for a single train. The first scenario considers a failure at b_{18}, corresponding to a failure within the first $star5$ segment,

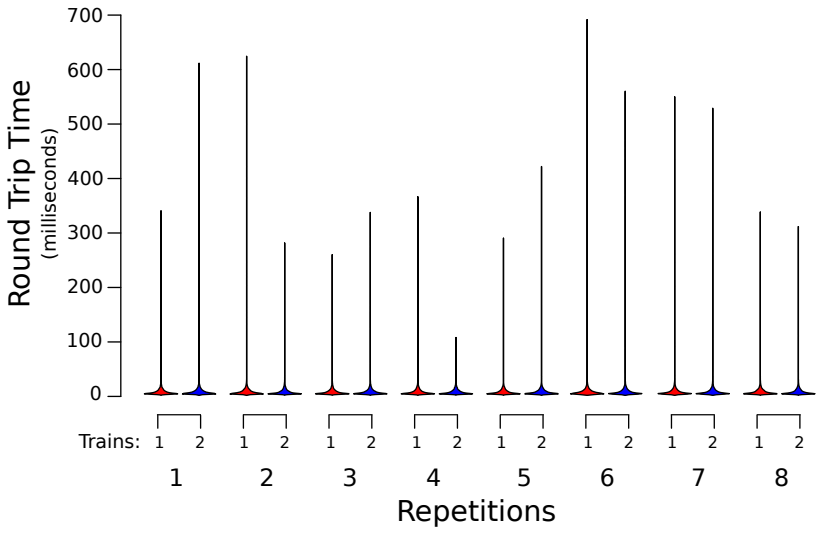
near the root of the segment. The reconfiguration of routes increases the maximum distance to the root in 2 hop. The second scenario considers a failure at b_{40}, which is the root node of the second segment of the *star*5 topology. The reconfiguration of routes increases the maximum distance to the root in 3 hops. The third considers a failure at b_{61}, near the root of the entire network (b_{67}). The maximum distance is not incremented, but the segment near the root of the network reaches the maximum distance (4 hops) of the whole network. The fourth scenario considers a failure at b_{97}, within the third segment of the *star*5 segment. The maximum distance is incremented in 1 hop. The last scenario considers a failure at b_{121}, the root node of the last *star*5 segment. The maximum distance to the root is incremented in 13 hops. Notice this latter scenario is the worst case scenario since when the root of the segment located at the end (or at the beginning) of the linear access network fails, the half of that segment (in our case $[n_{117}, n_{134}]$) must traverse the practically all its extension segment ($[g_{117}, g_{127}]$) through the Chordal-2 links and the linear access network links. Table 9.2 shown the GAL time and the percentage of ARPs retransmission for each evaluated scenario. Observe the minimum GAL time is almost the same for all the simulated cases (one very small difference at the b_{40} scenario). The maximum GAL time is also comparable to the maximum GAL time of the control scenario (No Failure), except for the scenarios when b_{40} and b_{61} have failed. These two scenarios have registered a lower percentage of retransmitted ARPs, yielding lower maximum GAL times. However, these times agree with the theoretical maximum for both measured percentages. When observing the average GAL time, we observe it is not affected significantly when compared with the control scenario. In summary, the presented results suggest that the failure of a single backbone node does not affect significantly the GAL time.

9.1.4 Round Trip Time

In this section we evaluate the Round Trip Time (RTT) measured at the application level. For that purpose, we use the same ICMP application previously described with the same predefined traffic conditions.

We evaluate this scenario considering two trains (departing from each end of the track) in order to describe the queuing effects when both trains cross each other in the middle of the track. In order to illustrate the dispersion of the measured RTT by each host on-board each train, we use **violin plots**, described by [Hintze & Nelson 1998]. These plots reflect the density of the data by depicting a wider area where observations are more dense (an a thin area where the data is less dense). Figure 9.6 shows the measured RTT by train for 8 representative trials from this evaluation (among 50 trials).

Violin plots show that almost all the measured RTT are clustered around the mean RTT of each sample ($\approx$ 8 ms). However, we note that some observations are very far away from the mean. These observations are considered outliers within each sample and they are depicted by the violin plot as a very thin line, indicating a very low density in their dispersion. Exploring the simulation results, we realize these outliers are produced when trains cross each other and trains are associated to the same WSAP, causing a condition



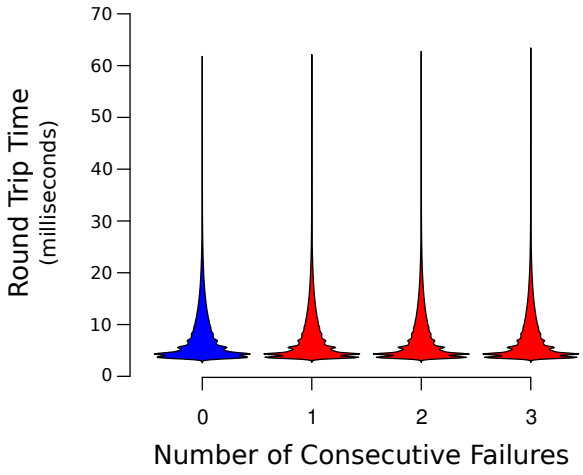
Train	Round Trip Time (ms)			
	Min	Max	Mean	Std.dev.
1	3.158	686.108	8.337	9.773
2	3.158	613.439	8.405	13.726

Figure 9.6: ICMP Round Trip Time (in milliseconds) for two trains **with** traffic.

of saturation on the wireless channel, which increases the RTT due queueing effects. The mean RTT is stable at around 8 ms, but its variability is rather large (standard deviation $\approx$ 10 ms) due to the outliers when trains cross each other. We estimate that the variability is about the half when these outliers are removed from the sample. In summary, when trains cross each other, the RTT is significantly affected due the saturation of the channel. As discussed in the previous section, both trains might be affected, depending on which train enters in second place into the same coverage area. In Figure 9.6 we observe the most representative trials illustrating how both trains are affected in different magnitudes.

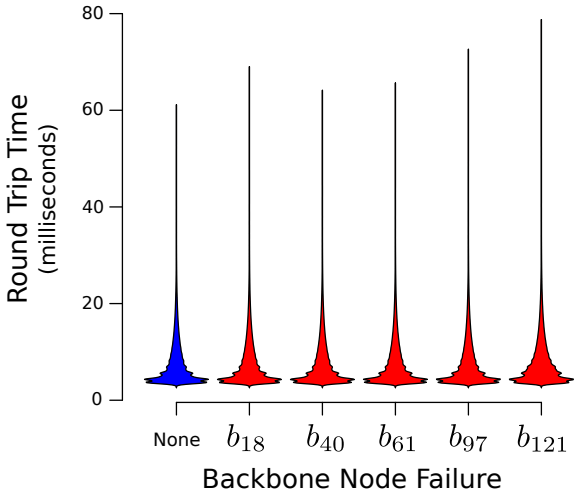
Furthermore, we evaluate the RTT for the same cases of failure in the access network as presented in the previous section (one train with 50 hosts and similar traffic conditions). Figure 9.7 shows the dispersion of the measured RTT by using violin plots. Notice the control scenario (without failures) describes more in detail the dispersion of the measured RTT values of each dense area (on the bottom of each violin) depicted by the previous violin RTT plot (Figure 9.6), which cannot be well appreciated due to the scale of the plot. The results obtained when evaluating up-to 3 contiguous WSAPs failures did not evidenced a significant change when compared to the control scenario without failures. Therefore, we conclude that the RTT is not affected by the reconfiguration of routes. This conclusion is reasonable when considering that the distance between contiguous gateways is of 3 or 4 access nodes.

Finally, we evaluate the measured RTT for the same failures in the backbone network considered by the evaluation of the GAL time. We use one train for this evaluation. Figure 9.8 shows the dispersion of the measured RTT by using violin plots.



Failure Scenario	Round Trip Time (milliseconds) Min	Max	Mean	Std.dev.
0 Failure	3.158	64.578	8.045	5.265
1 Failure	3.158	64.157	8.031	5.252
2 Failures	3.158	64.281	8.028	5.245
3 Failures	3.158	64.418	8.031	5.241

Figure 9.7: ICMP Round Trip Time (in milliseconds) considering up-to 3 consecutive failures in the access network.



Failure Scenario	Round Trip Time (milliseconds) Min	Max	Mean	Std.dev
None	3.158	64.578	8.045	5.265
b_{18}	3.158	68.453	8.031	5.230
b_{40}	3.158	64.567	8.048	5.241
b_{61}	3.158	65.082	8.027	5.230
b_{97}	3.158	72.054	8.036	5.251
b_{121}	3.158	78.200	8.254	5.200

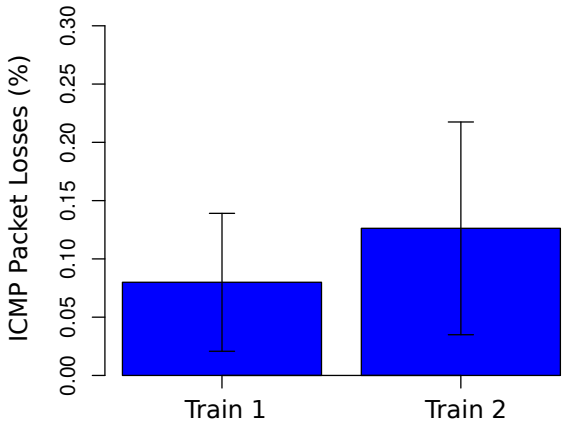
Figure 9.8: ICMP Round Trip Time (in milliseconds) considering a single failure in the backbone network.

Observe that the minimum is similar to the minimum presented for all the scenarios previously evaluated. However, the maximum is larger for some failure cases, which corroborates the increment in the number of hops to the root caused by the worst failure cases. We verify from this result that the failure at the root of a *star*5 segment located at the ends of the linear access network is the most severe failure for a single backbone node. Indeed, an increment of 13 hops in the maximum distance to the root does have an impact on the maximum measured RTT when traversing the affected segment of the access network (in this case the last segment defined by the *star*5 topology). Note that the maximum RTT is incremented in about 18 ms, and for the other failure cases the

increment is smaller. Furthermore, the mean RTT remains similar for all the cases, except for the last one. In summary, the reconfiguration of routes for all the cases, but the last, causes a negligible effect on the average RTT. However, the maximum RTT is increased significantly for the worst case failure of a single backbone node.

9.1.5 Packet Losses

In this section we aim at evaluating the packet losses perceived by the in-motion network at the application level. For this purpose, we measured the percentage of packet losses at the ICMP level for each host inside two trains along the trip, each one departing from each end of the track. In a similar way than before, we are interested in the moment when both trains cross each other since we have identified this case as a potential cause of packet losses. Figure 9.9 shows the packet losses by train when considering the same moderate condition of saturation already described before.



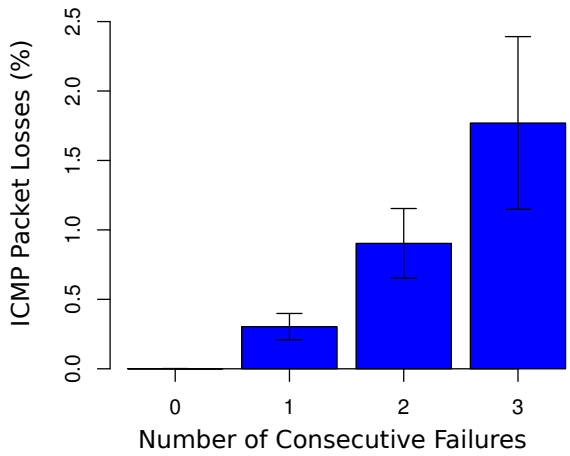
Train	ICMP Losses (%)		
	Max	Mean	Std.dev.
1	0.28	0.08	0.06
2	0.34	0.13	0.09

Figure 9.9: ICMP Packet lost (%) for two trains **with** traffic.

Notice that in average the percentage of losses is less than 0.25% of the traffic exchanged with the external host along the trip. This percentage agrees with the time window (12 seconds approximately) where both trains are associated to the same WSAP, fact we corroborated by exploring the simulation results. Let us call the "crossing coverage area" to the coverage area where the crossing event occurs. Thus, both trains are associated to the same WSAP. The difference in the packet losses perceived by each train is explained by observing which Train enters in second place into the crossing coverage area most of the times. In our simulations and considering all the trials, Train 2 enters in second place more times than Train 1. But note that the difference in the mean percentages of losses are about the same order of the measured standard deviation, which indicates that both trains might observe in average for all the trials the same packet losses.

Furthermore, we evaluate the packet losses considering failures in the access network. We evidence a direct relationship between the percentage of packet losses and the length of

the disconnected segment, which is expected. Figure 9.10 shows the percentages of packet losses for the three failure scenarios considered in previous evaluations (up-to 3 contiguous failures).



Failure Scenario	ICMP Losses (%)			
	Min	Max	Mean	Std.dev.
0 Failure	0.00	0.01	0.00	0.00
1 Failure	0.23	0.58	0.30	0.09
2 Failures	0.72	1.74	0.90	0.25
3 Failures	1.31	2.86	1.77	0.62

Figure 9.10: ICMP Packet losses (%) up-to 3 consecutive failures in the access network.

Considering a traffic stream (under a moderate condition of saturation) of 1400 seconds, a segment of 3 disconnected WSAPs causes as much as 2.5% of packet losses. When considering a vehicle speed of 20 m/s, the SD is disconnected from the network about 19 seconds, which agrees with the evidenced percentage of packet losses. This percentage of packet losses is not important when considering a continuous traffic stream such as a video surveillance feed along the trip. However, for an instantaneous connection, it might be significant. Recall that the SD is aggregating the traffic of all the hosts inside the in-motion network, therefore, when the disconnection occurs, all the on-going connections are affected depending on their traffic profile. For example, to lose 40 packets over 100 is significant, but it is not to lose 40 over 10000). This result suggest that as the percentage is low, we could consider a buffer to mitigate these losses. Thus, when observing the largest loss over the time, the capacity of this buffer should be the 2.5% of the traffic. In other words, for a data link rate of 11 Mbit/s for a time window of 1400 seconds, the buffer capacity should be approximately 50 MBytes. Note that a buffer of this size is not applicable for low delay (jitter) traffic profiles.

When evaluating the failure cases in the backbone network, we did not evidence any significant packet losses. Only independent losses were identified due to packet collisions. Therefore, we do not present further results on packet losses for backbone failure scenarios.

9.1.6 Conclusions

When comparing all the results presented in this section with respect to the GAL time, we conclude that the topology effects are negligible when considering a metro of 20 km long

track and an access network of 134 WSAP. These conclusions are valid for a moderated traffic condition, with failures in the access network up-to 3 consecutive WSAP and when considering a particular set of single backbone node's failures. For all these evaluated scenarios, we have seen that the average GAL time for 50 MAC addresses matches with the theoretical times described in Section 5.3.2.

The presented results on the the Round Trip Time (RTT) suggest that the mean RTT from the in-motion network to the external host is about 8 ms (with a minimum of 3 ms and a maximum of 64 ms). The evaluated failure cases in the access network and their reconfiguration of routes does not affect dramatically the RTT. On the contrary, the failures in the backbone network have an effect on the measured RTT. The failures at the root nodes of the star segments located at the edges of the linear access network are the worst case when considering a single backbone failure. However, the observed increments (overestimating the worst case, 20 ms) are not important when considering a forwarding delay of 5 μsec, which is a reasonable delay for a layer 2 switch nowadays.

The packet losses along the trip end-to-end of a metro train evidenced a low ratio of losses ($\approx 2.5\%$). Nevertheless, we must emphasize the fact that this percentage corresponds to a long term traffic stream of packets. When considering instantaneous connections, the speed of the metro (72 km/h) causes disconnection times up-to 19 seconds for 3 consecutive failures in the access network. This time will cause for sure a noticeable disconnection from the network. In addition, the use of a buffer to cope with this losses is not practical due to the excessive delay introduced by the capacity of the buffer. Perhaps for high-speed vehicles, a buffer might be applicable, but certainly not for a metro scenario.

In addition to the conclusions we have draw from our results, we identified a special case **when two trains cross each other** (and they are associated with the same WSAP). For this case, we did observe a degradation on all the measured parameters (GAL time, Round Trip Time and packet losses). This degradation is caused by the saturation of the WiFi channel used to serve the wireless up-link with the trains' Spiderman Device. The maximum GAL time might be incremented in about 30% (from 0.48 seconds to 0.60 seconds). The maximum RTT might suffer an increment up-to 800 ms under the traffic conditions considered by this evaluation. And with respect to the packet losses, while we found that they are low for a single crossing event, they might be larger when considering the fact that over the same track there are several metro trains circulating, therefore, the occurrence of trains crossing events is rather high.

In conclusion for the Metro evaluation, when considering a speed of $\approx$ 70 km/h, our results showed that the topology effects on the measured metrics are not significant. However, the case when two trains cross each other must be further studied, since the occurrence of these events might damage the QoS of the connections of passengers.

9.2 Train Scenario - Nice Marseille Railway

In the previous section we evaluated the effects of the proposed backbone network on the handover time and the QoS perceived by the in-motion network (Round Trip Time and

packet losses at the application level). For this evaluation, we considered a moderate condition of saturation on the wireless link between the in-motion and the access network. In this section we aim at studying the same effects, but for a longer linear access network. We consider the railway between Nice and Marseille, located at the south of France. This railway is **200 km long** and trains travel at **speeds up-to 200 km/h** because geographical constraints.

9.2.1 Backbone Network

For this train scenario, the linear access network consists in **1495 access nodes**. They are placed at regular intervals of 150 meters with their radio coverage overlapped 80 meters, as assumed in Section 3.2. We use the same strategy we presented in Section 8.4 to build a backbone network, but due to the reduction of the access network size (from 2000 to 1495) we found a better trade-off when deploying a $JC^2 - star7$ with $k = 2$ topology as the first one to deploy, and a Chordal-2 with $k = 3$ as the second topology. This change produces 49 segments of approximately 30 nodes; on each we deploy three layers of a Chordal-2 topology. Figure 9.11 shows the proposed backbone network in two part since it is too large to be illustrated as a whole. Access nodes are omitted intentionally and only backbone nodes are depicted.

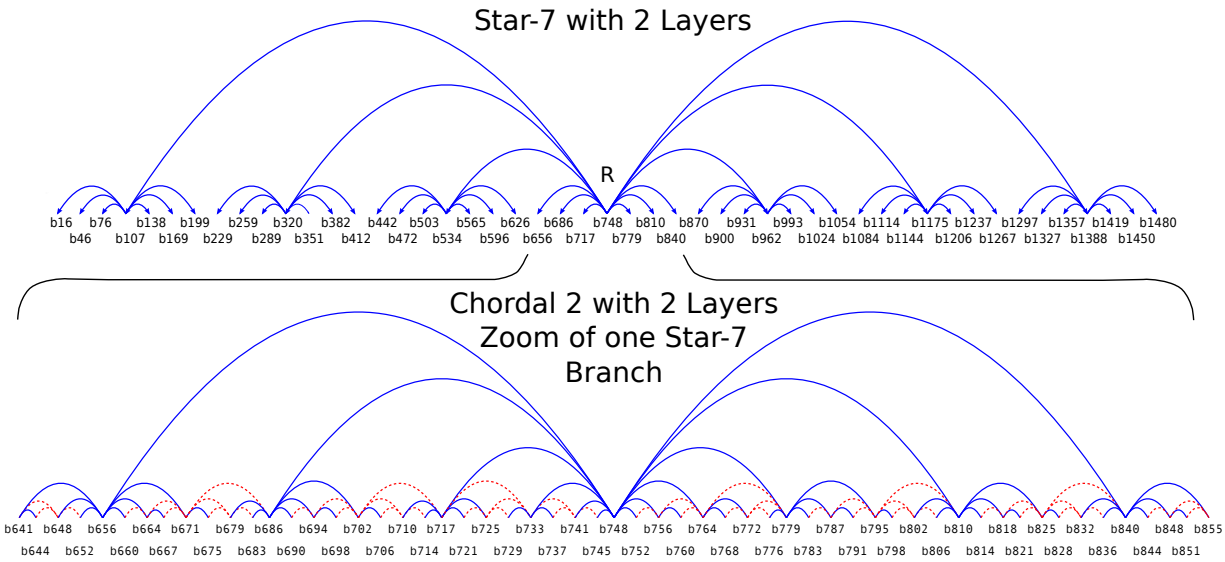


Figure 9.11: Nice-Marseille backbone network proposal

The first part of the figure shows the $JC^2 - star7$ and the second part shows only the middle star deployed at the second layer of the $star7$ topology. In this middle star, we observe 7 Chordal-2 segments, one by each star link created at layer $k = 2$. It is worth to mention that this backbone network has similar properties than the proposed network in Section 8.4.

- Number of gateways: 393 nodes (backbone nodes).
- Total length of links: $9628c = 1444.20$ km.
- Maximum number of hops to the root: 5 hops.
- Number of blocked links: 342 links (23.2% of the total length).
- Length of the blocked Links: $2237c = 335.55$ km.
- Min. num. of access nodes between gateways: 3 access nodes.
- Max. num. of access nodes between gateways: 4 access nodes.

In the same way we did for the Metro backbone network, we compute other network properties by simulating its deployment.

- Average number of hops to the root: 3.65151 hops.
- Maximum node's degree : 20
- Average node's degree : 2.9806
- Number of backbone nodes with degree equal or less than 4: 296 nodes.
- Length of the longest path to the root : $752c = 112.8$ km.

Note that about the 75% of the backbone nodes have a degree equal to or less than 4. This fact represents an important reduction in the aggregated cost of the backbone nodes. The highest degree is 20, meaning that a 24 port switch is the biggest layer 2 switch to use. In addition, despite the difference between the topology size (134 access nodes against 1495 access nodes), the average number of hops to the root is not that different (2.60 hops against 3.65 hops).

9.2.2 Simulated Scenario

The proposed infrastructure network is deployed along the Nice-Marseille railway with the same parameters used for the Metro evaluation: link's propagation delay, switches processing delay and radio configuration. Mobility is configured to simulate a train traveling at 60 m/s (216 km/h) without any stop along the trip. Figure 9.12 depicts the simulated scenario. Notice the backbone links are not depicted along the railway to improve the readability of the figure, but the propagation delay on links is calculated according to the linear distance along the railway.

The simulated time was 4000 seconds in order to simulate an end-to-end trip. We conducted the same simulated experiments (GAL time, RTT and packet losses) presented for the Metro evaluation, without and with traffic, considering 10 random failures up-to 3 access nodes, and 5 particular failure cases in the backbone network. We choose the

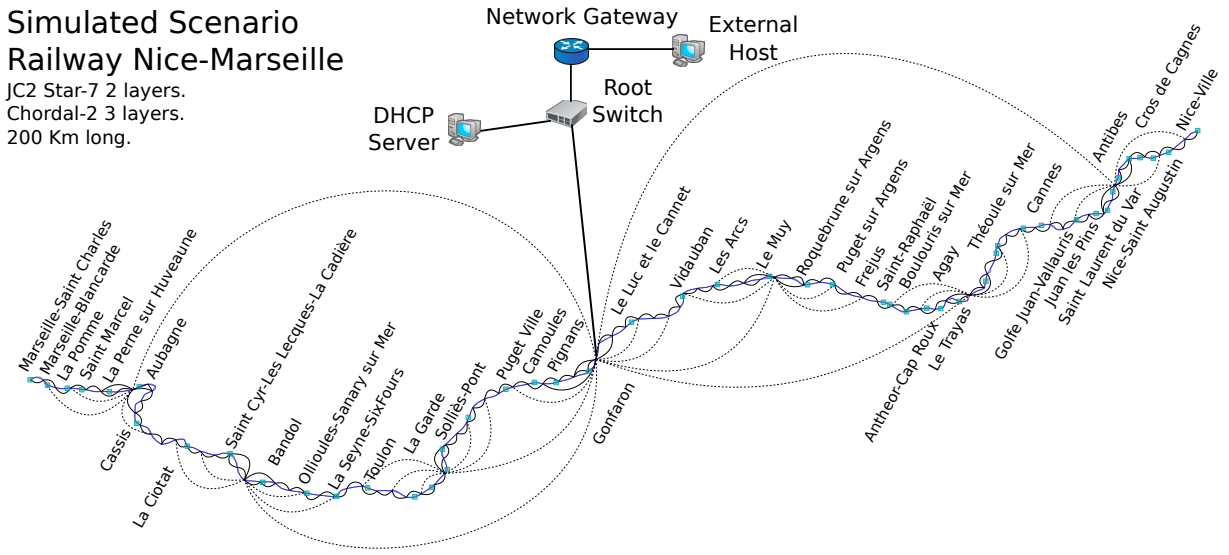


Figure 9.12: Nice-Marseille simulated scenario

backbone failed nodes from the left half (of the root node) of the topology, since the right half is symmetric, therefore the effects should be the same. Thus, the 5 scenarios are the following: the first scenario considers a failure at b_{16}. This backbone node is the root of a segment of 30 nodes created by the $star7$ topology at layer 2 in which a Chordal-2 ($k = 3$) is deployed. This segment is located at the left end of the linear access network, therefore, it is assimilated only from the right side (through b_{31}) when b_{16} fails. This failure increases the maximum distance of access nodes in 13 hops. The second scenario considers a failure at b_{46}. This node is the root node of the second segment of 30 nodes created by the $star7$ at $k = 2$. Contrarily to the first scenario, this segment is assimilated from both sides (b_{31} and b_{61}). The maximum distance is increased in 4 hops. The third failure case considers a failure at b_{107}. This node is the root of a segment of 217 nodes defined by the $star7$ at $k = 1$. This is one of the most critical failure for a single backbone failure (beside the root of the entire topology), since the maximum distance to the root is incremented in 25 hops. The fourth scenario considers a failure at b_{199}. This node is the root of the 7th segment of 30 nodes defined by the $star5$ topology at $k = 2$. This failure increases the maximum distance in 4 hops. The last scenario considers a failure at b_{320}. This node is the root of the second segment of 214 nodes created by the $star7$ topology at $k = 1$. This failure increases the maximum distance to the root in 9 hops.

9.2.3 Synthesis of Results and Conclusions

The GAL time meets the theoretical bounds. Train crossing events affect the percentage of retransmission of ARPs packets increasing the GAL time significantly. Failures in the access network does not affect the GAL time since the path between two contiguous WSAPs is not hardly affected by the reconfiguration of routes (gateways are 3 or 4 access nodes

apart). Most of the GAL operations are performed according to the theoretical bounds presented in Section 5.3.2 and some outliers might appears due to a higher percentage of ARPs retransmission. Failures in the backbone network also does not affect the GAL time, since the reconfiguration of routes plays in favor of the propagation of ARPs packets (routes between contiguous WSAP are shorter when using only the Chordal-2 links). However, we did observe some limited (less than 5) cases when the GAL procedure ended early. In these particular cases, we did not observe misrouted packets, but it does not mean that these losses might not happen. The cause of these early endings was the occurrence of the full scanning with an authentication timeout at the same time. The simultaneous occurrence of these two events reduce the handover time to less than maximum allowed handover time, triggering the early ending of the GAL.

The Round Trip Time from the in-motion network to the external host is about 8 ms (minimum 3 ms and maximum 64 ms) with a standard deviation of 5 ms. When two trains cross each other, this RTT might be increased up-to 800 ms (overestimating) within the time that both trains are associated to the same WSAP (considering a moderate condition of traffic). Failures in the access network does not affect dramatically the RTT, since the short distance between gateways minimize the impact of the reconfiguration of routes. A single failure in the backbone network might increase the RTT up-to 70 ms. But this increase is not large due to the fast forwarding time considered for nodes within the simulation (5 μsec). Larger forwarding times should have a larger impact on the observed RTT within the affected segment of access nodes. Nevertheless, nowadays middle range switches are able to forward packets with such a high rate that an important increment in the number of hops to the root will not affect significantly the observed RTT. On the contrary, the convergence of the routing protocol under such a failure conditions might be an issue. So, the convergence of an spanning tree based routing protocol, applied to an infrastructure network such as the one proposed by this thesis, should be further investigated.

Packet losses can be considered as negligible for this scenario over a long period of time. However, instantaneous connections are affected by the failure in the access network. The speed of the vehicle helps to reduce the impact of up-to 3 failed access nodes. However, the disconnected time (approximately 6 seconds) is still large enough as to consider a buffer to cope with this losses. Conversely, the packet losses perceived when two trains cross each other along the track are negligible due to the speed of trains. When both trains are traveling at high-speed in opposite directions, the time when both trains are associated to the same WSAP is so small than packets suffers a small increment in their RTT (due to queueing effects caused by the saturation of the channel) and possibly a brief buffer overflow might occur.

In conclusion, our results suggest that the backbone topology has not a big influence on the GAL time, Round Trip Time (RTT) or packet losses along a trip of 200 km. In case of failure, the routes reconfiguration has not affected these metrics in a dramatic way. But, we must emphasise that these results were obtained by considering a forwarding delay at each node of 5 μsec, which makes more easy to deal with an increment of 25 hops

(worst case) when reconfiguring the shortest path to the root. For larger forwarding delays, such increase in the maximum distance to the root might yield to larger round trip times. Nevertheless, if we assume a forwarding delay such that the RTT is incremented up-to 100 ms (or even 200 ms), we still have a reasonable delay when thinking in video or streaming applications. The problem would be to exhibit a delay of 600 ms or even 1 second, making this solution comparable to a satellite link. Furthermore, we have identified an additional factor which might affect the performance of these metrics, which is when two trains cross each other when traveling in opposite directions. The GAL time is affected by these events in a moderate way. The RTT is hardly affected and the packet losses are negligible thanks to the speed of the trains. In addition, the occurrence of these events is low within a train scenario.

9.3 Summary and Conclusions

In this chapter, we evaluated the communication from an in-motion network to an external host (or network without loss of generality) for two train scenarios: a metro line of 20 km long and a train line of 200 km long. We focus our evaluation on the integration of the proposed handover scheme (the **Spiderman Handover**) with the proposed infrastructure network for both scenarios. Our results suggest that the Spiderman handover is able to **work correctly within the infrastructure network**, propagating the routes without causing packet losses due to the handover operation or the misrouting of packets under traffic conditions. We obtained the same conclusions when considering up-to 3 consecutive failures in the access network and a particular set of failures in the backbone network. Nevertheless, the number of hosts inside the in-motion network might be large when considering a train scenario at 200 km/h with the coverage conditions assumed in Section 3.2. Under these conditions, we evidenced some cases when the *Spiderman Device* (SD) gets disconnected from the infrastructure network for a short period of time. While these events were limited, they exist. So, we suggest to keep the number of MAC address that participate in the handover operation at a low level (in example not more than 10). A low number of MAC addresses allows the handover scheme to deal with the occurrence of the full scanning and authentication/association timeouts. Techniques such as *Natting* or *Network masquerading* might help to deal with this problem.

The mean Round Trip Time measured for both scenarios were about 8 ms with a minimum of 3 ms and a maximum of 60 ms when considering a moderate condition of traffic (in average 1 packet in the transmission queue of the Spiderman Device). These delays are more than acceptable for an infrastructure network of the considered size. The failures in the access and backbone networks do not affect dramatically the mean RTT. A successful reconfiguration of routes and a low forwarding delay of packets within backbone and access nodes help to minimize the impact of failures on the observed RTT. However, when considering critical failure cases (the root of a star segment for instance), the maximum distance to the root might suffer an important increment, which might affect the convergence of the routing protocol to a working stable network. Therefore, this cases should be further inves-

tigated. Packet losses were also acceptable when considering the failure cases in the access network for long term traffic streams. However, for short connections the disconnected time caused by an interval up-to 3 access nodes (19 seconds for the metro scenario and 6 seconds for the train scenario) might be large enough to produce a perceptible error at the application level (even when considering the buffering of the data). Therefore, while the proposed handover scheme has successfully avoided packet losses, and the infrastructure network has successfully overcome the evaluated failures, **the problem of how to deal with the disconnected segment is still an open issue**. We showed in Chapter 7 that the probability to experience up-to 3 consecutive WSAPs failed might be significant when p is large (or at least in the order of 0.001) and there are 3 or 4 access nodes between consecutive gateways (the first step of the ladder effect). So, we could exploit the length of the train with two *Spiderman Devices* (one at each end of the train), in order to cope with one or perhaps two failed WSAP. Thus, in case of detecting a failed WSAP, we change the active SD to avoid the disconnected nodes. We will explore this type of solutions as further work.

Despite the findings already presented, we consider more important the conclusions we draw about **the case when two trains cross each other** along the railway. We have shown that when both trains are associated to the same WSAP, an important degradation of the handover time and the observed RTT occur due to the saturation of the WiFi channel. Packet losses might be not important when considering a single crossing event, But in a metro mobility scenario with several trains circulating along the track, these losses become important. A higher bandwidth WiFi link (such as 802.11g/n) can be used to mitigate this saturation by means of a better efficiency of the radio channel, but it only push the saturation limit forward. Therefore, a new approach is required. Perhaps a dual radio WSAP capable to serve each train on a different channel might solve the problem. This solution will be also investigated as further work.

Chapter 10
Conclusions

We classify the contributions of this thesis in two parts: 1) the horizontal handover of high-speed vehicles over a predictable trajectory, and 2) the design of an infrastructure network for a railway scenario. The results we presented through this thesis suggest that it is possible to reduce (or even eliminate) the packet losses caused by the handover operation when using a trackside infrastructure network based on a simple wireless access technology (such as WiFi). For the trackside infrastructure network, we proposed a combined hierarchical design, which exhibits a reasonable trade-off between the number of hops to reach the *Network Gateway*, the resilience of the network, and its deployment cost. However, this communication architecture poses new challenges, which require to be addressed in order to make such a system fully operable by a train operator company.

10.1 Summary of Contributions

The motivation of this thesis is to study a communication architecture to provide Internet connectivity for trains passengers only relying on an unlicensed wireless technology, such as WiFi. The challenges we addressed were two: the horizontal handover of an in-motion network at high-speeds (Part I of this thesis) and the design of an infrastructure network for a railway scenario (Part II).

In the first part, we presented a new handover scheme consisting in a handover algorithm (Chapter 4) and a routes updating procedure (Chapter 5). This handover scheme, called **The Spiderman Handover**, relies only on commodity WiFi hardware. It bases its operation only on the OSI Layer 2, allowing the operation of any Layer 3 protocol. The presented results suggest that the proposed scheme can successfully handle the handover of Ethernet devices up-to 350 km/h, without producing packet losses due to the handover operation. The routes updating procedure prevents the packet losses due to the misrouting of packets, and while it does not scale directly with the number of hosts, it can handle a large in-motion network by means of techniques such as Network Address Translation (NAT) or network masquerading (MASQ).

In the second part, we proposed a design of a backbone network capable of providing connectivity to a linear access network deployed along a railway (namely the infrastructure network). We studied some particular (recursive) backbone topologies, describing some of their network properties (Chapter 6). Then, we analyzed them comparing their properties in order to assess the trade-offs involved when choosing their parameters (Chapter 7). Finally, we proposed a **combined hierarchical design** for a backbone network (Chapter 8), focusing on a low delay when traversing the network (in terms of a low number of hops to

reach the root node or gateway), reasonable resilience (in terms of the ability of the network to cope with the most probable failures) and reasonable deployment cost (in terms of the number of backbone nodes and total length of links). Among the solutions we evaluated, the presented results suggest that the **combination of a Chordal topology with a Star topology** provides an infrastructure network exhibiting a low number of hops to the root and a reasonable resilience by an average deployment cost. This combined design is applicable for linear access network up-to 2000 nodes (300 km long) and possibly more.

Finally, by evaluating both contributions together (Chapter 9), we presented simulated results pointing out that it is technically possible to deploy a working large (bridged) trackside communication network in order to provide connectivity to several in-motion networks (such as trains) through commodity WiFi devices and layer 2 switches.

However, several challenges have been identified through the development of this thesis. These challenges must be further investigated in order to provide a solution that can be implemented and managed by a train operator. Therefore, this work is an initial step to propose an solution considering together the handover and the design problems of a trackside infrastructure network within a railway scenario.

10.2 Advantages and Disadvantages of the Proposed Solution

One of the main advantages of the proposed solution is that it operates only at the OSI layer 2. This feature becomes important when improving the logical design, traffic engineering and manageability of a communication network. For example, the logical design can be improved by the creation of different *Virtual LANs* (IEEE-802.1Q VLAN tagging), permitting the implementation of several virtual topologies, overlaid to the physical network topology, each one providing different QoS features according to the requirements of end-users. Concerning traffic engineering, the combination of VLANs and *multiple spanning trees* (IEEE 802.1Q-2003) allows the existence of different ways to reach the root of the network (one per tree), or even more, to have several regional roots in the network. Multiples spanning trees can be also used to provide load balancing in order to avoid the congestion of links near the root node of the network, effect that is well documented in the literature for spanning tree like routing protocols. In terms of manageability, we highlight the possibility to join several (bridged) Ethernet networks into a unique very large IP network. Each one of these bridged networks may cover a trajectory of a vehicle. For example, when joining several train lines' infrastructure networks across a country. This geographical division allows the decentralization of the network management and also diversify its operation. The handover between these regional networks can be easily handled by Mobile IP protocols, since the size of each regional network makes perfectly applicable a more time consuming handover scheme.

The mayor drawbacks we identify for the proposed solution are two: 1) maintainability of a large network deployed over a vast geographical area; and 2) the security of the

wireless up-link between the in-motion and the linear access networks. The first one could be mitigated by the implementation of a autonomous bootstrap protocol to auto-configure the WSAPs when deploying them. Thus, their replacement in case of failure can be done by non-specialized personnel, which helps to reduce the operational costs of the network. We showed some results on the probability of experiencing a disconnection from the network in terms of the length of an interval of contiguous failed WSAPs. Thus, their replacement operation can be performed in a bulk way only when observing an interval larger than the interval tolerated by the speed of the vehicle. However, still remains unanswered the question of how operable is such a network. The second drawback is about the security of the wireless link between the in-motion network and the access network. As the proposed solution is based on commodity WiFi hardware, security schemes are not bullet-proof. Security in this context can be seen from two points of view: 1) the up-link between the in-motion network and the access network is permanently lost due to jamming or other kinds of DoS attacks; and 2) the data exchanged between the in-motion network and the access node is compromised due to men-in-the-middle attacks or some other technique allowing the attacker to decode successfully the communications. The second one is likely less hard to solve since there the authentication/association process can be more sophisticated as the handover time is hidden from the handover process. Nevertheless, the first issue is wide unanswered. Perhaps to use a sort of spread spectrum for WiFi devices or maybe to reduce the WSAP coverage area to such a size that a jamming attack can not harm the communication.

10.3 Perspectives

In the development of this thesis, we identify several issues that require to be addressed in order to improve the completeness of the proposed solution. In the following we state some of these issues:

- The losses produced by the train crossing events leads us to suggest a dual radio WSAP. Thus, when both trains are associated to the same WSAP, each train can use a different channel to establish the in-motion network up-link. Therefore, the saturation identified when trains cross each other is avoided and the QoS of the in-motion network is preserved.

- The issue of dealing with a disconnected interval of one or two access nodes is still open. As the probability of observing such an event is independent of the number of access nodes between gateways, there is a real possibility to exploit the length of the train with two Spiderman Devices (SD). Thus, the in-motion network might cope with these failure cases by changing the active connection from one SD to the other.

- The convergence of spanning tree like protocols (such as the Rapid Spanning Tree Protocol - RSTP) when applied to the proposed infrastructure network is still not bounded. While the literature suggests that the RSTP is able to converge in less

than 30 seconds when the maximum distance to the root is less than 7 hops, we think that it is possible to propose improvements in order to provide a shorter (or at least bounded) convergence time for the initial setup of the tree, as well as for different worst failure cases scenarios.

In addition, we propose some further studies to continue this work in order to provide a better understanding of the communication problem within a railway scenario. We summarize some of them in the following:

- A study of the probability of observing exactly a certain number of times a disconnected interval of access nodes equal to or larger than a given number of access nodes. This study could provide a better notion about the overall disconnection time during the trip, and, at the same time, to have a an insight about the maintainability of the network. In this work, we only study the probability of experiencing at least one disconnected interval of a given number of access nodes (WSAPs).

- A study comparing the proposed design for an infrastructure network against other designs rules, such as a random graph or other hierarchical rings network designs, will help to identify better parameters, properties and metrics to analyze the performance of a linear infrastructure network.

- A study of the impact of using directional radio links or 3rd party networks (for example an ISP) to implement the first topology of the infrastructure network. In the two evaluated cases (Chapter 9), the star topology could be implemented by using other physical layer networks (or technologies) without affecting the resilience of the network, of course, as long as the Chordal-2 topology will be based on a wired trackside deployment. The feasibility of using radio links or 3rd party networks might help to reduce significantly the deployment cost of the infrastructure network.

Many further directions exist in order to continue this work. We only mentioned the ones we consider as important at this moment. But, certainly, we will continue the improvement of the presented results towards a better solution to provide a trackside communication system for trains, metros or urban buses.

Appendix A

Tables

A.1 Tables in Chapter 5

Hosts	GAL Time				G.ARPs sent		
	Min	Max	Mean	Sd	Min	Max	% Retrans.
1	0.0008	0.0081	0.001	0.0002	1	2	100%
5	0.0288	0.0363	0.030	0.0003	5	6	20%
10	0.0638	0.0843	0.064	0.0007	10	11	10%
20	0.1468	0.1680	0.147	0.0018	20	21	5%
50	0.3958	0.4241	0.397	0.0047	50	52	4%
100	0.8108	0.8455	0.815	0.0081	100	103	3%
150	1.2257	1.2740	1.235	0.0111	150	155	3.3%
200	1.3566	1.7028	1.636	0.0549	165	207	3.5%
250	1.3566	2.1321	1.898	0.2199	165	259	3.6%

Table A.1: GAL delay (in seconds) for different number of hosts **without** background traffic.

Hosts	GAL Time				G.ARPs sent		
	Min	Max	Mean	Sd	Min	Max	% Retrans.
1	0.0008	0.0281	0.002	0.0021	1	5	400%
5	0.0288	0.0873	0.034	0.0071	5	11	120%
10	0.0638	0.1566	0.075	0.0160	10	20	100%
20	0.1468	0.2884	0.192	0.0180	20	36	80%
50	0.3958	0.5801	0.455	0.0348	50	70	40%
100	0.8108	1.1462	0.911	0.0532	100	140	40%
150	1.2258	1.5943	1.362	0.0649	150	193	28%
200	1.3568	2.0962	1.753	0.1325	165	254	27%
250	1.3568	2.2875	1.937	0.2558	165	277	10%

Table A.2: GAL delay (in seconds) for different number of hosts **with** background traffic.

A.2 Tables in Chapter 7

Speed	Number of Failed Nodes (l^*)									
km/h	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
10	25.2	79.2	133.2	187.2	241.2	295.2	349.2	403.2	457.2	511.2
20	12.6	39.6	66.6	93.6	120.6	147.6	174.6	201.6	228.6	255.6
30	8.4	26.4	44.4	62.4	80.4	98.4	116.4	134.4	152.4	170.4
40	6.3	19.8	33.3	46.8	60.3	73.8	87.3	100.8	114.3	127.8
50	5.0	15.8	26.6	37.4	48.2	59.0	69.8	80.6	91.4	102.2
60	4.2	13.2	22.2	31.2	40.2	49.2	58.2	67.2	76.2	85.2
70	3.6	11.3	19.0	26.7	34.5	42.2	49.9	57.6	65.3	73.0
80	3.2	9.9	16.7	23.4	30.2	36.9	43.7	50.4	57.2	63.9
90	2.8	8.8	14.8	20.8	26.8	32.8	38.8	44.8	50.8	56.8
100	2.5	7.9	13.3	18.7	24.1	29.5	34.9	40.3	45.7	51.1
110	2.3	7.2	12.1	17.0	21.9	26.8	31.7	36.7	41.6	46.5
120	2.1	6.6	11.1	15.6	20.1	24.6	29.1	33.6	38.1	42.6
130	1.9	6.1	10.2	14.4	18.6	22.7	26.9	31.0	35.2	39.3
140	1.8	5.7	9.5	13.4	17.2	21.1	24.9	28.8	32.7	36.5
150	1.7	5.3	8.9	12.5	16.1	19.7	23.3	26.9	30.5	34.1
160	1.6	5.0	8.3	11.7	15.1	18.5	21.8	25.2	28.6	32.0
170	1.5	4.7	7.8	11.0	14.2	17.4	20.5	23.7	26.9	30.1
180	1.4	4.4	7.4	10.4	13.4	16.4	19.4	22.4	25.4	28.4
190	1.3	4.2	7.0	9.9	12.7	15.5	18.4	21.2	24.1	26.9
200	1.3	4.0	6.7	9.4	12.1	14.8	17.5	20.2	22.9	25.6
210	1.2	3.8	6.3	8.9	11.5	14.1	16.6	19.2	21.8	24.3
220	1.1	3.6	6.1	8.5	11.0	13.4	15.9	18.3	20.8	23.2
230	1.1	3.4	5.8	8.1	10.5	12.8	15.2	17.5	19.9	22.2
240	1.1	3.3	5.6	7.8	10.1	12.3	14.6	16.8	19.1	21.3
250	1.0	3.2	5.3	7.5	9.6	11.8	14.0	16.1	18.3	20.4
260	1.0	3.0	5.1	7.2	9.3	11.4	13.4	15.5	17.6	19.7
270	0.9	2.9	4.9	6.9	8.9	10.9	12.9	14.9	16.9	18.9
280	0.9	2.8	4.8	6.7	8.6	10.5	12.5	14.4	16.3	18.3
290	0.9	2.7	4.6	6.5	8.3	10.2	12.0	13.9	15.8	17.6
300	0.8	2.6	4.4	6.2	8.0	9.8	11.6	13.4	15.2	17.0

Table A.3: Disconnection time for different speeds and number of failed nodes

Bibliography

[Abuguba & Moldován 2006] S. Abuguba and I. Moldován. *Verification of RSTP convergence and scalability by measurements and simulations.* Broadband Europe 2006, Geneva, Switzerland, December 2006.

[Adya *et al.* 2004] A. Adya, P. Bahl, J. Padhye, A. Wolman and Lidong Zhou. *A multi-radio unification protocol for IEEE 802.11 wireless networks.* In Proceedings of 1st International Conference on Broadband Networks (BroadNets 2004), pages 344 – 354, October 2004.

[Aguado *et al.* 2008] M. Aguado, O. Onandi, P.S. Agustin, M. Higuero and E. Jacob Taquet. *WiMax on Rails.* IEEE Vehicular Technology Magazine, vol. 3, no. 3, pages 47 – 56, September 2008.

[Angskun *et al.* 2007] T. Angskun, G. Bosilca and J. Dongarra. *Binomial graph: A scalable and fault-tolerant logical network topology.* Parallel and Distributed Processing, pages 471–482, August 2007.

[Arden & Lee 1981] B.W. Arden and Hikyu Lee. *Analysis of Chordal Ring Network.* IEEE Transactions on Computers, vol. C - 30, no. 4, pages 291 – 295, April 1981.

[Arjona *et al.* 2008] A. Arjona, J. Kerttula and A. Yla-Jaaski. *Live Network Performance Challenge: FLASH-OFDM vs. HSDPA.* In Proceedings of 22nd International Conference on Advanced Information Networking and Applications (AINA 2008), pages 918 –925, March 2008.

[Beivide *et al.* 2003] R. Beivide, C. Martínez, C. Izu, J. Gutierrez, J. Gregorio and J. Miguel-Alonso. *Chordal Topologies for Interconnection Networks.* In Alex Veidenbaum, Kazuki Joe, Hideharu Amano and Hideo Aiso, editors, High Performance Computing, volume 2858 of *Lecture Notes in Computer Science*, pages 385–392. Springer Berlin / Heidelberg, 2003.

[Bermond *et al.* 2003] J-C. Bermond, S. Choplin and S. Pérennes. *Hierarchical Ring Network design.* Theory of Computing Systems, vol. 36, no. 6, pages 663–682, November 2003.

[Bianchi *et al.* 2003] G Bianchi, N Blefari-Melazzi, E Grazioni and S. *Internet access on fast trains: 802.11-based on-board wireless distribution network alternatives.* In Proceedings of 14th IST Mobile & Wireless Communication Summit, pages 15–18, July 2003.

[Billion & Van den Abeele 2007] J. Billion and D. Van den Abeele. *ICOM: A Communication Framework for Interoperable European Railways.* In Proceedings of the

7th International Conference on ITS Telecommunications. IEEE Computer Society, June 2007.

[Brik & Mishra 2005] V. Brik and S. Mishra A.and Banerjee. *Eliminating handoff latencies in 802.11 WLANs using multiple radios: applications, experience, and evaluation.* In Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement (IMC '05), pages 27–27, Berkeley, CA, USA, 2005. USENIX Association.

[Chen *et al.* 2008] Y. Chen, M. Chuang and C. Chen. *DeuceScan: Deuce-Based Fast Handoff Scheme in IEEE 802.11 Wireless Networks.* IEEE Transactions on Vehicular Technology, vol. 57, no. 2, pages 1126 –1141, March 2008.

[Chini *et al.* 2010] P. Chini, G. Giambene and S. Kota. *A survey on mobile satellite systems.* International Journal of Satellite Communications and Networking, no. 28, pages 29–57, August 2010.

[Chiruvolu *et al.* 2004] G. Chiruvolu, A. Ge, D. Elie-Dit-Cosaque, M. Ali and J. Rouyer. *Issues and approaches on extending Ethernet beyond LANs.* Communications Magazine, IEEE, vol. 42, no. 3, pages 80 – 86, March 2004.

[Cho & Pan 2009] C. Cho and J.Y. Pan. *Forecasting WiMAX System Earnings: A Case Study on Mass Rapid Transit System.* In Choong Hong, Toshio Tonouchi, Yan Ma and Chi-Shih Chao, editors, Management Enabling the Future Internet for Changing Business and New Computing Services, volume 5787 of *Lecture Notes in Computer Science*, pages 331–344. Springer Berlin / Heidelberg, September 2009.

[Chow *et al.* 2009] B. Chow, M. Yee, M. Sauer, A. Ng'Oma, M. Tseng and C. Yeh. *Radio-over-Fiber Distributed Antenna System for WiMAX Bullet Train Field Trial.* In Proceedings of Mobile WiMAX Symposium (MWS '09), pages 98–101. IEEE Computer Society, July 2009.

[Corvaja *et al.* 2004] R. Corvaja, A. Zanella, Dossim M., A. Tontoli and P. Zennaro. *Experimental Performance of the Handover Procedure in a WiFi Network.* In Proceedings of the 7th International Symposium on Wireless Personal Multimedia Communications (WPM'04), pages 12–15, September 2004.

[De Sousa 2006] A.F. De Sousa. *Improving Load Balance and Resilience of Ethernet Carrier Networks with IEEE 802.1s Multiple Spanning Tree Protocol.* In Proceedings of the International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06), pages 95–95. IEEE Computer Society, April 2006.

[Drake *et al.* 2010] S. Drake, J. Jaffe and R. Boggs. *Worldwide Mobile Worker Population 2009-2013 Forecast.* IDC Market Analysys Report - Excerpt (English Edition), June 2010.

[Durst *et al.* 1996] R. Durst, G. Miller and E. Travis. *TCP extensions for space communications.* In Proceedings of the 2nd annual international conference on Mobile computing and networking (MobiCom '96), pages 15–26. ACM Press, November 1996.

[Egevang & Francis 1994] K. Egevang and P. Francis. *RFC 1631: The IP Network Address Translator (NAT)*, May 1994. Status: INFORMATIONAL.

[Emmelmann *et al.* 2007] M. Emmelmann, S. Wiethoelter, A. Koepsel, C. Kappler and A. Wolisz. *Moving toward seamless mobility: state of the art and emerging aspects in standardization bodies.* Wireless Personal Communications, vol. 43, no. 3, pages 803–816, April 2007.

[Emmelmann *et al.* 2008] M. Emmelmann, T. Langgäertner and M. Sonnemann. *System design and implementation of seamless handover support enabling real-time telemetryhighly mobile users.* In Proceedings of the 6th ACM international symposium on Mobility management and wireless access (MobiWac '08), pages 1–8. ACM Press, October 2008.

[Emmelmann *et al.* 2009] M. Emmelmann, S. Wiethölter and H.T. Lim. *Influence of network load on the performance of opportunistic scanning.* In Proceedings of the 34th Conference on Local Computer Networks (LCN'2009), volume 20, pages 601 – 608. IEEE Computer Society, October 2009.

[Emmelmann 2000] M. Emmelmann. *TCP/IP Over Satellite.* Mark Emmelman personal Webpage [Online, accessed Oct 2010] `http://www.emmelmann.org/Library/Papers_Reports/docs/TcpIp_overSatellite/TcpIp_overSatellite.pdf`, 2000.

[Emmelmann 2005] M. Emmelmann. *Influence of velocity on the handover delay associated with a radio-signal-measurement-based handover decision.* In Proceedings of the 62nd Vehicular Technology Conference (VTC'2005-Fall), pages 2282–2286. IEEE Computer Society, September 2005.

[Emmelmann 2010] M. Emmelmann. Vehicular networking: Automotive applications and beyond, chapter System Design and Proof-of-Concept Implementation of Seamless Handover Support for Communication-Based Train Control. Intelligent Transportation Systems. Wiley-Blackwell, June 2010.

[Fadin *et al.* 1998] G. Fadin, H. Kirrmann and P. Umiliacchi. *ROSIN, Railway Open System Interconnection Network. Web Technologies for Railways.* In Proceedings Automation in Transportation, 1998.

[Faghani & Mirjalily 2009] F. Faghani and G. Mirjalily. *A New Ethernet Switching Method Based on Extended Forwarding Topology.* In Proceedings of the International Conference on Future Computer and Communication (ICFCC '09), pages 365–370. IEEE Computer Society, April 2009.

[Feremans *et al.* 2003] C. Feremans, M. Labbé and G. Laporte. *Generalized network design problems.* European Journal of Operational Research, vol. 148, no. 1, pages 1 – 13, January 2003.

[Fettweis & Irmer 2005] G. Fettweis and R. Irmer. *WIGWAM: System concept development for 1 Gbit/s air interface.* In 14th Wireless World Research Forum (WWRF'05), July 2005.

[Fokum & Frost 2010] Daniel T. Fokum and Victor S. Frost. *A Survey on Methods for Broadband Internet Access on Trains.* IEEE Communications Surveys & Tutorials, vol. 12, no. 2, pages 171–185, April 2010.

[Fowler & Zeadally 2006] S. Fowler and S. Zeadally. *Fast handover over micro-MPLS-based wireless networks.* In Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC '06), pages 181–186. IEEE Computer Society, June 2006.

[Gass *et al.* 2006] R. Gass, J. Scott and C. Diot. *Measurements of In-Motion 802.11 Networking.* In Proceedings of the 7th Workshop on Mobile Computing Systems & Applications (WMCSA'06), pages 69–74. IEEE Computer Society, August 2006.

[Gatti 2003] A. Gatti. *Trains as Mobile devices: the TrainCom project.* [Online, accessed Oct 2010] http://www.stellastellina.org/assets/applets/3540_WDC2002The_TrainCom_project.pdf, 2003.

[Georgopoulos *et al.* 2010] P Georgopoulos, J Moura, M Noor, B McCarthy and C. *Theoretical and Practical Survey of Backhaul Connectivity Options.* Technical report, Lancaster University, Computing Department, June 2010.

[Gondi 2009] V. Gondi. *Seamless Secured Roaming over Heterogeneous Wireless Networks.* PhD thesis, Université d'Evry Val d'Essonne, Laboratoire de Réseaux et Systèmes Multimédia, 2009.

[Greve *et al.* 2005] F. Greve, B. Lannoo, L. Peters, T. Leeuwen, F. Quickenborne, D. Colle, F. Turck, I. Moerman, M. Pickavet, B. Dhoedt and P. Demeester. *FAMOUS: A Network Architecture for Delivering Multimedia Services to FAst MOving USers.* Wireless Personal Communications, vol. 33, no. 3-4, pages 281–304, June 2005.

[Hempel *et al.* 2006] M. Hempel, H. Sharif, T. Zhou and P. Mahasukhon. *A wireless test bed for mobile 802.11 and beyond.* In Proceedings of the 2006 international conference on Communications and mobile computing (IWCMC '06), pages 1003–1008. ACM Press, June 2006.

[Hernandez & Helal 2001] E. Hernandez and A. Helal. *Examining Mobile-IP performance in rapidly mobile environments: the case of a commuter train.* Proceedings of the 26th Annual IEEE Conference on Local Computer Networks (LCN'2001), pages 365–372, November 2001.

[Hintze & Nelson 1998] J. Hintze and R. Nelson. *Violin Plots: A Box Plot-Density Trace Synergism.* The American Statistician, vol. 52, no. 2, pages 181–184, 1998.

[Huynh & Mohapatra 2007] M. Huynh and P. Mohapatra. *Metropolitan Ethernet Network: A move from LAN to MAN.* Computer Networks, vol. 51, no. 17, pages 4867 – 4894, August 2007.

[Huynh *et al.* 2009] M. Huynh, P. Mohapatra and S. Goose. *Spanning tree elevation protocol: Enhancing metro Ethernet performance and QoS.* Computer Communications, vol. 32, no. 4, pages 750–765, March 2009.

[IEEE 802.1D 2004] IEEE 802.1D. *Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks Amendment: Management Information Base (MIB) definitions for VLAN Bridges.* IEEE Computer Society - LAN/MAN Standards Committee, 2004.

[IEEE 802.1Q 2006] IEEE 802.1Q. *Local and metropolitan area networks Virtual Bridged Local Area Networks.* IEEE Computer Society - LAN/MAN Standards Committee, 2006.

[Ishizu *et al.* 2004] K. Ishizu, M. Kuroda and K. Kamura. *SSTP: an 802.1s extension to support scalable spanning tree for mobile metropolitan area network.* In Proceedings of the Global Telecommunications Conference (GLOBECOM'04), volume 3, pages 1500–1504. IEEE Computer Society, December 2004.

[Jang *et al.* 2009] K. Jang, M. Han, S. Cho, H. Ryu, J. Lee, Y. Lee and S. Moon. *3G and 3.5G wireless network performance measured from moving cars and high-speed trains.* Proceedings of the 1st workshop on Mobile internet through cellular networks (MICNET '09), page 19, September 2009.

[Johnson *et al.* 2004] D. Johnson, C. Perkins and J. Arkko. *Mobility Support in IPv6.* IETF, June 2004. [Standards Track RFC 3775].

[Judge 2005] P. Judge. *Case Study: 100 mph WiMax hits the rails to Brighton.* TechWorld.com [Online, accessed Oct. 2010], http://howto.techworld.com/mobile-wireless/1351/case-study-100-mph-wimax-hits-the-rails-to-brighton/, 2005.

[Kassab *et al.* 2010] Mohamed Kassab, Martine Wahl, Mauricio Casanova, M. Berbineau and Marina Aguado. *IEEE 802.11a performance for infrastructure-to-train communications in an underground tunnel.* In Proceedings of the 9th International Conference on Intelligent Transport Systems Telecommunications (ITST'09), pages 447–452. IEEE Computer Society, October 2010.

[Kim *et al.* 2004] H-S. Kim, S-H. Park, C-S. Park, J-W. Kim and S-J. Ko. *Selective Channel Scanning for Fast Handoff in Wireless LAN Using Neighbor Graph.* In Ignas

Niemegeers and Sonia Heemstra de Groot, editors, Personal Wireless Communications, volume 3260 of *Lecture Notes in Computer Science*, pages 629–629. Springer Berlin / Heidelberg, September 2004.

[Kitani *et al.* 2004] T. Kitani, N. Funabiki and T. Higashino. *A proposal of hierarchical chordal ring network topology for WDM networks.* In Proceedings of the 12th International Conference on Networks (ICON'2004), pages 605–609. IEEE Computer Society, November 2004.

[Kowal *et al.* 2010] M. Kowal, S. Kubal, P. Piotrowski and R-J. Zieliński. *Operational characteristic of wireless WiMax and IEEE 802.11x systems in underground mine environments.* International Journal of Electronics and Telecommunications, vol. 56, no. 1, pages 81–86, March 2010.

[Mahasukhon *et al.* 2007] P. Mahasukhon, M. Hempel, H. Sharif, T. Zhou, S. Ci and H-H. Chen. *BER Analysis of 802.11 b Networks under Mobility.* In Proceedings of the International Conference on Communications (ICC'07), pages 4722–4727. IEEE Computer Society, June 2007.

[Malekian 2008] R. Malekian. *The study of handover in mobile IP networks.* In Proceedings of the 3d International Conference on Broadband Communications, Information Technology & Biomedical Applications, pages 181–185. IEEE Computer Society, November 2008.

[Maureira *et al.* 2009] J-C. Maureira, P. Uribe, O. Dalle, T. Asahi and J. Amaya. *Component based approach using OMNeT++ for Train Communication Modeling.* In Proceedings of 9th International Conference on ITS Telecommunication (ITST'09), pages 441 – 446. IEEE Computer Society, October 2009.

[MeshDynamics] MeshDynamics. *MeshDynamics "Mines" New Outdoor Wireless Enterprise Opportunity.* Business Wire Magazine [Online, accessed Oct, 2010] http://www.businesswire.com/news/google/20080327005191/en/MeshDynamics-Mines-Outdoor-Wireless-Enterprise-Opportunity.

[Mirjalily *et al.* 2009] G. Mirjalily, F. Akhavan Sigari and R. Saadat. *Best Multiple Spanning Tree in Metro Ethernet Networks.* In Proceedings of the 2nd International Conference on Computer and Electrical Engineering (ICCEE '09), volume 2, pages 117–121. IEEE Computer Society, December 2009.

[Mishra *et al.* 2003] A. Mishra, M. Shin and W. Arbaugh. *An empirical analysis of the IEEE 802.11 MAC layer handoff process.* ACM SIGCOMM Computer Communication Review, vol. 33, no. 2, page 93, April 2003.

[Murray *et al.* 2007] D. Murray, M. Dixon and T. Koziniec. *Scanning delays in 802.11 networks.* In Proceedings of the International Conference onNext Generation Mo-

bile Applications, Services and Technologies. (NGMAST'07), pages 255–260. IEEE Computer Society, September 2007.

[Myers *et al.* 2004] A. Myers, E. Ng and H. Zhang. *Rethinking the service model: Scaling Ethernet to a million nodes.* In Proceedings of the ACM Workshop on Hot Topics in Networks (HotNets '04), November 2004.

[Ok *et al.* 2008] J. Ok, P. Morales and H. Morikawa. *Scanning through Authentication for IEEE 802.11 WLAN Fast Handoff.* IEICE technical report. Information networks, vol. 107, no. 483, pages 47–52, July 2008.

[Ott & Kutscher 2004] J. Ott and D. Kutscher. *The "drive-thru" architecture: WLAN-based internet access on the road.* In Proceedings of the 59th Vehicular Technology Conference, 2004. (VTC 2004-Spring), volume 5, pages 2615 – 2622. IEEE Computer Society, May 2004.

[Padmaraj *et al.* 2005a] M. Padmaraj, S. Nair, M. Marchetti, G. Chiruvolu and M. Ali. *Traffic engineering in enterprise ethernet with multiple spanning tree regions.* In Proceedings of the Systems Communications, pages 261 – 266, Montreal, Canada, August 2005.

[Padmaraj *et al.* 2005b] M. Padmaraj, S. Nair, M. Marchetti, G. Chiruvolu, M. Ali and A. Ge. *Metro Ethernet traffic engineering based on optimal multiple spanning trees.* In Proceedings of the 2nd IFIP International Conference on Wireless and Optical Communications Networks (WOCN 2005), pages 568 – 572, March 2005.

[Pallos *et al.* 2007] R. Pallos, J. Farkas, I. Moldovan and C. Lukovszki. *Performance of rapid spanning tree protocol in access and metro networks.* In Proceedings of the 2nd International Conference on Access Networks & Workshops (AccessNets '07), pages 1–8. IEEE Computer Society, August 2007.

[Perkins 2002] C. Perkins. *IP Mobility Support for IPv4.* RFC 3344 (Proposed Standard), August 2002. Updated by RFC 4721.

[Plummer 1982] D.C. Plummer. *RFC 826: Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware*, November 1982. Status: STANDARD.

[Pries & Heck 2004] R. Pries and K. Heck. *Performance Comparison of Handover Mechanisms in Wireless LAN Networks.* Technical report, University of Wüurzburg. Institute of Computer Science., Sydney, Australia, September 2004.

[Raghavendra *et al.* 2007] R. Raghavendra, E. Belding, K. Papagiannaki and K. Almeroth. *Understanding handoffs in large ieee 802.11 wireless networks.* Proceedings of the 7th ACM SIGCOMM conference on Internet measurement (IMC '07), pages 333–338, October 2007.

[Ramachandran *et al.* 2006] K. Ramachandran, S. Rangarajan and J.C. Lin. *Make-Before-Break MAC Layer Handoff in 802.11 Wireless Networks.* In Proceedings of the International Conference on Communications (ICC '06), volume 10, pages 4818 –4823. IEEE Computer Society, June 2006.

[Ramani & Savage 2005] I. Ramani and S. Savage. *SyncScan: practical fast handoff for 802.11 infrastructure networks.* In Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies. (INFOCOM 2005), volume 1, pages 675 – 684, March 2005.

[Ray *et al.* 2010] S-K. Ray, K. Pawlikowski and H. Sirisena. *Handover in Mobile WiMAX Networks: The State of Art and Research Issues.* IEEE Communications Surveys & Tutorials, vol. 12, no. 3, pages 376–399, April 2010.

[Rizvi *et al.* 2009] S.S. Rizvi, M-A. Khan and A. Riasat. *Active scanning: A better approach to reduce handover time at MAC layer for wireless networks.* In Proceedings of the 2nd International Conference on Computer, Control and Communication (IC4 2009), pages 1–4. IEEE Computer Society, February 2009.

[Rodriguez *et al.* 2004] P. Rodriguez, R. Chakravorty, J. Chesterfield, I. Pratt and S. Banerjee. *MAR: a commuter router infrastructure for the mobile Internet.* In Proceedings of the 2nd international conference on Mobile systems, applications, and services (MobiSys '04), pages 217–230. ACM Press, June 2004.

[Schena & Ceprani 2004] V. Schena and F. Ceprani. *FIFTH Project solutions demonstrating new satellite broadband communication system for high speed train.* In Proceedings of the 59th Vehicular Technology Conference (VTC 2004-Spring), volume 5, pages 2831 – 2835 Vol.5. IEEE Computer Society, May 2004.

[Sethom *et al.* 2004] K. Sethom, H. Afifi and G. Pujolle. *Wireless MPLS: a new layer 2.5 micro-mobility scheme.* In Proceedings of the 2nd international workshop on Mobility management & wireless access protocols (MobiWac '04), pages 64–71. ACM Press, October 2004.

[Shin *et al.* 2004] M. Shin, A. Mishra and W. Arbaugh. *Improving the latency of 802.11 hand-offs using neighbor graphs.* In Proceedings of the 2nd international conference on Mobile systems, applications, and services (MobiSys '04), pages 70–83. ACM Press, June 2004.

[Shingler *et al.* 2008] R. Shingler, G. Fadin and P. Umiliacchi. *From RCM to predictive maintenance: The InteGRail approach.* In Proceedings of the 4th IET International Conference on Railway Condition Monitoring, pages 1 – 5. IEEE Computer Society, June 2008.

[Sterbenz *et al.* 2010] J.P-G. Sterbenz, D. Hutchison, E-K. Çetinkaya, A. Jabbar, J-P. Rohrer, M. Schöller and P. Smith. *Resilience and survivability in communication*

networks: Strategies, principles, and survey of disciplines. Computer Networks, vol. 54, no. 8, pages 1245–1265, June 2010.

[Tse 2007] T. Tse. *Study of High-Speed Wireless Data Transmissions for Railroad Operation.* Technical report, U.S. Deparment of Transportation, April 2007.

[Varga 2001] A. Varga. *The OMNeT++ discrete event simulation system.* In Proceedings of the European Simulation Multiconference (ESM'2001), pages 319–324. SCS – European Publishing House, June 2001.

[Vassiliou & Zinonos 2010] V. Vassiliou and Z Zinonos. *An Analysis of the Handover Latency Components in Mobile IPv6.* Journal of Internet Engineering, vol. 3, no. 1, pages 230–240, January 2010.

[Velayos & Karlsson 2004] H. Velayos and G. Karlsson. *Techniques to reduce the IEEE 802.11b handoff time.* In Proceedings of the International Conference on Communications (ICC 2004), volume 7, pages 3844–3848. IEEE Computer Society, June 2004.

[Wiethoelter & Emmelmann 2010] S. Wiethoelter and M. Emmelmann. Modeling Handover from the Access Networks' Perspective, chapter 15, pages 341–356. Springer, 1st edition, October 2010.

[WifiRail] Customer Case Study WifiRail. *Global Wireless Leader Installs First High-Speed Commuter System.* Cisco System [Online, accessed Oct, 2010] http://www.cisco.com/en/US/prod/collateral/wireless/ps5678/ps6521/case_study_c36-532133.pdf.

[Wing 2010] D. Wing. *Network Address Translation: Extending the Internet Address Space.* IEEE Internet Computing, vol. 14, pages 66–70, August 2010.

[Xie *et al.* 2007] J. Xie, I. Howitt and I. Shibeika. *IEEE 802.11-based Mobile IP fast handoff latency analysis.* In Proceedings of the International Conference on Communications (ICC '07), pages 6055–6060. IEEE Computer Society, June 2007.

[Zhou *et al.* 2005] T. Zhou, H. Sharif, M. Hempel, P. Mahasukhon and Song C. *Performance of IEEE 802.11b in mobile railroad environments.* In Proceedings of the 62nd Vehicular Technology Conference (VTC-2005-Fall), volume 4, pages 2527 – 2531. IEEE Computer Society, September 2005.

[Zhou *et al.* 2009] T. Zhou, H. Sharif, M. Hempel, P. Mahasukhon, W. Wang and H-H. Chen. *Performance Study of a Mobile Multi-hop 802.11a/b Railway Network Using Passive Measurement.* Mobile Networks and Applications, vol. 14, no. 6, pages 782–797, December 2009.

[Zhu *et al.* 2010] Changping Zhu, Xingsong Deng, Jia Zhu, Lei Li, Xiaoyang Zeng, Hongzhen Yu and Shen Zhang. *Performance analysis of wireless local area networks (WLAN) in a coal-mine tunnel environment.* Mining Science and Technology (China), vol. 20, no. 4, pages 629–634, July 2010.