UNIVERSITAT ROVIRA I VIRGILI

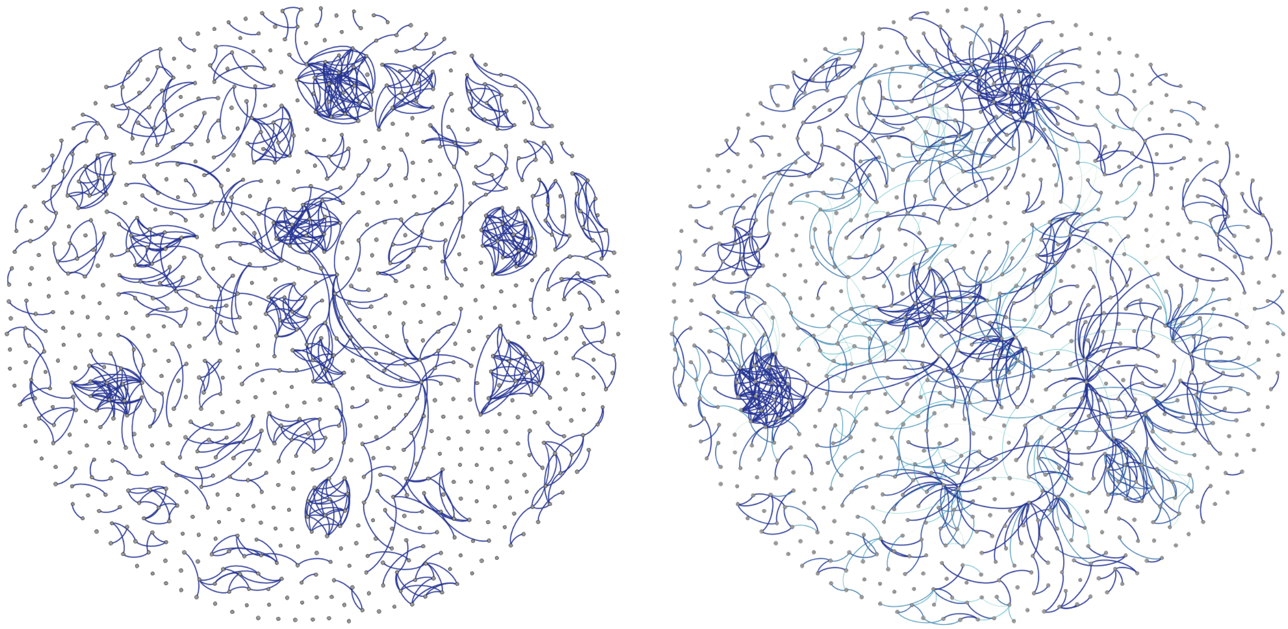
IMPROVING THE ROUTING LAYER OF AD HOC NETWORKS THROUGH PREDICTION TECHNIQUES

Pere Millán Marco

UNIVERSITAT
ROVIRA i VIRGILI

Improving the Routing Layer of Ad Hoc Networks Through Prediction Techniques

PERE MILLÁN MARCO



DOCTORAL THESIS
2018

UNIVERSITAT ROVIRA i VIRGILI

Improving the Routing Layer of Ad Hoc Networks Through Prediction Techniques

Pere Millán Marco

Doctoral Thesis by the
Departament d'Enginyeria Informàtica i Matemàtiques (DEIM)
Doctoral Programme in Computer Science and Mathematics of Security

Advisors:
Dr. Carlos Molina Clemente
Dr. Roc Meseguer Pallarès

Tarragona, Spain 2018

UNIVERSITAT
ROVIRA i VIRGILI

FAIG CONSTAR que aquest treball, titulat "**Improving the Routing Layer of Ad Hoc Networks Through Prediction Techniques**", que presenta **Pere Millán Marco** per a l'obtenció del títol de Doctor, ha estat realitzat sota la meva direcció al **Departament d'Enginyeria Informàtica i Matemàtiques** d'aquesta universitat.

HAGO CONSTAR que el presente trabajo, titulado "**Improving the Routing Layer of Ad Hoc Networks Through Prediction Techniques**", que presenta **Pere Millán Marco** para la obtención del título de Doctor, ha sido realizado bajo mi dirección en el **Departament d'Enginyeria Informàtica i Matemàtiques** de esta universidad.

I STATE that the present study, entitled "**Improving the Routing Layer of Ad Hoc Networks Through Prediction Techniques**", presented by **Pere Millán Marco** for the award of the degree of Doctor, has been carried out under my supervision at the **Departament d'Enginyeria Informàtica i Matemàtiques** of this university.

Tarragona, 18 de juny de 2018

Els directors de la tesi doctoral
Los directores de la tesis doctoral
Doctoral Thesis Supervisors

Dr. Carlos Molina Clemente

Dr. Roc Meseguer Pallarès

A mis padres

Delfina y Pedro

"Per fer les coses bé cal:
primer, l'amor a elles;
segon, la tècnica"

Antoni Gaudí (1852–1926)

Improving the Routing Layer of Ad Hoc Networks Through Prediction Techniques.
June 2018.

Pere Millán Marco
pere.millan@urv.cat

Departament d'Enginyeria Informàtica i Matemàtiques
Universitat Rovira i Virgili
Av. Països Catalans, 26 - ETSE/Campus Sescelades
43007 - Tarragona, Spain

Acknowledgments

First and foremost, I am so much grateful to my advisors, Carlos Molina and Roc Meseguer. Without their endless patience, excellents guidance, advice, and support, this long journey would simply have been impossible. I would also thank Eduard Montseny, the advisor in my first attempt to become a PhD.

Quiero agradecer también a Pedro y Delfina, por todos los esfuerzos y sacrificios que han realizado para que sus 3 hijos pudiéramos estudiar y labrarnos un buen futuro. Es la mejor herencia que nos pueden dejar.

També vull agrair a Pochy i Maximiliano pel seu suport i acompanyament en tota aquesta llarga aventura, encara que la major part del temps jo estigués només *"de cuerpo presente"*.

I would also like to thank Domènec Puig and Pedro García, for their funding support for my activities as PhD candidate, and to all my other colleagues at the Departament d'Enginyeria Informàtica i Matemàtiques (Universitat Rovira i Virgili) for their support and encouragement.

I am thankful also to my co-authors Sergio, Rodrigo, Esunly, Davide, Bart, Chris, Manos, Leandro, Javier, Matías.

Lastly, as Manos said: *to all those hours, that were not spent with the people and causes that deserved it.*

This work has been partially supported by the European Union through the projects *Community Networks Testbed for the Future Internet* (CONFINE): FP7-288535, and *A Community networking Cloud in a box* (Clommunity): FP7-317879, and also by Spanish government under contracts TIN2013-44375-R, TIN2013-47245-C2-1-R, TIN2013-47245-C2-2-R, TIN2012-37171-C02-02, TIN2016-77836-C2-1-R, TIN2016-77836-C2-2-R, and DPI2016-77415-R, and also by the Generalitat de Catalunya as Consolidated Research Groups 2014-SGR-881, 2017-SGR-688, and 2017-SGR-990.

Summary

EVERYDAY becomes more evident the key role that mobile computing and wireless technologies play in our daily activities. Being always connected, anytime, and anywhere is today more a necessity than a luxury. The ubiquitous computing scenarios created based on these technology advances allow people to provide and consume shared information. In these scenarios, the supporting communication networks are typically wireless and ad hoc.

The dynamic and changing characteristics of the ad hoc networks, makes the work done by the routing layer to have a high impact on the performance of these networks. It is very important for the routing layer to quickly react to changes that happen, and even be advanced to what will happen in the near future, by applying prediction techniques. This allows the routing layer to have a proactive approach (try to avoid a problem before it happens), which is more efficient than a reactive approach (which try to correct the problem when it is already present). In this context, the general research question that this thesis investigates is *What is the improvement achieved when we apply prediction to the routing layer of ad hoc networks?*

In order for the routing layer to do its job (to decide what is the best path for an information to reach its destination) it is necessary to know what paths or links are available among the network nodes (topological information) and how good these paths or links are (quality). For this reason, the general research question of the thesis is addressed in two key aspects: (1) the prediction of the topological information, and (2) the prediction of the quality of ad hoc networks.

This thesis investigates whether prediction techniques can improve the routing layer of ad hoc networks. As a first step in this direction, in this thesis we explored the potentiality of a History-Based Predictor (HBP) strategy to predict the Topology Control Information (TCI) generated by routing protocols. We demonstrated that there is a high opportunity for predicting the TCI, and this prediction can be just focused on a small subset of messages. Based on our findings we implemented the OLSR-HBP predictor and evaluated it with regard to the Optimized Link State Routing (OLSR) protocol. OLSR History-Based Predictor (OLSR-HBP) achieved important decreases of TCI (signaling overhead), without disturbing the network operation, and requiring a small and affordable amount of resources. Finally, regarding the impact of Prediction on the routing data for both Link and Path (or End-to-End) Quality information, we demonstrated that Time-series analysis is a promising approach to accurately predict both Link and End-to-End Quality in Community Networks.

Thesis Publications

- Millán, P., Molina, C., Meseguer, R., Ochoa, S. F. & Santos, R. *Using a History-Based Approach to Predict Topology Control Information in Mobile Ad Hoc Networks* in *Internet and Distributed Computing Systems* (eds Fortino, G., Di Fatta, G., Li, W., Ochoa, S., Cuzzocrea, A. & Pathan, M.) (Springer International Publishing, Cham, 2014), 237–249. ISBN: 978-3-319-11692-1. doi:10.1007/978-3-319-11692-1_21

- Millan, P., Molina, C., Medina, E., Vega, D., Meseguer, R., Braem, B. & Blondia, C. *Tracking and Predicting Link Quality in Wireless Community Networks* in *2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* CORE B (Oct. 2014), 239–244. ISBN: 978-1-4799-5041-6. doi:10.1109/WiMOB.2014.6962177

- Millan, P., Molina, C., Dimogerontakis, E., Navarro, L., Meseguer, R., Braem, B. & Blondia, C. *Tracking and Predicting End-to-End Quality in Wireless Community Networks* in *2015 3rd International Conference on Future Internet of Things and Cloud* (Aug. 2015), 794–799. ISBN: 978-1-4673-8103-1. doi:10.1109/FiCloud.2015.96

- Millan, P., Molina, C., Medina, E., Vega, D., Meseguer, R., Braem, B. & Blondia, C. Time Series Analysis to Predict Link Quality of Wireless Community Networks. *Comput. Netw.* **93.** Impact Factor: 1.446, Q2, 342–358. ISSN: 1389-1286 (Dec. 2015)

- Millán, P. *Improving Prediction in the Routing Layer of Wireless Networks Through Social Behaviour* in *2nd URV Doctoral Workshop in Computer Science and Mathematics* (eds Sánchez Artigas, M. & Valls Mateu, A.) (Publicacions URV, Tarragona, Nov. 2015), 5–8. ISBN: 978-84-8424-399-1

- Millan, P., Molina, C., Meseguer, R., Ochoa, S. F., Santos, R. & Medina, E. *OLSR-HBP: a History-Based Predictor for Topology Control Information in Mobile Ad Hoc Networks* **Submitted to** Wireless Communications and Mobile Computing, special issue on Topology Control in Emerging Mobile Networks, Hindawi/Wiley. ISSN: 1530-8677. doi:10.1155/6302 Impact Factor: 1.899, Q2. Oct. 2018

Other Publications

- Millán, P. *Real-Time Traffic-Scheduling in Underwater Acoustic Wireless Sensor Networks* in *3rd URV Doctoral Workshop in Computer Science and Mathematics* (eds Gómez, S. & Valls Mateu, A.) (Publicacions URV, Tarragona, Nov. 2016), 11–15. ISBN: 978-84-8424-495-0

- Santos, R., Orozco, J., Micheletto, M., Ochoa, S. F., Meseguer, R., Millan, P. & Molina, C. *Scheduling Real-Time Traffic in Underwater Acoustic Wireless Sensor Networks* in *Ubiquitous Computing and Ambient Intelligence* (eds García, C. R., Caballero-Gil, P., Burmester, M. & Quesada-Arencibia, A.) (Springer International Publishing, Cham, 2016), 150–162. ISBN: 978-3-319-48799-1. doi:10.1007/978-3-319-48799-1_19

- Santos, R., Orozco, J., Micheletto, M., Ochoa, S. F., Meseguer, R., Millan, P. & Molina, C. Real-Time Communication Support for Underwater Acoustic Sensor Networks. *Sensors* **17.** Impact Factor: 2.677, Q1. ISSN: 1424-8220. doi:10.3390/s17071629 (2017)

Contents

Acronyms

ANSN Advertised Neighbor Sequence Number

CN Community Network

CONFINE Community Networks Testbed for the Future Internet

CPU Central Processing Unit

DL Deep Learning

DM Data Mining

DSDV Destination Sequenced Distance Vector

DSR Dynamic Source Routing

EB Edge Betweenness

ECDF Empirical Cumulative Distribution Function

EED End-to-End Delay

EER End-to-End Retransmissions

EtEQ End-to-End Quality

ETOP Expected number of Transmissions On a Path

ETX Expected Transmission Count

EWMA Exponentially Weighted Moving Average

GPR Gaussian Process for Regression

gTCm generated TC message

HBP	History-Based Predictor
HD	History Depth
HoPS	Holistic Packet Statistics
IoT	Internet of Things
KMPR	Kinetic Multipoint Relaying
kNN	k-Nearest Neighbours
LAN	Local Area Network
LQ	Link Quality
LQE	Link Quality Estimators
LQI	Link Quality Indication
MAE	Mean Absolute Error
MANET	Mobile Ad hoc NETwork
MARA	Metric-Aware Rate Adaptation
MGR	Mobile Gambler's Ruin
ML	Machine Learning
MSN	Message Sequence Number
MPR	Multi-Point Relay
ND	Node Degree
NIC	Network Interface Card
NLQ	Neighbour Link Quality
OLSR	Optimized Link State Routing
OLSR-HBP	OLSR History-Based Predictor
OLSRp	OLSR with Prediction
OSPF	Open Shortest Path First
PSR	Packet Success Rate

pTCm	predicted TC message
RBR	Rule-Based Regression
RMSE	Root Mean Squared Error
RNP	Required Number of Packets
RSS	Received Signal Strength
RSSI	Received Signal Strength Indication
RT	Regression Tree
RTT	Round-Trip Time
SLAW	Self-similar Least Action Walk
SNR	Signal-to-Noise Ratio
SRQ1	Specific Research Question 1
SRQ2	Specific Research Question 2
SVM	Support Vector Machine
TC	Topology Control
TCI	Topology Control Information
WEED	Weighted End-to-End Delay
WMEWMA	Window Mean with Exponentially Weighted Moving Average
WMCN	Wireless Mesh Community Network
WMN	Wireless Mesh Network
WSN	Wireless Sensor Network

Chapter 1

Introduction

1.1 Background

An *ad hoc network* is a decentralized type of Local Area Network (LAN) where every individual device (node) communicates directly with each other. In classical LANs, there are a pre-existing centralized infrastructure (such as routers in wired networks or access points in wireless networks). "Ad hoc" in Latin means "for this purpose" and, by extension, improvised. In ad hoc networks, nodes join "on the fly" for a specific purpose (such as when a computer wirelessly connects to a printer). These networks support mobility and are very well suited for many difficult situations, such as rescue missions, natural disasters, emergency military operations, vehicular communications, etc.

The operation of ad hoc networks is distributed and decentralized. Routing (forwarding data to other nodes) is made dynamically by the nodes, according to the actual connectivity status and routing algorithm.

From a technological point of view, ad hoc networks can be classified in the following types:

- *Mobile Ad hoc NETworks (MANETs).* A MANET is a self-made network of mobile devices with wireless connections.
- *Wireless Mesh Networks (WMNs).* A WMN is a communications network of radio nodes with a mesh topology. The network nodes can be laptops, smartphones, and other wireless devices. The mesh network, using routers

and gateways, transmits data among the wireless nodes. The communication is within the mesh network and not to the internet.

- *Wireless Sensor Networks (WSNs).* A WSN contains sensor-based devices to monitor environmental or physical parameters such as temperature, pressure, sound, etc. WSNs have application in many areas, such as traffic control, vehicle detection, greenhouse monitoring, etc.

The open movement has also reached ad hoc networks, from the point of view of use and management. Wireless Mesh Networks (WMNs) [91] allow local communities to build their own network infrastructures, known as *Community Networks (CNs)*, providing affordable inter-networking with the Internet and including isolated rural communities worldwide [77]. Sharing resources, such as infrastructure or Internet access, is encouraged at all levels [41, 87] to lower the cost of network infrastructures and services. Community Networks are distributed, large-scale, and decentralized networking infrastructures composed of several nodes, links, and services where the resources are made available to a group of people living in the same area. Networks of this kind are extremely diverse and dynamic because they are composed of decentralized nodes and they mix wired and wireless technologies, several routing schemes, and diverse services and applications [6]. The network is managed using an open peering agreement, which avoids barriers for the participation in the network. Governance, knowledge, and ownership of the network are open. These networks are, therefore, not only decentralized but also owned and managed by community members. In addition, they grow dynamically with regards to the number of links, their capacity, and services provided. Some relevant examples of Community Networks include guifi.net [9] or FunkFeuer [30]. Among many others, the guifi.net Community Network exemplifies how communities can develop their own network infrastructures as a commons [9], using several interconnected WMNs. This results in a large-scale and heterogeneous network which uses diverse routing protocols in several network zones.

The main problems in ad hoc networks [24] are routing and characteristic of wireless communication. In infrastructure's networks a node can communicate with all nodes in the same cell. In ad hoc a node can communicate only with nodes in its area. This node can communicate with other nodes, but a routing algorithm is necessary. Unlike wired communication, wireless networks have transmission problem with data transmission such as, possibility of asymmetric connections and higher interferences.

The dynamic and changing characteristics of the ad hoc networks, makes the work done by the routing layer to have a high impact on the performance of these networks. It is very important for the routing layer to quickly react to changes that happen, and even be advanced to what will happen in the near future, by applying prediction techniques. This allows the routing layer to have a proactive approach (try to avoid a problem before it happens), which is more efficient than a reactive approach (which try to correct the problem when it is already present). In order for the routing layer of the ad hoc network to do its job (to decide what is the best path for an information to reach its destination) it is necessary to know what paths or links are available between the nodes (topological information) and how good these paths or links are (quality).

In this context, our **overall research question** is: *What is the improvement achieved when we apply prediction to the routing layer of ad hoc networks?*

1.2 Problem Statement

In ad hoc networks, the operation of the routing layer plays a vital role in the performance achieved by these networks. In this context, the general research question, of *what is the improvement achieved when we apply prediction to the routing layer of ad hoc networks*, can be addressed in two key aspects of the information used by the routing layer: (1) the topological information and (2) the quality of the connections among network nodes. These two key aspects give rise to the **two specific research questions** that we address in this thesis:

Specific Research Question 1 (SRQ1): What impact does prediction have on the topological information used by the routing layer?

Specific Research Question 2 (SRQ2): What impact does prediction have on the quality of ad hoc networks?

To study these two aspects in depth, we will analyze real use cases to determine the impact achieved by the prediction techniques. On the one hand, *SRQ1* focus on the prediction of the topological information needed by the routing layer to know what nodes are within reach of other nodes on the network, to route data communications. The use case we choose for *SRQ1* is the OLSR protocol, that is optimized for mobile ad hoc networks. On the other hand, *SRQ2* focus on the prediction of the quality of communications (both link and end-to-end) to route data communications through nodes with higher quality. To assess the improvements achieved, the use case we choose for *SRQ2* is Community Networks.

1.2.1 Topological Information Prediction

Regarding *SRQ1*, the focus is on the prediction of the topological information used by the routing layer. Several social computing participation strategies (e.g. crowdsensing and crowdsourcing), use mobile ad hoc or opportunistic networks to support user's activities. The unreliability and dynamism of the communication links provided by these kinds of networks makes the routing protocols a key component to be able to achieve efficient and reliable data communication. Often the routing capabilities come at expenses of flooding the network with TCI, which can overload the communication links, and dramatically increase the energy consumption of the participating devices.

We want to analyze if predicting the network topology can reduce the number of control messages sent through the network, and their impact on energy consumption and available bandwidth. First, it is necessary to determine the predictability limits of the history-based prediction, and the performance of different history lengths. Then the implementation of a real predictor will allow us to assess the performance achieved.

1.2.2 Quality Prediction

With respect to *SRQ2*, we want to analyze the quality both locally (link) and globally (end to end, or full path), in a Community Network scenario. Community Networks have emerged under the mottos “break the strings that are limiting you”, “don't buy the network, be the network” or “a free net for everyone is possible”. Such networks create a measurable social impact as they provide to the community the right and opportunity of communication. As any other network that mixes wired and wireless links, the routing protocol must face several challenges that arise from the unreliable nature of the wireless medium.

LQ tracking helps the routing layer to select links that maximize the delivery rate and minimize traffic congestion. Moreover, LQ prediction has proved to be a technique that surpasses LQ tracking by foreseeing which links are more likely to change its quality. In this step of the thesis, we will focus on LQ prediction by means of a time series analysis. We will apply this prediction technique in the routing layer of large-scale, distributed, and decentralized networks. We want to determine if it is possible to accurately predict the Link Quality of the instances, both in the short and the long terms. Particularly, we will analyze the behaviour of the links globally to identify the best prediction algorithm and metric, the impact of lag windows in the results, the prediction accuracy some time steps ahead into the future, the degradation of prediction over time, and the correlation of prediction with topological features. Moreover, we will also analyze the behaviour of links individually to identify the variability of LQ prediction between links, and the variability of LQ prediction over time. Then, we will also present an optimized prediction method that will consider the knowledge about the expected LQ values.

EtEQ or Path Quality extends the LQ concept to the full communication path (between sender and receiver) and it is computed based on the quality, i.e. ETX, of the individual links that conform the communication path. In this last step of the thesis, we want to analyze if our previous research on LQ is applicable to the full communication path (EtEQ tracking and prediction) and determine what differences exist between individual LQ and EtEQ. To the best of our knowledge, no previous work explores EtEQ prediction in the routing layer of large-scale, distributed, and decentralized systems. EtEQ tracking helps the routing layer to select paths that maximize the delivery rate and minimize traffic congestion. We believe that EtEQ prediction can be a technique that surpasses EtEQ tracking by foreseeing which paths are more likely to change quality. We focus on EtEQ prediction by means of time-series analysis. We will apply this prediction technique in the routing layer of large-scale, distributed, and decentralized networks. We want to demonstrate if it is possible to accurately predict End-to-End Quality with an small average Mean Absolute Error (MAE). Particularly, we will analyze the path properties and path ETX behavior to identify the best prediction algorithm. Moreover, we will analyze the EtEQ prediction accuracy some steps ahead in the future and also its dependency on the hour of the day.

1.3 Experimental Methodology

Aiming to answer the two presented specific research questions, our work has followed an empirical quantitative methodology, based on the design science problem proposed by [73]. This methodology is:

- **Collection of data**: To characterize the problem of what routing information is available and are likely to be predicted, we performed simulations with different mobility models to get traces of Topology Control Information with varied parameters. We also used real data about quality, collected from actual Community Networks.

- **Definition of research objectives**: The collected data, together with the literature reviewed, helped us to clearly define the objectives of our work.

- **Design and Implementation**: We developed the different parts of the research following an experimental design approach, specifying variables, measurement methods, and requirements, and also selecting appropriate technologies.

- **Experimentation and Simulation**: For the evaluation of our work we performed experiments with synthetic and real network data, depending on the nature of the problems addressed. Additionally, simulations were performed to demonstrate the impact of variables that were not considered in our proposed design. These simulations do not rely on an existing network simulator, but are implementations of analytical methods that represent different scenarios, and the results were computed based on input data from the existing network components.

1.4 Contributions

1.4.1 Prediction of Topology Control Information

The operation of the OLSR routing protocol can be improved by applying History-Based Predictor techniques (OLSR-HBP). This is the main conclusion that we can draw when we address *SRQ1: What impact does prediction have on the topological information used by the routing layer?* In this way, we have analyzed the performance of a History-Based strategy for predicting the Topology Control Information (TCI) generated by routing protocols for mobile ad hoc and opportunistic networks. This analysis was performed by simulating several mobile collaboration scenarios. The results obtained indicate that the History-Based Predictor (HBP) strategy contributes to reduce the traffic on these networks with a minimal increase on energy in the mobile devices, supporting mobile collaborative activities.

First, we have observed that most of the time, for low densities of nodes, a message has already appeared in the past. This percentage falls when considering a network with a higher node density. This demonstrates that the upper bound limits of the HBP strategy remain high for a wide variety of interaction scenarios, which make us expecting important benefits for mobile collaborative applications that use these networks as communication support.

Second, the results also show that few messages contribute significantly to the total percentage of messages delivered through the network. This means that there is a high opportunity for predicting the TCI, and this prediction can be just focused on a small subset of messages. Finally, we have identified the role played by different History-Depth patterns, prediction policies, confidence mechanism, and the combination of several approaches at the same time, after analyzing the behavior of HBP mechanisms under several scenarios.

In order to experimentally assess the previous analytical results, we implemented the HBP approach over the OLSR protocol (OLSR-HBP). We executed OLSR-HBP to measure diverse metrics on real network conditions. Regarding the percentage of control messages predicted, its value is lower than the potentiality obtained from the analytical results, because OLSR-HBP only can skip a control message transmission when all the destination nodes are able to make a correct prediction (*global hit*). Our approach is independent of the specific routing protocol and, therefore, it can be adapted to routing protocols other than OLSR.

We also evaluated the impact of OLSR-HBP in the network performance and conditions. The OLSR-HBP executions we performed did not affect the network operation. Throughput and Latency metrics performs almost the same as the original OLSR protocol. Regarding the resources used by OLSR-HBP (Energy and Memory), the Energy consumption slightly increases with respect to OLSR. The decrease in TCI achieved by OLSR-HBP does not translate in energy savings, because the number of packet transmissions usually increases (but with less messages per packet). With respect to the amount of Memory used to store TCI in each node, the OLSR-HBP memory requirements can be considered scarce for the amount of memory available on current common devices acting as communication nodes. Also, these low memory requirements allow OLSR-HBP to be implemented in current communication nodes. Conversely, other routing proposals based on Deep Learning, require off-node computations, with higher computing resources and additional communication overheads.

In summary, OLSR-HBP achieves important decreases of TCI (signaling overhead), without disturbing the network operation, and OLSR-HBP requires a small and affordable amount of resources. Our approach is deterministic and with much fewer resource requirements, in comparison with other statistical proposals (as Machine or Deep Learning).

The work related with this contribution is presented in Chapter 2 and was originally reported in [66, 67].

1.4.2 Link-Quality Prediction

Time-series analysis is a promising approach to accurately predict Link Quality values in Community Networks. This is the main conclusion that we can draw when we address *SRQ2: What impact does prediction have on the quality of ad hoc networks?* with a first focus on Link Quality (LQ). Therefore, we demonstrate that Time-series analysis can be used to improve the performance of the routing protocol by providing information to make appropriate and timely decisions to maximize the delivery rate and minimize traffic congestion.

We analyzed results from four learning algorithms that model time series: Support Vector Machine (SVM), k-Nearest Neighbours (kNN), Regression Tree (RT), and Gaussian Process for Regression (GPR). All algorithms achieved high percentages of success when predicting the next future value of the LQ, with RT being the best one. Moreover, these results were obtained just considering those links that experienced variations. Therefore, the prediction accuracy could have been even better if all the network links were included. In addition, we show that the prediction of values that are more than one step ahead in time (and not just the next value) also achieves high success ratios. We also observed that the size of the training data-set is a key factor to achieve high accuracy in the predictions. The bigger the size of the data set, the smaller the degradation of the error over time.

The global analysis of the LQ behaviour we performed, allowed us to identify the best prediction algorithm (RT) and metric (Mean Absolute Error), and to understand the impact of lag windows in the prediction (although we were not able to determine the best lag window size). It also helped us to evaluate the accuracy of prediction some time steps ahead into the future (it seems possible to predict the LQ some steps ahead), the degradation of prediction over time (the degradation follows a linear function: the larger the size of the training dataset, the smaller the error), and the correlation of prediction with some topological features (node degree does not present correlation with LQ, but there are indications to the fact that edge betweenness produces changes in the LQ; however, it is not straightforward to apply this correlation to the prediction process).

We also analyzed the behaviour of links individually, to identify the variability of the LQ prediction between links and over time. Finally, we also enhanced the prediction method by taking into consideration our previous knowledge about the expected LQ values. Hence, we can claim that the optimization strategies based on the correction of the predicted value lead to significant results improvement.

The efforts related with this contribution are presented in Chapter 3 and were originally reported in [64, 65].

1.4.3 End-to-End Quality Prediction

Time-series analysis is a promising approach to accurately predict End-to-End Quality values in Community Networks. This is the main conclusion that we can draw when we address *SRQ2: What impact does prediction have on the quality of ad hoc networks?* with a last focus on end-to-end (or path) quality. In this way, we demonstrate that time series analysis can be used to improve the performance of the routing protocol by providing information to make appropriate and timely decisions to maximize the delivery rate and minimize traffic congestion.

The dataset we used in our analysis shows quite significant fluctuations in the temporal evolution of the number of paths (routing entries). This contrasts with the general assumption in mesh networks that the number of paths is stable. The persistence of paths in our dataset is very stable for the majority of paths, but there are some paths with lower persistence values, and those paths create the temporal fluctuations in the number of paths.

Regarding the Expected Transmission Count (ETX) behavior, the average ETX path quality is more stable at night, and presents more fluctuations during the morning. In non-working days, this behavior is less apparent. We assume this is due to the variation of network traffic and interference during the day, leading to packet loss. Hence, the accuracy of prediction depends on the day and time when it is applied. On the other side, we determine the most frequent number of Hops and ETX values. The dispersion of ETX according to the number of Hops is minimal at both ends of the number of Hops range. However, the dispersion is significant and relevant at the middle range of the number of Hops. Therefore, the accuracy of the predictions would be better or worse depending on the number of Hops.

We present results from four well known learning algorithms that model time series. All of them achieved high percentages of success when predicting the next value of the EtEQ. We also analyzed the error variability and found that three of them presented similar performance, whereas the other performs worse due to outliers with larger errors. We have also observed differences in the prediction behavior during day and during night, as it happens with actual ETX values.

The main results related with this contribution are presented in Chapter 4 and were originally reported in [63].

1.5 Thesis Structure

The rest of this thesis is organized as follows.

Chapter 2 discusses the impact of predicting the Topology Control Information (TCI) generated by routing protocols in unreliable networks. We implement a real predictor in order to empirically assess the performance achieved. The predictor is applied to the OLSR routing protocol and follows a History-Based approach that uses information of the nodes past behavior. As a result, we propose and evaluate the OLSR History-Based Predictor (OLSR-HBP), using as baseline for this evaluation the original OLSR protocol (without prediction). Our study also determines the predictability limits of the proposed strategy, assuming that a TCI message can be correctly predicted if it appeared at least once in the past. We analyze three different history durations.

In Chapter 3, we focus on Link Quality prediction by means of a time series analysis. We apply this prediction technique in the routing layer of large-scale, distributed, and decentralized networks. Particularly, we analyze the behaviour of the links globally, to identify the best prediction algorithm and metric, the impact of lag windows in the results, the prediction accuracy some time steps ahead into the future, the degradation of prediction over time, and the correlation of prediction with topological features. Moreover, we also analyze the behaviour of links individually to identify the variability of LQ prediction between links, and the variability of LQ prediction over time. Finally, we also present an optimized prediction method that considers the knowledge about the expected LQ values.

Chapter 4 is devoted to End-to-End Quality (EtEQ) by means of time-series analysis. We apply this prediction technique in the routing layer of large-scale, distributed, and decentralized networks. We demonstrate that it is possible to accurately predict EtEQ with an small average Mean Absolute Error (MAE). Particularly, we analyze the path properties and path ETX behavior to identify the best prediction algorithm. Moreover, we analyze the EtEQ prediction accuracy some steps ahead in the future and also its dependency on the hour of the day.

Chapter 5 concludes and summarizes the findings of this thesis and draws future directions that deserve further work.

Chapter 2

Control Information Prediction

This chapter presents a study that quantifies the impact of predicting the Topology Control (TC) messages generated by routing protocols in wireless ad hoc networks. The underlying idea behind this prediction is that some topological changes in these networks have happened before in the past, therefore, a History-Based Predictor (HBP) can take advantage of these patterns following a simple and low-cost History-Based approach. This will allow us to answer the *Specific Research Question 1 (SRQ1): What impact does prediction have on the topological information used by the routing layer?* First, we show the potential benefits of this approach using simplified communication scenarios. Then, in order to determine the impact of this topology prediction strategy in real scenarios, a real predictor is implemented and added to the Optimized Link State Routing (OLSR) protocol. The prediction capability of this component, named OLSR History-Based Predictor (OLSR-HBP), is evaluated using a common use case for static and mobile scenarios. Almost any routing protocol for mobile ad hoc networks can take advantage of this prediction approach as a way to reduce network traffic, and consequently, the energy consumption on the devices participating in these networks.

2.1 Introduction

Everyday becomes more evident the key role that mobile computing and wireless technologies play in our daily activities. Being always connected, anytime, and anywhere is today more a necessity than a luxury. The ubiquitous computing scenarios created based on these technology advances allow people to provide and consume shared information [1, 68] like the status of the traffic in a particular area

[2, 14, 74], the security level of a specific neighborhood [33, 35], or the location of interesting places to visit during a touristic activity [17, 37, 46].

In these scenarios, the supporting communication networks are typically wireless and ad hoc. Therefore, the provision of the application' services makes sense only if consumers (i.e., the end-users) are located near to the service providers. For instance, during lunch time restaurants located in a pedestrian area can deliver special offers irradiating messages to the smartphone of the passers-by, motivating them to have lunch or dinner in their business premises. If the service providers (i.e., the restaurants) use WiFi to deliver these messages, then the process will require message routing to reach the devices of potential clients that are located more than one-hop distance from the message sender.

The routing protocols used in these mobile scenarios must be simple, efficient, reliable and must have the capability of quickly adapting themselves to changes on the network topology [85, 90, 97, 98]. Thus, they try to maximize the reachability of the target nodes, by consuming as few energy of the network as possible. To deal with these topology changes, the routing protocol exchanges TC messages with a certain frequency. Quick detection of topology changes requires high frequency of TC messages, at the cost of additional signaling overhead. Therefore, novel solutions are required to reduce the number of TC messages exchanged among the network nodes, and consequently, decrease the network traffic and energy consumption on the participating devices; particularly, on mobile devices that depend on their battery. These solutions should keep the routing performance of the current protocols.

In previous works, we have shown that the traffic generated by a routing protocol, like Optimized Link State Routing (OLSR) [20], grows almost exponentially with the number of nodes if different node densities are considered in the analysis [58, 59]. Therefore, a network with a large number of nodes, requires to exchange a huge amount of TC messages to inform the nodes about the topology changes. These messages not only overloads the communication links, but also increases the energy consumption of the nodes. We have also shown that the problem of delivering much control information through the network can be addressed using predictions [58, 60]. Particularly, the OLSRp protocol was proposed to eliminate redundant control information, and thus, to reduce CPU and energy consumption in mobile ad hoc networks. This prediction mechanism is based on the assumption that the last TC message sent by a node will probably be repeated during the next round of information delivery. Such a proposal shown positive results in a simplified communication scenario, but its real capabilities need to be determined through more complex scenarios that consider the diversity of variables affecting the TC message exchange in mobile ad hoc networks.

In order to quantify the benefits of this prediction approach in real-world scenarios, we implemented a real predictor of the network topology and we added it to the OLSR protocol. We named this predictor OLSR-HBP and we evaluated its performance using simulations that involved several stationary and mobile scenarios. These scenarios are representative of some everyday life activities, where people move freely around a certain area, and eventually interact with other people.

Our approach is a deterministic one, a History-Based tree. This method is far away of some new proposals that are based on Machine and Deep Learning techniques. The predictions we make do not require a previous training step to generate a model. Another important issue is the consumption of resources in terms of CPU and memory. Our proposal uses a simple tree data-structure. In this case, the computation power needed to update the History-Based tree is very small. As a result, the memory and CPU requirements of the proposal are feasible for actual wireless routers.

Several research studies have addressed the issue of network overload caused by the TC messages of the routing protocols in ad hoc networks. Some of these works have analyzed the effect of tuning some parameters of the routing protocols [31, 32, 40, 88], showing that increasing the frequency of control messages allows a quicker detection of topology changes at the cost of additional signaling overhead. There are also trade-offs on power consumption, available bandwidth, latency, and throughput.

Prediction is a well-known technique that has been applied successfully in several areas of computer science [34, 50, 55, 84, 89], including routing protocols. In general, if the percentage of right predictions is high enough, the overall performance of the routing protocols can be significantly improved. However, this process typically introduces complexity to routing protocols in the form of additional hardware and software, required to make and validate predictions. Prediction can also introduce time penalties to the system, mainly when the rate of mispredictions is high.

Prediction techniques have been applied to different aspects of routing protocols: mobility prediction and reliability of the network topology [19, 80, 86], link-quality prediction [11, 45, 63–65], and network traffic reduction [58–60].

Mobility prediction has also been applied to ad hoc networks and mobile scenarios. The Mobile Gambler's Ruin (MGR) algorithm [80] deals with global mobility prediction, to identify nodes that are more likely to disconnect in the near future. The MGR aims to keep continuous connections among mobile devices in ad hoc networks. Mobility prediction has also been locally applied to estimate the link expiration time between adjacent mobile nodes [86] and determine if a node moves from its current position to the next location within a certain period of time [19]. In the former case, prediction helps reconstruct routes before they expire and, in the latter, prediction facilitates resource reservation and route maintenance decisions.

Link Quality (LQ) prediction is used in combination with LQ tracking [26, 45, 93, 96] to determine which links are more suitable to change its behaviour in the future. This helps the routing protocol to select higher-quality links, which maximize delivery rate and minimize traffic congestion. Different Link Quality Estimators (LQE) metrics [15, 28, 83], can be used in isolation or can also be combined [8, 51, 57, 76] to build a more accurate estimation, and to select the most suitable nodes when making routing decisions. MetricMap [94] is a routing protocol for Wireless Sensor Networks that uses a learning-enabled LQ assessment method in which a Machine Learning algorithm is applied to classify Link Qualities, and offline-collected rules are used to make LQ predictions at runtime. Other related works [59, 63] are focused on the use of time-series analysis to predict future LQ and Path Quality values in large Wireless Mesh Community Networks.

The Kinetic Multipoint Relaying (KMPR) protocol [38] also focuses on reducing the amount of redundant retransmissions while broadcasting a message through a mobile network. The KMPR approach is based on reducing the number of nodes that are allowed to forward the message. Mobility prediction is applied to detect when a change in the neighborhood is about to happen, allowing a fast adaptation to topology changes. Recent works have begun to apply Deep Learning (DL) architectures and algorithms to network traffic control systems. The authors of [27] develop and analyze the results of a new DL application for intelligent routing operations of a backbone network. The results obtained outperform conventional routing strategies, like the Open Shortest Path First (OSPF). There are other works that focus on routing strategies based in DL [43]. In the case of routing in Wireless Sensor Networks, DL has been used extensively and efficiently [4, 25, 29].

All these previous works apply a *probabilistic* approach. By contrast, the approach presented in this chapter is *deterministic*, i.e., given the same input data, the routing decision will always be the same. Our approach has also a low cost in terms of CPU resources, because it uses simple memory-based trees to aggregate in addition to Topology Control Information. Furthermore, this prediction approach keeps the default parameter values of the routing protocols, which facilitates its implementation and portability to a variety of other protocols. In this sense, our proposal provides flexibility to improve the performance of the predictor by tuning these parameters.

The main contributions of this chapter are the following:

- An in-depth analysis of the potentiality and limits of using a History-Based approach to predict TCI information. We determine the upper-bound limit and the ratios of accuracy.
- The implementation of a real predictor of the network topology that was added to the OLSR protocol (OLSR-HBP), in order to quantify the benefits of this prediction approach in real-world scenarios.
- The evaluation of OLSR-HBP performance using simulations that involved several stationary and mobile scenarios (that are representative of some everyday life activities, where people move freely around a certain area, and eventually interact with other people).
- The analysis of diverse history durations (short, medium, long) and its impact on the results and performance.
- An analysis to determine whether or not the decrease in TCI messages does negatively affect the data transfer capabilities of the network, and what are the extra resources needed by the predictor.

The rest of this chapter is organized as follows. Section 2.2 presents the experimental methodology used in this chapter. Section 2.3 corresponds to the analysis of results, starting with the study of a static scenario in Subsection 2.3.1, then Subsection 2.3.2 analyses the potentiality of our proposal in dynamic scenarios, and we close with Subsection 2.3.3 where we implement the predictor and asses its performance in dynamic scenarios. Finally, section 2.4 summarizes the conclusions of this chapter.

2.2 Experimental Methodology

In this section we apply the methodology presented in section 1.3. Subsection 2.2.1 explains the data collection process. In subsection 2.2.2 we define the research objectives. Then, in subsection 2.2.3 we explain the design and implementation step of the methodology. Finally, subsection 2.2.4 presents the experiments and simulations performed.

2.2.1 Data Collection: Mobility Models and Scenarios

We performed simulations with different mobility models to get traces of Topology Control Information (TCI) with varied parameters.

The simulation setup was configured to represent a real-world use case of mobile collaboration activities. The physical space was a square open area of 300x300 meters, which can represent a beach or a park, where people are free to move and eventually interact with other people. In these scenarios, people represent friends, relatives, or acquaintances, such as service providers, who can remain stationary (e.g., during a picnic) or can be walking (with or without a clear direction). In a context like this, a mobile collaborative application that detects the presence of people in the area, can be used to promote face-to-face interactions and service provision among them, as proposed in [92].

The simulations performed considered devices using WiFi to detect other nodes and to exchange control information. Mobile networks composed of 10, 20, 30, and 40 nodes were simulated. The nodes were randomly deployed in the area, and their behavior alternated between some stationary periods and others in which they moved at 1 m/s (walking), 2 m/s (trotting), 4 m/s (running), and 6 m/s (bicycling).

Different mobility models [13, 47] were simulated using the BonnMotion simulator [5]. These models are quite representative of the mobility patterns of people performing outdoor activities.

- The *Random Walk* model considers people moving randomly in terms of direction and speed within a certain area; e.g., people in a park, where each person can be walking, running, or bicycling without using formal paths.
- The *Nomadic* model considers people moving in groups, from one location to another. This is representative of guided tours, e.g., at a city downtown.
- Finally, in the *Self-similar Least Action Walk (SLAW)* model people move quite randomly, but it considers previous movements (speed and direction) to determine the new ones. Unlike other models, the speed in this model cannot be parameterized, and it assumes a default value of 1 m/s (walking speed). SLAW is also effective to model casual encounters among community members; e.g., students at the university campus or friends in a theme park.

The capabilities of the nodes used in the simulations are equivalent to most common smartphones. These devices have an effective WiFi range of approximately

80 meters in open areas. In such range, we can expect quite stable ad hoc communication among devices and a bandwidth of at least 50 Kbps, which is appropriate to support reliable interactions among mobile nodes.

2.2.2 Research Objectives: Control Information Prediction

The research objective of this chapter is to answer *SRQ1: What impact does prediction have on the topological information used by the routing layer?* For this reason, we focus on the prediction of the topological information needed by the routing layer to know what nodes are within reach of other nodes on the network, to route data communications. The use case we choose for SRQ1 is the Optimized Link State Routing (OLSR) protocol, that is optimized for mobile ad hoc networks.

We want to analyze if predicting the network topology can reduce the number of control messages sent through the network, and their impact on energy consumption and available bandwidth. First, it is necessary to determine the predictability limits of the History-Based Prediction, and the performance of different history durations. Then the implementation of a real predictor will allow us to assess the performance achieved.

2.2.3 Design & Implementation: History-Based Predictor

2.2.3.1 Conceptual Idea

The conceptual idea of our proposal is depicted in Figure 2.1. Neighbor nodes transmit TCI messages between them. Many of these TCI messages are redundant and, thus, can be predicted. We add our predictor (green box in Fig. 2.1) between the Routing Layer and the Network Layer of both the source and destination nodes. Based on historical TCI, our predictor determines if every TCI message can be skipped from transmission at the source node, because it can be predicted at the destination node. When an expected TCI is not received at destination node, this means that a prediction should be made, and the predictor at destination node generates the "missing" TCI message, injecting it into the upper Routing Layer (as if the TCI message had actually been received).

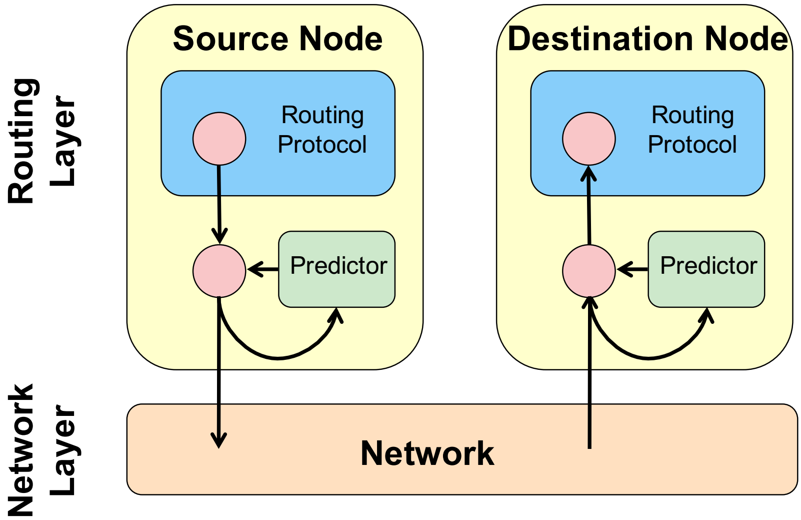


Figure 2.1: Network architecture of our proposal.

2.2.3.2 Theoretical Framework

This subsection presents the conceptual framework of the proposed History-Based Predictor (HBP). The basic idea behind the proposed predictor is that many Topology Control (TC) messages interchanged between nodes in ad hoc networks are redundant and can be predicted. Therefore, a predictor component is implemented as a transparent communication intermediary between the routing and the network layers, as shown in Figure 2.2. Based on historical information, the predictor determines if a TC message can be predicted at destination and as a result, the message transmission could be prevented. When a TC expected at destination is not received, a prediction is made to generate the "missing" message (pTCm). This message is passed to the upper routing layer as if the TC message had actually been received through the network. This means that the predictor reduces the amount of control traffic (gTCm) transmitted through the network, without modifying the signaling information and how it is processed by the nodes.

The HBP method is focused on the prediction of a topological state that has already appeared in the past. It considers that each node keeps a recent history of the TC messages received from its neighbors. When a prediction process is required, the topological history is used as input to determine the future topology, preventing the delivery of signaling information through the network.

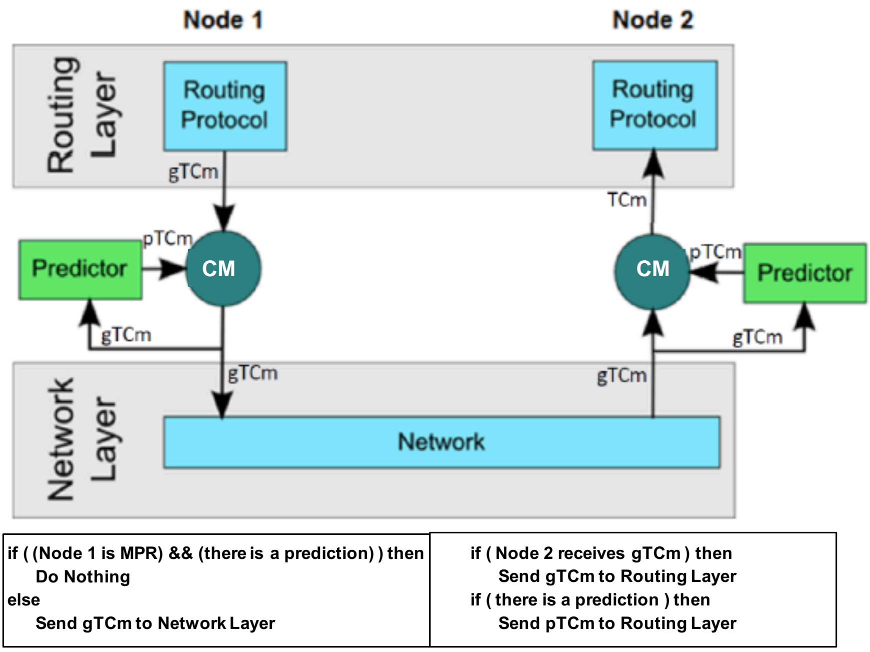


Figure 2.2: Network architecture including OLSR with Prediction (OLSRp).

In HBP, each node keeps historical information, which is locally updated and stored in a table. The proposed HBP approach uses unbounded tables, because they provide flexibility for the identification of the mobility patterns of the nodes, and thus, to determine future network topologies.

By analyzing the nodes' tables, it can be observed that each mobility pattern corresponds to a sequence of TC messages (one or more) that the node making the prediction has seen in the past (and has registered in its local table). The tables also record statistical information that helps the prediction process to select one option among several candidate mobility patterns. These statistics include both, the most frequent and the last message of the pattern. In short, an entry of the table will be composed by the following items:

- An input: the TC messages representing the pattern.
- An output: the list of control messages that appeared after each pattern.
- Statistical information related to every output, which helps to predict the next TC message.

Figure 2.3 shows an example of the prediction table, that assumes patterns with two TC messages. At the top of the table we can observe a sequence with TC messages A, B, and C. There are six entries with all the pairs of messages that appeared in the sequence. The first column contains the patterns of two consecutive messages. The second column contains all the messages appeared after the pattern. The third column, records the number of times each message appeared. Finally, the fourth column marks the last message appeared after the pattern.

AAAABABAACBAABBAB

Pattern	Next	Count	Last
AA	A	2	
	B	2	#
	C	1	
AB	A	2	
	B	1	#
AC	B	1	#
BA	A	2	
	B	2	#
BB	A	1	#
CB	A	1	#

Figure 2.3: Example of prediction table for HBP.

In order to analyze the usefulness of this historical information for the prediction process, the proposed HBP approach defines the *History Depth (HD)* metric. This metric can be computed as the number of TC messages that composes a mobility pattern. For instance, if we consider a table with HD=1, the number of messages that identify each pattern is one, and there will be one entry in the table for every TC message appeared in the past. It seems reasonable to think that a larger HD leads to high prediction accuracy, but low prediction opportunities, because the message sequences (patterns) are longer.

According to how the messages of the patterns are selected to make predictions, we analyze three different versions of the HBP approach: *last value* (uses the last/most-recent message of the pattern), *most-frequent value* (uses the most-frequent message), and *random value* (uses any randomly selected message).

The proposed HBP method includes a confidence mechanism to determine the likelihood of a right prediction. The purpose of this mechanism is to avoid incorrect predictions, which can skew the network topology map and decrease the reliability of the process. The confidence mechanism used determines if each message of the output had already been predicted in the past. If so, a counter is incremented by 1. Otherwise, it is decremented by 1. The counter is initialized to 1, and it can take values in the range from 0 to 3. A prediction is considered to have a high confidence value (high certainty in a right prediction) if the counter is equal to or higher than 2.

Although applying the confidence mechanism can improve the prediction accuracy, the opportunities of performing predictions are fixed if we consider a fixed/constant HD. Therefore, in this chapter we analyze a HBP approach, the *Prediction Tree*, in which the HD is not initially fixed. In the Prediction Tree the maximum HD is assumed to make a prediction. When it is not possible to make such an assumption (e.g., because there are no entries in the table, or there is not enough certainty in the prediction), the HD will be decreased by 1. Then, the Prediction Tree attempts to make a new prediction, but using a shorter pattern. This will be repeated until the HD metric reaches 0. This approach was analyzed both, with and without a confidence mechanism.

To understand the limits of the HBP approach and determine how far a particular version of this approach is from the best performance, different metrics were analyzed. The repetition of TC messages over time was quantified, as well as the maximum prediction accuracy of the different HBP versions. Moreover, the number of incorrect predictions (for various HD values) that could be correctly predicted were also quantified. We assume that if a particular TC message has appeared in the past at least once, it could be correctly predicted.

This chapter also analyzes the representativeness of the most-frequent messages with respect to the whole set of messages received by a node over time. This give us an understanding about how difficult is to make right predictions, and which is the amount of data (historical information) that must be tracked to be able to make these predictions.

To experimentally validate the performance of the HBP approach in a real world scenario, we implemented a predictor on top of the Optimized Link State Routing (OLSR) protocol [20]. The predictor, named OLSR History-Based Predictor (OLSR-HBP), was evaluated in stationary and dynamic network scenarios, and the results were analyzed using diverse network metrics.

The nodes in OLSR networks periodically exchange routing information to keep a map of the network topology. The Multi-Point Relays (MPRs) are the network nodes selected for propagating the topology information. In OLSR, there are two types of control messages: HELLO and Topology Control (TC). HELLO messages allow each node to discover its neighboring nodes, and to obtain information about the state of the links with them. TC messages allow MPR nodes to disseminate neighbor information throughout the network.

2.2.3.3 Predictor Implementation

OLSR-HBP was implemented by extending the C++ code of the NS-3 OLSR module [70]. Its implementation considers TC sequences at both origin and destination nodes. Origin nodes broadcast TC messages to different destination nodes. Because of mobility, it is necessary to keep separate TC sequences for each pair of origin-destination nodes, and the prediction should be made for each separate sequence. Every node should keep updated TC sequences for the nodes that receive TC messages from it, and also for the nodes that send TC messages to it.

OLSR-HBP at origin also considers that the next TC message can be omitted (not transmitted) if all destination nodes can predict correctly (hit) this same TC message from their corresponding individual sequences. We call this a *global hit*. If the target nodes do not receive a TC message within a time limit, they should consider that the next TC message predicted is a hit, which should be used to update the topology information.

TC messages contain diverse information, particularly the IPv4 address of the originator node, the IPv4 addresses of its neighbors, the Advertised Neighbor Sequence Number (ANSN), a number that is incremented every time the topology information of the node changes, and the Message Sequence Number (MSN), incremented every time a node sends a new control message).

When predicting a new TC message at the destination node, all this information should be generated. The originator's address and the list of neighbors are generated using the historical information stored in the nodes' tables. To generate the ANSN, we use the same last ANSN value received from the origin node. To produce MSNs we introduced "gaps" in the sequence of MSNs, so that the destination nodes can detect these "gaps" and use them as the MSN for the generated TC. If the last MSN received is not consecutive to the previous MSN received, a gap is detected, and the missing MSN is put in a small buffer. When a new TC message is generated, its MSN value is taken from this buffer. If the buffer were empty, we use an MSN value equal to 0.

We added data structures to the NS-3 OLSR module to keep the information of the individual sequences of TC messages for every pair of origin-destination nodes that interchange TC messages. OLSR-HBP uses these data structures to detect *global hits* at origin nodes and therefore, to prevent broadcasting this TC message. At destination nodes, the predictor uses a timer for every possible origin node. If the timer expires and its expected TC message is not received, a new message is generated and processed as if it had been received from the origin.

The potential problems with the proposed approach are the following: (1) if a TC message that had been broadcasted is lost during transmission, OLSR-HBP at destination will wrongly assume that it was a *global hit* and thus, a wrong TC message will be generated; and (2) the predictor is not 100% accurate due to the delay in updating the routing information at destination, caused by the extra time required for a new TC message to arrive from the origin node before a prediction can be made.

2.2.4 Experiments & Simulations

In order to determine the performance of the HBP approach, we designed and simulated several interaction scenarios using NS-3 [78]. This simulation tool allowed us to model these scenarios, collect statistics, define initial network topologies, configure wireless network interfaces, and set the mobility patterns of the nodes. Every simulation performed in this study lasted 1, 4, or 24 hours. The simulations used the default configuration of the OLSR [20] protocol. This configuration considers the delivery of HELLO messages every 2 seconds, and TC messages every 5 seconds.

The simulation setup was configured to represent a real-world use case of mobile collaboration activities. The details of this use case have already been presented in subsection 2.2.1.

2.3 Analysis of Results

2.3.1 Benchmark Analysis in Static Scenarios

In this subsection we compare the use of a deterministic predictor with both, the standard OLSR protocol, and the use of Deep Learning (DL) techniques presented in [27]. Results from this comparison provide a benchmark against which the performance of the proposed OLSR-HBP can be measured.

In [27], the authors use the Open Shortest Path First (OSPF) routing protocol as benchmark method, whilst our baseline is OLSR (similar to OSPF, but it is used for ad hoc networks). The data traffic rate of the test scenario for the DL technique varied between 7.68 Mbps to 14.4 Mbps. Similarly, we considered 1 KByte ping data packets, sent every 10, 1, and 0.1 seconds, which corresponds to a range from 0.6 to 61 MBps of data traffic (due to network congestion, the maximum received was about 21 MBps). We also considered the same static scenario: a 4×4 wireless mesh backbone network, where 12 edge-routers generate packets for other edge routers, and 4 inner-routers just forward packets. Table 2.1 summarizes the main results of this study.

	OLSR			OLSR-HBP		
Ping period	10 s	1 s	0.1 s	10 s	1 s	0.1 s
%TC msgs pred @dest	-	-	-	99.8%	99.3%	63.4%
Min. % of pings recvd	100%	29.9%	10.8%	100%	28.3%	16.1%
Avg. % of pings recvd	100%	44.3%	34.5%	100%	49.7%	34.3%
Max. % of pings recvd	100%	64.7%	59.7%	100%	82.1%	69.0%
Min. Energy/node (J)	4.5	20.6	167.6	+5.4%	+3.0%	+24.1%
Avg. Energy/node (J)	8.6	32.5	361.1	+1.9%	+7.2%	+4.1%
Max. Energy/node (J)	14.0	46.9	566.3	+2.1%	+4.5%	-3.6%

Table 2.1: Results of OLSR and OLSR-HBP in a static scenario.

In the considered static scenario, OLSR-HBP outperforms the classic OLSR routing protocol in several performance metrics. Table 2.1 shows that the number of TC messages that can be correctly predicted (and thus not transmitted) achieves very high values (more than 99%), except for the congestion case, where the metric decreases to about 63%. Regarding the metric of ping messages received, the predictor receives between -0.2% up to +5.4% the average amount of pings received by OLSR. The minimum ping messages received by a node, ranges between -1.6% to +5.3% with respect to OLSR. In the case of the maximum value of pings received, our proposal always equals or improves the OLSR maximum values (up to +17.4%). Finally, for the metric of energy consumption per node, OLSR-HBP increases this metric in +4.4% on average, +14.2% on the nodes with minimum energy consumption, and just +1.0% on the nodes with maximum energy consumption. The only case when this metric is better for OLSR-HBP than OLSR, is the maximum consumption in the case of ping congestion (one ping message every 0.1 second), where the energy consumed is -3.6% that of OLSR. We analyze this energy behaviour in more depth in Subsection 2.3.3.4.

An important issue that was not considered in [27] is the consumption of resources in terms of CPU and memory. It is well-known that DL algorithms require a lot of Central Processing Unit (CPU) power and memory space. For this reason, the DL computation is performed on additional outsourced hardware (not in the router, which lacks the required resources). By contrast, the proposed OLSR-HBP is based on a deterministic predictor that uses a simple Tree data-structure. In this case, the computation power needed is very small (just enough to update the number of TC messages in the Tree). The amount of memory needed to store the Tree data-structure is typically just about 16 KBytes, with a few cases of up to 450 KBytes. As a result, the memory and CPU requirements of the method proposed in this chapter are feasible for real routers. It is also important to notice that, contrary to the DL technique, the implemented predictor works both in static and in dynamic environments where the nodes have mobility. The dependence on external hardware to perform DL computation requires a fast and reliable connectivity, which is not feasible in most dynamic environments.

Three network-performance metrics were evaluated and compared. The first metric is the signaling overhead, which is significantly lower for the DL routing than for OSPF (between 8% and 14%).

For OLSR-HBP the signaling overhead is less than 1% for 10 and 1 second ping periods. However, when there is network congestion, the overhead is about 55%.

Regarding the second compared metric, total throughput, DL techniques obtain between 31% and 72% higher throughput than OSPF. With OLSR-HBP there is a more moderate improvement of about 12% in average, with some cases achieving up to 50% better throughput than OLSR.

The last performance metric evaluated was the average per-hop delay. The DL method achieves delays from 65% to 72% lower than OSPF. With prediction techniques, there is a slight increase, in the range from 2% to 9%, in the OLSR average per-hop delays. Considering these results we can conclude that, if some test conditions are controlled, the performance of the proposed prediction approach is similar to the DL approach. In addition, some of the disadvantages of the proposed predictor, can be compensated by the fact that it can be applied to mobility scenarios, and also has very low computational and memory costs. In short, we can claim that DL techniques are more suitable for static scenarios, whereas OLSR-HBP is more appropriate for dynamic scenarios.

2.3.2 Prediction Opportunities in Dynamic Scenarios

2.3.2.1 Quantifying Predictability Limits

In order to determine the predictability upper-bound limit achievable by the HBP approach, we have assumed unbounded memory for the nodes, and also checking if a TCI message has ever appeared in the past. Figure 2.4 shows how the three mobility models behave when considering several node densities (from 10 to 40 nodes) and a similar average mobility speed (1 m/s) for the nodes in every model. Notice that for the scenario with 10 nodes, about 80% of the time the TC message to be predicted has already appeared in the past. This upper-bound limit is extremely high. Therefore, the potentiality of predict right the TCI is also high.

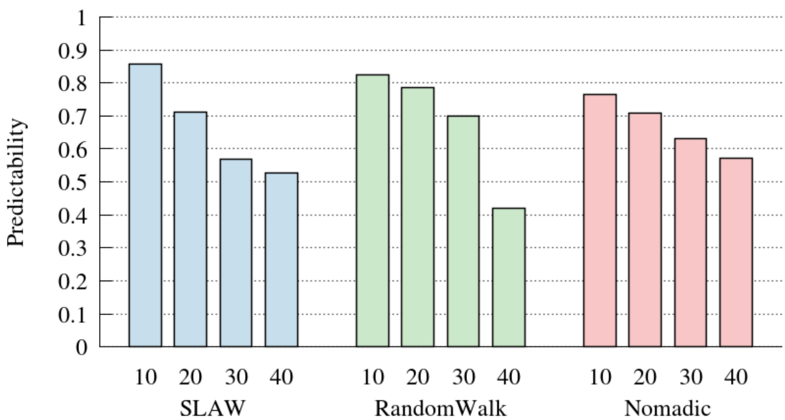


Figure 2.4: Predictability limits.

Besides that, we can see that there are no significant differences among the prediction capability in the three mobility models. This would be indicating that the prediction capability of the HBP approach does not depend on the mobility model being used by the nodes. In fact, we could expect similar results even in scenarios where the nodes use several mobility patterns.

The results also show that the prediction limits decrease when the node density increases. This is a result that can be expected, since a high-density network has many communication links that need to be correctly predicted; therefore, an important amount of control messages must be delivered through the network, and a fewer number of patterns (in percentage) are identifiable by the nodes.

In order to determine the role that the speed of nodes is playing in the results obtained, we established maximum speeds to the nodes. The nodes can randomly assume a certain speed (1 m/s, 2 m/s, 4 m/s or 6 m/s) for short time periods, and then make a new assumption for the next period. The results obtained have shown that the nodes' speed does not affect the prediction upper-bound limit of the HBP approach. Therefore, we can say that this limit always ranges between 50% and 80%. Moreover, the predictability limit decreases when the node density increases, and there is not a significant difference among the mobility models used by the nodes.

2.3.2.2 Frequency of the Observed Control Messages

Concerning the control messages that most frequently appear in the history kept by the nodes, Figure 2.5 shows a curve illustrating the results. The curve considers, from left to right, the most-frequent messages. The X-axis indicates (in percentage) the number of different messages that appear most frequently, with respect to the total number of messages observed (i.e. messages that were recorded in the historical information of the nodes). For instance, in the scenario with 30 nodes, there are a 30% of different control messages that represent 70% of the (total) messages observed. Notice that Y-axis and X-axis are both normalized.

The most important result shown in Figure 2.5 is that there is a small subset of messages that are representative of most messages delivered by the nodes through the network. In other words, the control messages belonging to this small subset, traverse the network many times, therefore they have high representativeness. Although it seems that the combination of multiple nodes will produce a huge number of possibilities, in reality just with a few number of messages we can obtain most of the messages that a network produces. Notice that this result does not depend on the node density in the network.

Although the values shown in Figure 2.5 were obtained considering a SLAW mobility model and a speed of 1 m/s, the simulations performed using the other mobility models have shown the same distribution of results.

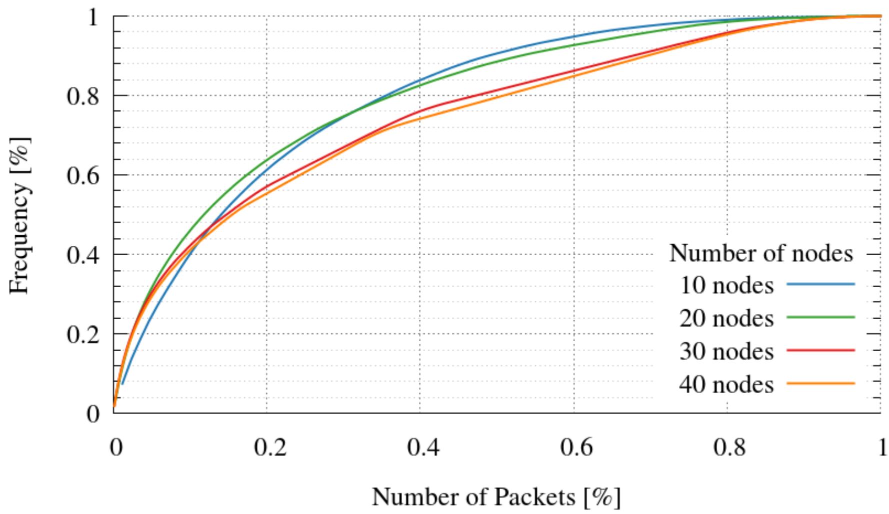


Figure 2.5: Frequency of the observed control messages.

Summarizing the first results we have obtained in previous subsection 2.3.2.1 and in this subsection 2.3.2.2, we have determined the maximum potential of using History-Based Prediction. This first study have shown that about 80% of the time the control messages to be predicted had already appeared in the past, which confirms that the potential of correctly predict TC information is high. Results from this previous study also has shown that the prediction capability of the HBP approach does not depend on the mobility model used by the nodes. Moreover, the predictability is independent of the nodes' speed, but decreases for high node densities. This first study also demonstrated that there is a small subset of TC messages of frequent occurrence in the network, which are representative of the majority. All these results proved that the opportunities for predictions are high in a diversity of dynamic application scenarios.

2.3.2.3 History-Based Prediction

In order to perform a more comprehensive evaluation of the performance of this proposal, we evaluated different versions of the HBP approach. We also identified four typical cases to analyze when making a prediction. In the first case, *nopred*, there is no prediction because there was no an entry in the table that matches the current pattern. This would be the case with probably the lowest occurrence, as it just happens the first time that a pattern appears. The second case, *hit*, means that a prediction is made (i.e., the pattern matches with an entry in the table) and it is correct (i.e., the message associated to that entry is the next expected message). The third case, *missNoPred*, indicates that a prediction is made. Although, it is wrong, it is impossible to make a correct prediction, as the next expected message never appeared in the past with that pattern. Finally, the fourth case, *missPred*, denotes that a prediction is made and it is wrong, but the message could be correctly predicted as the next expected message appeared with that pattern at least once in the past.

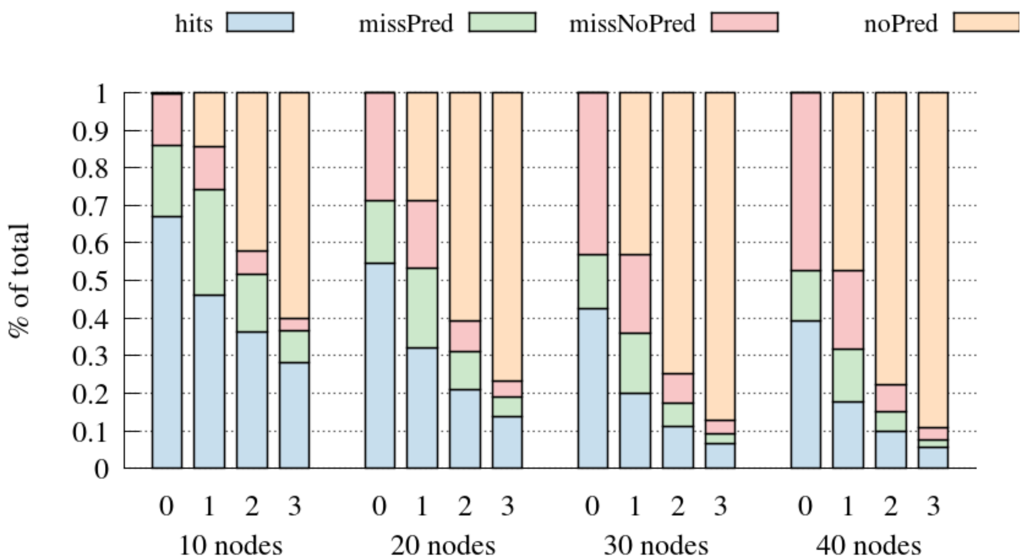


Figure 2.6: History-Based prediction using a SLAW mobility model, last-value policy, and History Depth (HD) in the range 0-3.

Figure 2.6 shows the HBP performance considering these prediction cases in a scenario with a SLAW mobility, and node density ranging from 10 to 40 nodes. The HBP approach assumes the last-value policy as the selected prediction mechanism, and a HD in the range from 0 to 3.

These results indicate that the largest percentages of *hits* are achieved with *HD=0*, but these cases also present important percentages of *misses* (i.e., *missPred* and *missNoPred*). The results also show the effects of the predictability limits, which reduces the number of *hits* and *misses* when the node density increases. The equivalence of *missNoPred* (*HD=0*) and *noPred* (*HD=1*) can be explained due to the fact that, for *HD=1*, the mechanism cannot predict the first time that a message appears; i.e., this is *missNoPred* for *HD=0*.

Figure 2.7 shows the effect of using different prediction policies with a SLAW mobility model, for a network with 10 nodes. For this analysis we considered three different durations of history windows, corresponding to 1, 2, and 3 rounds of TCI delivery (X-axis). These window durations establish the amount of historical information used to identify the patterns.

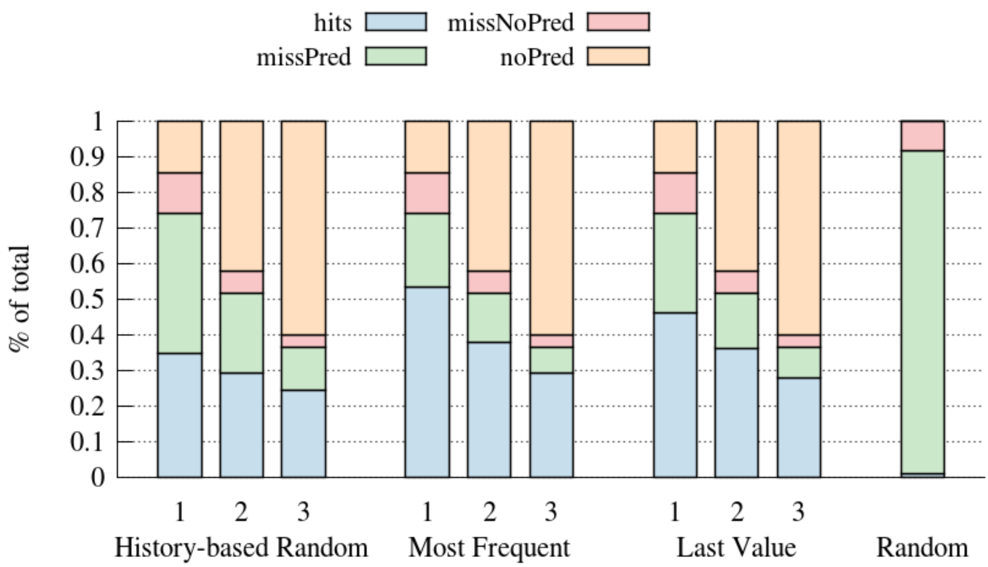


Figure 2.7: History-Based Prediction using different prediction policies.

When using pure Random policy it is always possible to make a prediction (even without history information), but most predictions are *missPred*. The use of historical information clearly allows achieving better results, even using a history-based Random strategy. This can be considered as the baseline, and demonstrates the importance of using the historical information to make more accurate predictions.

Finally, Figure 2.8 shows the effect of using different mobility models with a speed of 1 m/s, for a network with 10 nodes. For this analysis we considered three different durations of history window (1, 2, and 3). We can observe that the behavior of the three mobility models is quite similar (less than a 10% of difference), with a significant increase in *noPred* cases for longer history values. These results confirm that there are no significant differences among the prediction capability in the three mobility models.

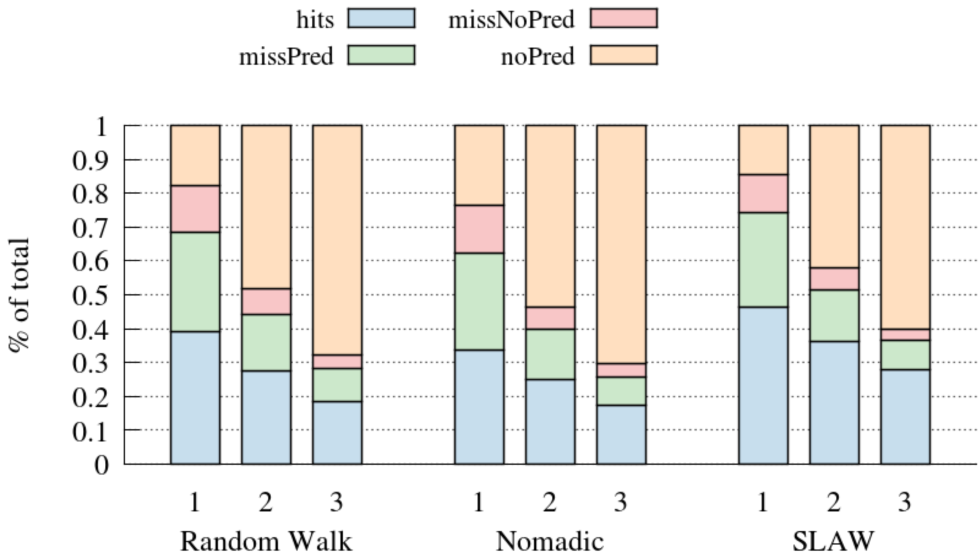


Figure 2.8: History-Based Prediction using different mobility-models.

2.3.2.4 History-Based Prediction Using a Confidence Mechanism

Notice also that in all the analyzed approaches, a prediction is always made (besides the *noPred* case), and this prediction can be a hit or a miss. In the case of a hit, the direct benefit is a reduction of the network traffic and energy consumption, because the producer will not send the control message through the network and the consumer will assume that its predicted TC message (pTCm) is correct. However, every miss prediction has a cost for the nodes, since the producer will detect that the prediction is not correct (because it has the current message and this message does not correspond to the one that returns the predictor). Therefore, this node will send the correct message through the network. Meanwhile, the consumer will assume an incorrect prediction, as long as the correct message does not arrive. This elapsed time would not be long, but it could be long enough to route some data messages, assuming a low accuracy of the network topology map. Therefore, the challenge to address is to always predict in certainty scenarios and not predict in the others.

This strategy can be implemented by including a mechanism that adds two more cases to the previous four. Now a prediction is made if the predictor has enough confidence. Otherwise, the prediction would be a hit (*noConfidence/hit*) or a miss (*noConfidence/miss*). Our aim is to maximize *noConfidence/miss* with a minimum *noConfidence/hit*. Figure 2.9 shows the behavior of this confidence mechanism, considering the six cases (two of them consider no confidence). The results were obtained using a Nomadic mobility model, in a network with 10 nodes, which had a maximum speed of 1 m/s. A last-value strategy was chosen to make the predictions.

The results indicate that there are few predictions when using confidence. However, most of them are hits and there are few misses (the miss ratio is minimized). Most of the predictions were not made because there was not enough confidence value (mainly *noConfidence/miss* with little *noConfidence/hit)*. This indicates that by using a confidence mechanism we can minimize the prediction errors.

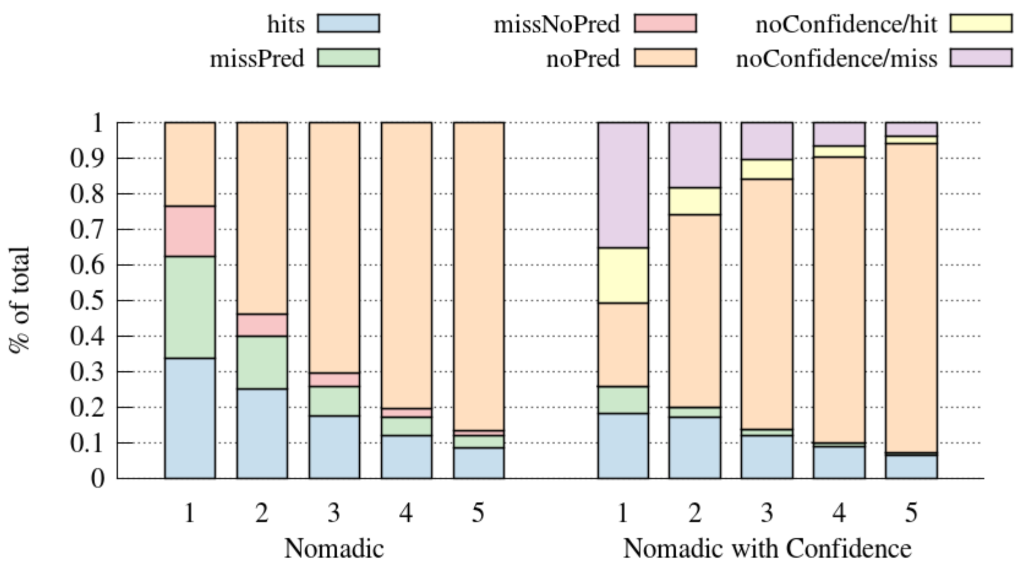


Figure 2.9: History-Based Prediction using a 2-bit confidence-mechanism.

2.3.2.5 Dynamic History-Depth

With the aim of improving the total number of hits, we relaxed the condition of fixed History-Depth patterns. This way we can have more opportunities to correctly predict the next Topology Control Information (TCI) message. When a pattern has no previous history and/or not enough confidence, we decrease in 1 the History Depth and check again if a prediction can be made (with the same selection policy), as a way to minimize the *noPred* cases. We call this method *Dynamic History-Depth* (or *Tree*).

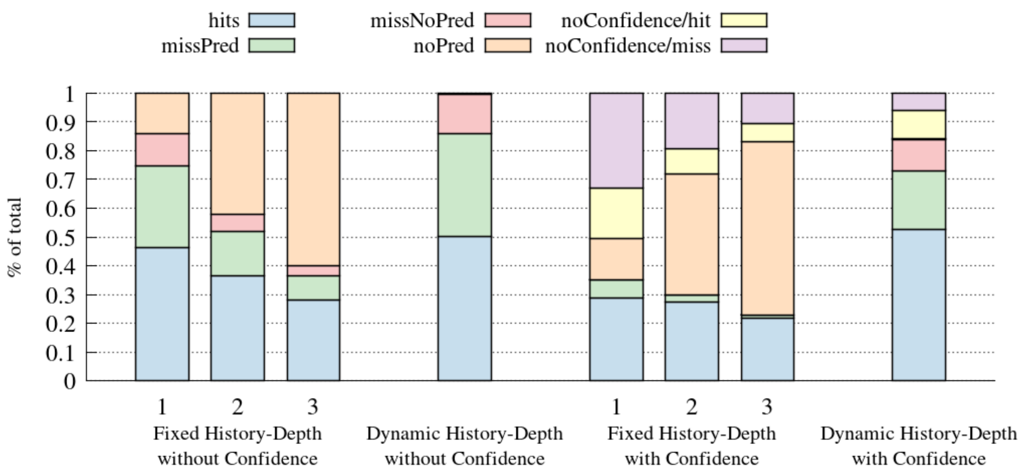


Figure 2.10: Fixed History-Depth versus Dynamic History-Depth (Tree).

Figure 2.10 shows the results of using *Fixed* versus *Dynamic History-Depth* with and without confidence (using SLAW mobility, 10 nodes, 5 as maximum HD, maximum speed of 1 m/s, and last-value policy). We can see that the Tree method minimizes *noPred*. Therefore, the same percentage of hits is maintained, but the overall number of hits is significantly increased. When we include a confidence mechanism (right side of Figure 2.10), Fixed History-Depth shows a decrease in the number of hits and misses. However, Tree with Confidence achieves better results, maximizing hits and minimizing misses.

2.3.2.6 Historical-Data Windows

We have also performed an analysis to determine the effect on the prediction results for different durations of historical-data windows. Historical-data windows represent the amount of historical data used for the predictions. We analyzed historical-data windows of 1, 4, and 24 hours. This means that the historical data used by the predictor is cleared every 1, 4, or 24 hours respectively. The aim of this study is to determine if there is any relationship between the historical-data windows and the prediction results.

Our experiments were performed from 24 hours of network-data traces. We split these 24 hours of historical data in 6 fragments of 4 hours, and also in 24 fragments of 1 hour. Then, we analyzed the results of every individual fragment. We computed the average values of the 6 fragments of 4 hours, and of the 24 fragments of 1 hour. The results presented below correspond to the average values for 1 and 4 hours fragments, and also for the whole 24 hours of historical data.

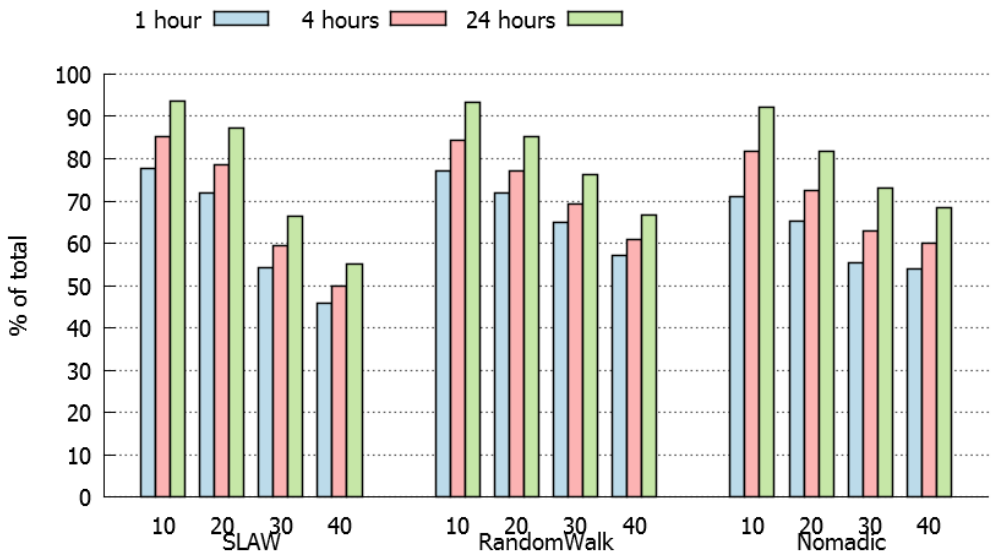


Figure 2.11: Predictability limits (1 hour / 4 hours / 24 hours).

Figure 2.11 presents the predictability limits for 10 to 40 nodes, and for the 3 mobility models considered, but taking into account historical-data windows of 1 h, 4 h, and 24 h. As expected, we can observe that predictability limits increase with the duration of the window (the 24 hours window achieves the highest predictability limits, whereas the 1 hour window presents the lowest values). These results can be explained due to the fact that in a 24 hours time-period, there is a higher opportunity to find TCI messages that have already appeared in the past. Therefore, from the point of view of higher predictability limits, it is preferable to use historical-data windows of 24 hours.

To further investigate the effect of the duration of the historical-data windows in the prediction results, we analyzed the impact of different prediction algorithms. Figure 2.12 shows the results obtained for the "Last-value" and "Most-frequent" prediction algorithms, with and without a confidence mechanism (2-bit confidence and no-confidence). Other parameter settings are: SLAW mobility model, 10 nodes, and dynamic HBP (Tree) with 5 as maximum HD.

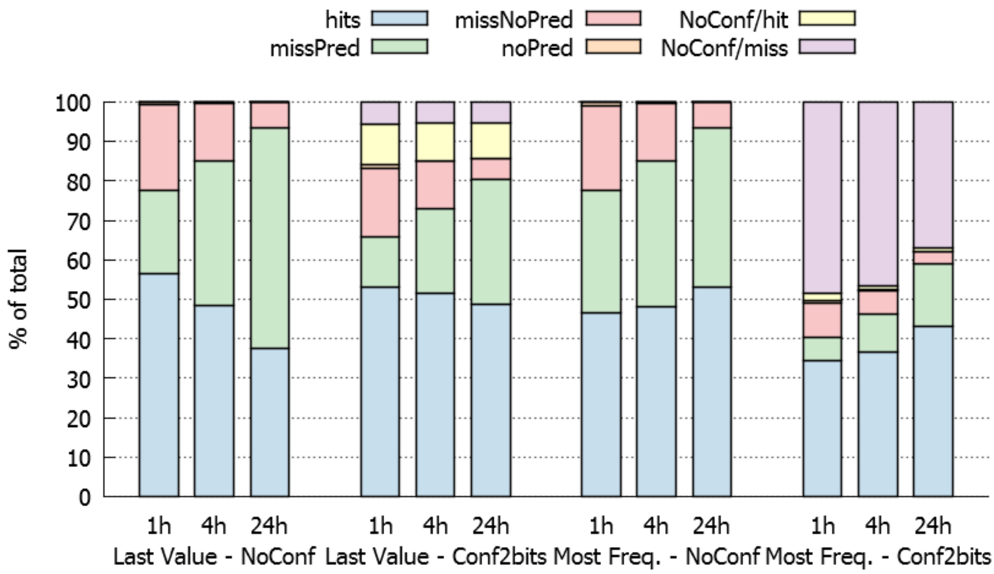


Figure 2.12: History-Based Prediction using a SLAW mobility model with 10 nodes, and last-value and most-frequent policies, with and without confidence (1 hour / 4 hours / 24 hours).

On the one hand, we observe that using the Last-value prediction policy (left half of Figure 2.12), it is better to use short historical-data windows (1 hour), because longer values (4 and 24 hours) decrease the percentage of hits and increase the percentage of total misses. This behavior can be explained because longer windows have more variability and fewer opportunities to repeat last values. On the other hand, the Most-frequent prediction policy (right half of Figure 2.12) takes advantage of longer historical-data windows (24 hours), thus increasing the percentage of hits. The explanation for this is that when having short windows, there is not enough historical data (repeated values appeared in the past) so that the Most-frequent policy could predict accurately. Nevertheless, for 24 hours of historical-data windows, the opportunities to find values that had already appeared in the past are higher. The conclusion from Figure 2.11, a potential analysis on the TC message originator nodes, was that longer windows have higher predictability potential. Nevertheless, Figure 2.12 corresponds to real experimental data using OLSR-HBP, and the highest number of hits is achieved with the shortest window (1 hour) and Last-Value policy. For this reason, we conclude that in a real scenario, shorter windows achieve better results, as depicted in Figure 2.12.

Regarding the confidence method, the aim is to reduce the number of wrong predictions (misses) made by the predictor because it has not enough confidence. When applying confidence (second and fourth graphs in Figure 2.12), we assure that the predictions that are not made due to low confidence are mainly misses. This comes at the cost of a reduction in the percentage of hits. By looking at the purple values in Figure 2.12, we can observe that the number of predictions that are not made (due to lack of confidence and that would therefore be misses) are much higher for the Most-frequent than for the Last-value approach. Notice that for the Last-value policy, it is counterproductive to use confidence, because the majority of predictions that are not made would have been hits (yellow values in Figure 2.12).

Finally, a new research line opens to investigate what is the trade-off between the number of hits and the number of misses. That is, is it better to increase the percentage of hits at the cost of higher number of misses (when not applying confidence)?, or would it be better to decrease the percentage of misses but also the percentage of hits (when applying confidence)? In a real-world scenario, this will depend on the time that benefits from hits, the additional time spent with misses, and the time consumed when no prediction is made because of the lack of confidence.

2.3.3 Applying HBP in Dynamic Scenarios

2.3.3.1 Experimental Framework

We also compared the performance of OLSR-HBP with diverse parameter settings. For each specific set of parameters, we executed 10 test variants, so we can compute the average, standard deviation, minimum, and maximum values. In the next subsections, we present the results obtained. The parameter settings are specified in Table 2.2.

Regarding the confidence, we run some initial tests, first with 0 bits (no confidence) and then with confidence with 2 bits, but we finally discarded the latter case. The reason to discard it was that the small number of errors produced at destination nodes, when no confidence was applied (about 0.5% of the TC messages predicted/generated at destination), were “amplified” by the confidence mechanism. This mechanism produced 40 times more wrong predictions at destination and 42% less TC messages omitted (not sent) at origin.

Parameter	Values used at execution
Mobility	Static Grid 4×4 (baseline), SLAW
Surface/deploy area	120×120 m (Grid 4×4), 300×300 m (SLAW)
Number of nodes	10 (baseline), 20
Execution time	1 hour (baseline), 4 hours
Confidence	0 bits (no confidence)
Prediction algorithm	Last Value
Data traffic	1 KByte periodic pings, from 1st to 2nd half of nodes
Ping periods	no pings, 10 seconds, 1 second, 0.1 seconds

Table 2.2: OLSR-HBP execution parameters.

2.3.3.2 Control Traffic Decrease

The main focus of this chapter is to decrease the number of TC messages sent through the network. As it was explained in subsection 2.2.3.3, we can skip a TC broadcast when all the destination nodes (that would receive this TC message) are able to correctly predict it (*global hit*). If just one of the destination nodes is unable to predict the TC message, it has to be broadcasted. This requirement of a *global hit* will produce worse results than those presented in section 2.3.2, where the hit was considered just at the origin node.

Figure 2.13 plots the number of TC broadcasts for five different cases. The values shown in this figure were normalized, representing the average number of TC broadcasts per node and per hour. The bottom values depicted in blue represent the number of broadcasts that are actually sent through the network. The broadcasts in green at top correspond to global hits and therefore to TC messages that are not transmitted (they can be correctly predicted at the destination nodes). The Grid case (represented on the leftmost side of the figure), shows the best result in terms of percentage of global hits. The reason behind this good result is that this case corresponds to a static mobility model, which implies that after an initial transmission of a few TC messages, the network topology does not change, and therefore the remaining TC messages are all the same (and can be correctly predicted at destination). The other four cases with SLAW mobility model, 10 or 20 nodes, and 1 or 4 hours of historical-window duration, show lower percentages of skipped TC broadcasts (between 40% and 54%). Although the values are normalized, we can observe that the cases with 20 nodes have more broadcasts because both, the possible combinations of nodes, and the number of MPR nodes that broadcast TC messages, are higher.

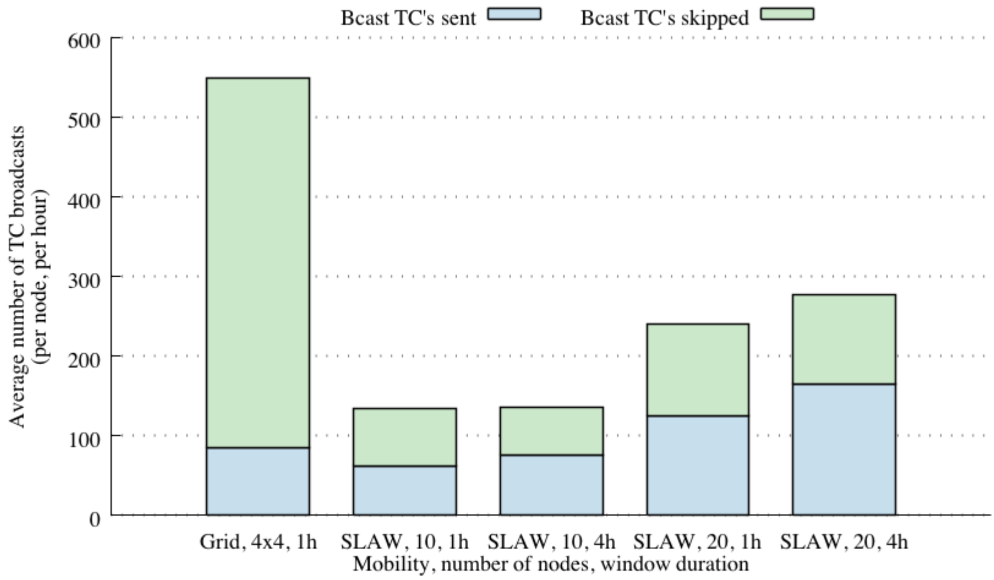


Figure 2.13: Average number of TC broadcasts (per node, per hour).

If we compare the results shown in Figure 2.13 with those of the predictability limits represented in Figure 2.11, the percentages of 70-85% of predictability limits for SLAW, 10, 20 nodes, and 1, 4 hours, decrease now with OLSR-HBP to 40%-54%. This can be explained due to the fact that not all hits at origin nodes (Figure 2.11) are global hits (Figure 2.13). Another difference between these two figures is that in Figure 2.13 the prediction percentages increased with longer historical-data windows, but OLSR-HBP performs better with shorter historical-data windows, as shown in Figure 2.11. This means that for our proposal it is more beneficial to work with shorter historical-data windows (1 hour is better than 4 hours). To apply this, the predictor could delete the historical data every hour.

Figure 2.14 shows the results from the perspective of the receiver of the TC messages. The absolute number of messages at destination is greater than that at origin (Figure 2.13) because every single broadcast message can be received by several nodes (we accumulate every TC message received at each node). The trends observed in Figure 2.13 are repeated now, but with higher percentages of predictable TC messages at destination. In fact, SLAW with 10 nodes and 1 hour historical-data window achieves 85.15% of prediction success, which is slightly worse than the 89.74% of the static case (Grid 16×16). For the other cases with SLAW mobility, the prediction percentage decreases with higher number of nodes and longer historical-data windows, down to 82.83% (10 nodes, 4 hours), 76.20% (20 nodes, 1 hour), and 71.32% (20 nodes, 4 hours).

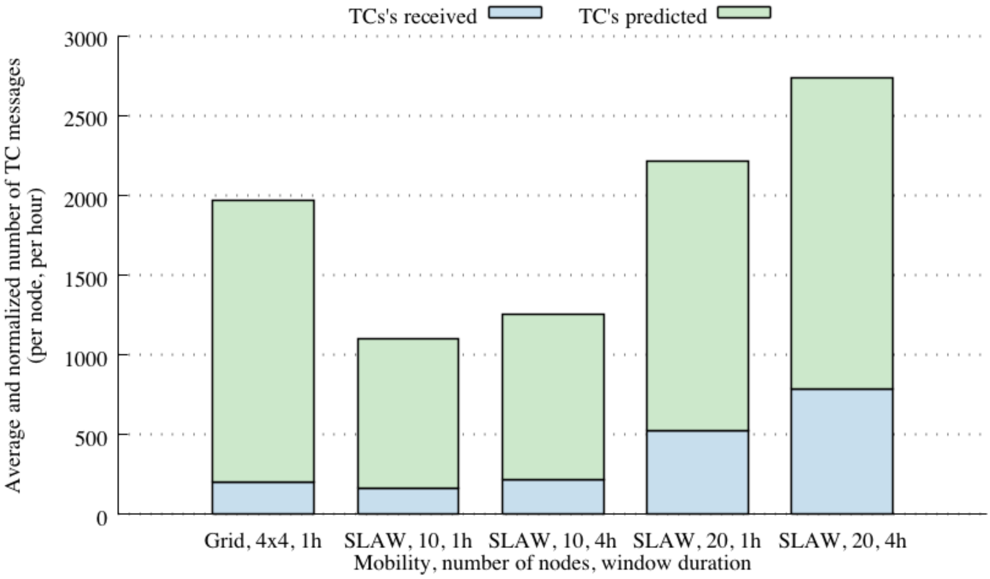


Figure 2.14: Average and normalized number of TC messages received/predicted at destination nodes.

2.3.3.3 Data Transfer (delivery rate) Capabilities

To confirm that OLSR-HBP does not negatively affect the data transmission through the network, we performed some additional tests. To simulate data traffic, we used periodic ping requests with 1 Kbyte payload each. We divided the network nodes into two halfs. The first half sent periodic ping requests to one node in the second half (every pair of nodes transferred pings among them). The ping periods used were 10, 1, and 0.1 seconds (T10, T1, and T.1 respectively). The metrics analyzed were the delivery rate (percentage of ping requests returned to the sender) and Round-Trip Time (RTT) (elapsed time between the instant when the request is sent and when the response is received), which corresponds to the latency. We compared the metric values between the original OLSR (where all TC messages are sent) and OLSR-HBP (where some TC messages were not sent, because they can be predicted at destination).

Figure 2.15 shows a table with the percentages of ping deliveries (number of pings received divided by number of pings sent). We consider the same five

	Grid 16x16 nodes 1 h			SLAW 10 nodes 1 h			SLAW 10 nodes 4 h		
	T10	T1	T.1	T10	T1	T.1	T10	T1	T.1
OLSR	100,0%	73,95%	37,76%	19,38%	16,99%	16,89%	21,00%	18,01%	17,66%
OLSR-HBP	83,6%	60,95%	38,58%	19,52%	16,91%	16,77%	21,02%	17,72%	17,35%
Improvement	-16,4%	-17,6%	2,2%	0,7%	-0,5%	-0,8%	0,1%	-1,6%	-1,8%

	SLAW 20 nodes 1 h			SLAW 20 nodes 4 h		
	T10	T1	T.1	T10	T1	T.1
OLSR	22,14%	16,84%	16,77%	32,54%	21,24%	20,86%
OLSR-HBP	22,02%	16,75%	16,33%	32,06%	20,63%	19,94%
Improvement	-0,5%	-0,5%	-2,6%	-1,5%	-2,9%	-4,4%

Figure 2.15: Ping Delivery Rates (in %).

test cases of subsection 2.3.3.2 for both OLSR and OLSR-HBP. The first row represents the baseline values, and corresponds to the Delivery Percentage of OLSR (DP_{OLSR}). The second row contains the Delivery Percentage ($DP_{OLSR-HBP}$) of the implemented predictor. Finally, the third row computes the improvement achieved. It is computed as $Improvement = (DP_{OLSR-HBP} - DP_{OLSR})/DP_{OLSR}$. As a result, positive improvement values mean that OLSR-HBP achieves higher Delivery Percentage, whereas negative improvement values mean that the predictor performs worse than the original OLSR. On the one hand, we can observe that OLSR-HBP performs similarly than OLSR, with a maximum of -4.4% reduction in the Delivery Rate (for SLAW mobility with the highest number of nodes, longest history-data windows, and highest ping frequency). There are even a couple of cases where the predictor Delivery Rate is up to 0.7% better than OLSR. On the other hand, for the Static/Grid case, the improvement values range between -4.0% and 2.2%, and the best OLSR-HBP behavior is achieved when the ping frequency is maximum. The average improvement value is -1.22% (almost negligible). We can conclude that OLSR-HBP reaches and keeps the Delivery Rates of the baseline OLSR.

The second metric we analyzed was the average Round-Trip Time (RTT) of the pings. Figure 2.16 shows the RTT times (in milliseconds) for the five cases considered, and the percentage of improvement achieved. If we denote RTT_{OLSR} and $RTT_{OLSR-HBP}$ as the RTT values for OLSR and OLSR-HBP, respectively, then we can define the RTT improvement as $RTT_{Improvement} = (RTT_{OLSR} - RTT_{OLSR-HBP})/RTT_{OLSR}$. Due to the fact that the RTT is better for smaller values, $RTT_{Improvement}$ is positive when $RTT_{OLSR-HBP}$ is smaller than RTT_{OLSR}, and negative otherwise. The columns with ping periods of 0.1 s and history-data window durations of 4 hours are marked as "#N/A" because the average RTT

	Grid 16x16 nodes 1 h			SLAW 10 nodes 1 h			SLAW 10 nodes 4 h		
	T10	T1	T.1	T10	T1	T.1	T10	T1	T.1
OLSR	11,55	28,50	47,74	10,15	10,59	6,76	10,39	10,85	#N/A
OLSR-HBP	20,30	33,80	47,97	9,49	10,49	7,39	12,47	12,08	#N/A
Improvement	-75,7%	-18,6%	-0,5%	6,5%	0,9%	-9,4%	-20,0%	-11,4%	#N/A

	SLAW 20 nodes 1 h			SLAW 20 nodes 4 h		
	T10	T1	T.1	T10	T1	T.1
OLSR	15,00	21,23	13,06	14,70	32,29	#N/A
OLSR-HBP	13,54	22,27	13,16	14,02	33,44	#N/A
Improvement	9,7%	-4,9%	-0,8%	4,6%	-3,6%	#N/A

Figure 2.16: Ping Round-Trip Times (RTT, in ms).

computation uses 16-bit sequences (icmp_seq) and at a rate of 0.1 seconds for 4 hours, the number of pings overflows the 16-bit capacity. This means that the RTT computation is wrong. For other cases where the RTT can be computed, we can observe better values (less RTT) in 4 of 13 cases. Nevertheless, there are 9 cases where the RTT is worse for OLSR-HBP. The RTT differences in percentage ranges from 9.7% (good) to -75.7% (bad), with a global average of -9.46%, which is reduced to just -2.8% for SLAW mobility cases, where the worse value is -20.0%. It was not possible for us to determine the exact reasons of these variations, because they are probably caused by a combination of diverse factors.

On one hand, the decrease in the number of TC messages in the network releases communication resources that can be used to send data with less delay, and therefore with a good RTT. The loss of some TC messages or wrong/miss predictions at destination nodes, may produce alterations in the routing information and configuration. Consequently, ping messages can be routed to a path with more hops (and therefore longer RTT values).

2.3.3.4 Energy Consumption

Another potential benefit of the decrease in the number of TC messages sent, is the energy savings. To compute the reduction in the amount of energy consumed for transmissions and receptions, we used the NS-3 energy model WiFiRadioEnergy [71]. In this model, we configured the Transmitter and Receiver electric current to 0.38 and 0.313 Amperes respectively [95]. The rest of the parameters were set to 0 Amperes. This way we focus the energy analysis on the power needed to just send and receive data wirelessly.

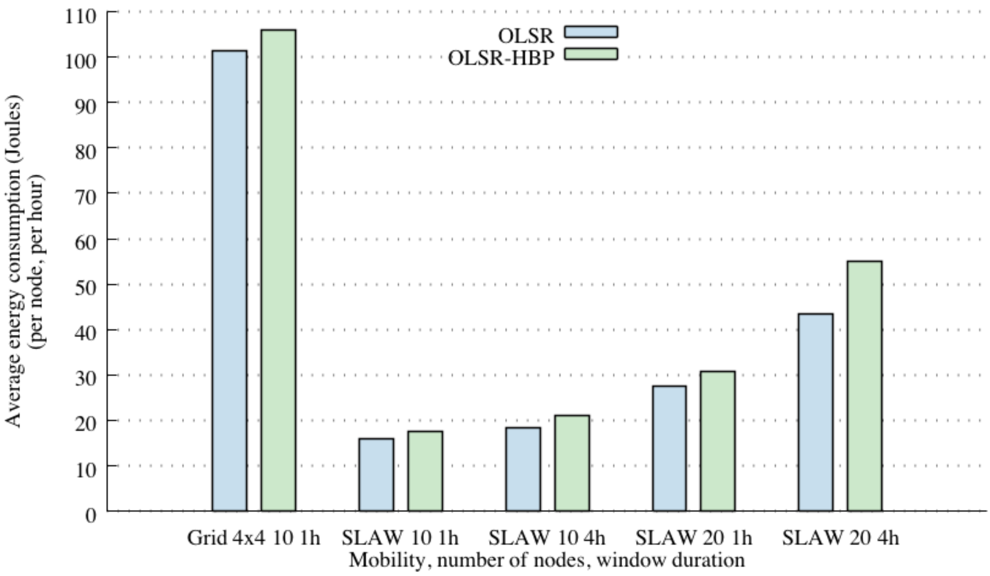


Figure 2.17: Average energy consumption (in Joules) per node and per hour.

Figure 2.17 shows the average normalized energy-consumption for the five test cases. The values correspond to the average energy consumption (in Joules) per node and per hour, for the four periods of data specified in Table 2.2. We performed 10 different executions for each of the 4 ping periods (40 executions in total for every energy value in Figure 2.17). The difference among the 10 executions was the initial location of the nodes and, therefore, also the mobility paths, due to the impact of initial location in the SLAW mobility model.

It can be observed that there is an increase in energy consumption of OLSR-HBP with respect to OLSR. These increases are in the range 4.34% to 27.3% with an average of 13.45%. The 4×4 Grid case shows the lowest increase of 4.34%, whilst for SLAW cases, the values increased with higher number of nodes and longer window durations. In light of these results, we must identify the cause of these increases, because we would expect that less TC messages sent should mean less energy consumption. We observed that the predictor generated more network packet transmissions (with less messages per packet). The energy used for network communications depends more on the number of packets sent, than on the size of each individual packet. In the original OLSR, TC messages and HELLO messages are usually transmitted (and forwarded) in the same packet. However, OLSR-HBP should wait an additional time to receive the next TC message. If not received within this extra period of time, the receiver node should predict the next TC message and forward it. This forwarding is performed when there are no additional HELLO messages to include in the same packet, and therefore more packets (with less messages per packet) are transmitted. This transmission of a higher number of packets results in an increment in the energy consumption.

2.3.3.5 Memory Overhead

The last analysis performed was to compute the amount of memory occupation. Our aim is to compute the amount of memory used by each node to store all the historical data needed by OLSR-HBP. It uses basically two data structures: a vector to store the identifiers (ids) of the TC messages, and a table to store the HBP information (such as in Figure 2.3). Each TC message contains an IPv4 address (4 bytes) to identify the originator node, and a list of the IPv4 addresses of its neighbor nodes. The greatest number of neighbors contained in a TC message was 8 (for 10 nodes cases), and 14 (for 20 nodes cases). The id assigned to each TC message requires 2 bytes. The table that contains the HBP information has entries with data for the pattern (a sequence of up to 5 ids), and the id of the most recent TC message received for this pattern. To compute the amount of memory used by OLSR-HBP, we did not consider the confidence bits, nor the counters used by the Most-frequent algorithm.

Figure 2.18 plots the average amount of memory (in KBytes) used per node and per hour, considering the five test cases. Each average value is computed from the same 40 executions than in the previous section. The Grid 4×4 is the case with less memory requirements, because the number of different TC messages is very small. The memory used by OLSR-HBP in this last case is just about 128 KBytes (KB) per node.

The SLAW mobility produces more variety on the TC messages and therefore higher memory requirements. SLAW with 10 nodes needs 190 KB (for the 1 hour window) and 199 KB per hour (for the 4 hours window, per hour; meaning a total of 796 KB for the whole 4 hours window). When the historical-window lasts 4 hours, there is a 4.8% increment in the memory requirements with respect to the case when the historical-window lasts 1 hour. This slight increase may be caused by a greater variety of TC messages present when the historical-window lasts longer.

In the cases of SLAW with 20 nodes, the average memory occupation increases to 652 KB (1 hour window), and 930 KB per hour (4 hours window; 3,720 KB in total, a little less than 4 MBytes). These values represent an increase of 3.4 and 4.7 times

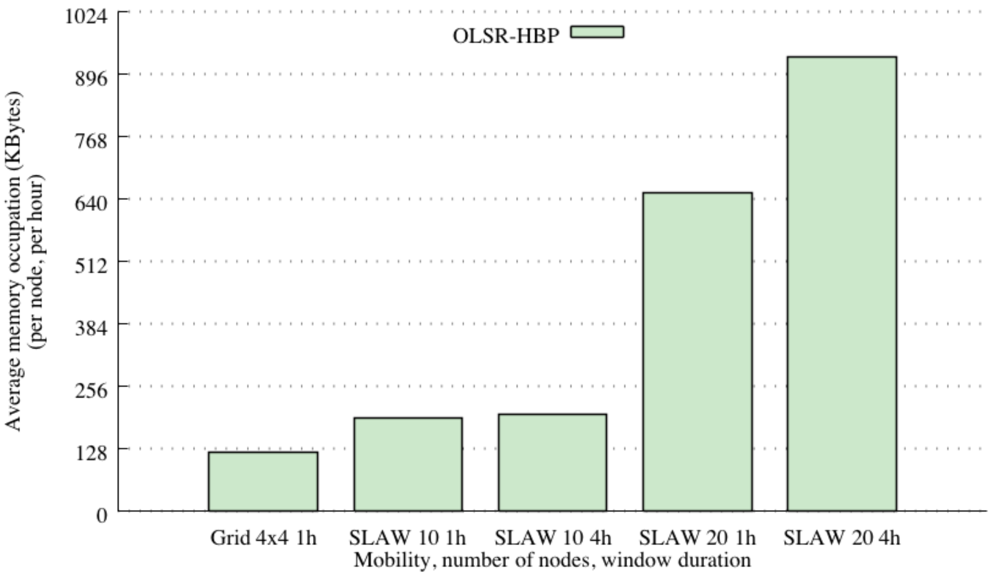


Figure 2.18: Average memory occupation (in KBytes) per node and per hour.

(for 1 hour and 4 hours historical-windows, respectively) the values of the 10-node cases. This means that the memory occupation has a quadratic relationship with the number of nodes (when the number of nodes doubles, the memory occupation quadruples). The increase in the average memory used between 1 and 4 hours is 24%. This value is larger than the previous 4.8% obtained for the 10-node cases. However, this increment is consistent with the quadratic increase observed in the number of nodes dimension.

In summary, considering the worst case (20 nodes and 4 hours), the amount of memory needed is 3.63 MBytes per node. This value is affordable for the memory capacities we can currently found on standard communication devices. Moreover, if the historical information is cleared every hour (as it is recommendable taking into account the results of subsection 2.3.3.2), the average memory requirements per node are reduced to just 420 KBytes. The memory requirements of OLSR-HBP are much lower than those required by Deep Learning routing methods and its associated data structures [4, 25, 27, 43].

2.4 Conclusions

This chapter evaluated the performance of a History-Based Predictor (HBP) strategy, used to predict the Topology Control (TC) messages generated by routing protocols in mobile ad hoc and opportunistic networks. The prediction strategy allows decreasing the number of TC messages exchanged through the network, and consequently reducing the traffic and energy consumption at the participant nodes. The efficient use of the resources of networks and devices is particularly relevant in mobile collaboration scenarios, where many people use devices that depend on their battery to participate; e.g., in crowdsourcing activities, social exergames, and emergency relief efforts.

We started our study determining the prediction performance of our proposal, just considering the potential prediction capability in simple network scenarios. Then we extended this first work determining both, the potential and the effective prediction capability of our proposal. This second part of the study involves real world scenarios.

The results of our research show that the potential prediction capability of the HBP strategy, for low node-density mobile-networks, is around 80% of the times when a TC message has already appeared in the past. This percentage falls to 50% when considering a mobile network with a higher node-density. This analysis also proved that just few messages contribute significantly to the total percentage of messages delivered through the network. This suggests that there is a good opportunity for predicting TC information, and that this prediction can just be focused on a small subset of messages.

In order to experimentally evaluate the proposal, i.e., to determine the effective prediction capability of the HBP strategy, a real predictor was implemented and added to the OLSR routing protocol. This predictor, named OLSR-HBP, was evaluated under realistic network conditions. The experimental results showed a percentage of control messages predicted that ranges between 86% and 92% for static scenarios, and between 40% and 55% for mobile scenarios. The values of effective prediction for mobile scenarios are lower than the highest values in the potential prediction (up to 80%). This can be explained by the fact that OLSR-HBP can only skip a control message transmission when all the destination nodes are able to make a correct prediction (global hit).

The network performance in terms of throughput and latency was similar to the original OLSR protocol, with -1.2% average throughput variations and -2.8%

average latency variations. This means that the execution of the predictor did not affect the network operation.

The resources used by OLSR-HBP, in terms of energy and memory, were also evaluated. Interesting results were obtained regarding the memory used to store the TC information in each node. In the worst-case scenario, there was a memory utilization of about 3.6 MBytes, with a typical case of just 420 KBytes. This leads to the conclusion that the memory requirements of OLSR-HBP can be considered minimal, by taking into consideration the amount of memory resources available on nowadays mobile devices. These low memory requirements confirm the suitability of OLSR-HBP to be implemented in mobile devices that act as communication nodes in mobile collaborative applications. By contrast, other routing proposals based on Deep Learning, require off-node computations, with higher computing resources and additional communication overheads.

In summary, OLSR-HBP achieves important decreases of signaling overhead, without disturbing the network operation, and requiring a small and affordable amount of resources. Our approach is deterministic and with much lower resource requirements, in comparison with other statistical proposals (such as Machine or Deep Learning).

Chapter 3

Link Quality Prediction

In order to answer the *Specific Research Question 2 (SRQ2): What impact does prediction have on the quality of ad hoc networks?*, in this chapter we focus on Link Quality (LQ) prediction by means of a Time-series analysis. We apply this prediction technique in the routing layer of large-scale, distributed, and decentralized networks. Particularly, we analyze the behaviour of the links globally to identify the best prediction algorithm and metric, the impact of lag windows in the results, the prediction accuracy some time steps ahead into the future, the degradation of prediction over time, and the correlation of prediction with topological features. Moreover, we also analyze the behaviour of links individually to identify the variability of LQ prediction between links, and the variability of LQ prediction over time. Finally, we also present an optimized prediction method that considers the knowledge about the expected LQ values.

3.1 Introduction

Community Networks (CNs) are large-scale, distributed, and decentralized networking infrastructures with several nodes, links, and services. This kind of networks are extremely diverse and dynamic, because of their decentralized nodes, their mix of wired and wireless technologies, their several routing schemes, and their diverse services and applications [6]. Those networks are made available to a community of people living in the same area. The network management is based on an open peering agreement, which avoids barriers for the network participation. Ownership, governance, and knowledge are also open (community members own and manage these networks). Community Networks (i.e., FunkFeuer [30] and guifi.net [9]) grow dynamically with regards to the number of links.

Community Networks features (large, heterogeneous, dynamic, decentralized) raise challenges of interest for researchers [12]. One of the most important challenge is the effect of the asymmetrical features and unreliability of wireless communications on network performance and routing protocols. Many metric-based routing protocols for mesh networks that track Link Quality (LQ) and select higher-quality links have been proposed to minimize traffic congestion and maximize delivery rate [26, 45, 75, 96]. Hence, when routing packets through an unreliable network, LQ tracking is definitely a key method to apply. Moreover, routing algorithms should avoid weak links as soon as possible [79], and whenever possible [93]. LQ estimation [8] (or prediction [51]) is an approach that increases the improvements in routing performance achieved by LQ tracking. Usually, real-time metrics do not provide enough information to detect the degradation or activation of a link at the right moment. Therefore, prediction techniques are needed to foresee LQ changes and take the appropriate measures.

LQ tracking has been previously applied in several scenarios in different ways [26, 45, 75, 93, 96] to select higher-quality links that maximize delivery rate and minimize traffic congestion. LQ prediction/estimation is used in addition to LQ tracking to determine beforehand which links are more likely to change their behaviour. As a result, the routing layer can make better decisions at the appropriate moment. Link Quality Estimators (LQE) [8, 51] are in charge of measuring the quality of the links between nodes, based on physical or logical metrics. Physical metrics focus on the received signal quality and logical metrics focus on the percentage of lost packets. LQE with metrics like Link Quality Indication (LQI) [28], Signal-to-Noise Ratio (SNR) [53], or Received Signal Strength Indication (RSSI) [83] fit in the former category, whereas metrics like Required Number of Packets (RNP) [15], Expected Transmission Count (ETX) [23, 69], or Packet Success Rate (PSR) [96] fit in the latter. It is also important to notice, as it is shown in [8], that a slight variation in SNR may affect the LQ, changing it from good to bad in a bursty way. All these metrics can be used by LQE in isolation or as a combination of some of them [8, 51, 76] to select the more suitable neighbour nodes when making routing decisions.

The Four-Bit Wireless Link Estimation method [28] collects information of several layers (physical, link, and network) to make LQ predictions that can be decoupled from particular layer implementations while remaining efficient and accurate. A Kalman Filter approach [83] estimates the Received Signal Strength (RSS), then takes this value to evaluate the SNR, and finally approximates the Packet Success

Rate (PSR) in a way that can adapt fast enough to the temporal dynamics of the links. There are some other studies [53] that conclude that by including SNR information, wireless LQ prediction is significantly improved. The Holistic Packet Statistics (HoPS) [76] approach applies Exponentially Weighted Moving Average (EWMA) filters to calculate several metrics that help evaluate the short-term and long-term LQ, its variation, and evolution over time. On the other hand, the Window Mean with Exponentially Weighted Moving Average (WMEWMA) estimator [96] predicts LQ of sensor networks based on the Packet Success Rate (PSR) in a simple, memory efficient, and fast reacting way.

The analysis of the most important network properties for the design of efficient routing protocols leads to the conclusion that the Required Number of Packets (RNP) that must be sent before a packet is received, is a precise LQ estimator [15]. Based on this metric, distributed and centralized routing algorithms are proposed to improve the efficiency of networks with low power links. ETX has also been proved to be a good metric for LQ estimation [23], particularly with larger networks with longer paths. For instance, when the Destination Sequenced Distance Vector (DSDV) and Dynamic Source Routing (DSR) routing algorithms are adapted to include the ETX metric, the overall performance of the network is significantly improved. The ETX_ANT algorithm [69] is a simple approach that predicts ETX values a few seconds in advance. This estimation assumes a linear regression method, that is applied to a variable number of last ETX measurements.

Special attention must be paid to the MetricMap mechanism [94]. MetricMap is fundamentally a routing protocol for Wireless Sensor Network (WSN) that uses a learning-enabled method for LQ assessment. Based on the observation that high traffic rates make tracking Link Qualities more difficult, this protocol uses prediction methods to estimate them in advance. In a first stage, a machine-learning algorithm is applied to classify Link Qualities. Two types of classifiers are evaluated: a decision tree and a rule-based classifier. The data used to train both classifiers was preclassified offline, based on a LQ indicator and other metrics that represent some features of the nodes. In a second stage, the MetricMap routing protocol estimates the LQ at runtime by replacing the current traffic information with the rules collected offline from the classifiers. Results show that MetricMap can achieve a significant improvement on the data delivery rate in high traffic rate applications. MetricMap is the most similar work to the one presented in this chapter, as it uses Time-series analysis to improve the routing protocol; however, there are some significant differences:

- They evaluate a small WSN, whereas we evaluate a large Wireless Mesh Community Network (WMCN).

- They give only a hint of the potential of Time-series analysis to predict LQ. In contrast, we perform a detailed and deep analysis of this potential.
- They apply a Time-series analysis to predict current LQ values, while we use a Time-series to predict future LQ values.
- They use a cross-validation method, which uses a subset of the sample data to validate the LQ estimation. We, on the other hand, use new data to validate the LQ estimation.

In this chapter, we present a LQ analysis and prediction of the FunkFeuer WMCN [30]. To the best of our knowledge, no previous work explores LQ prediction in the routing layer of large scale, distributed, and decentralized systems. The main contributions of this chapter are the following:

- The use of Time-series analysis to estimate LQ in the routing layer for real-world WMCNs.
- Clear evidence that LQ values computed through Times-series algorithms can make accurate predictions in WMCNs.
- A detailed global analysis of links to identify the best prediction algorithm and metric, the impact of lag windows on it, its accuracy some time steps ahead into the future, its degradation over time, and its correlation with topological features, showing the potential of Time series to estimate LQ.
- A detailed analysis of individual links to identify the variability of LQ prediction between links and over time, showing the potential of Time series to estimate LQ.
- A hybrid prediction approach that combines a number of prediction algorithms along with a value saturating technique.

The rest of this chapter is structured as follows. Section 3.2 presents the experimental methodology used in this chapter. Section 3.3 corresponds to the analysis of results, starting with the global results presented in subsection 3.3.1, then subsection 3.3.2 describes the individual link and nodes results, and we close with subsection 3.3.3 where we propose an enhanced predictor. Finally, in Section 3.4 we provide some concluding remarks.

3.2 Experimental Methodology

In this section we apply the methodology presented in section 1.3. Subsection 3.2.1 explains the data collection process. In subsection 3.2.2 we define the research objectives. Then, in subsection 3.2.3 we explain the design and implementation step of the methodology. Finally, subsection 3.2.4 presents the experiments and simulations performed.

3.2.1 Data Collection: FunkFeuer Network and Open Data Set

FunkFeuer [30] is a free experimental WMCN deployed in several locations in Austria. This network is a non-commercial project maintained by computer enthusiasts that install WiFi antennas across rooftops. Currently, there are around 2000 wired and wireless links, and every week new nodes are added to the network. It uses the OLSR-NG routing protocol, which expands the capabilities of the OLSR protocol, and makes it highly scalable. In fact, some members of the FunkFeuer network are actively involved in the olsr.org open source project as developers, testing the protocol in the network.

We used open data-sets from the FunkFeuer network, available through the Community Networks Testbed for the Future Internet (CONFINE) Project open-data platform [21]. The data set is composed of OLSR information such as routing tables and network topology data of 404 nodes with 2095 links, collected during 7 days in the period from April 28^{th} to May 4^{th}, 2014. The data-set instances were sampled every 5 minutes. It is important to notice that the total number of nodes is large. Every node has about 3.5 neighbours on average (degree), and that the largest of the shortest paths in the network (diameter) is 16. This means that there are several paths where packets have to go through a relatively high number of hops in order to reach their destination. The routing protocol must, therefore, react quickly to any change in the network topology since this will be critical to achieve high performance.

3.2.2 Research Objectives: Link Quality

The research objective of this chapter is to answer *SRQ2: What impact does prediction have on the quality of ad hoc networks?*, with regard to Link Quality (LQ). For this reason, we focus on LQ prediction for the FunkFeuer network by means of a Time-series analysis. A recent study [54] has analyzed the FunkFeuer network focusing on link layer properties, topological patterns, and routing performance.

ETX [23] is a link metric that measures the expected number of data transmissions required to send a packet over that link, and is widely used in several mesh network protocols. The ETX of a particular link is computed as: $ETX = 1/(LQ \times NLQ)$, where LQ and NLQ stand for the "Link Quality" and the "Neighbour Link Quality" of that link, respectively. The Optimized Link State Routing (OLSR) protocol uses the ETX to choose, for each device and packet, the next hop. The LQ assumed by OLSR is defined as the fraction of successful packets (HELLO) that were received by a node from a given neighbour within a certain time window, while the Neighbour Link Quality (NLQ) is the fraction of successful packets that were received by the neighbour within a time period. We focus on predicting the LQ, as the NLQ is directly derived from the neighbour-node LQ, and the ETX can be easily computed using both predictions.

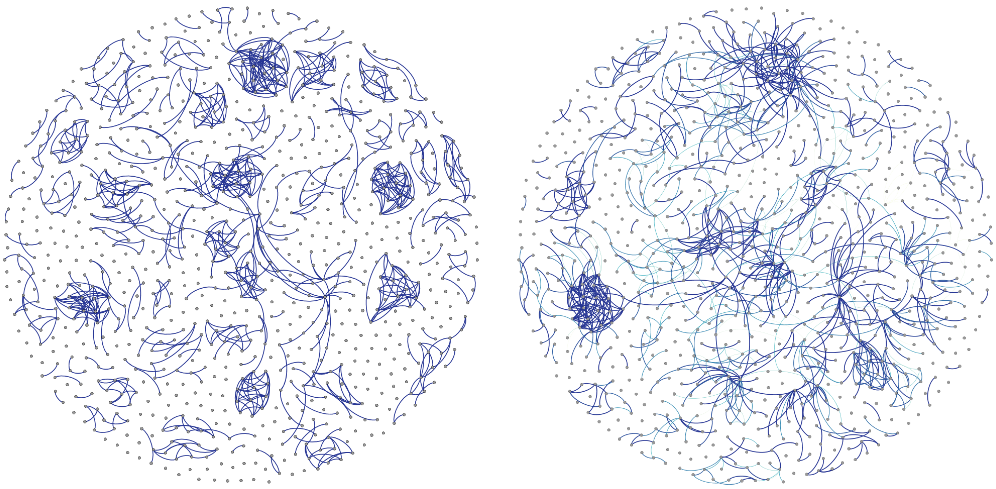


Figure 3.1: (left) Links with maximum Link Quality (LQ=1), and (right) Links with LQ<1.

Figure 3.1 illustrates the nodes and links from the open data-set scenario we used. On the left side, all the nodes are represented, but only links whose quality is always equal to one (maximum quality) are drawn. On the right side of Figure 3.1, all nodes and also those links that have experienced some variations in the Link Quality (LQ<1) at any instant over time are represented. For our simulations we only considered the latter links, which represent approximately half of the overall number of links (1068 of 2095). Moreover, we also discarded those links that did not have enough samples to perform the Time-series analysis. Therefore, our study only considered 1032 links. Notice that the prediction results will only be given for those links that present variations in the LQ (LQ<1). If we considered all links of nodes in the network, we would have obtained higher prediction accuracy; however, predicting the behaviour of links with unaltered LQ is trivial. For that reason, we have excluded these links from our study.

3.2.3 Design and Implementation: Time-Series Analysis

A Time series is a set of data collected over time with a natural temporal ordering. It differs from typical Data Mining or Machine Learning applications where the ordering of data points within a data set is not important. Time-series analysis is the process of using statistical techniques to model and explain a time-dependent series of data points. Similarly, Time-series forecasting is a method that uses a model to generate predictions (forecasts) of future events based on known past events. In our case, we used more than one prediction algorithm, so that we do not rely on a specific learning technique. We applied four of the best well-known approaches [3]: Support Vector Machine (SVM), k-Nearest Neighbours (kNN), Regression Tree (RT), and Gaussian Process for Regression (GPR).

- The Support Vector Machine (SVM) algorithm has recently become one of the most popular and widely used method in Machine Learning. It performs a linear or nonlinear division of the input space, and builds a prediction model that assigns target values into one or another category.

- The k-Nearest Neighbours (kNN) algorithm is one of the most simple Machine Learning algorithms, as it makes no assumptions on the underlying data distribution. This algorithm takes the k data-points closest to the target value, and picks the most common one.

- Regression Tree (RT) is a type of decision-tree algorithm where the target value can have continuous values. This method recursively partitions the data space, and runs a simple prediction model within each partition. The main advantages of these tree algorithms are that they produce fast results, and that they are resistant to irrelevant values.

- Finally, the Gaussian Process for Regression (GPR) algorithm is a very flexible approach that can easily deal with complex data-sets. In this model the output is a normal distribution, denoted by the mean and variance terms. The target value is represented by this mean value, and the variance can be interpreted as a measure of its confidence.

3.2.4 Experiments and Simulations: Training / Test, Metrics, Plots, and Topological Features

We applied training and test-sets validation to evaluate the predictive accuracy of the models. After a model is processed using the training set, it is tested by making predictions against the test set. For this purpose, we used the Weka workbench system [36], a framework that incorporates a variety of learning algorithms and some tools for the evaluation and comparison of the results. Weka has a dedicated environment for Time-series analysis, that allows forecasting models to be developed and evaluated. The Weka's Time-series framework takes a Machine Learning or Data Mining approach to model Time-series by transforming the data into a form that can be processed by standard propositional learning-algorithms. To do so, it removes the temporal ordering of individual inputs, by encoding the time dependency via additional input fields. These fields are sometimes referred to as "lagged" variables.

Usually, classification studies assess the predictive power of their model by using Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE), both widely used in related work. MAE is a common method to evaluate the performance of prediction approaches, that gives the same weight to all individual differences. This metric is calculated through the following formula: $MAE = sum(abs(predicted - actual))/N$. RMSE is another frequently used metric, but gives a relatively high weight to large errors. In this case, the formula is the following: $RMSE = sqrt(sum((predicted - actual)^2)/N)$.

Boxplots are classic representations of a statistical distribution of values. A box is drawn around the region between the first and third quartile, and a horizontal line at the median value. Whiskers extend from the box to the lowest and highest value within 1.5 times the interquartile range of the lower and upper quartile, respectively. Data points that lie outside these limits are independently drawn.

The analysis of topological features of a network (degree, strength, distance, betweenness, etc.) may help to understand its behaviour, and therefore, to improve the efficiency of transferring information through nodes and links. In this chapter, we focus on node degree and edge betweenness (in terms of hopcount) to correlate them with the LQ. The former defines the number of links/edges connected to a particular node. The latter defines the number of the shortest paths that go through one particular link or edge.

3.3 Analysis of Results

3.3.1 Global Analysis of Links and Nodes

The first step of our analysis considers all links and nodes globally. This will allow us to get an overview of the behaviour of the nodes with LQ variations. In subsequent sections, we will analyze them individually, in order to identify different link classes according to their individual behaviour.

3.3.1.1 Comparison of learning algorithms based on Time series

The main aim of this chapter is to explore whether Time-series analysis and prediction can be used to predict the next LQ value. As stated before, we applied four well-known approaches: Support Vector Machine (SVM), k-Nearest Neighbours (kNN), Regression Tree (RT), and Gaussian Process for Regression (GPR).

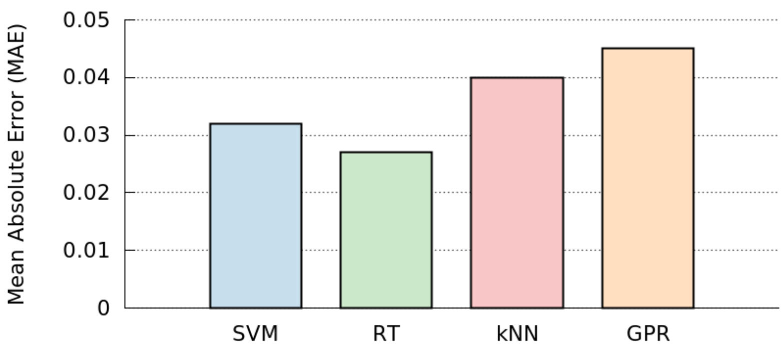


Figure 3.2: Average Mean Absolute Error (MAE) of the links.

Figure 3.2 shows the average Mean Absolute Error (MAE) per link using a training data-set of 1728 instances (6 days, 1 instance every 5 minutes), a test data-set of 288 instances (1 day), and a lag window composed of the last 12 instances. This test was performed to verify whether Time-series learning-algorithms could predict consecutive LQ values. These results show that we achieved the best accuracy for the Regression Tree (RT), and the worst for Gaussian Process for Regression (GPR). Notice that the maximum LQ value is 1 and, therefore, the MAE per link is 2.7% for RT and 4.5% for GPR. We applied a T-test to mean values for independent samples (at 95% confidence level) in order to compare the classification algorithms using the MAE. After this analysis, p-values smaller than 0.05 indicate that the means are significantly different, and therefore, we would reject the null hypothesis of no difference between the means. Consequently, we can claim that RT is a good candidate to make predictions.

We analysed the error variability of each algorithm and represented the results using boxplots. The four algorithms achieved a similar performance for most of the links, as shown in Figure 3.3 (both MAE and RMSE). Although the median, first quartile, and third quartile values are similar for all of them, there are some outliers with large errors. These outliers increase the average values and change the overall evaluation of the algorithms. Comparing MAE and RMSE results, we can observe that the behaviour for the four algorithms is similar using MAE and RMSE, although RMSE values are higher. For brevity reasons, we select MAE as the comparative metric, unless in any particular study the RMSE results were very different.

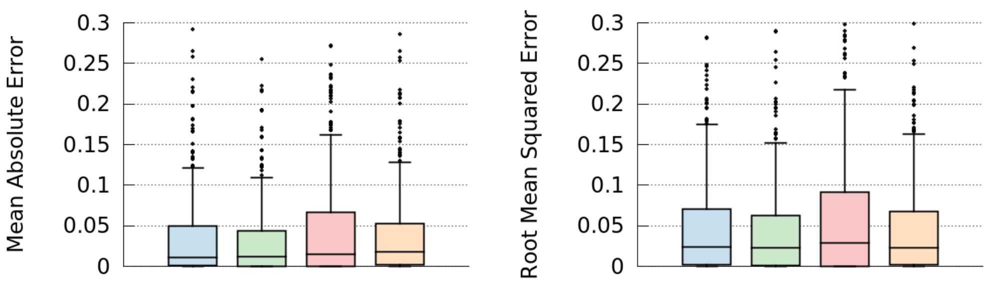


Figure 3.3: (left) Mean Absolute Error (MAE), and (right) Root Mean Squared Error (RMSE) of the LQ predictions, both as boxplots.

3.3.1.2 Analysis of the Impact of the Lag Window Size

Lagged variables are the main mechanism by which we can capture relationships between past and current values of a series using propositional learning-algorithms. They create a “window” or “snapshot” over a time period. Basically, the number of lagged variables determines the window size (i.e. the number of previous samples used to make a new prediction, with samples every 5 minutes).

This analysis was performed to check the impact of the lag window in the prediction of the next LQ value. Figure 3.4 shows the average MAE per link of the RT algorithm using the same experimental setup as in the previous test (1728 and 288 instances for training and testing, respectively) but in this experiment we used a

Figure 3.4: RT average Mean Absolute Error of the LQ predictions.

lag window size ranging from 1 to 24 instances. We obtained good results using window sizes between 3 and 18 (Figure 3.4). The worst results were obtained for window sizes of 1 and 24. Nevertheless, these results are similar or even better than the results obtained by the other algorithms. Thus, we can sustain the claim that RT is the best candidate.

We also analyzed the error variability for each window size, and concluded that all window sizes achieved a similar performance for most of the links. Although the values for the median and the first quartile are similar for all window sizes, the values of the third quartile and the outliers slightly differ. These differences in the variability of errors lead to the differences in the average MAE. Finally, we tried to find the best lag window size for mean values using the T-test for independent samples (at 95% confidence level). After this analysis, we could not reject the null hypothesis at 95% of significance. Consequently, our results do not provide clear evidence about what is the best window size.

3.3.1.3 Prediction of Some Steps Ahead

This analysis was performed to explore if Time-series analysis and prediction can be used to predict the value of LQ some time steps ahead into the future.

Figure 3.5 shows the average MAE of links. It shows the results of the RT algorithm using the same setup that the baseline experiment (a lag window size of 12 instances, a training data-set of 1728 instances and a test data-set of 288 instances) and then predicting from 1 to 8 time steps into the future. The results obtained were good for all the tests. As we can observe, the average MAE grows

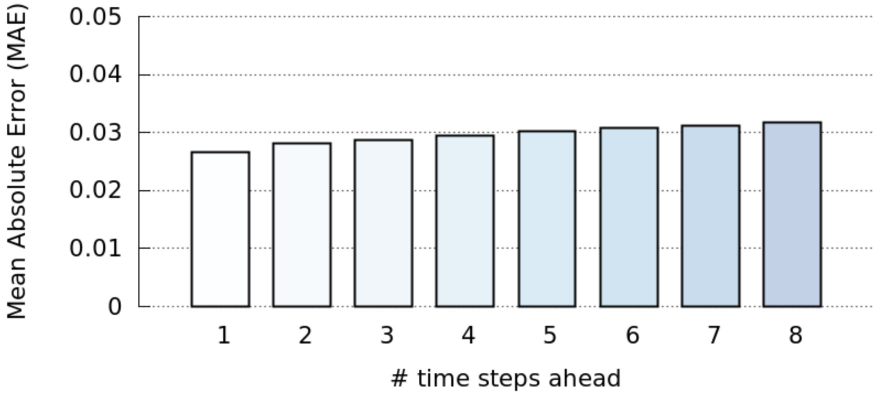


Figure 3.5: RT average MAE of the LQ predictions.

very slowly. It seems possible to affirm that we could predict successfully the LQ several steps ahead in time. We also analyzed the variability of errors for each number of steps ahead. Although the values for the median and the first quartile are similar for all steps ahead considered, the values of third quartile and outliers grow with the number of steps. These differences in the variability of errors lead to the differences in the average MAE.

3.3.1.4 Degradation of the Regression-Tree (RT) Model over Time

This experiment was performed to evaluate the accuracy of the prediction models over time. Figure 3.6 shows the average MAE of the overall network and its approximation to a linear function. It shows the results of the RT algorithm using the same setup as the baseline experiment (a lag window size of 12 instances and a training data-set of 288 instances) but using a test data-set ranging from 144 (1/2 day) to 1728 (6 days) instances. We used linear regression to compute the parameters and estimate the goodness-of-fit, obtaining these parameters: slope = 0.0212 and b = 0.0132 (depicted by a line in Figure 3.6). Clearly, we can affirm that a linear function can be used to model the degradation of the RT over time.

We have also observed that the variability of errors increases linearly with the number of instances of the test data-set. For this reason, it is important to train the model again after a certain period. Due to the fact that both the MAE and the error variability follow a linear function, we could easily determine a trade-off between the resulting error, and the frequency of updates to the model.

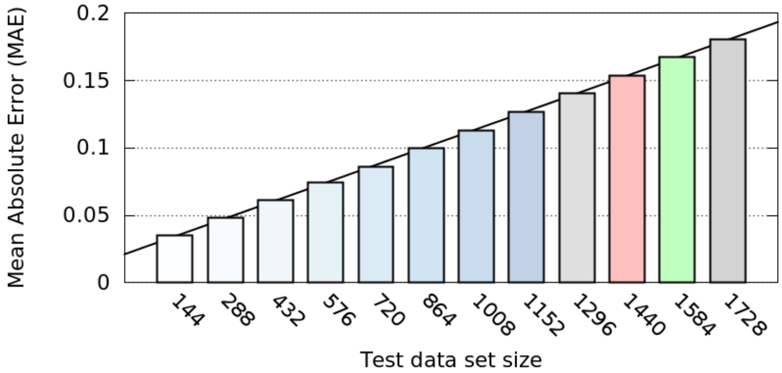


Figure 3.6: RT Mean Absolute Error of the whole network for several test data-set sizes.

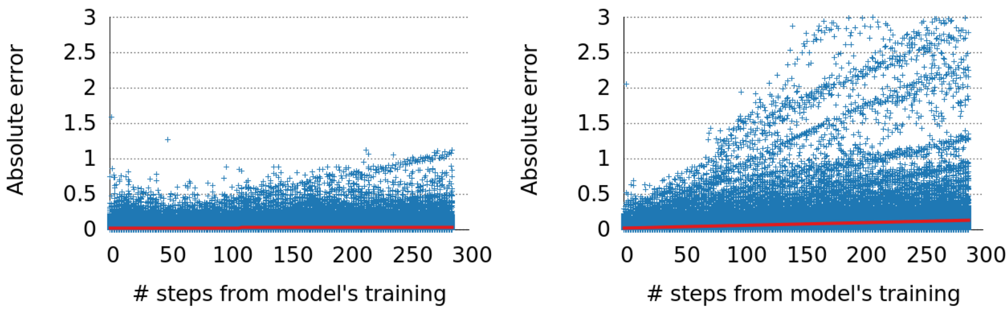


Figure 3.7: RT absolute error with training data-set sizes of 1728 (left) and 228 (right) instances.

Figure 3.7 depicts the evolution of the absolute LQ prediction error over time. In both figures, the RT model was trained at time 0. The left side shows 288 values predicted after training with 1728 instances. As we can observe, most of the errors are within the range from 0 to 1.5. On the other hand, the right side also shows 288 values predicted, but after training with only 288 instances. In this second case, the absolute error dispersion is higher, and the range is also larger (from 0 to 3). The red lines in both figures represent the linear regression of the absolute error. The slope of this regression is 3.8 10^{-4} and 4.8 10^{-4} for the left and right figure, respectively. It is important to notice that, in the second case, the value of the slope is an order of magnitude higher. Both figures show the impact of the size of the training data-set on the prediction error. We can observe that the larger the size of the training data-set, the smaller the error.

3.3.1.5 Analysis of Topological Features

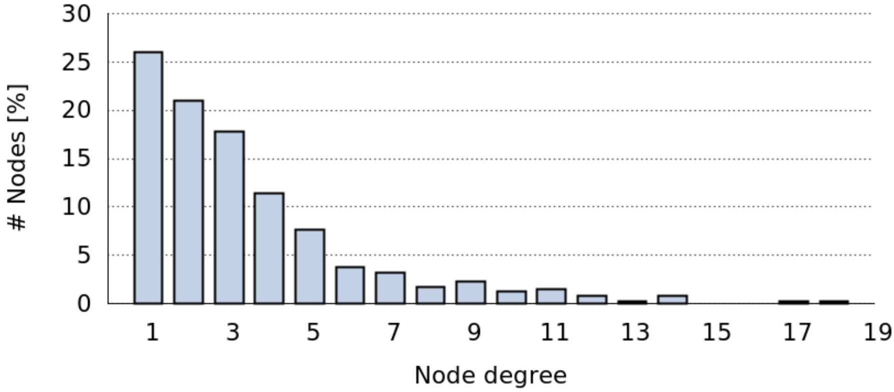


Figure 3.8: Histogram of node degree (for a total of 489 nodes).

In this section we present the results obtained after analyzing whether topological features of links and nodes are related with the behaviour of the LQ. The first topological feature we analyzed was the Node Degree (ND). Figure 3.8 shows a histogram of the degrees of the network nodes. Notice that the majority of nodes (almost 90%) have degrees within the range from 1 to 5. We also have a significant number of nodes with degrees within the range from 6 to 12. Finally, we only have a small number of nodes with degrees higher than 12.

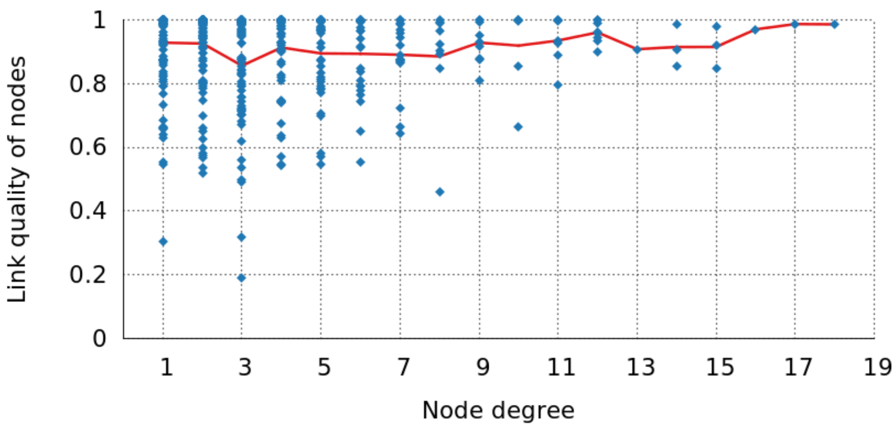


Figure 3.9: Node Degree vs. average LQ (the red line corresponds to the mean value for each degree).

Figure 3.9 depicts the average LQ of nodes (computed as the average of LQ values on each edge of the node). The mean value for each ND is depicted by the red line. We can observe that there are no significant differences of average LQ for different NDs. The worst LQ value corresponds to ND 3, with a mean LQ value of about 0.86. NDs of 9 and above present slightly better values of LQ, but the number of nodes is very small to be statistically relevant. Hence, we do not observe a clear correlation between ND and average LQ.

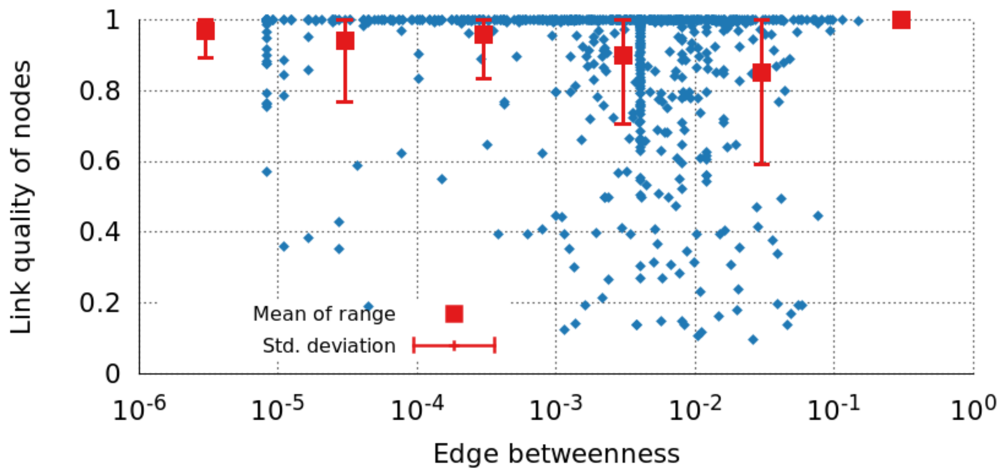


Figure 3.10: Edge Betweenness vs. Link Quality.

The next topological feature we analyzed was the Edge Betweenness (EB). This metric measures how often an edge appears on the shortest paths between nodes in the network. Figure 3.10 plots the LQ that corresponds to different EB values (represented in the figure as the amount of the shortest path length that traverses a node, normalized, with logarithmic scale). In this figure we identify six value intervals (from 10^{-6} to 0). Interval 1 (from 10^{-6} to 10^{-5}) presents a low dispersion of LQ values with a large concentration of them around 0.9. Interval 2 (from 10^{-5} to 10^{-4}) and interval 3 (from 10^{-4} to 10^{-3}) present a moderate dispersion of LQ values, the majority of them having LQ=1 and a few of them having LQ between 0.2 and 0.9. In interval 4 (from 10^{-3} to 10^{-2}) and interval 5 (from 10^{-2} to 10^{-1}) most nodes also have LQ=1, but in this case there is a high dispersion and a significant number of values are between 0.1 and 0.9. Finally, Interval 6 is not relevant as it only deals with a few isolated values. With these results, we can conclude that there is a clear correlation between LQ and EB.

In order to analyze the behaviour of our predictor with respect to the EB metric, we represented the RT prediction accuracy versus the Mean Absolute Error, as shown in Figure 3.11. We can observe that, in general, the behaviour is good for the whole range of EB values. However, for the two EB values with more concentration, the MAE is slightly worse (about 95% of prediction accuracy). When comparing the behaviour observed in figures 3.10 and 3.11, we can conclude that our predictor can successfully follow the behaviour pattern displayed by the Edge Betweenness metric.

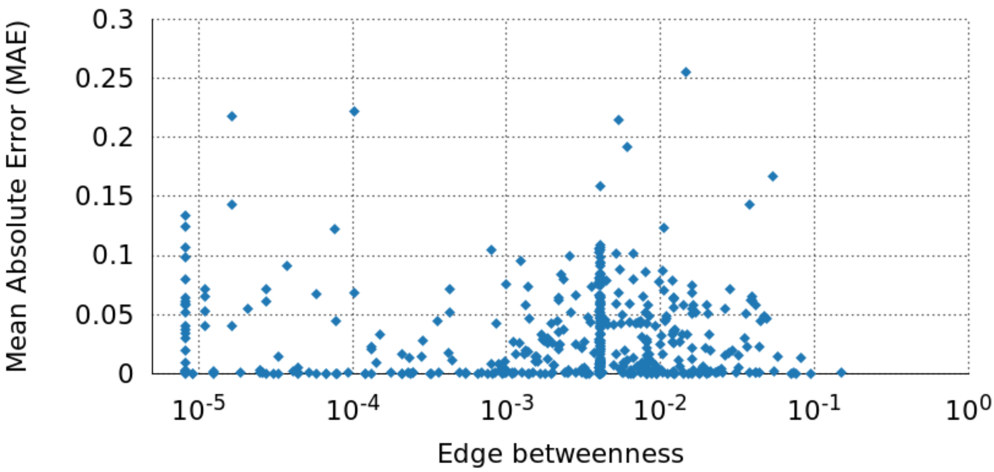


Figure 3.11: Edge Betweenness vs. RT prediction Mean Absolute Error.

3.3.1.6 Lessons Learned

The global analysis of nodes allowed us to determine the general behaviour of the LQ by considering the entire network as a whole. We have observed that some individual groups of nodes and links seem to present a divergent behaviour. This opens a new research line to further analyze the behaviour of such groups, in order to be able to characterize the observed behaviour heterogeneity. Nonetheless, from the global analysis of nodes we performed in the previous sections, we are able to draw some interesting lessons and conclusions:

- RT is the prediction method that performs best, followed by SVM.
- Globally, the MAE and the RMSE metrics show an equivalent behaviour.
- Different lag window sizes produce slightly different results, but we could not determine an optimal size.

- The degradation of RT over time follows a linear function, and the larger the size of the training dataset used, the smaller the resulting error will be.
- Regarding the topological features, the Node Degree does not present a correlation with the LQ, but there are indications that the Edge Betweenness affects the LQ results. However, our results are not conclusive enough to allow us to apply this correlation in the prediction process.

It is also important to notice that any LQ values should lie in the range 0 to 1. Therefore, the predicted values that are greater than 1 or lower than 0, should be considered to be exactly 1 or 0 respectively. Moreover, if the LQ is very poor, it makes no sense to make a prediction (it would be better to disable the predictor). These measures can be applied to improve the prediction results.

3.3.2 Analysis of Individual Links and Nodes

In this section, we perform an analysis of individual links and nodes. Our aim is to identify and analyze data individually, in order to classify the nodes and links according to their behaviour. Thus, we try to answer questions such as: "Do some nodes and links show a similar performance?", and "Is it possible to group nodes and links in clusters according to similarities in their behaviour?" The first part of this section analyzes the variability of the LQ prediction among links. The last part, on the other hand, analyzes the variability of the LQ prediction over time. The aim is to determine if this prediction remains constant with time or if it presents variations and, in the latter case, try to classify the behaviour of these variations.

3.3.2.1 Variability of Link Quality (LQ) Prediction among Links

As an initial analysis, we classified the links that presented particular LQ values in different groups. Figure 3.12 shows a graph with the results considering both all links (blue line), and only those links that present variations in the LQ (red line).

If we take into account the whole set of links, we can identify three different ranges of LQ values. The biggest group (80% of the links) contains link qualities higher than 90%. A second group of links (10%) contains LQs between 70% and 90%. The remaining 10% of the links present LQs below 70%. Similarly, if we consider only the links that present variations in the LQ, these percentages decrease slightly but they seem to follow the same trend. In this case, 60% of the links show LQs

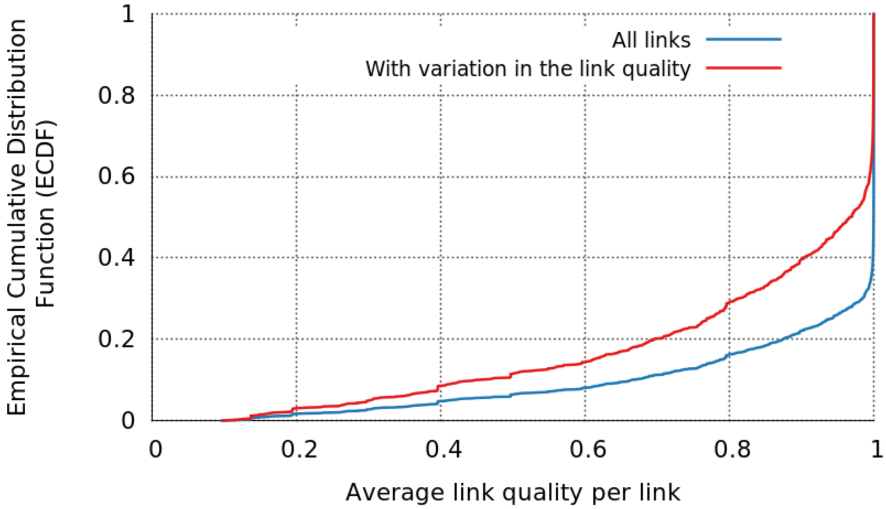


Figure 3.12: ECDF of the average LQ per link.

of 90% or more, 20% of the links show LQs between 70% and 90% and finally, the remaining 20% of the links present LQs under 70%. Therefore, we have a whole range of LQ values but the majority of them are high.

The next analysis was intended to evaluate the behaviour of the four algorithms (SVM, RT, kNN, and GPR) for variations in the LQ in steps of 0.1, and therefore, to check if the behaviour changes for any particular range of LQ values (i.e., to verify if RT is not the best algorithm for a specific range). This study extends the global analysis presented in Figures 3.2 and 3.3 (left). Figure 3.13 shows the results obtained.

As we can observe, RT is the best solution for LQ<0.9, but in the range from 0.9 to 1, SVM shows a slightly better performance. Table 3.1 shows a more detailed analysis of the data for RT and SVM for LQ values in the 0.9-1 range.

	Min.	**1st qu.**	**Median**	**Mean**	**3rd qu.**	**Max.**	**St. dev.**
RT	0.000	0.0002	0.0020	0.0129	0.0154	0.5167	0.03105
SVM	0.000	0.0004	0.0015	0.0130	0.0125	1.4290	0.06172

Table 3.1: MAE in the 0.9-1 LQ range for RT and SVM algorithms.

These results show that SVM has better (lower values) median and 3rd quartile values, the mean values are almost the same for both methods, and SVM performs

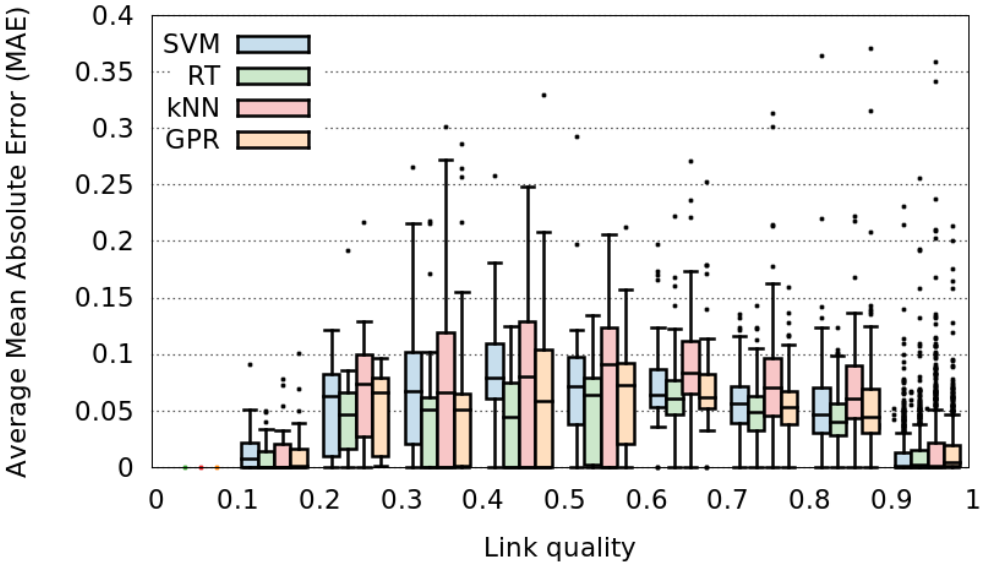


Figure 3.13: Average Mean Absolute Error versus Link Quality in steps of 0.1.

worse in the 1st quartile, maximum, and standard deviation results. This behaviour can be explained due to the fact that SVM presents some very high values that produce a significant increase in the mean and standard deviation values. Despite this, results in Figure 3.13 confirm that, for 87.5% of the links, SVM performs better than RT in the 0.9-1 LQ range.

3.3.2.2 Variability of Link Quality (LQ) Prediction over Time

The last LQ analysis performed was aimed at evaluating how the LQ values of each link vary over time. Figure 3.14 represents the evolution of the LQ values of a sample link over a period of 24 hours. We can observe that the LQ values on this link are not constant: most of the time the values are above 0.85, but at certain times, they drop below 0.7.

In order to include time components in our study, first we need to select the right mathematical tools. We therefore, need to analyze Time-series data (the evolution of LQ values over time) and, then, classify the links into groups (clustering) according to their behaviour.

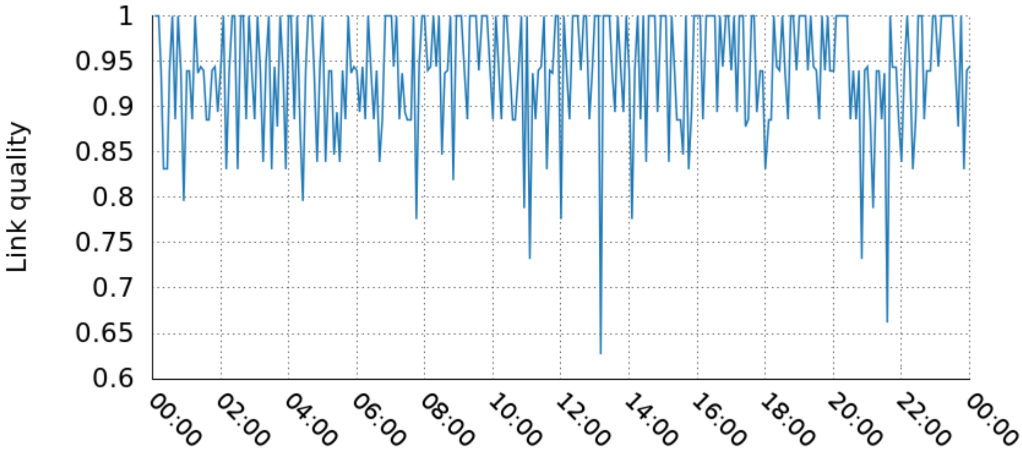


Figure 3.14: Example of the variation of the LQ of a link over a period of 24 hours.

The analysis and comparison of Time series is a common topic in the literature [18, 49]. The classical Discrete Fourier Transform used to characterize Time series, has been improved by the use of Wavelets, and more precisely, Haar wavelets [16]. The Haar wavelet captures the shape of Time series better than the Fourier Transform (within the context of similarity search). The Haar transform decomposes a discrete signal into two subsignals of half its length. One subsignal is a moving average, or trend; the other subsignal is a moving difference, or fluctuation. The Haar wavelet has a major drawback: the basic functions for the Haar wavelet are not smooth (i.e., they are not continuously differentiable). On the contrary, the Haar wavelet approximates any signal by a ladder-like structure. In addition, the number of features we have to use is high (this property is referred to as "slow convergence").

In our case, we used nine features to apply the Haar wavelet to our analysis of the LQ Time series. However, due to the fact that the Haar window is only two elements wide, when a big change appears between an even and an odd value, the high-frequency coefficients will not reflect such a change. Therefore, the Haar wavelet was unable to capture high-frequency changes in our LQ time series, which are very common in our data-set (LQ values that change suddenly between 1 and 0 or vice versa).

The Daubechies wavelet is similar to the Haar wavelet in the sense that it also computes the moving averages and differences via scalar products, but the scaling signals and wavelets are different. Daubechies wavelets use overlapping windows, so that all high-frequency changes are included. For our analysis, we used Daubechies wavelets, with a 4-element window, and 9 features. On the other side, K-means is the most popular partitioning method and requires prior specification of the number of clusters to extract. We used this method to classify our LQ time series in different clusters, with the main purpose of determining whether or not we can achieve better prediction accuracy for a specific cluster.

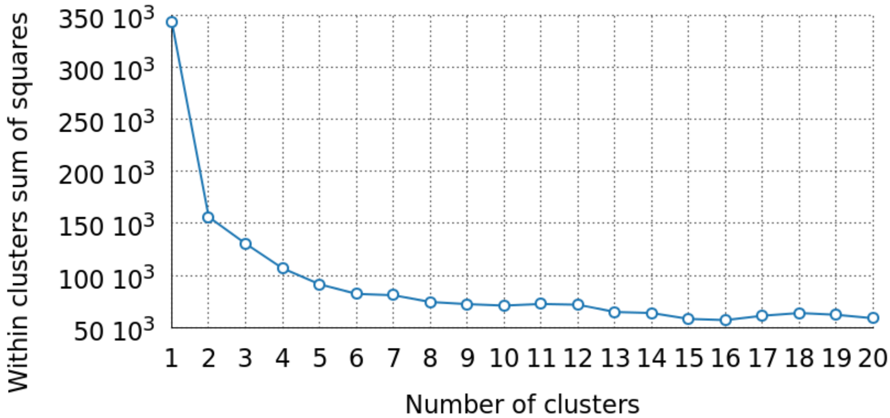


Figure 3.15: Possible number of clusters for LQ Time-series classification.

Our first task, in order to identify the different kinds of links (according to the temporal evolution of their LQ) was to determine the best number of clusters. To do so, we selected only those links that presented changes in the LQ (we excluded links with a constant value of LQ=1). Figure 3.15 shows the "*within clusters sum of squares*" for different possible numbers of clusters (K), up to 20. It is important to notice that although small sum values are better, because the elements within a cluster are more similar to each other, there is not a clear criterion to decide which is the "best" number of clusters. According to the figure, it would be appropriate to try K values of 6, 8, or 10. However, we have to take into consideration that we are not only interested in an analytical interpretation of the common features of the links within a cluster, but also in a human-readable interpretation, and when the number of clusters is large, the amount of elements belonging to some clusters are too small, making it more difficult for humans to interpret the criteria used to classify the links that belong to each cluster. For this reason, we choose K=4.

The next step involved applying the K-means algorithm, with K=4 clusters. In Figure 3.16 we can observe a sample of the LQ time series for each one of the four clusters (notice that axes are the same as Figure 3.14). The main features of the links in each cluster are listed below:

- Cluster 1 (618 links): links with LQ very close to 1 and small oscillations.
- Cluster 2 (148 links): links with large oscillations in LQ.
- Cluster 3 (206 links): links with small LQ oscillations, but LQ values far from 1.
- Cluster 4 (55 links): links with ON/OFF behaviour (with small oscillations).

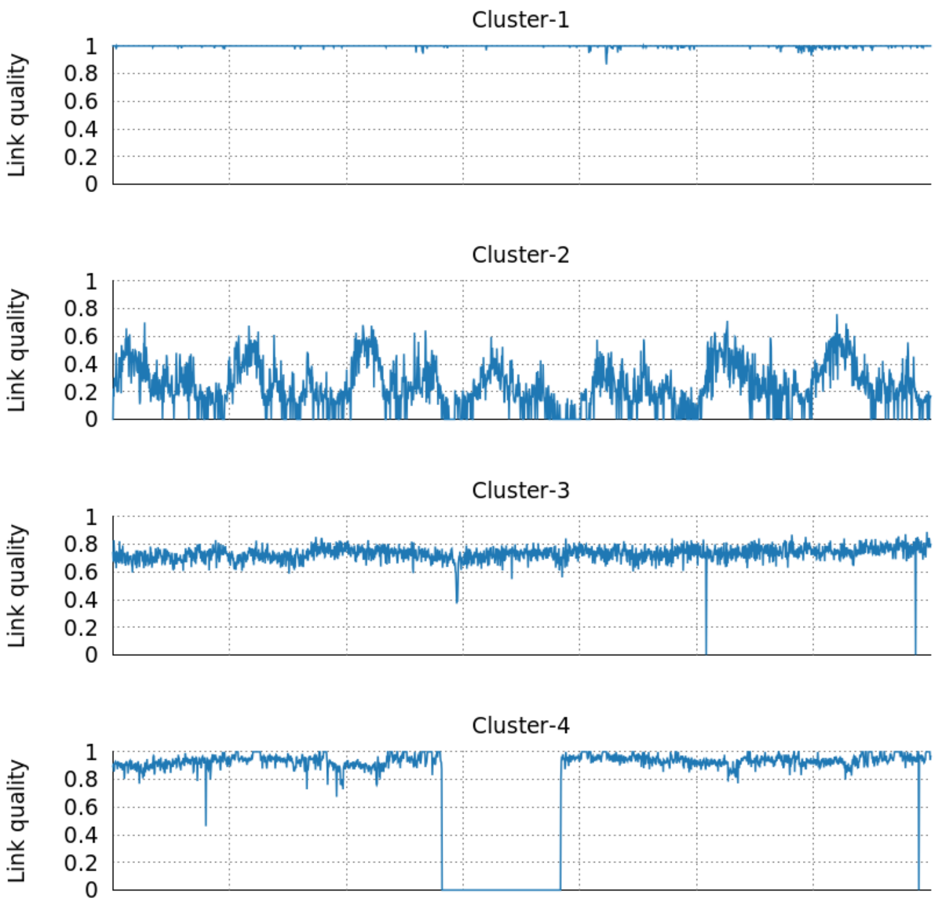


Figure 3.16: Sample of LQ Time series for each of the 4 clusters from K-means algorithm.

Once we identified the four clusters, we could analyze their behaviour. Figure 3.17 contains the boxplots of the average LQ for each one of the four cluster types. We can observe that clusters 1 and 4 achieve the best LQ values (closer to 1), cluster 3 has LQ values around 0.7, whereas cluster 2 is the worst (with average LQ values of just 0.4). This implies that the prediction would be better for clusters 1 and 4, but we will find more problems in clusters 2 and 3. When a link in cluster 4 is in the OFF state, the link does not exist and it does not have LQ values, thus the prediction is disabled.

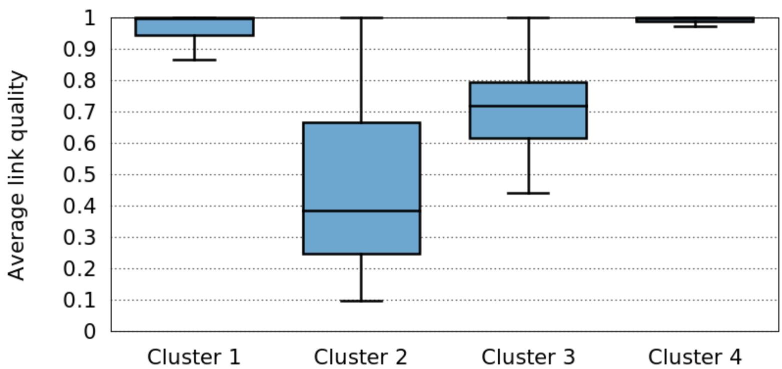


Figure 3.17: Average Link Quality of the links of each cluster, depicted as box plots.

We have also analyzed how well each one of the four prediction methods (SVM, RT, kNN, and GPR) performs for each of the four cluster types. The results are available in Figure 3.18. Comparing these results with the ones depicted in Figures 3.2 and 3.3, we confirm that the global results previously obtained, where RT was the best prediction method, are now still valid for clusters 2 and 3. For cluster 1, nevertheless, both RT and SVM have a similar performance. A different result is obtained for cluster 4, where RT is the third best method, after kNN (the second best) and SVM (the best). Previously, in Figure 3.13, we had also observed a case with SVM performing better than RT, but now the difference is even more significant.

In order to further analyze the results presented in Figure 3.18, we considered each of the four clusters separately, and evaluated each one of them for increasing LQ values in steps of 0.1 (as done previously for Figure 3.13). The results are presented in Figure 3.19. To compare the different prediction methods, we use the 3rd quartile (the top part of each box) because it represents the majority of the links (75% of the error values lie below the 3rd quartile)

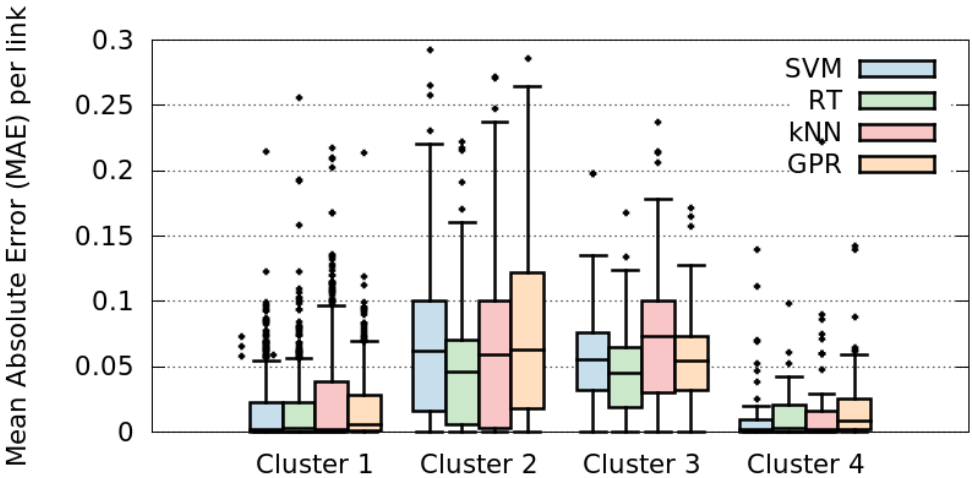
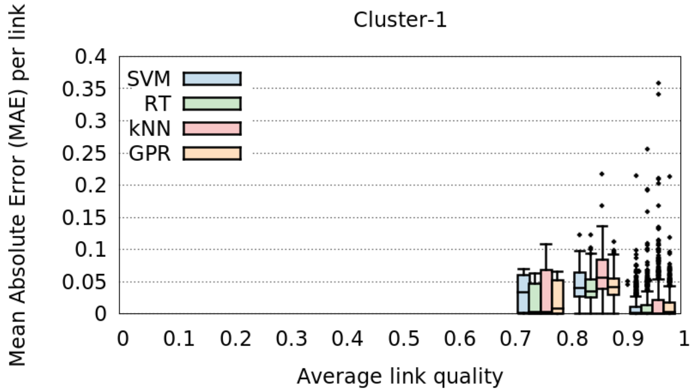


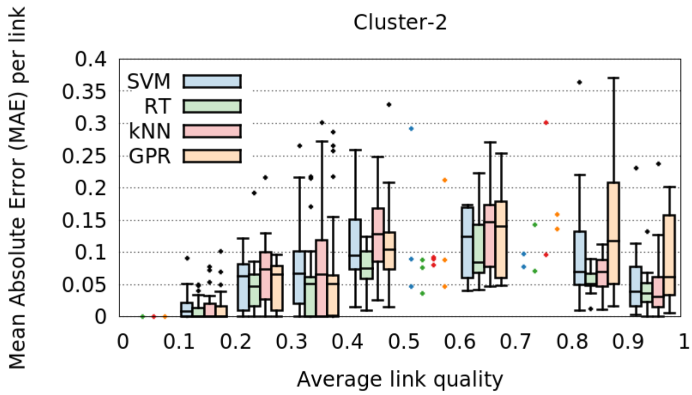
Figure 3.18: Prediction error of each method for each cluster, depicted as boxplot.

The results can be analyzed for each cluster:

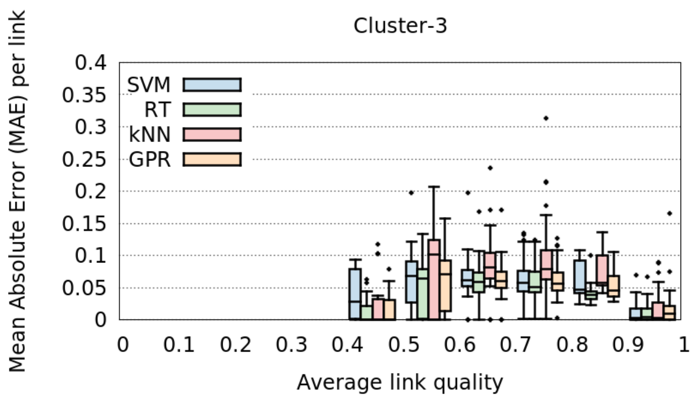
- Cluster 1: LQ values are in the range from 0.7 to 1. For LQ values below 0.9, RT achieves the best predictions. However, in the range from 0.9 to 1, SVM is again the best algorithm.
- Cluster 2: here we find a broader range of LQ values, from 0.1 to 1. In all ranges, RT performs best than any of the other prediction methods.
- Cluster 3: this is the cluster with the second wider range, with LQ values between 0.4 and 1. Similar to cluster 1, in this case it is also better to use RT for LQ values below 0.9, but switching to SVM for LQ values above 0.9.
- Cluster 4: this is the cluster with the narrowest LQ range, from 0.8 to 1. Also in this cluster, it is better to apply RT or SVM for values below or above 0.9 respectively.



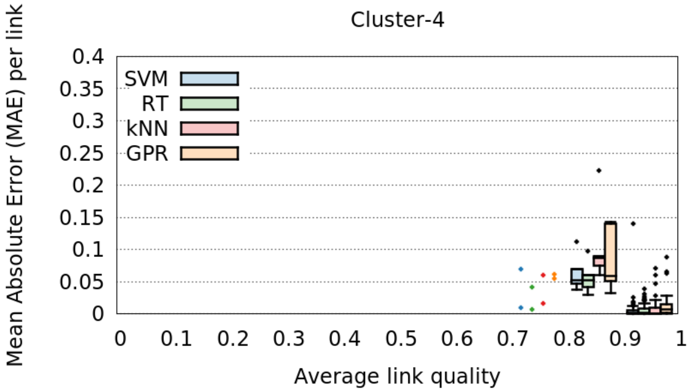
(a) Cluster 1 average MAE vs. LQ



(b) Cluster 2 average MAE vs. LQ



(c) Cluster 3 average MAE vs. LQ



(d) Cluster 4 average MAE vs. LQ

Figure 3.19: Average Mean Absolute Error vs. LQ (in steps of 0.1), for each prediction method and cluster, depicted as box plots.

We should note that although SVM is better than RT in some cases, it takes a more aggressive prediction approach, trying to follow very accurately any change, but generating more outliers and therefore, making more errors. On the contrary, RT is more conservative when predicting changes, and for that reason produces a lower number of errors than SVM.

Comparing Figure 3.13 and Figure 3.19, we can observe:

- The boxplots in Figure 3.13 with LQ values between 0 and 0.4 correspond to cluster 2 in Figure 3.19.
- The boxplots in Figure 3.13 with LQ values in the range from 0.6 to 0.8 correspond to cluster 3 in Figure 3.19.
- The boxplots in Figure 3.13 with LQ values above 0.8 mainly correspond to cluster 1 (because cluster 1 contains 618 links, whereas cluster 4 only contains 55 links).

3.3.2.3 Lessons Learned

The LQ prediction analysis we performed individually on the links, provided us with some interesting results. This helped us reach the following conclusions:

- In general, the RT predictor achieves the best results.

- The majority of the LQ values lie within the range from 0.9 to 1.
- For clusters 1, 3, and 4, the SVM predictor performs better than the RT for LQ values within the range from 0.9 to 1.
- For cluster 2, RT always produces the best results for the whole range of LQ values.
- To include topological features in the predictor does not improve the results.
- The predictor generates LQ values out of the range from 0.0 to 1.0, which offers an opportunity to improve the predictor.

3.3.3 Enhanced Predictor

Based on the observations made in the previous section, we propose two orthogonal improvements to the prediction process: (1) saturation, and (2) RT/SVM. On the one hand, we have seen that all four predictors analyzed in this chapter make predictions of LQ values by applying methods and computations that produce unbounded values. Therefore, we want to study how much the prediction process improves when we limit the predicted values to the "logical" range from 0.0 to 1.0. Thus, if the prediction produces values above 1 or below 0, we "saturate" the predicted value (i.e., all predictions above 1 would be 1, and all predictions below 0 would be 0). Obviously, this simple improvement will produce smaller errors, and will result in a better overall performance of the prediction process. On the other hand, we have also seen that sometimes the SVM predictor performs better than the RT predictor (LQ values within the range from 0.9 to 1 for clusters 1, 3, and 4), but sometimes the RT predictor performs better than SVM (the whole range of LQ values for cluster 2). Therefore, we propose to select the best of these two prediction approaches depending of the current cluster and range of values.

In order to test the various improvements, starting from the *Base* method (RT without saturation), we add the saturation process (RT Sat.), then a combination of RT and SVM (RT/SVM) and, finally, optimizations to both algorithms (RT/SVM Sat.). The analysis was performed with the parameters shown in Table 3.2.

A comparison of the results obtained for the *Base* and the three *Optimized* methods, together with the improvement percentage (computed as $(Base - Optimized)/Base$) are presented in Table 3.3 and Figure 3.20.

If we analyze the results presented in Figure 3.20 and Table 3.3, we can conclude the following:

Parameter	Values used for the analysis
Maximum lag window size	12
Training data-set size	1728
Test data-set size	288
Metric	Absolute Error of LQ overall predictions

Table 3.2: Optimization analysis parameters.

	RT (base)	**RT Sat %Improv.**	**RT/SVM %Improv.**	**RT/SVM Sat. %Improv.**
2nd qu.	0.0040	25%	62.5%	75.0%
3rd qu.	0.0298	2.0%	3.0%	4.0%
87.5% per.	0.0645	1.7%	0.2%	1.1%
Max.	8.2130	87.8%	0.0%	87.8%

Table 3.3: Results of the Base (RT) and *Optimized* methods (Sat.: with saturation, RT/SVM: combination of the RT and the SVM algorithms, for clusters and ranges where SVM performs better than RT). %Improv.: percentage of improvement over the *Base* method. The results correspond to the maximum and representative quartiles.

- For the 50% of the links that had good prediction results (2nd quartile), all improvements have a significant impact on the results, especially when combining the RT and the SVM algorithms.

- For links that displayed the worst behaviour (with an absolute error close to the maximum value) only the saturation strategy produces some improvements.

- For the rest of the links, which achieved a moderate prediction performance (those with values from the 3rd quartile to almost the maximum), any of the strategies proposed showed significant improvements in the results.

- According to Figure 3.20, by applying the saturation strategy we can observe an increment in the Empirical Cumulative Distribution Function (ECDF) for absolute errors smaller than 10^{-5} (in practice, this means an absolute error equal to 0). Therefore, we can claim that the optimization strategies based on the correction of the predicted value produce a significant improvement in the results.

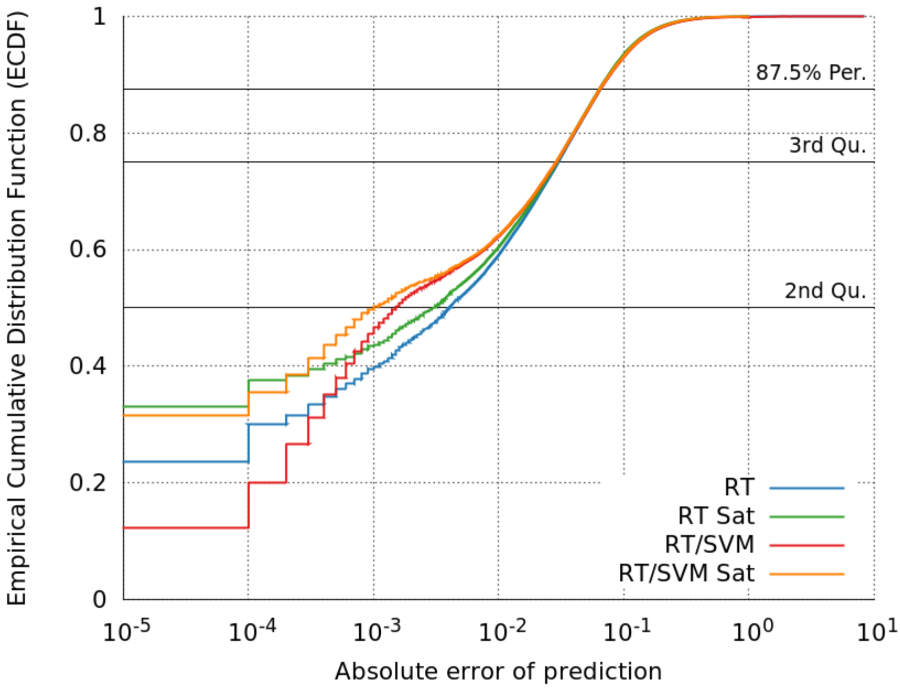


Figure 3.20: ECDF of the absolute error of the overall LQ predictions for the Base (RT) and the three enhanced methods.

3.4 Conclusions

This chapter has been devoted to answer *SRQ2: What impact does prediction have on the quality of ad hoc networks?*, with regard to Link Quality prediction. We have demonstrated that Time-series analysis is a promising approach to accurately predict LQ values in Community Networks. This technique can be used to improve the performance of the routing protocol by providing information to make appropriate and timely decisions to maximize the delivery rate and minimize traffic congestion.

We analyzed results from four learning algorithms (Support Vector Machine, k-Nearest Neighbours, Regression Tree, and Gaussian Process for Regression that model Time series. All algorithms achieved percentages of success between 95% and 98% when predicting the next future value of the LQ, with the Regression Tree being the best one. Moreover, these results were obtained just considering those links that experienced variations. Therefore, the prediction accuracy could have been even better by including all the network links.

In addition, we showed that the prediction of values that are more than one step ahead in time (and not just the next value) also achieves high success ratios, between 97% and 98%. We also observed that the size of the training data set is a key factor to achieve high accuracy in the predictions. The bigger the size of the data-set, the smaller the degradation of the error over time.

The global analysis of the LQ behaviour we performed in this chapter, allowed us to identify the best prediction algorithm (Regression Tree) and metric (Mean Absolute Error), and to understand the impact of lag windows in the prediction (although we were not able to determine the best lag window size). It also helped us to evaluate the accuracy of prediction some time steps ahead into the future (it seems possible to predict the LQ some steps ahead), the degradation of prediction over time (the degradation follows a linear function: the larger the size of the training data-set, the smaller the error), and the correlation of prediction with some topological features (node degree does not present correlation with LQ, but there are indications to the fact that edge betweenness produces changes in the LQ; however, it is not straightforward to apply this correlation to the prediction process).

We also analysed the behaviour of links individually to identify the variability of the LQ prediction between links (for LQ values within range 0.9-1, SVM performs better than RT in 87.5% of links), and over time (we identified four kinds of links according to the results of a Time-series clustering; and SVM continued performing better than RT in three of the four clusters).

Finally, we also enhanced the prediction method taking into consideration our previous knowledge about the expected LQ values. To perform this, the optimized predictor applied an approximation (saturation) of the predicted LQ values to keep them within the range from 0 to 1. Combining RT, SVM, and saturation, we achieved improvements of up to 75% for the second quartile, but for links that displayed the worst behaviour, only the saturation strategy produced some improvements. Therefore, we can claim that the optimization strategies based on the correction of the predicted value, lead to a significant improvement in the results.

Chapter 4

End-to-End Quality Prediction

In this chapter, the answer to the *Specific Research Question 2 (SRQ2): What impact does prediction have on the quality of ad hoc networks?*, is extended from Link Quality (LQ) to End-to-End (full path) Quality. This chapter analyzes the use of Time-series analysis to predict End-to-End Quality (EtEQ). We apply this prediction technique in the routing layer of large-scale, distributed, and decentralized networks. We demonstrate that it is possible to accurately predict EtEQ with an small average Mean Absolute Error (MAE). Particularly, we analyze the path properties and path Expected Transmission Count (ETX) behavior to identify the best prediction algorithm. Moreover, we analyze the EtEQ prediction accuracy some steps ahead in the future, and also its dependency on the hour of the day.

4.1 Introduction

Community Networks (CNs) are large-scale, distributed, and decentralized networking infrastructures with several nodes, links, and services. This kind of networks are extremely diverse and dynamic, because of their decentralized nodes, their mix of wired and wireless technologies, their several routing schemes, and their diverse services and applications [6]. Those networks are made available to a community of people living in the same area. The network management is based on an open peering agreement, which avoids barriers for the network participation. Ownership, governance, and knowledge are also open (community members own and manage these networks). Community Networks (i.e., FunkFeuer [30] and guifi.net [9]) grow dynamically with regards to the number of links.

CNs features (large, heterogeneous, dynamic, decentralized) raise challenges of interest for researchers [12]. One of the most important challenges is the effect of the asymmetrical features and the unreliability of wireless communications on network performance and routing protocols. Many metric-based routing protocols for mesh networks that track LQ and select higher-quality links, have been proposed to minimize traffic congestion and maximize delivery rate [26, 45, 75, 96]. Hence, when routing packets through an unreliable network, LQ tracking is definitely a key method to apply. Moreover, routing algorithms should avoid weak links as soon as possible [79], and whenever possible [93]. LQ estimation [8] (or prediction [51]) approach increases the improvements achieved by LQ tracking in routing performance. Usually, real-time metrics do not provide enough information to detect the degradation or activation of a link at the right moment. Therefore, prediction is needed to foresee LQ changes and take the appropriate measures.

End-to-End Quality (EtEQ) or Path Quality extends the LQ concept to the full communication path (between sender and receiver) and it is computed based on the quality (ETX) of the individual links that conform the communication path. In this chapter, we want to analyze if our previous work on LQ [65] (Chapter 2) is applicable to the full communication path (EtEQ tracking and prediction) and determine what differences exist between individual LQ and EtEQ. To the best of our knowledge, no previous work explores EtEQ prediction in the routing layer of large-scale, distributed, and decentralized systems.

Prediction techniques have been applied in several ways, such as routing traffic reduction [39, 59, 60], energy efficient routing [44, 52, 56], or LQ estimation.

LQ tracking has been previously applied in several scenarios in different ways [26, 45, 75, 96] to select higher quality links that maximize delivery rate and minimize traffic congestion. Link Quality Estimators (LQE) [8, 51] are in charge of measuring the quality of the links between nodes based on physical or logical metrics. Physical metrics focus on the received signal quality, and logical metrics focus on the percentage of lost packets. LQE with metrics like Link Quality Indication (LQI) [28], Signal-to-Noise Ratio (SNR) [53], or Received Signal Strength Indication (RSSI) [83] fit in the former category, whereas metrics like Required Number of Packets (RNP) [15], Expected Transmission Count (ETX) [23, 69], or Packet Success Rate (PSR) [96] fit in the latter. All these metrics can be used by LQE in isolation or as a combination of some of them [8, 51, 76] to select the more suitable neighbor nodes when making routing decisions. LQ prediction is used in addition to LQ tracking to determine beforehand which links are more likely to change their behavior. Although LQ prediction is not identical to EtEQ prediction, some of the above techniques can be very similar [15, 23, 28, 53, 76, 96]. This relationship is even more direct in the case of ETX EtEQ, studied in this chapter, which is a linear function of the ETX LQ. As a result, the routing layer can take better decisions at the appropriate moment.

End-to-End Quality has not been widely considered in the past. In fact, the main efforts have been focused on determining the cost of a single link, and then extended to the full path by assuming the cost of a path as the sum of costs of several links. For instance, Metric-Aware Rate Adaptation (MARA) [72] has been proposed as a method to make more accurate routing decisions by combining route quality evaluation and automatic rate selection. To do this, MARA computes ETX of every link at every available rate, by estimating metrics as SNR, packet error rate, and probe packet size. End-to-End Retransmissions (EER) operation model [10] claims that the total cost of a path cannot be expressed as a linear sum of individual link costs. Therefore, variations of this simple formulation are proposed, such as incorporating error rate in the link cost. This led the authors to achieve significant energy savings compared to traditional minimum energy approaches. Expected number of Transmissions On a Path (ETOP) [42] has been proposed as a path metric to determine reliable end-to-end packet delivery. In the same way as EER, this chapter assumes that the cost of a path does not only rely on the quality of individual links, but also on their relative position on the path. Finally, End-to-End Delay (EED)/Weighted End-to-End Delay (WEED) [48] is another approach that was designed as a link/path metric to select paths with minimum end-to-end delay and high network throughput, but considering load balancing of routing. In any case, there is no work concerning prediction of Path Quality in a Wireless Mesh Community Network (WMCN).

There are some relevant works that must be paid special attention as they are related to our study: Wang, et al [94], Maccari and Cigno [54], Cunha et al [22] and Millán et al [65]. Wang et al [94] introduces the MetricMap mechanism, that is fundamentally a routing protocol for Wireless Sensor Networks that uses a learning-enabled method for LQ assessment. Based on the observation that high traffic rates make tracking Link Qualities more difficult, this protocol uses prediction methods to estimate them in advance. In a first stage, a Machine Learning algorithm is applied to classify Link Qualities. Two types of classifiers are evaluated: a decision tree and a rule-based classifier. The data used to train both classifiers was preclassified offline, based on a LQ indicator and other metrics that represent some features of the nodes. In a second stage, the MetricMap routing protocol estimates the LQ at runtime by replacing the current traffic information with the rules collected offline from the classifiers. Results show that MetricMap can achieve a significant improvement on the data delivery rate in high traffic-rate applications.

Maccari and Lo Cigno [54] have considered the FunkFeuer network focusing on link-layer properties, topological patterns, and routing performance. They have analyzed the quality of the routes, and proposed a couple of techniques to select the Multi-Point Relay (MPR) nodes in the Optimized Link State Routing (OLSR) protocol. Traditionally, routing algorithms assume that mesh networks are fairly stable, but we have also observed that this is not completely true. Therefore, MPR selection should consider the path variability of a node instead of selecting them by agreement. We also analyze the quality of routes, but focused on estimating its future quality, to improve the routing layer in order to select links that maximize the delivery rate and minimize traffic congestion.

Cunha et al [22] proposed a simple strategy for improving routing in the Internet domain, that moves in two ways: first, it detects path changes (NN4 approach) and then it remaps these paths once a change is detected (DTRACK approach). Therefore, DTRACK adapts path sampling rates to minimize the number of missed changes, based on NN4's predictions. This predictor is based on Rule-Fit, a well-known machine learning technique, that takes into account inputs as route prevalence, route age, number of past route changes, and number of times a route appeared in the past. Results show that NN4 is not highly accurate, but it demonstrates the potential of prediction to improve the routing layer when making routing decisions.

Finally, Millán et al [65] analyze the behavior of LQ prediction in the routing layer of large-scale, distributed, and decentralized systems, composed of many nodes, links, content, and services. In summary, the main contributions of this work are (1) the employ of Time-series analysis to estimate LQ in the routing layer for real-world WMCN, (2) the detailed evaluation of results, assuming several learning algorithms to show the potentiality of Time-series analysis for estimating LQ in short and in long term, and (3) the evidence that LQ computed from Time series can be used to accurately predict future values in WMCN. This work is the most similar to ours in this Chapter 4, as both deal with Time-series analysis to improve the routing protocol, but in this chapter we focus on EtEQ instead of LQ. To the best of our knowledge, this is the first attempt to predict EtEQ in WMCNs.

The main contributions of this chapter are the following:

- A detailed analysis of path properties and path ETX behaviour in WMCNs, showing that Path Quality prediction is possible and meaningful.

- The use of Time-Series analysis to estimate EtEQ in the routing layer for real-world WMCNs.

- Clear evidence that EtEQ values computed through Time-Series algorithms can make accurate predictions in WMCNs.
- A detailed analysis of the prediction accuracy for the next step considering also the hour of the day, and for some steps ahead in the future.

The rest of this chapter is structured as follows. Section 4.2 presents the experimental methodology used in this chapter. Section 4.3 corresponds to the analysis of results, starting with the data set analysis for both path and ETX behaviour in subsection 4.3.1; then subsection 4.3.2 compares four well-known learning algorithms based on Time series; in subsection 4.3.3 we analyze EtEQ Prediction with Rule-Based Regression; finally, in subsection 4.3.4 we explore if it is possible to predict EtEQ values some time-steps ahead into the future. To finish this chapter, section 4.4 provides some conclusions.

4.2 Experimental Methodology

In this section we apply the research methodology presented in section 1.3. Subsection 4.2.1 explains the data-collection process. In subsection 4.2.2 we define the research objectives. Then, in subsection 4.2.3 we explain the design and implementation step of the methodology. Finally, subsection 4.2.4 presents the experiments and simulations performed.

4.2.1 Data Collection: FunkFeuer Network and Open Data Set

FunkFeuer [30] is a non-commercial project maintained by computer enthusiasts that install WiFi antennas across rooftops in several places of Austria, that are relatively close to each other (Vienna, Graz, Weinviertel, and Bad Ischl). Currently, there are around 2000 wired and wireless links, and every week new antennas are added to the network. FunkFeuer uses the OLSR-NG routing protocol, which expands the capabilities of the OLSR protocol and makes it highly scalable. In fact, some members of the FunkFeuer network are actively involved in the olsr.org open source project as developers, testing the protocol in the network. Furthermore, the FunkFeuer network maintains open data-sets, available also through the Community Networks Testbed for the Future Internet (CONFINE) Project open-data platform [21], which were used in this chapter. The chosen dataset is composed of OLSR information such as routing tables and network-topology data, collected during 7 days in the period from April 28th to May 4th, 2014. The largest of the shortest paths in the network (diameter) is 18. This means that there are several paths where packets have to go through a relatively high number of Hops, in order to reach their destination. The routing protocol must, therefore, react quickly to any change in the network topology, since this will be critical to achieve high performance.

As stated before, FunkFeuer assumes OLSR, a link-state routing protocol where every node maintains a connectivity map for all the network. Exploiting this OLSR property, FunkFeuer publishes its complete network information from the point of view of a single node (ego-network). While convenient for data collection, this method comes with the downside that the data-set is biased and does not represent the real network-state, since the time for event propagation throughout the network is not negligible. In other words, the higher the distance between a node and an event that happens in the network, the later this event will be present in the nodes global-view. Therefore, prediction of path changes can improve local node routing decisions, since it can provide the node with an estimation about the future local and remote events.

4.2.2 Research Objectives: End-to-End Quality

The research objective of this chapter is to answer *SRQ2: What impact does prediction have on the quality of ad hoc networks?*, by extending it from Link Quality to End-to-End (or Full Path) Quality. For this reason, we focus on End-to-End Quality (EtEQ) prediction for the FunkFeuer network by means of a Time-series analysis.

ETX [23] is an active-probing link-metric, designed for MANETs and widely used in mesh protocols, based on estimating the bidirectional loss ratios of a link. The ETX value of a link is the number of expected transmissions needed to send a packet over the link, and is computed as follows: $ETX = 1/(LQ \times NLQ)$, where LQ and NLQ stand for the "Link Quality" and the "Neighbour Link Quality" of that link, respectively. The ETX of a path is defined as the sum of the ETX value of the links that compose the path. As a result, ETX is always greater or equal to the actual number of Hops in the path. The difference between the path ETX and the number of Hops of the path is the expected number of losses.

Figure 4.1 illustrates the temporal evolution of the number of paths (routing entries) for a period of 8 days. It can be observed a quite significant fluctuation on the number of paths seen by one node, of about 7%.

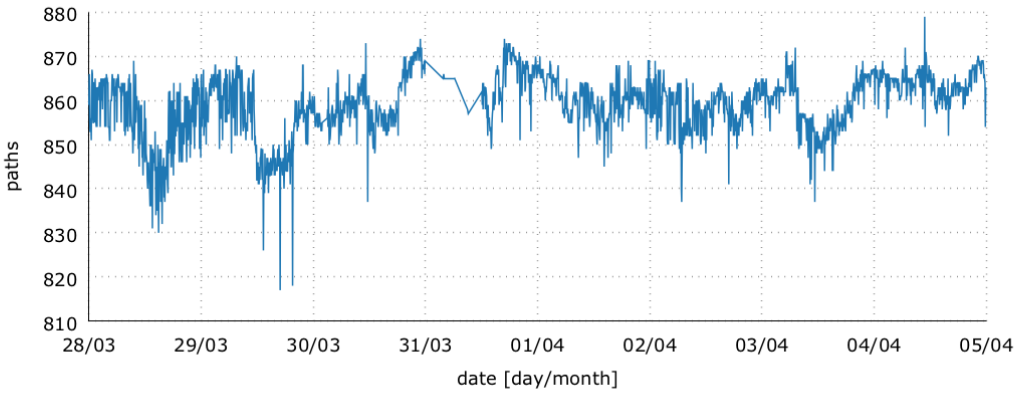


Figure 4.1: Temporal evolution of the number of paths.

The OLSR protocol uses ETX to choose, for each device and packet, the next hop. Concerning physical links, the LQ assumed by OLSR is defined as the fraction of successful packets (HELLO) that were received by a node from a given neighbor within a certain time window, while the NLQ is the fraction of successful packets that were received by the neighbor within a time period. Concerning paths, OLSR computes the ETX of all possible paths from the source to the destination, as described above, and chooses the one with minimum ETX value. That is to say, the ultimate decision to be made by OLSR will be about the selected paths; therefore, the final metric value that will be the subject of comparison will relate to the whole path. As a result, prediction of the path ETX will allow more efficient routing decisions in an unstable environment, taking also into account the ego-network measurement effect previously explained. It is important to point out here that LQ, as defined by ETX and studied in this chapter, ignores the parameters of transmitted-packets size, as well as link transmission rate. Consequently, this chapter considers that the significant Path Quality parameter is packet loss.

It is well known that selecting high quality links in real-world networks, composed by wireless channels with unpredictable conditions, is a big challenge for achieving high delivery rate and performance. Our research goal in this chapter is to assess if the improvements previously achieved by applying LQ tracking and prediction techniques, are also achievable when considering the full communication path (End-to-End Quality). To evaluate the potential benefits of this proposal, we first analyze the characteristics of a well known, free, and experimental WMCN, that deals with the Optimized Link State Routing (OLSR) protocol to maintain the network topology.

4.2.3 Design and Implementation: Time-Series Analysis

As stated before, we deal with Time-series analysis to estimate LQ in the routing layer for real-world WMCNs. To do this, we assume the FunkFeuer experimental network and an OLSR data-set of several nodes and links.

A Time series is a set of data collected over time with a natural temporal ordering. It differs from typical Data Mining or Machine Learning applications, where the ordering of data points within a data set is not important. Time-series analysis is the process of using statistical techniques to model and explain a time-dependent series of data points. Similarly, Time-series forecasting is a method that uses a model to generate predictions (forecasts) of future events based on known past events. In our case, we used more than one prediction algorithm, so that we do not rely on a specific learning technique. We applied four of the best well-known approaches that behave better for this study [3]: SVM, kNN, RT, and RBR.

- The Support Vector Machine (SVM) algorithm has recently become one of the most popular and widely used methods in Machine Learning. It performs a linear or nonlinear division of the input space, and builds a prediction model that assigns target values into one or another category.
- The k-Nearest Neighbours (kNN) algorithm is one of the most simple machine learning algorithms as it makes no assumptions on the underlying data-distribution. This algorithm takes the k data-points closest to the target value, and picks the most common one.
- Regression Tree (RT) is a type of decision-tree algorithm where the target value can have continuous values. This method recursively partitions the data space and runs a simple prediction model within each partition.
- Finally, the Rule-Based Regression (RBR) algorithm is similar to a decision-tree approach, but it is a stronger model that provides rules that are often potentially more predictive.

4.2.4 Experiments and Simulations: Training/Test, Learning Algorithms, Framework, and Metrics

We applied training and test-sets validation to asses the predictive accuracy of the models. After a model is processed using the training set, it is tested by making predictions against the test set. For this purpose, we used the Weka-workbench system [36], a framework that incorporates a variety of learning algorithms, and some tools for the evaluation and comparison of the results. Weka has a dedicated environment for Time-series analysis that allows forecasting models to be developed and evaluated. The Weka's Time-series framework takes a Machine Learning or Data Mining approach to model Time-series by transforming the data into a form that can be processed by standard propositional learning algorithms. To do so, it removes the temporal ordering of individual inputs, by encoding the time dependency via additional input fields.

Usually, classification studies assess the predictive power of their model by using Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE), both widely used in related work. We assume MAE in our experiments, as it is a common method to evaluate the performance of prediction approaches, that gives the same weight to all individual differences. This metric is computed through the following formula: $MAE = sum(abs(predicted - actual))/N$.

4.3 Analysis of Results

4.3.1 Data Set Analysis

In this subsection we present the performed data-set analysis, that provided us with a better insight into ETX and path behaviour, but also helped us to confirm our hypotheses that Path Quality prediction is possible and meaningful.

4.3.1.1 Path Behavior

Our first study analyses the temporal evolution of the number of paths (routing entries). As stated before, Figure 4.1 shows this evolution in a period of 8 days. We can observe that the fluctuation of the number of paths seen by one node is quite significant (about 7% of variation). This observation contrasts with the general assumption in mesh networks that the number of paths is stable. Multi-Point Relay (MPR) selection, that is normally made by consensus, does not take into account this observed variation and, therefore, the accuracy of finding the best routes can be compromised. Our Time-series-based analysis will consider this variation to make accurate predictions of future EtEQ.

Figure 4.2 plots the fraction of snapshots over the total for which the path has been present, namely the *Persistence* of each path. Considering that the snapshots were taken every 5 minutes, we observe that about 10% of paths lie below 60%

of persistence, but the remaining 90% of paths are very stable. We notice that the 10% of unstable paths can explain the variations in Figure 1. Although the amount of unstable paths is significant, a path with very low persistence cannot be useful for predicting the behavior of stable paths. Therefore, taking into account that the data set represents 8 days, for the analysis done in the following sections we ignore paths that are stable less than 2 days in total (persistence below 25%).

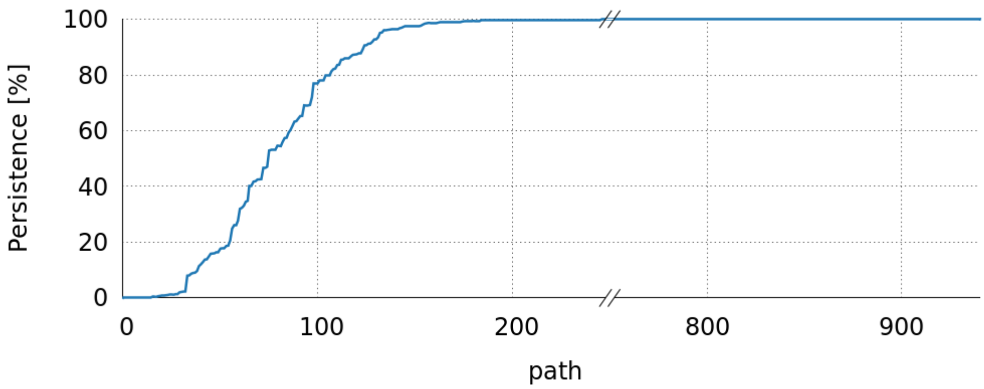


Figure 4.2: Persistence of paths.

4.3.1.2 ETX Behavior

To understand the behavior of ETX Path Quality, we made some additional measurements. Notice that in the data-set introduced above, the corresponding EtEQ is ETX Path Quality.

Figure 4.3 shows the evolution of average ETX Path Quality inside a day for the 5 working days included in the data set. From this figure, it can be seen that ETX path values are more stable at night, between 2 a.m. and 8 a.m. Throughout the day, between 9 a.m. and 1 a.m., there are more fluctuations, reaching differences of even 1 expected transmission. Variation of actual ETX path values can be significantly higher, but this information is hidden by the average value. For non-working days this behavior is less apparent. Similar patterns are often found in network usage data sets, therefore we can assume that the reasons for the depicted behavior is the variation of network traffic and interference during the day, leading to packet loss. This observation suggests that we possibly have to expect different prediction accuracy depending the day and time when prediction is applied.

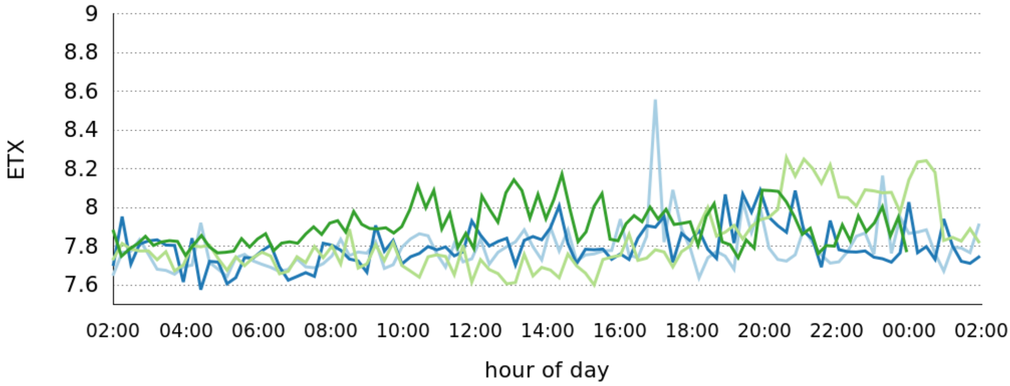


Figure 4.3: Temporal evolution of the average ETX of paths.

To analyze the relation between path Hop-count and its corresponding ETX, Figure 4.4 plots the frequency of each of these values. We can observe that Hops and ETX overlap between the values 1 and 4. That means that the quality of short paths is maximum. However, for values 5 or higher, ETX path values present deviations that can be translated as decreased EtEQ and a large number of packet retransmissions needed. Notice that the more frequent values of Hops and ETX (more than 5% each) are in the range from 5 to 10.

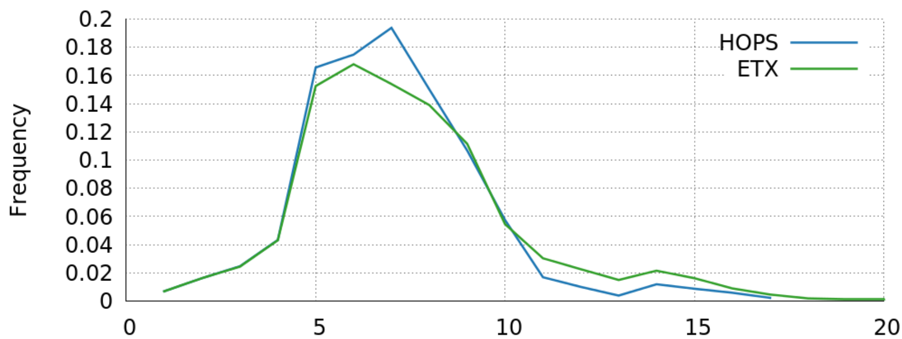


Figure 4.4: Frequency of path lengths based on hop count and ETX.

In our last study, we analyze the dispersion of ETX according to the number of Hops. In this way, Figure 4.5 contains the boxplots of ETX, for individual number of Hops. The ETX values are normalized with the corresponding number of Hops to provide a better idea of the scale of the deviation, taking into account that ETX deviation is not correlated with the number of Hops. We can see that ETX dispersion is minimal for the range from 1 to 4 Hops, it is significant for range 5 to

9 Hops, and relevant for 10 to 13 Hops. For 14 Hops and above, ETX dispersion is minimal again. Therefore, we can conclude that the prediction would be more or less accurate depending on the number of Hops.

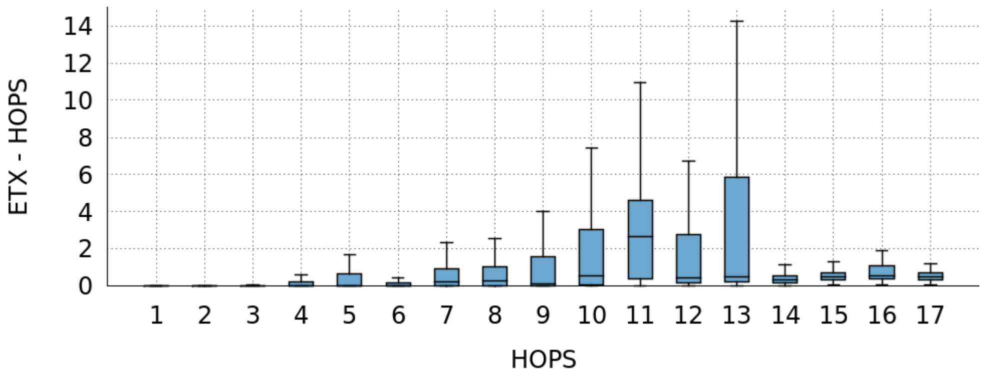


Figure 4.5: Distribution of ETX of paths versus the number of Hops, normalized by corresponding number of Hops.

From the above preliminary observations and data-set studies, we reached the conclusion that from a node's ego-network point-of-view there is a non-negligible amount of unstable paths for which the ETX path value shows a significant deviation from the optimal value. Therefore, investigating the EtEQ behavior as well as finding efficient EtEQ prediction strategies based on Time-series analysis will result to valuable input for routing in WMCNs.

4.3.2 Comparison of learning algorithms based on time series

As stated before, we want to explore whether Time-series analysis can be used to predict future End-to-End Quality (EtEQ) values. To do this, we applied four well-known approaches: SVM, kNN, RT, and RBR.

Figure 4.6 shows the average Mean Absolute Error (MAE) per path using a training data-set of 2016 instances (7 days), a test data-set of 288 instances (1 day), and a lag window composed of the last 12 instances. This test was performed to verify whether Time-series learning-algorithms could predict consecutive EtEQ values. These results show that we achieved the best accuracy for the Rule-Based Regression (RBR), and the worst for k-Nearest Neighbours (kNN). Regression Tree (RT) and Support Vector Machine (SVM) also moves very close to RBR results. Notice that the maximum EtEQ value is 1 and therefore, the MAE per

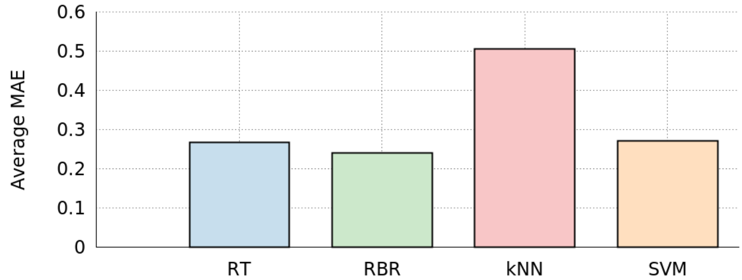


Figure 4.6: Average Mean Absolute Error (MAE) of the paths.

link is 2.4% for RBR and 5% for kNN. We applied a T-test to mean values for independent samples (at 95% confidence level) in order to compare the classification algorithms using the MAE. After this analysis, p-values smaller than 0.05 indicate that the means are significantly different, and therefore, we would reject the null hypothesis of no difference between the means. Consequently, we can claim that Time-series analysis achieves high percentages of success, and that among them, RBR seems to be the best candidate to make predictions.

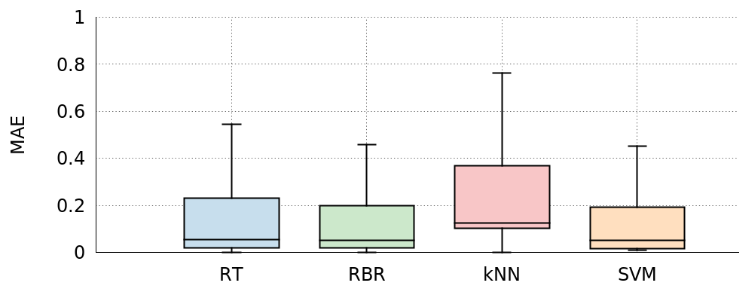


Figure 4.7: Mean Absolute Error (MAE) of the EtEQ predictions as boxplot.

We also analyzed the error variability of each algorithm and represented the results using boxplots. Three of the four algorithms achieved similar performance for most of the links (RT, RBR, and SVM), as shown in Figure 4.7. Although, RT may present some outliers, the differences among median, 1st quartile, and 3rd quartile are minimal. On the other side, kNN presents different behavior compared to the others. In this case, outliers present larger errors that increase the average values and change the overall evaluation of the algorithm. In the rest of the chapter, we assume the RBR algorithm to show the potential benefits of predicting EtEQ by means of a Time-series analysis.

4.3.3 EtEQ Prediction with Rule-Based Regression

We proceed next to analyze more in depth the EtEQ using the RBR algorithm, in order to discover how can we reach a satisfactory level of prediction.

Figure 4.8 presents boxplots of the MAE of path ETX prediction with RBR, versus the number of Hops corresponding to the paths. We used the same training data set as subsection 4.3.2. Even though the dispersion of the error seems high in specific cases (for example 10, 11, and 13 hops), the results follow the dispersion pattern of actual ETX values as described in figure 4.5, though in a much smaller scale. In other words, even though ETX values for 10, 11, and 13 hop paths have a high dispersion, our prediction manages to successfully predict a big percentage of the fluctuations. For instance, the dispersion of 13 hop path-ETX is 6, while the error of its prediction has a maximum value of 3. For the rest of the paths, the MAE has maximum value less than 1, resulting to a meaningful prediction.

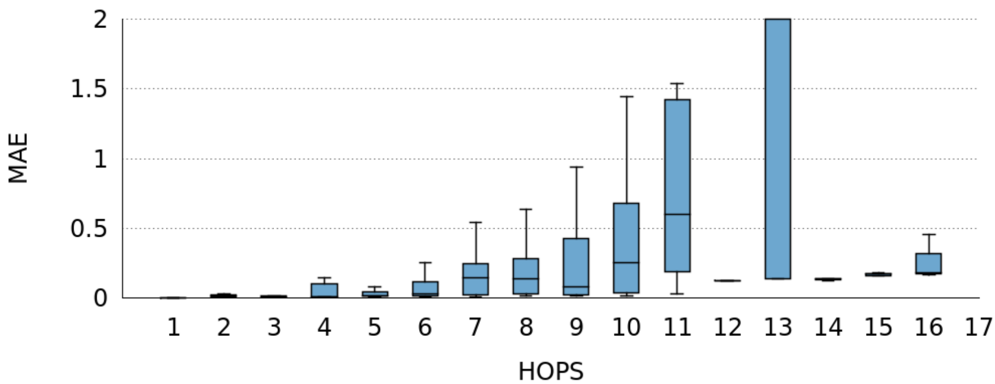


Figure 4.8: Distribution of RBR Mean Absolute Error as boxplot.

Figure 4.9 provides a more detailed analysis of the prediction accuracy. We can see that the average ETX value and the average prediction value are very close, even overlapping during the first half of prediction test. A better estimation for the deviation of the individual path values is given by the average absolute-error line. Notice that the deviation remains less than 0.5 throughout the whole prediction, which is a great achievement. Nevertheless, the potential impact of this small error in routing decisions can be further studied.

Another characteristic of the prediction revealed by Figure 4.9 is that after 100 steps of prediction (between 8 a.m. and 8:30 a.m.) the absolute error presents an increasing trend. We assume that this outcome is an effect of the actual ETX

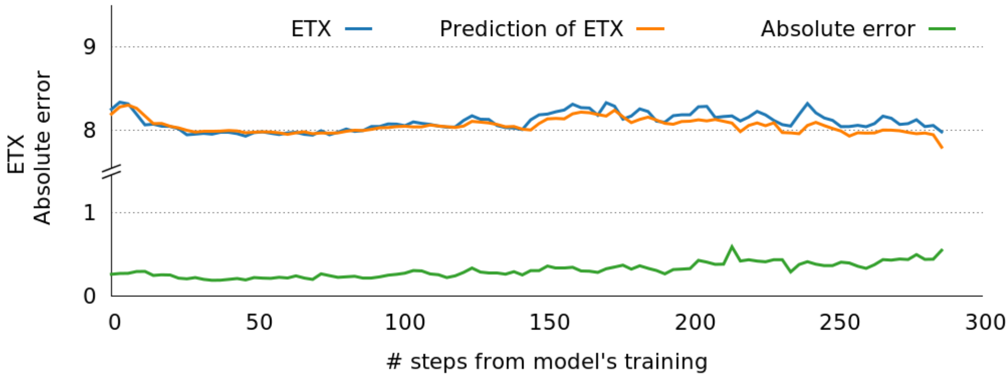


Figure 4.9: Evolution of the average ETX, the average prediction and the average absolute error.

oscillations depicted in Figure 4.3, and not the result of a possible degradation by time of the prediction accuracy. In order to verify this assumption we performed two more prediction tests. From 12 a.m. to 12 p.m. (Figure 4.10.a) and from 12 p.m. to 12 a.m. (figure 4.10.b), using as training data-set the 2016 more recent instances (7 days before prediction starts), as test data-set 144 instances (half a day), and a lag window composed of the last 12 instances. The results obtained are almost identical to the results of Figure 4.9, leading to the conclusion that the ETX oscillations are indeed affecting the prediction. This opens a new research line to explore how accuracy could be increased by deploying two different predictors, for day and night.

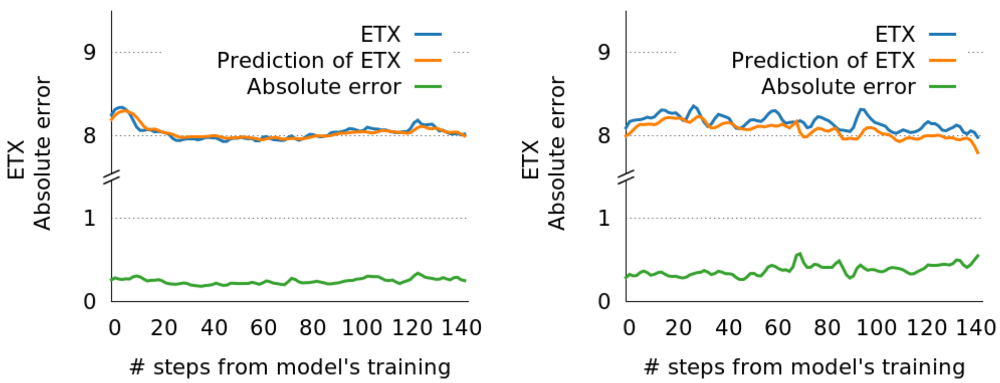


Figure 4.10: Evolution of the average ETX, the average prediction and the average absolute error for: a)12 am-12 pm, b)12 pm-12 am

4.3.4 Prediction of Some Steps Ahead

This analysis was performed to explore if Time-series analysis and prediction can be used to predict the value of EtEQ some time-steps ahead into the future.

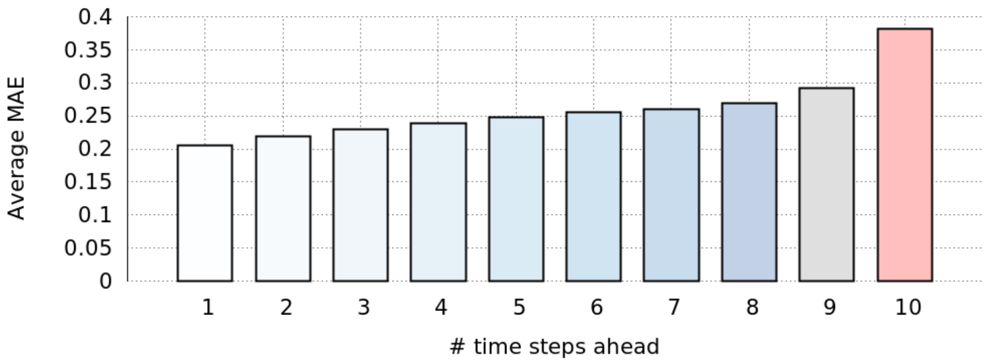


Figure 4.11: RBR average MAE of the EtEQ predictions.

Figure 4.11 shows the average MAE of paths. It shows the results of the RBR algorithm using the same setup that the baseline experiment (a lag window size of 12 instances, a training data-set of 2016 instances, and a test data-set of 288 instances), and then predicting from 1 to 10 time steps into the future. The results obtained were good for the majority of the tests. As we can observe, the average MAE grows very slowly. It seems possible to conclude that we could successfully predict the EtEQ several steps ahead in time.

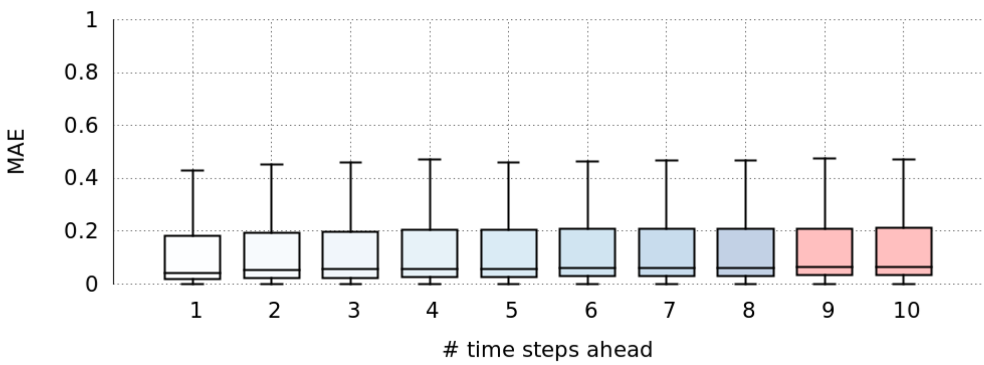


Figure 4.12: RBR MAE of EtEQ predictions, depicted as boxplots.

Once more, we analyzed the variability of errors for each number of steps ahead using a boxplot, shown in Figure 4.12. Although the values for the median and the first quartile are similar for all steps ahead considered, the values of third quartile and outliers (not depicted) grow with the number of steps. These differences in the variability of errors lead to the differences in the average Mean Absolute Error (MAE).

4.4 Conclusions

This chapter has been devoted to answer *SRQ2: What impact does prediction have on the quality of ad hoc networks?*, with regard to End-to-End Quality (EtEQ) prediction. We have demonstrated that Time-series analysis is a promising approach to accurately predict EtEQ values in Community Networks. This technique can be used to improve the performance of the routing protocol, by providing information to make appropriate and timely decisions, to maximize the delivery rate and minimize traffic congestion.

The data-set we used in our analysis shows quite significant fluctuations (about 7%) in the temporal evolution of the number of paths (routing entries). This contrasts with the general assumption in mesh networks that the number of paths is stable. The persistence of paths in our data-set is very stable for 90% of paths, but there are 10% of paths with persistence values below 60%, and those paths create the temporal fluctuations in the number of paths. Our study includes paths with persistence above 25%.

Regarding the Expected Transmission Count (ETX) behavior, the average ETX Path Quality is more stable at night, and presents more fluctuations during the morning. In non-working days, this behavior is less apparent. We assume this is caused by the variation of network traffic and interference during the day, leading to packet loss. Hence, the accuracy of prediction depends on the day and time when it is applied.

On the other side, the most frequent number of Hops and ETX values are in the range from 5 to 10. The dispersion of ETX according to the number of Hops is minimal between 1 and 4 Hops, and for more than 14 Hops. However, the dispersion is significant between 5 and 9 Hops, and relevant for 10 to 13 Hops. Therefore, the accuracy of the predictions would be better or worse depending on the number of Hops.

We have presented results from four well known learning algorithms that model Time series: Support Vector Machine (SVM), k-Nearest Neighbours (kNN), Regression Tree (RT), and Rule-Based Regression (RBR). All of them achieved high percentages of success, with average Mean Absolute Error values per link between 2.4% and 5% when predicting the next value of the EtEQ.

We also analyzed the error variability and found that three of the learning algorithms presented similar performance (RT, RBR, and SVM), whereas kNN performs worse due to outliers with larger errors. A more detailed study of RBR prediction shows an average absolute-error less than 1. We have also observed differences in the prediction behavior during day and during night, as it happens with actual ETX values.

Chapter 5

Conclusions & Future Work

Prediction is a technique that has been widely used in many research fields. In this thesis we investigated on the usefulness and benefits of applying prediction to ad hoc networks. More specifically, we focused on prediction techniques applied to the routing layer of ad hoc networks. We have shown that prediction techniques can help the routing layer to quickly react to the changes that happen, and even to be prepared in advance to the near future. This allows ad hoc networks to have a proactive approach, which is more efficient than a reactive approach. The overall contribution of this thesis is the answer to the general research question about *What is the improvement achieved when we apply prediction to the routing layer of ad hoc networks?*

5.1 Overview of Contributions

We determined what is the improvement achieved when prediction is applied to the routing layer of ad hoc networks. In order for the routing layer to do its job (to decide what is the best path for an information to reach its destination) it is necessary to know what paths or links are available between the network nodes (Topological Information) and how good these paths or links are (Quality). In our effort to address the general research problem, we decomposed it into two major research questions, based on the above two key aspects of the information used by the routing layer. In one side, we determined *what impact does prediction have on the topological information used by the routing layer (SRQ1)*. On the other side, we determined *what impact does prediction have on the quality of ad hoc networks (SRQ2)*.

Concerning the prediction of Topology Control Information (TCI), we applied a History-Based approach in order to analyze its prediction potential. The results of these analysis demonstrated that a History-Based Predictor (HBP) strategy is useful to predict the TCI generated by routing protocols for mobile ad hoc and opportunistic networks (first contribution). The upper-bound limits of the HBP strategy remain high for a wide variety of interaction scenarios. There is a high opportunity for predicting the TCI, and this prediction can be just focused on a small subset of messages. Then, we evaluated this analytical results with the implementation of the HBP predictor over the Optimized Link State Routing protocol (OLSR-HBP). This is the second contribution. OLSR-HBP achieves important decreases of TCI (signaling overhead), without disturbing the network operation, and requiring a small and affordable amount of resources. Our approach is deterministic, and with much lower resource requirements, compared to other statistical proposals (as Machine or Deep Learning [27]).

Regarding the impact of prediction on the routing data for both Link and Path (or End-to-End) Quality information, we have demonstrated that Time-series analysis is a promising approach to accurately predict both Link and End-to-End Quality in Community Networks (third contribution). This technique can be used to improve the performance of the routing protocol, by providing information to make appropriate and timely decisions to maximize the delivery rate and minimize traffic congestion.

Overall, we are confident that these three main contributions provide a satisfactory answer to our two specific research questions. We believe that we have paved the way to apply prediction techniques to improve the operation of the routing layer of ad hoc networks.

5.2 Application to Other Environments

The aim of this thesis was to assess the impact of prediction on ad hoc networks. Our main focus was on the prediction of the information used by the routing layer of these networks.

In order to perform our research, we analyzed the potential of the prediction techniques in diverse scenarios, and we carried out a number of simulations to evaluate the impact of these prediction techniques. Furthermore, we are currently implementing our proposal, that will allow us to determine the computational cost of the predictions, how to be used by routing protocols, and if the addition of other sources of information (i.e. NIC parameters) could improve the predictions.

The studies we performed on Community Networks (CNs) and the results we obtained, are based on the Optimized Link State Routing (OLSR) protocol. This is not a specific protocol just for CNs, but a generalistic protocol. For this reason, we firmly believe that our results and conclusions are extensible and generalizable to other types of ad hoc networks. All the evidences we gather about CNs are sufficiently general to be applicable to other types of ad hoc networks, according to the experience that we achieved in other phases of this thesis.

5.3 Future Research Directions

We have applied prediction to the routing layer of ad hoc networks and we have determined its benefits and its impact on the performance of such networks. Prediction is still a very promising approach and it can be further applied in the routing layer. Next we are going to enumerate some topics and ideas that deserve future work.

Regarding our findings on TCI prediction, we identified 3 possible new research lines.

- (1) The reduction of the amount of TCI transmitted through the network, opens an opportunity to decrease the routing parameters values, in order to improve and to sooner detect changes in the network topology, particularly in high mobility scenarios. The extra TCI generated due to the shorten of the period of TCI transmissions, could be assumed by the reduction in transmitted TCI because of prediction. But further analysis would be required, to assess the effect of this parameter tuning.

- (2) To investigate what is the trade-off between the number of hits and the number of misses. That is, is it better to increase the percentage of hits at the cost of higher number of misses (when not applying confidence)?, or would it be better to decrease the percentage of misses, but also the percentage of hits (when applying confidence)? In a real-world scenario, this will depend on the time that benefits from hits, the additional time spent with misses, and the time consumed when no prediction is made because of the lack of confidence.

- And (3) the analysis of the prediction performance in ad hoc and opportunistic networks involving heterogeneous environments. Addressing these scenarios will allow developers to deal with the stated concerns in IoT-based solutions.

As future work from our findings about Link and Path Quality prediction:

- We plan to identify which Links and Paths contribute most to the error in the Quality prediction and to understand what factors make it more difficult to predict the behaviour of these links and paths.
- We also want to analyze if there is a subset of Links and Paths that provides real trends in Link and Path Quality. We plan to identify which Links and Paths contribute most to the errors in the Quality prediction, and to understand what factors make it more difficult to predict them.
- Moreover, we believe that social analysis and user profiles may help to understand Link and Path Quality behaviour and its prediction.
- We also want to study the impact of errors in routing decisions.
- A new research line is open to explore how accuracy could be increased by deploying two different predictors, for day and night.
- On the other hand, we plan to improve the prediction process by discarding those links and paths whose relation between Quality values and prediction accuracy is above a certain threshold. This discarding can be done at the very beginning (static discarding) or at any given time (dynamic discarding).
- Finally, from our findings about Path/End-to-End Quality prediction, we want to extend this analysis to other Community Networks [7, 9] to evaluate if the observed behaviour could be generalized.

Chapter 6

References

1. Aggarwal, C. C. & Abdelzaher, T. in *Managing and Mining Sensor Data* 237–297 (Springer US, Boston, MA, 2013). ISBN: 978-1-4614-6309-2. doi:10.1007/978-1-4614-6309-2_9 (cit. on p. 11).
2. AlOrabi, W. A., Rahman, S. A., El Barachi, M. & Mourad, A. Towards on demand road condition monitoring using mobile phone sensing as a service. *Procedia computer science* **83,** 345–352. ISSN: 1877-0509 (2016) (cit. on p. 12).
3. Alsheikh, M. A., Lin, S., Niyato, D. & Tan, H. P. Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications. *IEEE Communications Surveys Tutorials* **16.** Fourth Quarter, 1996–2018. ISSN: 1553-877X (2014) (cit. on pp. 53, 84).
4. Arroyo-Valles, R., Alaiz-Rodriguez, R., Guerrero-Curieses, A. & Cid-Sueiro, J. *Q-Probabilistic Routing in Wireless Sensor Networks* in *Intelligent Sensors, Sensor Networks and Information (ISSNIP)* (2007), 1–6. doi:10.1109/ISSNIP.2007.4496810 (cit. on pp. 14, 44).
5. Aschenbruck, N., Ernst, R., Gerhards-Padilla, E. & Schwamborn, M. *BonnMotion: A Mobility Scenario Generation and Analysis Tool* in *ICST Conference on Simulation Tools and Techniques (SIMUTools)* (2010), 51:1–51:10. ISBN: 978-963-9799-87-5. doi:10.4108/ICST.SIMUTOOLS2010.8684 (cit. on p. 16).
6. Avonts, J., Braem, B. & Blondia, C. *A questionnaire based examination of community networks* in *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (Oct. 2013), 8–15. doi:10.1109/WiMOB.2013.6673333 (cit. on pp. 2, 47, 77).
7. AWMN. *Athens Wireless Metropolitan Network* (accessed February 23, 2018) (2010). http://www.awmn.net/ (cit. on p. 98).

8. Baccour, N., Koubâa, A., Mottola, L., Zúñiga, M. A., Youssef, H., Boano, C. A. & Alves, M. Radio Link Quality Estimation in Wireless Sensor Networks: A Survey. *ACM Trans. Sen. Netw.* **8,** 34:1–34:33. ISSN: 1550-4859 (Sept. 2012) (cit. on pp. 14, 48, 78).

9. Baig, R., Roca, R., Navarro, L. & Freitag, F. *guifi.net: A Network Infrastructure Commons* in *Proceedings of the Seventh International Conference on Information and Communication Technologies and Development* (ACM, Singapore, Singapore, 2015), 27:1–27:4. ISBN: 978-1-4503-3163-0. doi:10.1145/2737856.2737900 (cit. on pp. 2, 48, 77, 98).

10. Banerjee, S. & Misra, A. *Minimum Energy Paths for Reliable Communication in Multi-hop Wireless Networks* in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing* (ACM, Lausanne, Switzerland, 2002), 146–156. ISBN: 1-58113-501-7. doi:10.1145/513800.513818. http://doi.acm.org/10.1145/513800.513818 (cit. on p. 79).

11. Bote-Lorenzo, M. L., Gómez-Sánchez, E., Mediavilla-Pastor, C. & Asensio-Pérez, J. I. Online machine learning algorithms to predict link quality in community wireless mesh networks. *Computer Networks* **132,** 68–80 (2018) (cit. on p. 13).

12. Braem, B., Blondia, C., Barz, C., Rogge, H., Freitag, F., Navarro, L., Bonicioli, J., Papathanasiou, S., Escrich, P., Baig Viñas, R., Kaplan, A. L., Neumann, A., Vilata i Balaguer, I., Tatum, B. & Matson, M. A Case for Research with and on Community Networks. *SIGCOMM Comput. Commun. Rev.* **43,** 68–73. ISSN: 0146-4833 (July 2013) (cit. on pp. 48, 78).

13. Camp, T., Boleng, J. & Davies, V. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing* **2,** 483–502. ISSN: 1530-8677 (2002) (cit. on p. 16).

14. Cerotti, D., Distefano, S., Merlino, G. & Puliafito, A. A Crowd-Cooperative Approach for Intelligent Transportation Systems. English. *IEEE Transactions on Intelligent Transportation Systems* **18,** 1529–1539. ISSN: 15249050 (2017) (cit. on p. 12).

15. Cerpa, A., Wong, J. L., Potkonjak, M. & Estrin, D. *Temporal Properties of Low Power Wireless Links: Modeling and Implications on Multi-hop Routing* in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing* (ACM, Urbana-Champaign, IL, USA, 2005), 414–425. ISBN: 1-59593-004-3. doi:10.1145/1062689.1062741 (cit. on pp. 14, 48, 49, 78).

16. Chan, F. K.-P., Fu, A. W.-C. & Yu, C. Haar wavelets for efficient similarity search of time-series: with and without time warping. *IEEE Transactions on Knowledge and Data Engineering* **15,** 686–705. ISSN: 1041-4347 (May 2003) (cit. on p. 66).

17. Chang, S. E. & Shen, W.-C. Exploring smartphone social networking services for mobile tourism. *International Journal of Mobile Communications* **16,** 63–81 (2018) (cit. on p. 12).

18. Chaovalit, P., Gangopadhyay, A., Karabatis, G. & Chen, Z. Discrete Wavelet Transform-based Time Series Analysis and Mining. *ACM Comput. Surv.* **43,** 6:1–6:37. ISSN: 0360-0300 (Feb. 2011) (cit. on p. 66).

19. Chen, I. & Verma, N. *Simulation Study of a Class of Autonomous Host-Centric Mobility Prediction Algorithms for Wireless Cellular and Ad Hoc Networks* in *Annual Simulation Symposium (ANSS), Orlando, Florida, March 30 - April 2, 2003* (2003), 65–72. doi:10.1109/SIMSYM.2003.1192799 (cit. on pp. 13, 14).

20. Clausen, T. & Jacquet, P. *Optimized Link State Routing Protocol (OLSR)* RFC 3626. Oct. 2003. doi:10.17487/RFC3626 (cit. on pp. 12, 21, 23).

21. CONFINE Project. *FunkFeuer - CONFINE Open Data* (accessed February 19, 2017) (2013). http://opendata.confine-project.eu/group/funkfeuer (cit. on pp. 51, 81).

22. Cunha, I., Teixeira, R., Veitch, D. & Diot, C. Predicting and Tracking Internet Path Changes. *SIGCOMM Comput. Commun. Rev.* **41,** 122–133. ISSN: 0146-4833 (Aug. 2011) (cit. on pp. 79, 80).

23. De Couto, D. S. J., Aguayo, D., Bicket, J. & Morris, R. *A High-throughput Path Metric for Multi-hop Wireless Routing* in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking* (ACM, San Diego, CA, USA, 2003), 134–146. ISBN: 1-58113-753-2. doi:10.1145/938985.939000 (cit. on pp. 48, 49, 52, 78, 82).

24. Dipobagio, M. *An Overview on Ad Hoc Networks* in (Institute of Computer Science (ICS), Freie Universität Berlin, 2009). https://www.mi.fu-berlin.de/inf/groups/ag-tech/teaching/2008-09_WS/S_19565_Proseminar_Technische_Informatik/dipobagio09overview.pdf (cit. on p. 2).

25. Dong, S., Agrawal, P. & Sivalingam, K. *Reinforcement Learning Based Geographic Routing Protocol for UWB Wireless Sensor Network* in *the Global Communications Conference (GLOBECOM)* (Nov. 2007), 652–656. doi:10.1109/GLOCOM.2007.127 (cit. on pp. 14, 44).

26. Draves, R., Padhye, J. & Zill, B. Comparison of Routing Metrics for Static Multi-hop Wireless Networks. *SIGCOMM Comput. Commun. Rev.* **34,** 133–144. ISSN: 0146-4833 (Aug. 2004) (cit. on pp. 14, 48, 78).

27. Fadlullah, Z. M., Tang, F., Mao, B., Kato, N., Akashi, O., Inoue, T. & Mizutani, K. State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow's Intelligent Network Traffic Control Systems. *IEEE Communications Surveys and Tutorials* **19,** 2432–2455 (2017) (cit. on pp. 14, 24, 25, 44, 96).

28. Fonseca, R., Gnawali, O., Jamieson, K. & Levis, P. *Four-Bit Wireless Link Estimation* in *Proceedings of the Sixth ACM Workshop on Hot Topics in Networks (HotNets-VI)* (Oct. 2007). `http://conferences.sigcomm.org/hotnets/2007/papers/hotnets6-final131.pdf` (cit. on pp. 14, 48, 78).

29. Forster, A. & Murphy, A. L. *FROMS: Feedback Routing for Optimizing Multiple Sinks in WSN with Reinforcement Learning* in *Intelligent Sensors, Sensor Networks and Information (ISSNIP)* (2007), 371–376. doi:`10.1109/ISSNIP.2007.4496872` (cit. on p. 14).

30. FunkFeuer Wien. *FunkFeuer, a free, experimental network in Vienna* (accessed February 19, 2017) (2003). `https://www.funkfeuer.at/` (cit. on pp. 2, 48, 50, 51, 77, 81).

31. Gomez, C., Garcia, D. & Paradells, J. *An OLSR parameter based study of the performance of real ad-hoc network environments* in *European Wireless Conference - Next Generation wireless and Mobile Communications and Services* (2005), 1–6 (cit. on p. 13).

32. Gomez, C., Garcia, D. & Paradells, J. *Improving performance of a real ad-hoc network by tuning OLSR parameters* in *IEEE Symposium on Computers and Communications (ISCC)* (June 2005), 16–21. doi:`10.1109/ISCC.2005.83` (cit. on p. 13).

33. González-Reverté, F., Díaz-Luque, P., Gomis-López, J. M. & Morales-Pérez, S. Tourists' Risk Perception and the Use of Mobile Devices in Beach Tourism Destinations. *Sustainability* **10.** 2018. ISSN: 2071-1050. doi:`10.3390/su10020413` (2018) (cit. on p. 12).

34. Hadjiefthymiades, S. & Merakos, L. Proxies + Path Prediction: Improving Web Service Provision in Wireless–Mobile Communications. *Mobile Networks and Applications* **8,** 389–399. ISSN: 1572-8153 (Aug. 2003) (cit. on p. 13).

35. Hagan, M., Zhang, N. & Kaye, J. '. *Safe Mathare: A Mobile System for Women's Safe Commutes in the Slums* in *Human-computer Interaction with Mobile Devices and Services Companion (MobileHCI)* (2012), 47–52. ISBN: 978-1-4503-1443-5. doi:`10.1145/2371664.2371675` (cit. on p. 12).

36. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I. H. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.* **11,** 10–18. ISSN: 1931-0145 (Nov. 2009) (cit. on pp. 54, 85).

37. Hardy, A., Hyslop, S., Booth, K., Robards, B., Aryal, J., Gretzel, U. & Eccleston, R. Tracking tourists' travel with smartphone-based GPS technology: a methodological discussion. *Information Technology & Tourism* **17,** 255–274. ISSN: 1943-4294 (Sept. 2017) (cit. on p. 12).

38. Härri, J., Filali, F. & Bonnet, C. *Kinetic Multipoint Relaying: Improvements Using Mobility Predictions* in *IFIP TC6 Active and Programmable Networks (IWAN)* (2005), 224–229. doi:10.1007/978-3-642-00972-3_22 (cit. on p. 14).

39. Härri, J., Filali, F. & Bonnet, C. *Kinetic Multipoint Relaying: Improvements Using Mobility Predictions* in *Active and Programmable Networks* (eds Hutchison, D., Denazis, S., Lefevre, L. & Minden, G. J.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2009), 224–229. ISBN: 978-3-642-00972-3. doi:10.1007/978-3-642-00972-3_22 (cit. on p. 78).

40. Hiyama, M., Kulla, E., Ikeda, M., Barolli, L. & Takizawa, M. *Investigation of OLSR Behavior for Different Hello Packets Intervals in a MANET Testbed* in *International Conference on Advanced Information Networking and Applications (AINA)* (2013), 183–188. doi:10.1109/AINA.2013.141 (cit. on p. 13).

41. (ITU), I. T. U. *Trends in Telecommunication Reform 2008: Six Degrees of Sharing* (accessed May 7, 2018) (2008). http://handle.itu.int/11.1002/pub/807b3dbf-en (cit. on p. 2).

42. Jakllari, G., Eidenbenz, S., Hengartner, N., Krishnamurthy, S. V. & Faloutsos, M. *Link Positions Matter: A Noncommutative Routing Metric for Wireless Mesh Network* in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications* (Apr. 2008), 744–752. doi:10.1109/INFOCOM.2008.125 (cit. on p. 79).

43. Kato, N., Fadlullah, Z. M., Mao, B., Tang, F., Akashi, O., Inoue, T. & Mizutani, K. The Deep Learning Vision for Heterogeneous Network Traffic Control: Proposal, Challenges, and Future Perspective. *IEEE Wireless Communications* **24,** 146–153. ISSN: 1536-1284 (2016) (cit. on pp. 14, 44).

44. Kim, D., Garcia-Luna-Aceves, J. J., Obraczka, K., Cano, J. C. & Manzoni, P. Routing mechanisms for mobile ad hoc networks based on the energy drain rate. *IEEE Transactions on Mobile Computing* **2,** 161–173. ISSN: 1536-1233 (Apr. 2003) (cit. on p. 78).

45. Koksal, C. E. & Balakrishnan, H. Quality-Aware Routing Metrics for Time-Varying Wireless Mesh Networks. *IEEE Journal on Selected Areas in Communications* **24,** 1984–1994. ISSN: 0733-8716 (Nov. 2006) (cit. on pp. 13, 14, 48, 78).

46. Law, R., Chan, I. C. C. & Wang, L. A comprehensive review of mobile technology use in hospitality and tourism. *Journal of Hospitality Marketing & Management* **27,** 626–648 (2018) (cit. on p. 12).

47. Lee, K., Hong, S., Kim, S. J., Rhee, I. & Chong, S. *SLAW: A New Mobility Model for Human Walks* in *IEEE INFOCOM 2009* (Apr. 2009), 855–863. doi:10.1109/INFCOM.2009.5061995 (cit. on p. 16).

48. Li, H., Cheng, Y., Zhou, C. & Zhuang, W. *Minimizing End-to-End Delay: A Novel Routing Metric for Multi-Radio Wireless Mesh Networks* in *IEEE INFOCOM 2009* (Apr. 2009), 46–54. doi:10.1109/INFCOM.2009.5061905 (cit. on p. 79).

49. Liao, T. W. Clustering of time series data—a survey. *Pattern Recognition* **38,** 1857–1874. ISSN: 0031-3203 (2005) (cit. on p. 66).

50. Lou, W. & Lu, H. *Efficient Prediction of Web Accesses on a Proxy Server* in *Proceedings of the Eleventh International Conference on Information and Knowledge Management* (ACM, McLean, Virginia, USA, 2002), 169–176. ISBN: 1-58113-492-4. doi:10.1145/584792.584823 (cit. on p. 13).

51. Lü, L. & Zhou, T. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications* **390,** 1150–1170. ISSN: 0378-4371 (2011) (cit. on pp. 14, 48, 78).

52. Luo, Y., Wang, J. & Chen, S. *An energy-efficient DSR routing protocol based on mobility prediction* in *2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)* 3 pp.-446 (2006). doi:10.1109/WOWMOM.2006.25 (cit. on p. 78).

53. Ma, Y. *Improving wireless link delivery ratio classification with packet SNR* in *2005 IEEE International Conference on Electro Information Technology* (May 2005), 6. doi:10.1109/EIT.2005.1626960 (cit. on pp. 48, 49, 78).

54. Maccari, L. & Cigno, R. L. A week in the life of three large Wireless Community Networks. *Ad Hoc Networks* **24.** Modeling and Performance Evaluation of Wireless Ad-Hoc Networks, 175–190. ISSN: 1570-8705 (2015) (cit. on pp. 51, 79, 80).

55. Makker, S. & Rathy, R. Web Server Performance Optimization using Prediction Prefetching Engine. *International Journal of Computer Applications (0975–8887)* **23,** 19–24 (June 2011) (cit. on p. 13).

56. Maleki, M., Dantu, K. & Pedram, M. *Lifetime prediction routing in mobile ad hoc networks* in *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.* **2** (Mar. 2003), 1185–1190. doi:10.1109/WCNC.2003.1200539 (cit. on p. 78).

57. Martínez, V., Berzal, F. & Talavera, J. C. C. A Survey of Link Prediction in Complex Networks. *ACM Comput. Surv.* **49,** 69:1–69:33 (2017) (cit. on p. 14).

58. Medina, E., Messeguer, R., Molina, C. & Royo, D. *OLSRp: Predicting Control Information to Achieve Scalability in OLSR Ad Hoc Networks* in *Mobile Networks and Management (MONAMI)* (2010), 225–236. doi:10.1007/978-3-642-21444-8_20 (cit. on pp. 12, 13).

59. Meseguer, R., Molina, C., Ochoa, S. F. & Santos, R. *Reducing energy consumption in human-centric wireless sensor networks* in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (Oct. 2012), 1473–1478. doi:10.1109/ICSMC.2012.6377943 (cit. on pp. 12–14, 78).

60. Meseguer, R., Molina, C., Ochoa, S. F. & Santos, R. Energy-Aware Topology Control Strategy for Human-Centric Wireless Sensor Networks. *Sensors* **14,** 2619–2643. ISSN: 1424-8220 (2014) (cit. on pp. 12, 13, 78).

61. Millán, P. *Improving Prediction in the Routing Layer of Wireless Networks Through Social Behaviour* in *2nd URV Doctoral Workshop in Computer Science and Mathematics* (eds Sánchez Artigas, M. & Valls Mateu, A.) (Publicacions URV, Tarragona, Nov. 2015), 5–8. ISBN: 978-84-8424-399-1 (cit. on p. vii).

62. Millán, P. *Real-Time Traffic-Scheduling in Underwater Acoustic Wireless Sensor Networks* in *3rd URV Doctoral Workshop in Computer Science and Mathematics* (eds Gómez, S. & Valls Mateu, A.) (Publicacions URV, Tarragona, Nov. 2016), 11–15. ISBN: 978-84-8424-495-0 (cit. on p. viii).

63. Millan, P., Molina, C., Dimogerontakis, E., Navarro, L., Meseguer, R., Braem, B. & Blondia, C. *Tracking and Predicting End-to-End Quality in Wireless Community Networks* in *2015 3rd International Conference on Future Internet of Things and Cloud* (Aug. 2015), 794–799. ISBN: 978-1-4673-8103-1. doi:10.1109/FiCloud.2015.96 (cit. on pp. vii, 9, 13, 14).

64. Millan, P., Molina, C., Medina, E., Vega, D., Meseguer, R., Braem, B. & Blondia, C. Time Series Analysis to Predict Link Quality of Wireless Community Networks. *Comput. Netw.* **93.** Impact Factor: 1.446, Q2, 342–358. ISSN: 1389-1286 (Dec. 2015) (cit. on pp. vii, 8, 13).

65. Millan, P., Molina, C., Medina, E., Vega, D., Meseguer, R., Braem, B. & Blondia, C. *Tracking and Predicting Link Quality in Wireless Community Networks* in *2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* CORE B (Oct. 2014), 239–244. ISBN: 978-1-4799-5041-6. doi:10.1109/WiMOB.2014.6962177 (cit. on pp. vii, 8, 13, 78–80).

66. Millán, P., Molina, C., Meseguer, R., Ochoa, S. F. & Santos, R. *Using a History-Based Approach to Predict Topology Control Information in Mobile Ad Hoc Networks* in *Internet and Distributed Computing Systems* (eds Fortino, G., Di Fatta, G., Li, W., Ochoa, S., Cuzzocrea, A. & Pathan, M.) (Springer International Publishing, Cham, 2014), 237–249. ISBN: 978-3-319-11692-1. doi:10.1007/978-3-319-11692-1_21 (cit. on pp. vii, 7).

67. Millan, P., Molina, C., Meseguer, R., Ochoa, S. F., Santos, R. & Medina, E. *OLSR-HBP: a History-Based Predictor for Topology Control Information in Mobile Ad Hoc Networks* **Submitted to** Wireless Communications and Mobile Computing, special issue on Topology Control in Emerging Mobile Networks, Hindawi/Wiley. ISSN: 1530-8677. doi:10.1155/6302 Impact Factor: 1.899, Q2. Oct. 2018 (cit. on pp. vii, 7).

68. Mizouni, R. & El Barachi, M. *Mobile phone sensing as a service: Business model and use cases* English. in *International Conference on Next Generation Mobile Applications, Services, and Technologies* (Prague, 2013), 116–121. ISBN: 9781479920198. doi:10.1109/NGMAST.2013.29 (cit. on p. 11).

69. Naimi, S., Busson, A., Vèque, V., Slama, L. B. H. & Bouallegue, R. *Anticipation of ETX Metric to Manage Mobility in Ad Hoc Wireless Networks* in *Ad-hoc, Mobile, and Wireless Networks* (eds Guo, S., Lloret, J., Manzoni, P. & Ruehrup, S.) (Springer International Publishing, Cham, 2014), 29–42. ISBN: 978-3-319-07425-2. doi:10.1007/978-3-319-07425-2_3 (cit. on pp. 48, 49, 78).

70. ns-3 project. *API of the ns-3 OLSR module* https://www.nsnam.org/doxygen/group__olsr.html (2018) (cit. on p. 22).

71. ns-3 project. *ns-3 Energy Models* https://www.nsnam.org/doxygen/group__energy.html (2018) (cit. on p. 41).

72. Passos, D. & Albuquerque, C. V. N. A Joint Approach to Routing Metrics and Rate Adaptation in Wireless Mesh Networks. *IEEE/ACM Transactions on Networking* **20,** 999–1009. ISSN: 1063-6692 (Aug. 2012) (cit. on p. 79).

73. Peffers, K., Tuunanen, T., Rothenberger, M. & Chatterjee, S. A Design Science Research Methodology for Information Systems Research. *J. Manage. Inf. Syst.* **24,** 45–77. ISSN: 0742-1222 (Dec. 2007) (cit. on p. 5).

74. Rahman, S. A., Mourad, A., Barachi, M. E. & Orabi, W. A. A novel on-demand vehicular sensing framework for traffic condition monitoring. *Vehicular Communications* **12,** 165–178. ISSN: 2214-2096 (2018) (cit. on p. 12).

75. Rault, T., Bouabdallah, A. & Challal, Y. Energy efficiency in wireless sensor networks: A top-down survey. *Computer Networks* **67,** 104–122. ISSN: 1389-1286 (2014) (cit. on pp. 48, 78).

76. Renner, C., Ernst, S., Weyer, C. & Turau, V. *Prediction Accuracy of Link-Quality Estimators* in *Wireless Sensor Networks* (eds Marrón, P. J. & Whitehouse, K.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2011), 1–16. ISBN: 978-3-642-19186-2. doi:10.1007/978-3-642-19186-2_1 (cit. on pp. 14, 48, 49, 78).

77. Rey-Moreno, C., Roro, Z., Tucker, W. D., Siya, M. J., Bidwell, N. J. & Simo-Reigadas, J. *Experiences, Challenges and Lessons from Rolling out a Rural WiFi Mesh Network* in *Proceedings of the 3rd ACM Symposium on Computing for Development* (ACM, Bangalore, India, 2013), 11:1–11:10. ISBN: 978-1-4503-1856-3. doi:10.1145/2442882.2442897 (cit. on p. 2).

78. Riley, G. F. & Henderson, T. R. in *Modeling and Tools for Network Simulation* 15–34 (Springer Berlin Heidelberg, 2010). doi:10.1007/978-3-642-12331-3_2 (cit. on p. 23).

79. Rodríguez-Covili, J., Ochoa, S. F., Pino, J. A., Messeguer, R., Medina, E. & Royo, D. A communication infrastructure to ease the development of mobile collaborative applications. *Journal of Network and Computer Applications* **34.** Control and Optimization over Wireless Networks, 1883–1893. ISSN: 1084-8045 (Nov. 2011) (cit. on pp. 48, 78).

80. Rosa, F. D., Malizia, A. & Mecella, M. Disconnection prediction in mobile ad hoc networks for supporting cooperative work. *IEEE Pervasive Computing* **4,** 62–70. ISSN: 1536-1268 (July 2005) (cit. on p. 13).

81. Santos, R., Orozco, J., Micheletto, M., Ochoa, S. F., Meseguer, R., Millan, P. & Molina, C. Real-Time Communication Support for Underwater Acoustic Sensor Networks. *Sensors* **17.** Impact Factor: 2.677, Q1. ISSN: 1424-8220. doi:10.3390/s17071629 (2017) (cit. on p. viii).

82. Santos, R., Orozco, J., Micheletto, M., Ochoa, S. F., Meseguer, R., Millan, P. & Molina, C. *Scheduling Real-Time Traffic in Underwater Acoustic Wireless Sensor Networks* in *Ubiquitous Computing and Ambient Intelligence* (eds García, C. R., Caballero-Gil, P., Burmester, M. & Quesada-Arencibia, A.) (Springer International Publishing, Cham, 2016), 150–162. ISBN: 978-3-319-48799-1. doi:10.1007/978-3-319-48799-1_19 (cit. on p. viii).

83. Senel, M., Chintalapudi, K., Lal, D., Keshavarzian, A. & Coyle, E. J. *A Kalman Filter Based Link Quality Estimation Scheme for Wireless Sensor Networks* in *IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference* (Nov. 2007), 875–880. doi:10.1109/GLOCOM.2007.169 (cit. on pp. 14, 48, 78).

84. Smith, J. E. *Retrospective: A Study of Branch Prediction Strategies* in *25 Years of the International Symposia on Computer Architecture (Selected Papers)* (ACM, Barcelona, Spain, 1998), 22–23. ISBN: 1-58113-058-9. doi:10.1145/285930.285940 (cit. on p. 13).

85. Spyropoulos, T., Rais, R. N. B., Turletti, T., Obraczka, K. & Vasilakos, A. Routing for disruption tolerant networks: taxonomy and design. *Wireless Networks* **16,** 2349–2370. ISSN: 1572-8196 (Nov. 2010) (cit. on p. 12).

86. Su, W., Lee, S.-J. & Gerla, M. Mobility prediction and routing in ad hoc wireless networks. *International Journal of Network Management* **11,** 3–30. ISSN: 1099-1190 (2001) (cit. on pp. 13, 14).

87. The European Parliament and the Council of the European Union. *Directive 2014/61/EU of the European Parliament and of the Council of 15 May 2014 on measures to reduce the cost of deploying high-speed electronic communications networks* (accessed May 7, 2018) (2014). http://ec.europa.eu/newsroom/dae/document.cfm?doc_id=7390 (cit. on p. 2).

88. Valera, A. & Tan, H. P. *Analysis of Hello-based link failure detection in wireless ad hoc networks* in *Personal, Indoor and Mobile Radio Communications (PIMRC)* (2012), 669–674. doi:10.1109/PIMRC.2012.6362868 (cit. on p. 13).

89. Vanderwiel, S. P. & Lilja, D. J. Data Prefetch Mechanisms. *ACM Comput. Surv.* **32,** 174–199. ISSN: 0360-0300 (June 2000) (cit. on p. 13).

90. Vasilakos, A. V., Zhang, Y. & Spyropoulos, T. *Delay Tolerant Networks: Protocols and Applications* 1st. ISBN: 9781439811085 (CRC Press, Inc., Boca Raton, FL, USA, 2011) (cit. on p. 12).

91. Vega, D., Baig, R., Cerdà-Alabern, L., Medina, E., Meseguer, R. & Navarro, L. A technological overview of the guifi.net community network. *Computer Networks* **93.** Community Networks, 260–278. ISSN: 1389-1286 (2015) (cit. on p. 2).

92. Vergara, C., Ochoa, S. F., Gutierrez, F. J. & Rodríguez-Covili, J. *Extending Social Networking Services toward a Physical Interaction Scenario* in *Ubiquitous Computing and Ambient Intelligence - (UCAmI)* (2012), 208–215. doi:10.1007/978-3-642-35377-2_29 (cit. on p. 16).

93. Vural, S., Wei, D. & Moessner, K. Survey of Experimental Evaluation Studies for Wireless Mesh Network Deployments in Urban Areas Towards Ubiquitous Internet. *IEEE Communications Surveys Tutorials* **15.** First Quarter, 223–239. ISSN: 1553-877X (2013) (cit. on pp. 14, 48, 78).

94. Wang, Y., Martonosi, M. & Peh, L.-S. *A New Scheme on Link Quality Prediction and Its Applications to Metric-based Routing* in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems* (ACM, San Diego, California, USA, 2005), 288–289. ISBN: 1-59593-054-X. doi:10.1145/1098918.1098958 (cit. on pp. 14, 49, 79).

95. Warden, P. *Smartphone Energy Consumption* https://petewarden.com/2015/10/08/smartphone-energy-consumption/ (2018) (cit. on p. 41).

96. Woo, A., Tong, T. & Culler, D. *Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks* in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems* (ACM, Los Angeles, California, USA, 2003), 14–27. ISBN: 1-58113-707-9. doi:10.1145/958491.958494 (cit. on pp. 14, 48, 49, 78).

97. Youssef, M., Ibrahim, M., Abdelatif, M., Chen, L. & Vasilakos, A. V. Routing Metrics of Cognitive Radio Networks: A Survey. *IEEE Communications Surveys Tutorials* **16.** First Quarter 2014, 92–109. ISSN: 1553-877X (cit. on p. 12).

98. Zeng, Y., Xiang, K., Li, D. & Vasilakos, A. V. Directional routing and scheduling for green vehicular delay tolerant networks. *Wireless Networks* **19,** 161–173. ISSN: 1572-8196 (Feb. 2013) (cit. on p. 12).

List of Figures

List of Tables

UNIVERSITAT
ROVIRA i VIRGILI