UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament d'Enginyeria Telemàtica

Contribution to the Improvement of the Performance of Wireless Mesh Networks Providing Real Time Services

by
Andrés Marcelo Vázquez Rodas

Ph.D. Advisor:
Dr. Luis J. de la Cruz Llopis

Thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Telematics
in the
Department of Telematics Engineering

Barcelona, December 2014

A mi hija Marcela,
a mi esposa Alexandra,
a mis padres,
a mis hermanos,
a mi Abuelita,
mi amor hacia ustedes es infinito.

Agradecimientos

En primer lugar quiero expresar mi más sincera gratitud a Luis, mi director de tesis, que desde mi primer instante en Barcelona fue el principal soporte, no solamente a nivel académico sino también a nivel personal. Su dedicación, comprensión, sus valiosas orientaciones y directrices han sido imprescindibles, han facilitado y hecho posible el desarrollo de este trabajo. Mi agradecimiento también al grupo de Servicios Telemáticos (SERTEL) del Departamento de Ingeniería Telemática, en especial a Mónica y Emilio. De la misma manera mi gratitud a la Prof. Isabelle Guérin Lassous por haber hecho posible mi estancia en el laboratorio LIP de la Ecole Normale Supérieure (ENS) de Lyon, Francia. Dicha experiencia fue muy enriquecedora.

Un agradecimiento especial a mis adoradas esposa Alexandra e hija Marcela, que sacrificando su comodidad y propios ideales han estado siempre a mi lado escuchándome, apoyándome y llenándome de amor. Gracias a ustedes los días difíciles se han hecho más llevaderos y la alegría de los momentos buenos ha resultado incomparablemente amplificada. Siempre estaremos juntos y estoy seguro que esta experiencia y travesía lejos de los nuestros nos han unido y solidificado más que cualquier otra.

Mi eterno agradecimiento a mis padres y hermanos que, como si fuera poco, a más de su amor, ejemplo y apoyo incondicional siempre me han dado más de lo que realmente merezco. Me resulta imposible encontrar las palabras que reflejen y expresen mi infinito amor y gratitud a ustedes. Además, una mención especial merece tu visión padre, que siempre creíste en mí y me has motivado constantemente, hasta el punto de convencerme que esto era factible.

A todos los familiares y amigos que desde la distancia nos han hecho llegar sus palabras de aliento y soporte. A mi compañero de departamento Luis, por su amistad y por las incontables tertulias técnicas que se han plasmado en publicaciones.

Finalmente agradecer a mi país Ecuador que a través de la Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT) ha financiado parte de mi investigación y estancia en Barcelona. De la misma manera al Ministerio de Educación, Cultura y Deporte de España que bajo el proyecto COPPI (TEC2011-26491) ha financiado mi estancia doctoral en Francia.

Abstract

In the present days, people expectations for ubiquitous connectivity is continuously growing. Modern cities are now moving towards the smart city paradigm. Electricity companies aims to become part of smart grids. Internet is no longer exclusive for humans, we now assume the Internet of everything. Under these premises, we consider that Wireless Mesh Network (WMN) technology has a set of valuable features (self-forming, self-configuration, robustness, resilience, self-healing, easy deployment and maintenance, among others) that will make it an important part of such challenging environments. Besides, WMNs can be use in less favored and rural areas thanks to their comparatively low-cost deployment. This is socially relevant since it facilitates the digital divide reduction and as consequence could help to improve the population quality of life. Research and industrial communities have been working these last years in open or proprietary mesh solutions. Standardization efforts and real deployments establish a solid starting point.

Within this framework, it is expected that WMNs constitute a supporting part for the emergence of an unlimited number of new applications and services from a variety of fields: community and neighborhood networking, intelligent transportation systems, health and medical systems, public safety and surveillance, disaster management and rescue operations, advanced metering, etc. For all these cases, the growing needs of users for real-time and multimedia information is currently evident. On the basis of these precedents, this thesis proposes a set of contributions to improve the performance of an application service of such type and to promote the better use of two critical resources (memory and energy) of WMNs.

In particular, for the offered service, this work focuses on a Video on Demand (VoD) system. One of the requirements of this system is the high capacity support. This is mainly achieved by distributing the video contents among various distribution points which in turn consist of several video servers. Each client request that arrives to such video server cluster must be handled by a specific server in a way that the load is balanced. For such task, this thesis proposes a mechanism to appropriately select a specific video server such that the transfer time at the cluster could be minimized. The proposal takes into account possible heterogeneous systems and it is scalable and easy to implement.

On the other hand, mesh routers that creates the mesh backbone, in the most practical situations, are equipped with multiple interfaces from different technologies and channel types (dedicated and shared). For such devices, a very important resource is the amount of memory intended for buffers. The quality of service perceived by the users are largely affected by the size of such buffers. This is because important network performance parameters such as packet loss probability, delay, and channel utilization are highly affected by the buffer sizes. Therefore, an efficient use of memory for buffering, in addition to facilitate the mesh devices scalability, also prevents the problems associated with excessively large buffers. Most of the current works associate the buffer sizing problem with the dynamics of TCP congestion control mechanism. Since this work focuses on real time services, in which the use of TCP is unfeasible, this thesis proposes a dynamic buffer sizing mechanism mainly dedicated for such real time flows. The approach is based on the maximum entropy principle and allows that each device be able to dynamically self-configure its buffers to achieve more efficient memory utilization. At the same time, the maximum admissible packet loss probability must be guaranteed. The proper performance of the proposal has been extensively evaluated in wired and wireless interfaces. Classical infrastructure-based wireless and multi-hop mesh interfaces have been considered.

Simulation results for different scenarios, traffic load conditions, channel considerations, etc. confirm the benefits of the proposed approach.

Finally, when the WMN is built by the interconnection of end-user hand-held devices, energy is a limited and scarce resource, and therefore any approach to optimize its use is valuable. For this case, this thesis proposes a topology control mechanism based on centrality metrics from social network analysis. The main idea is that, instead of having all the user devices executing routing functionalities, just a subset of nodes are selected for this task. To this end, we evaluate different centrality metrics, form both centralized and distributed perspectives. In addition to the commonly used random mobility models we include the analysis of the proposal with a socially-aware mobility model which takes into account human behavior and social information and generates networks with a clear community structure. Experimental results show the effectiveness of our mechanism since the energy consumption is considerably reduced without compromising the overall network performance.

Resumen

En la actualidad, las expectativas de las personas para acceder a una conectividad ubicua están creciendo continuamente. Las ciudades modernas están trabajando para alcanzar el paradigma de ciudades inteligentes. Las compañías eléctricas tienen como meta el llegar a ser parte de redes eléctricas inteligentes. Internet ha dejado de ser exclusivo de las personas y ahora asumimos el Internet de todo. Bajo estas premisas, se considera que las redes inalámbricas de malla (WMNs) poseen un valioso conjunto características (auto-creación, auto-configuración, robustez, fácil despliegue y mantenimiento, flexibilidad, entre otras) que las harán una parte importante de dichos entornos tan desafiantes. Además, las WMNs pueden utilizarse en zonas rurales y menos favorecidas debido a su despliegue comparativamente más económico. Esto es socialmente relevante ya que facilita la reducción de la brecha digital y como consecuencia puede ayudar a mejorar la calidad de vida de la población. En estos últimos años, centros de investigación y sectores industriales han estado trabajando en soluciones de redes de malla abiertas o propietarias. Los esfuerzos de estandarización y los despliegues de redes reales establecen un punto de partida sólido.

En este marco de referencia, se espera que las WMNs den soporte a la aparición de un número importante de nuevas aplicaciones y servicios, de una variedad de campos: redes de vecindario y comunitarias, sistemas de transporte inteligente, sistemas médicos y de salud, seguridad y vigilancia pública, operaciones de rescate y de emergencia, etc. En todos estos casos, es evidente la necesidad cada vez mayor que tienen los usuarios de disponer de información multimedia y en tiempo real. En base a estos precedentes, esta tesis propone un conjunto de contribuciones para mejorar el funcionamiento de un servicio de este tipo y promover un uso eficiente de dos recursos críticos de las WMN: memoria y energía.

En particular, para el servicio ofrecido, este trabajo se centra en un sistema de video bajo demanda. Uno de los requerimientos de estos sistemas es el de soportar capacidades elevadas. Esto se consigue principalmente, distribuyendo los contenidos de video entre diferentes puntos de distribución, los cuales a su vez están formados por varios servidores. Cada solicitud de un cliente que llega a dicho conjunto de servidores debe ser manejada por un servidor específico, de tal forma que la carga sea balanceada. Para esta tarea, esta tesis propone un mecanismo que selecciona apropiadamente un servidor de manera que el tiempo de transferencia del sistema pueda ser minimizado. La propuesta toma en cuenta la posible heterogeneidad del sistema y es escalable y fácil de implementar.

Por otra parte, en los casos más prácticos, los enrutadores de malla que crean la red troncal están equipados con múltiples interfaces de diferentes tecnologías y tipos de canal (dedicados y compartidos). Un recurso muy importante para estos dispositivos es la cantidad de memoria destinada a las colas de transmisión. La calidad de servicio percibida por los usuarios está altamente influenciada por el tamaño de dichas colas. Esto es debido a que parámetros importantes del rendimiento de la red como la probabilidad de pérdida de paquetes, el retardo, y la utilización del canal se ven muy afectados por el tamaño de las colas. Por lo tanto, un uso eficiente de la memoria de las colas, a más de facilitar la escalabilidad de los equipos, también previene de los problemas asociados a colas excesivamente largas. La mayoría de los trabajos actuales asocian el problema de dimensionamiento de las colas con la dinámica del mecanismo de control de congestión del TCP. Debido a que este trabajo se enfoca en servicios en tiempo real, en los cuales no es factible usar TCP, esta tesis propone un mecanismo de dimensionamiento dinámico de colas dedicado principalmente a dichos flujos en tiempo real.

La propuesta está basada en el principio de máxima entropía y permite que los dispositivos sean capaces de auto-configurar sus colas y así lograr un uso más eficiente de la memoria. Al mismo tiempo, debe garantizarse un valor máximo para la probabilidad de pérdida de paquetes. El funcionamiento adecuado de la propuesta ha sido evaluado extensamente en interfaces cableadas e inalámbricas. Se han considerado redes inalámbricas clásicas basadas en infraestructura y redes de malla multi-salto. Los resultados de simulación para diferentes escenarios, condiciones de tráfico, consideraciones de canal, etc. confirman los beneficios de la propuesta.

Finalmente, cuando la WMN se construye a través de la interconexión de los dispositivos portátiles de los usuarios, la energía es un recurso limitado y escaso, y por tanto cualquier propuesta para optimizar su uso es muy valorada. Para estos casos, esta tesis propone un mecanismo de control de topología basado en métricas de centralidad provenientes del análisis de redes sociales. La idea principal es que en lugar de que todos los dispositivos realicen funciones de enrutamiento, solo un subconjunto de nodos es seleccionado para esta tarea. Para esto, se evalúan diferentes métricas de centralidad, desde una perspectiva centralizada y otra distribuida. A más de los modelos de movilidad aleatorios, comúnmente utilizados, se incluye el análisis de la propuesta con modelos de movilidad que toman en cuenta el comportamiento humano y la información social y como resultado generan redes con una clara estructura de comunidades. Los resultados de los experimentos muestran la efectividad del mecanismo ya que el consumo de energía se reduce considerablemente sin comprometer el rendimiento global de la red.

Acronyms

ABS	Adaptive Buffer Sizing
AIFS	Arbitration InterFrame Space
AIFSN	Arbitration InterFrame Space Number
AODV	Ad hoc On-Demand Distance Vector Routing
BDP	Bandwidth Delay Product
BSS	Basic Service Set
cdf	cumulative distribution function
CDS	Connected Dominating Set
CLB	Combination Load Balancing
CSAR	Connection Strength Aware Routing
CW	Contention Window
DIFS	Distributed (coordination function) InterFrame Space
DP	Distribution Point
DQL	Dynamic Queue Limit
EAF	Egocentric Advertising Frame
EDCA	Enhanced Distributed Channel Access
EWMA	Exponentially Weighted Moving Average
FCFS	First Come First Served
GRT	Global Random Trial
HWMP	Hybrid Wireless Mesh Protocol
IBSS	Independent Basic Service Set
IFS	Inter Frame Space
IP	Internet Protocol
ITS	Intelligent Transportation System
LBF	Least Busy Fit
LRD	Long Range Dependencies
MAC	Medium Access Control
MBSS	Mesh Basic Service Set
MCCA	MCF Controlled Channel Access
MCF	Mesh Coordination Function
MCL	Mesh Connectivity Layer
MLE	Maximum Likelihood Estimate
MPM	Mesh Peering Management
NAF	Neighborhood Advertising Frame
NIC	Network Interface Card
pdf	probability density function
PREP	Path Reply
PREQ	Path Request
QoS	Quality of Service
RANN	Root Announcement
RRT	Repeated Random Trials
RSF	Router Selection Frame

RSI	Resource Sharing Index
RTP	Real-time Transport Protocol
SNA	Social Network Analysis
STA	Station
STR	Single Random Trial
TCP	Transport Control Protocol
TUT	Topology Updating Time
TXOP	Transmission Opportunity
UDP	User Datagram Protocol
VANET	Vehicular Ad hoc Network
VBR	Variable Bit Rate
VoD	Video on Demand
VoIP	Voice over Internet Protocol (IP)
WLAN	Wireless Local Area Network
WMN	Wireless Mesh Network
WSN	Wireless Sensor Network

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Smart environments, smart devices, smart interaction, computing anytime and anywhere ..., the accelerated development of information technologies and mobile devices results in people and/or "things" increasingly dependent on the on-line services offered through private (intranets) or public (Internet) networks. In such scenario, Wireless Mesh Networks (WMNs) have evolved as a cost effective possible solution for the uninterrupted access of users to networking facilities. Valued features like robustness, reliability, resilience, easy deployment and maintenance, self-forming and self-configuration, make WMNs an important alternative to achieve an always-on connectivity.

Besides, WMNs allow the interconnection of the existing networks technologies such as Internet, wireless local area networks, cellular networks, metropolitan area networks, vehicular networks, sensor networks, and personal and body area networks [1]. As a consequence, a wide variety of potential applications arise:

- Broadband home and neighborhood networking. For instance, community networks can be used to reduce the digital divide [2], and promote not only low cost Internet connectivity but also have a more private and neutral communication network [3, 4, 5].
- Support to intelligent transportation systems, mainly as backhaul of vehicular networks [6, 7, 8].
- Building automation and interconnection of home area networks with smart grid communications [9, 10, 11].
- Public safety, security surveillance, disaster management and rescue operations [12, 13, 14].
- Multimedia and real time services [15, 16, 17].
- Health and medical systems.

Actually, there are a lot of ongoing projects on wireless mesh networks in universities, research labs, and industry. Some remarkable reported academic research WMN testbeds are the following:

- MIT Roofnet from the Massachusetts Institute of Technology which consist of 37 mesh nodes distributed over an urban segment of Cambridge, Massachusetts with an area of around 4 km^2 [18, 19, 20].
- VMesh wireless network at University of Thessaly, Volos, Greece [21].
- UCSB MeshNet from the University of California at Santa Barbara consisting of 30 indoor nodes covering several floors inside a building [22].
- Berlin Roofnet (BRN) from the Humboldt University, an experimental IEEE 802.11 based mesh network with around 50 indoor nodes to mainly provide Internet access and VoIP services [23].
- Wray WMN deployed by researchers of Lancaster University in the Wray village located close to the campus in northwest England [2].
- ReMesh 9-nodes mesh network deployed at the Fluminense Federal University campus in the city of Niterói, Brazil. [24].
- Quail Ridge Wireless Mesh Network (QuRiNet) is a 35-nodes wireless mesh testbed for ecological and environmental research. It was deployed by the University of Carolina in the Quail Ridge natural reserve [25].
- Heraklion metropolitan WMN that covers an area of around 60 km^2 with 16 multi-radio mesh nodes in the City of Heraklion, Crete [26].
- BilMesh, an indoor 802.11 b/g mesh networking testbed from Bilkent University of Turkey [27].

As industry and commercial examples it can be mentioned:

- The Google WiFi network which consists of over 500 outdoor mesh nodes deployed in Mountain View, CA and provides free public access to the networking facilities [28].
- The Mesh Connectivity Layer (MCL), an ad-hoc routing and link quality measurement module developed by Microsoft researchers [29, 30].
- The Cisco enterprise wireless mesh solution [31].
- The recently discontinued Mad City Broadband commercial wireless mesh network (MadMesh) which consist on around 250 distributed in Madison, Wisconsin [32].
- The ABB Tropos wireless mesh network communication solutions [33, 34] able to work in extreme environmental conditions.
- The Motorola mesh wide area networks solutions [35].
- The Aruba AirMesh multiservice mesh networks [36].

There are also some real and operational implementations examples of WMNs which are mainly promoted by city councils or non-profit citizen communities:

- Barcelona WiFi-Mesh network deployed by the Barcelona City Council as part of the Smart City project. This project aims to achieve the smart management of all the municipal services operating in the city, including the traffic, security cameras, parking meters, bollards that give access to streets, police's PDAs, and citizens Internet access.

- Guifi.net is an open free and neutral large-scale community mesh network originated in a rural area of Catalonia Spain, currently consisting of 26537 operative nodes [37, 38, 39].

- FunkFeuer Net is a free, experimental network deployed in Viena, Austria [40].

- Athens Wireless Metropolitan Network (AWMN) [41].

- These last tree community networks and other efforts are currently integrated and extended by the Community-Lab European integration project [42, 43, 44].

Basically, a WMN is a multi-hop wireless network composed by mesh routers as the core nodes that provide radio mesh connectivity to commonly mobile and energy constrained mesh clients. Generally, the mesh routers have limited mobility, multiple wireless interfaces and do not suffer of energy constraint problems. Their main function is to forward the client packets to gateways which have an Internet connection in a multi-hop manner. The WMNs architecture has been classified into three main groups [1]. First, infrastructure meshing, in which the static mesh routers form an infrastructure, providing backbone to traditional clients, and allowing different network technologies integration. Second, client meshing in which client nodes simultaneously provide end user applications and routing functionalities. The lack of infrastructure and high mobility at clients level brings as consequence one of the most challenging tasks that is to select the most appropriate routing algorithms, protocols, and metrics that overcome the high variable network conditions with respect to traffic and topology. Finally, the Hybrid WMNs (Fig. 1.1) is a combination of infrastructure and client meshing and probably the most applicable and interesting architecture. In this type, the infrastructure allows multiple technologies interconnection while the client meshing increases the coverage area and reliability.

On the other hand, real time and multimedia services such as VoIP, videoconferencing, video streaming, on-line gaming, IPTV, and video on demand (VoD), are becoming the most demanding applications by the Internet users. Then, due to the increasing computing capacity of mobile devices, the growing availability of wireless channel bandwidth, and the meaningful progress in WMN technology, it is expected that current wireless mesh networks support this kind of services and applications. In general, this real time and multimedia services present demanding requirements for end-to-end delay and packet loss ratio. For instance, in video-conference services, the end-to-end delay is directly associated with the human perception of interaction and so with the end user level of satisfaction about the service. In turn, VoD services, besides delay and loss ratio needs, exhibit high bandwidth requirements that must also be considered. This thesis mainly focuses on real time services in which the information must be consumed at the same time that it is generated. Therefore, connection-oriented protocols like TCP are unfeasible and hence UDP is the preferred transport protocol to be used.

Despite of the large amount of research in WMN and real time multimedia services, there are still open and challenging issues regarding, to name a few, load balancing, efficient resource management and scalability, which motivate this thesis.

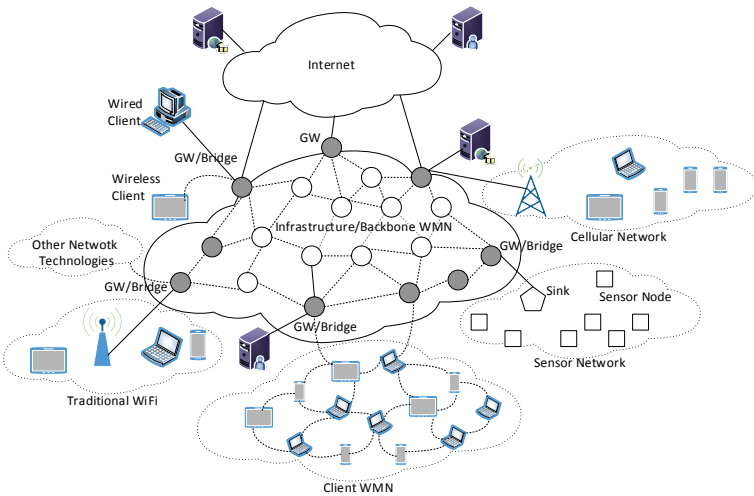


Figure 1.1: WMN general architecture.

1.1 Objectives

The current impact of real time services and WMN technology encourage us to study and make a contribution to each of the main components of an hybrid WMN architecture providing real time services (Fig. 1.1). Thus, the main objective of this dissertation consists of two parts. The first is devoted to improve the performance of an application service, while the second aims to promote the efficient management of two critical network resources which are memory and energy. These goals are more precisely described as follows:

- **Application service.** For the offered service this thesis focuses on a VoD system. Due to scalability and capacity issues, in most of the practical situations, the video distribution points consist of a set of video servers grouped into clusters. With this consideration in mind, we aim to propose a mechanism to appropriately select a video server such that the transfer time at the cluster could be minimized.

- **Infrastructure meshing.** In most real cases, infrastructure mesh routes are equipped with multiple interfaces from a variety of technologies and from both wired and wireless channels. In this case, the memory resources intended for buffers are of paramount importance. Fundamental network performance parameters such as the packet loss probability, end-to-end delay, utilization of transmission channels, etc., are highly influenced by the buffer size of network interface cards. These parameters directly affect the Quality of Service (QoS) perceived by end users. A dynamic buffer sizing method can provide optimal memory allocation, and also helps to prevent exaggerated delays and other problems associated with excessively large buffers. The aim in this case is therefore, to propose and evaluate a dynamic buffer sizing mechanism that allows an efficient memory utilization of multiple-interface mesh routers. This way, mesh devices must be able to autonomously self-configure their interface buffers.

  It is worth to mention that, although the main target are mesh routers from the infrastructure case, this mechanism would be also valuable for client meshing devices

with memory-constrained resources.

- **Client meshing.** The increasing availability of smart devices, embedded with a big amount of sensors, make that we focus our attention to client WMN built by the interconnection of user hand-held devices. The energy resource is one of the major concern of this kind of devices. The purpose at this point is, accordingly, to propose and evaluate a topology control mechanism to reduce the energy consumption of these devices without affecting the overall network performance.

The methodology to fulfill these objectives is based on the use of analytical models whenever possible and their verification with well-known network simulators.

1.2 Summary of Contributions

This section summarizes the major contributions of this thesis:

- In the context of a VoD system, this thesis makes an analysis of the statistical properties of multiplexed sequences of real video traces. The analysis of transmission and transfer times are also included. On this basis, for a cluster of video servers we propose a server selection mechanism that minimizes the average packet transfer time. Besides, the proposal is scalable and easy to implement.

- For a better use of the memory resources intended for buffering, we propose a straightforward method to dynamically adapt the buffer size of network devices. The mechanism is based on the maximum entropy principle and requires parameters which could be effortless measured by each device. The starting point is an easy to follow approximation to the solution of the G/G/1 and G/G/1/K queues via maximum entropy and its application to the dynamic buffer sizing problem. Extensive simulations and numerical results verifies the utility of the proposal for dedicated channels and wired interfaces. The analysis of the input parameter requirements, and the inclusion of additional measures to minimize the introduced mechanism overload is also covered. Traffic differentiation is a key aspect that has been included in this analysis. Once the effectiveness of the proposal was verified for dedicated channels, we extend the buffer sizing method to wireless devices working over shared channels. The algorithm implementation on well-know network simulators allows the validation of the method for both, traditional infrastructure-based wireless interfaces and multi-hop mesh interfaces. Different scenarios, traffic load conditions, channel considerations, etc., have been used to confirm the proper performance of the buffer sizing approach. The definition of a metric to quantify the memory utilization efficiency and its respective evaluation is also included. Finally, it has been made evident the advantages of multiple-interface devices when they are equipped with the ability of efficient memory management and self-configuration of their buffers.

- To reduce the energy consumption of client meshing devices, this thesis proposes a topology control mechanism based on centrality metrics from social network analysis. Different centrality measures have been evaluated, from both centralized and distributed perspectives. In addition to commonly used random mobility models, we include the analysis of socially-aware mobility models which take into account the human behavior

and social relationships. Details of a practical implementation are also specified. The effectiveness of the approach has been verified by the assessment of the network performance using extensive simulations under different operation modes and set-up conditions.

1.3 Resulting Publications

Most of the contents of this dissertation have been published in the following journals and conferences:

JCR Journal Publications:

- L. J. de la Cruz Llopis, A. Vázquez-Rodas, M. Aguilar Igartua, E. Sanvicente Gargallo, "Load splitting in clusters of video servers", *Computer Communications*, vol. 35, no. 8, pp. 993 – 1003, 2012. [45].

- A. Vázquez-Rodas, L. J. de la Cruz Llopis, M. Aguilar Igartua, E. Sanvicente Gargallo, "Dynamic buffer sizing for wireless devices via maximum entropy", *Computer Communications*, vol. 44, no. 8, pp. 44 – 58, 2014. [46].

- A. Vázquez-Rodas, L. J. de la Cruz Llopis, "A centrality-based topology control protocol for wireless mesh networks", *Ad Hoc Networks*, vol. 24, Part B, pp. 34 – 54, 2015. [47].

Latindex Journal Publication:

- R. Rumipamba, A. Vázquez-Rodas, L. J. de la Cruz Llopis, E. Sanvicente Gargallo, "Dynamic Buffer Size Allocation in Wireless Mesh Networks for Non-Elastic Traffic", *Revista Politécnica*, vol. 34, no. 2, pp. 17 – 26, 2014. [48].

Refereed Conferences:

- A. Vázquez-Rodas, L. J. de la Cruz Llopis, "Topology control for wireless mesh networks based on centrality metrics", in *Proceedings of the 10th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, PE-WASUN '13*, New York, NY, USA: ACM, 2013, pp. 25–32. [49].

- A. Vázquez-Rodas, L. J. de la Cruz Llopis, M. Aguilar Igartua, E. Sanvicente Gargallo, "Dimensionado dinámico de buffers para flujos de tráfico diferenciados no elásticos", in *Proc. XI Jornadas de Ingeniería Telemática (JITEL 2013)*, pp. 213 – 220, 2013. [50].

Additionally, it is worth to mention some publications which are not included in this dissertation. They are the result of a collaborative work with the department colleagues and are related with the second and third contributions respectively:

Collaborative Refereed Conferences:

- L. Urquiza-Aguiar, A. Vázquez-Rodas, C. Tripp-Barba, M. Aguilar, L. J. de la Cruz Llopis, E. Sanvicente, "MAX-MIN based buffer allocation for VANETs", in *Wireless Vehicular Communications (WiVeC), 2014 IEEE 6th International Symposium on*, pp. 1–5, Sept. 2014 [51].

- L. Urquiza-Aguiar, A. Vázquez-Rodas, C. Tripp-Barba, A. Mohamad Mezher, M. Aguilar Igartua, and L. J. de la Cruz Llopis, "Efficient deployment of gateways in multi-hop ad-hoc wireless networks", In *Proceedings of the 11th ACM Symposium on Performance Evaluation of Wireless Ad hoc, Sensor, and Ubiquitous Networks, PE-WASUN '14*. ACM, New York, NY, USA, pp. 93-100. [52].

1.4 Outline of this Thesis

This thesis is organized as follows. Chapter 2 provides the background for wireless mesh networks technology and its most important protocols. The next chapters (from 3 to 5) presents, respectively, the three previously described contributions and all of them includes the corresponding state of the art review for each case. The organization details of these chapters is shown in the following.

Chapter 3 presents, for a VoD service, a mechanism to appropriately select a server from a clusters of video servers. To explain this mechanism, this chapter includes a study of video traffic generation and the analysis of the packet transmission and transfer times. The validation of the proposal by means of simulations and the details of a practical implementation is also included.

Chapter 4 focuses on the dynamic buffer sizing contribution to improve the memory utilization efficiency of network devices. After the state of the art review, this chapter presents the approximation to the solution of the G/G/1 and G/G/1/K queues via maximum entropy and its application to the buffer sizing problem. This chapter continues with an extensive analysis and validation of the proposal for dedicated channels, which includes: the traffic load description, the input parameters considerations, the analysis of the overload introduced by the mechanism and the adopted solution to reduce it, and the service differentiation considerations. The second part of this chapter is dedicated to wireless interfaces, including the method extension considerations, and the results for the comprehensive simulation-based evaluation. This last includes different scenarios for both traditional WLAN and mesh interfaces, diverse traffic load and channel conditions, the definition of a metric to quantify the memory utilization efficiency and its evaluation, and the multi-interface devices assessment.

Chapter 5 is devoted to the improvement of the energy efficiency of client WMNs by means of a topology control approach. In addition to the state of the art review, this chapter provides a background on centrality metrics from social network analysis. The presented proposal consists of a centralized and a distributed version. For this last, the practical implementation details are included. This chapter also provides the experimental results of the performance evaluation for different operation modes and for several metrics. In addition to random networks, the last part of this chapter covers the new trends in human mobility models, the resulting community-structured networks, and its implications for the topology control approach.

Finally, Chapter 6 summarizes the main conclusions of this dissertation and depicts the possible improvements and future lines of work.

Chapter 2

Background on Wireless Mesh Networks

This chapter makes a review of the main features of wireless mesh network technology. Specifically, the standardized IEEE 802.11 mesh networking technology and protocols are covered.

2.1 Wireless Mesh Networks main features

WMNs are self-forming multi-hop networks mainly devoted to provide ubiquitous and wireless connectivity to mesh clients through a set of commonly multiple-interface mesh routers.

On its more practical hybrid architecture (Fig. 1.1) WMNs consist of fixed and mobile nodes. The fixed mesh routers, generally powerful devices equipped with more than one radio interfaces, create a resilient backbone to provide Internet access and interoperability with any other network technology. A especial gateway functionality must be included in one or more mesh router(s) for such external interconnection.

In turn, mesh clients are commonly mobile and resource constrained devices that could provide, in addition to end user interface, routing and forwarding capabilities. The main objective of client meshing is to dynamically extend the coverage of the mesh backbone.

The fast, simple and cost-effective deployment is one of the main features of WMNs. This is possible thanks to the self-organizing and self-configuring capabilities, which in addition allow the network growth without considerable effort. The mesh nodes must automatically detect their peers and establish the network connectivity in an unsupervised manner. Besides, the WMN must be robust enough to cope with network topology dynamism, node failures, poor channel conditions, etc. Such self-healing characteristic joint with the inherent high resilience (there are in general multiple available paths) make WMNs able to provide reliable, fault tolerant, scalable, and high performance communications.

The protocols required to accomplish these tasks could be implemented in the network layer or as extensions of the link layer and as open or proprietary solutions. A common factor, in any case, is the fact that most of the research and commercial mesh networking approaches are based on IEEE 802.11 commodity hardware. Since the standardization facilitates the integration, compatibility, and interoperability of devices from different vendors, this dissertation focuses

on the standardized IEEE 802.11s-based mesh networking technology. The next section covers the main features and protocols of this kind of WMNs.

2.2 IEEE 802.11s-based WMNs

The IEEE 802.11s mesh networking task group worked since 2004 to enhance the MAC of IEEE 802.11 in order to support multi-hop mesh networking. The amendment was finally included in the IEEE Std. 802.11-2012 [53]. One of the main features is the definition of a path selection and forwarding mechanism at the link layer instead of the traditional layer-3 routing. Many other functionalities are included as: mesh discovery, peering management, security, beaconing and synchronization, mesh coordination function, power management, channel switching, extended addressing, interworking with external technologies, congestion control, and emergency service support [53]. In the following, after the description of the IEEE 802.11 mesh architecture, this overview focuses only on the facilities most relevant to our work. In addition to the standard itself, more complete surveys can be found in [54, 55, 56, 57, 58].

2.2.1 IEEE 802.11 Mesh Basic Service Set (MBSS) Architecture

A MBSS consists of the following physical and/or logical architectural components (Fig. 2.1):

- **Mesh Stations (Mesh STAs)** include mesh functionalities that allow multi-hop communication. They participate in the creation and operation of the mesh cloud. After the establishment of wireless links with their neighbor STAs, these nodes could be sources, destinations or forwarders of data traffic.
- **Mesh Gate** is a logical component that allows the interconnection among different mesh basic service sets and the integration with other traditional non-mesh infrastructure-based WLANs. This last type of STAs can access the mesh services through a mesh STA with gate and access point (AP) functionalities. It is possible that several mesh gates coexist in a mesh network.
- **Mesh Portal**, another logical component required for the interconnection with external non-IEEE 802.11 network technologies.

Gate, portal or access point logical components could be collocated in a single mesh device. The next sections overview the main mesh functionalities and protocols required to establish and maintain a MBSS.

2.2.2 Mesh Discovery

The mesh discovery procedure can be carried out by a passive scanning of the Beacon frames that every station sends periodically, or by an active scanning process using Probe Request/Response frames. In both cases, the scanning process provides the mesh profile information. The mesh profile specifies the configuration attributes of a mesh network and contains the identifiers of:

- the mesh (Mesh ID),
- the path selection protocol,

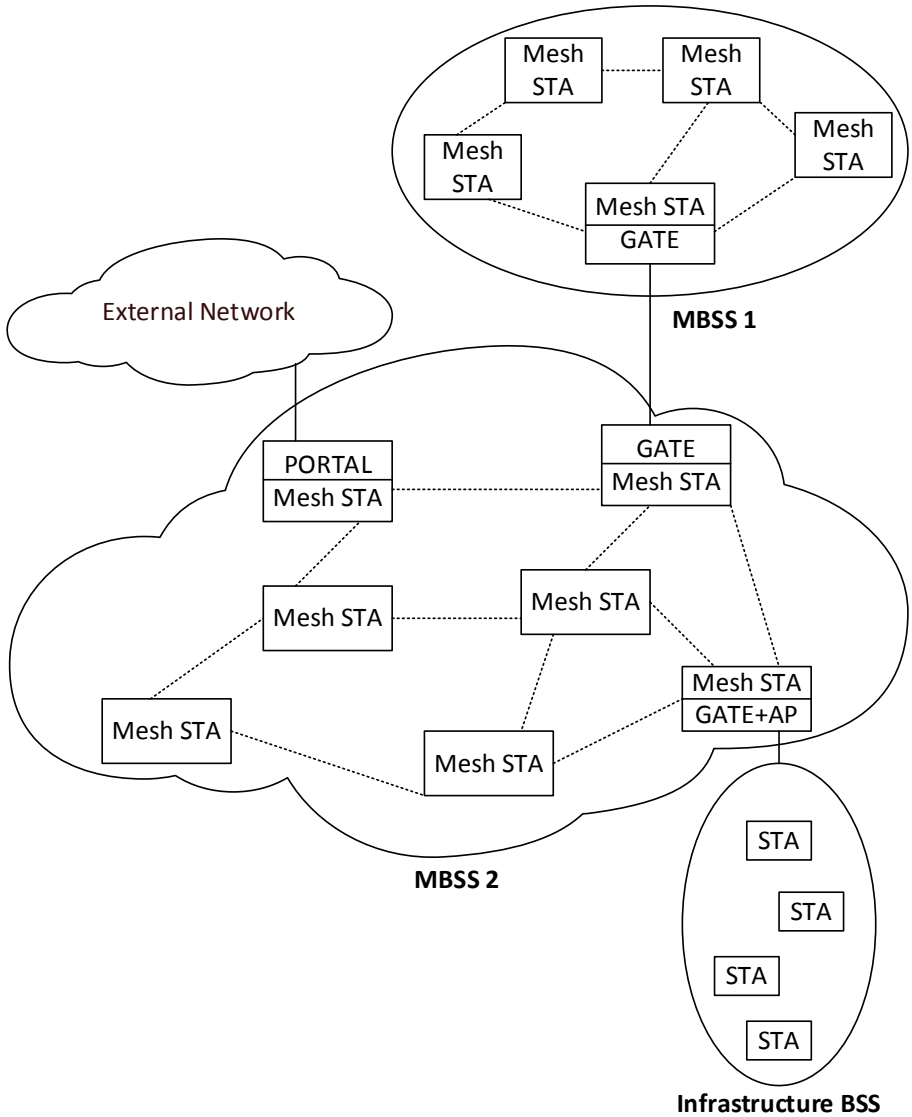


Figure 2.1: Physical and/or logical architectural components of a MBSS.

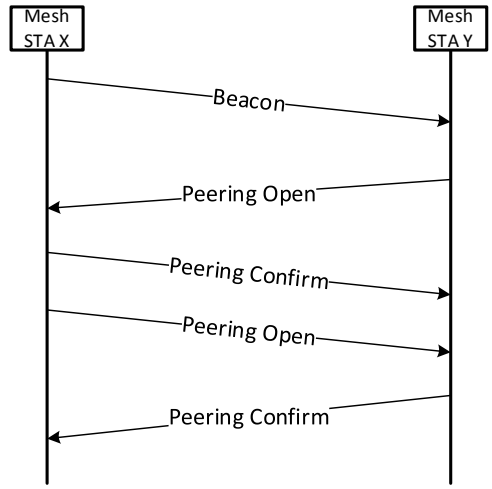


Figure 2.2: Mesh peer link establishment handshake.

- the path selection metric,
- the congestion control mode,
- the synchronization method, and
- the authentication protocol.

All the mesh STAs inside a MBSS must configure and use the same mesh profile. Each mesh STA shall inform regarding the capability or not of establishing additional mesh peerings. This is done through a flag present in the Mesh Configuration element which is contained in the mesh control frames.

After the discovery of a potential candidate mesh peer, and before data interchange, mesh stations must establish and maintain mesh peer links with their one-hop neighbors. This is done by means of the mesh peering management protocol (MPM).

2.2.3 Mesh Peering Management (MPM) Protocol

The mesh peering management protocol uses Mesh Peering Open, Mesh Peering Confirm, and Mesh Peering Close management frames to open, manage and close links between neighbor mesh stations. All established peer links must be bidirectional. For this, a peer link is established only if both involved stations have sent Peering Open requests and have successfully received Peering Confirm responses. Fig. 2.2 illustrates a successful peer link establishment procedure.

A mesh peer is identified by the set of localLinkID (unique integer generated by the mesh STA), localMAC, peerMAC, and peerLinkID (integer generated by the peer mesh STA and transmitted in the peering management frames). Once a peer link has been established and properly identified, the involved stations could start to interchange path selection or data frames.

Although the standard does not specify when to close a peer link, the ns-3 [59] peering management protocol implementation [60], used throughout this work, utilize a number of consecutive beacon loss or data transmission failures to detect peer link breakdowns.

2.2.4 Mesh Coordination Function (MCF)

For channel access, mesh STAs use the mesh coordination function (MCF). It consist of a contention-based channel access, specifically the Enhanced Distributed Channel Access (EDCA) mechanism described in the following, and an optional controlled channel access (MCCA) which is a reservation based channel access method.

The default IEEE 802.11 Distributed Coordination Function (DCF) provides only best effort services. In this case, all traffic types compete in the same way for channel access. Real-time applications requires to guarantee certain QoS parameters like bandwidth, delay and delay jitter. To accomplish with these requirements the IEEE 802.11e group proposes the EDCA protocol [53]. This mechanism improves and extends the original features of DCF and it is the mandatory MAC scheme for mesh STAs.

EDCA provides service differentiation and different priorities to four classes of services including: voice (VO), video (VI), best effort (BE) and background (BK). The different traffic classes are called Access Categories (ACs) and each of them has one specific priority queue. Additionally, in order to achieve priority differentiation, every AC has four important channel access parameters, which are:

- the minimum Contention Window (CWmin),
- the maximum Contention Window (CWmax),
- the Arbitration inter-frame space (AIFS), and
- the Transmission Opportunity (TXOP).

CWmin and CWmax define the contention window range used for the backoff process. In the EDCA scheme, CW will be reset to CWmin immediately after a successful packet transmission. If there is a collision, CW will be doubled until reaches CWmax.

The maximum allowed transmission time is defined by the TXOP limit: once a station accesses to the medium it can transmit one or more frames during TXOP. Finally, instead of using a fixed Distributed Inter Frame Space (DIFS), in EDCA an Arbitrary IFS (AIFS) is applied. The AIFS for a given AC is determined by the following equation [53]:

$$\text{AIFS[AC]} = \text{AIFSN[AC]} \times \text{aSlotTime} + \text{aSIFSTime} \tag{2.1}$$

AIFSN is the AIFS number determined by the AC and the physical settings. The highest priority will be given to the AC with smallest AIFS. Table 2.1 shows the default parameters for the different ACs [53].

As previously said, each AC has its own queue and it is possible that one or more classes tries to access to the medium at the same time, which is known as “internal contention”. In this case, packets from highest priority queue (AC_VO) are served first due to their lowest backoff time.

On the other hand, MCCA aims to optimize the efficiency of frame exchanges inside the mesh network by reducing the contention. This work only focuses on mesh STAs working with EDCA since it is the default and mandatory mechanism.

Table 2.1: Default EDCA parameters.

AC	CWmin	CWmax	AIFSN	TXOP Limit
AC_VO	3	7	2	1504 µs
AC_VI	7	15	2	3008 µs
AC_BE	15	1023	3	0
AC_BK	15	1023	7	0

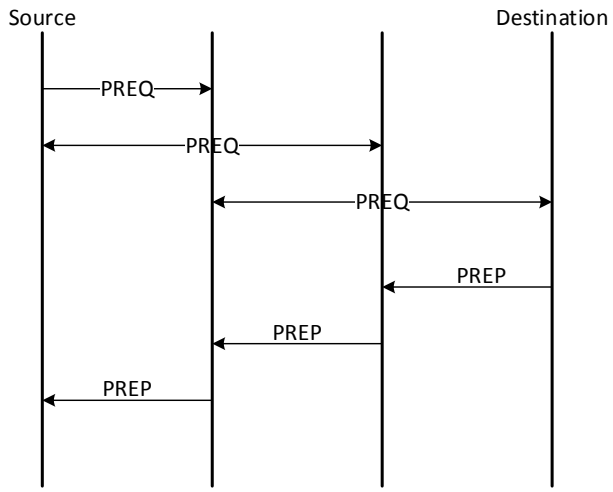


Figure 2.3: On demand mode of HWMP path discovery procedure.

2.2.5 Mesh Path Selection and Forwarding

The key mesh functionality to provide layer-two multi-hop communications is the link layer routing or mesh path selection mechanism. The Hybrid Wireless Mesh Protocol (HWMP) is the mandatory and default path selection protocol. HWMP is an AODV-inspired [61] protocol that combines a reactive on-demand path selection with a proactive tree building mode. These two modes can be used concurrently, to allow its use in a diversity of environments and requirements. It can be considered that the proactive mode extends the default on-demand functionalities.

In the **On-demand** mode (Fig. 2.3), when a mesh STA requires a path to another one, it broadcast a Path Request (PREQ) frame. Intermediate mesh STAs propagate the PREQ to their peers after creating or updating its own path information towards the source. Obviously, the metric field of the PREQ is updated with a cumulative value on each propagation step. Loop-free connectivity is ensured by a sequence number mechanism which is managed by each mesh STA. When the request reaches the target mesh STA, it responses with a unicast Path Replay (PREP) frame back to the source and it also creates the corresponding reverse path. In this way, mesh STAs communicate between them using bidirectional, best metric end-to-end paths. A faster path establishment is possible when intermediate STAs are allowed to return an already known PREP to the source. This is managed through the Target Only (TO) control flag available in the PREQ frames. TO flag set to zero means that intermediate mesh nodes are able to generate PREPs for available paths.

On the other hand, for the **Proactive** mode, at least one station must be configured as root

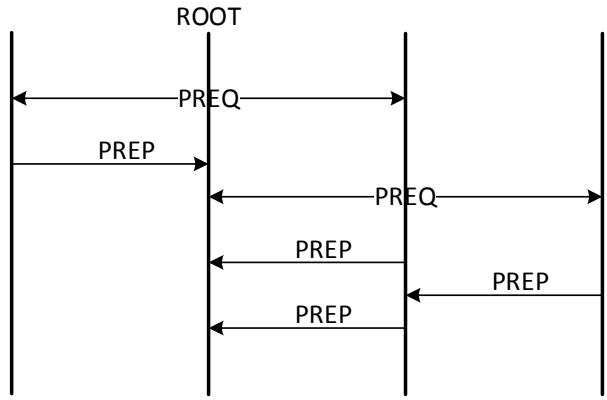


Figure 2.4: Proactive PREQ-based mode of HWMP path discovery procedure.

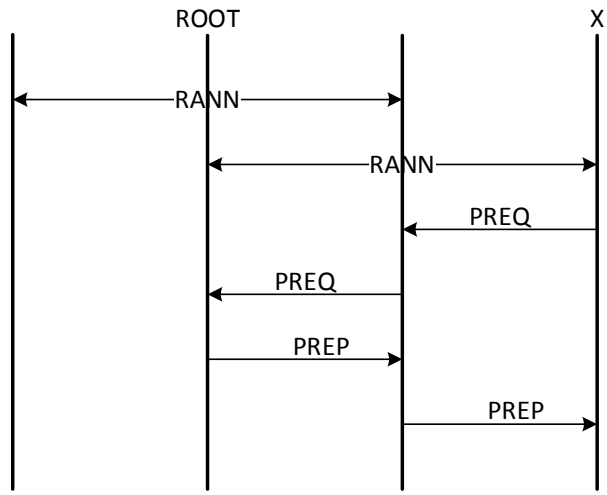


Figure 2.5: Proactive RANN-based mode of HWMP path discovery procedure.

mesh STA. The proactive tree to the root can be built in two different ways: using a proactive PREQ or by means of root announcement (RANN) frames.

For the proactive PREQ mechanism (Fig. 2.4), the root periodically broadcasts PREQ frames with TO set to one. When such proactive PREQ reaches a mesh STA, this node creates or updates the path to the root. Before propagating the proactive PREQ such STA updates and records the metric and hop count to the root. This way all mesh STAs become aware of the root presence and the distance to it. Thanks to the sequence number mechanism, only newer or better metric paths are taken into account when multiple PREQs are received. A proactive PREQ flag controls whether a mesh receiving a PREQ must respond with a proactive PREP frame or not. When this flag is set to one the recipient mesh STA shall send a PREP and so the path from the root to the STA is established. On the other hand, if the flag is set to zero, the mesh STA only send a proactive PREP when it has data to transmit to the root.

In the case of using the proactive RANN mechanism (Fig. 2.5), the root periodically

propagates RANN frames containing path metrics to the root. Nevertheless, there is no creation of forwarding paths after the reception of a RANN. If a RANN receiver (For instance, X station in Fig. 2.5) want to establish a path to the root, it shall send a unicast PREQ to the root. The root replies with a PREP to create the bidirectional path.

The standard also allows that mesh stations may include alternative path selection protocols and metrics. However, just one path selection protocol and metric shall be active at a given time.

The path selection protocol can use the default and mandatory **airtime link metric** to select the best path to a destination. The metric objective is to estimate the amount of channel resources required to transmit a frame over a specific link. This metric is computed according to the following equation [53]:

$$c_a = \left[O + \frac{B_t}{r}\right] \frac{1}{1 - e_f} \tag{2.2}$$

Where O is the channel access overhead, which depends on the PHY type and includes frame headers, training sequences, access protocol frames, etc. B_t is the test frame size (the recommended value is 8192 bits). r is the date rate in Mbps at which mesh STA would transmit the test frame and e_f is the measured frame error rate for the test frame.

Chapter 3

Load Splitting in Clusters of Video Servers

Nowadays, video on demand is one of the services more highly appreciated and demanded by customers. As the number of users increases, the capacity of the system that provides these services must also be increased to guarantee the required quality of service. An approach to that end is to have available several video servers at various distribution points in order to satisfy the different incoming demands (video server cluster). When a movie demand arrives to such a cluster, a load balancing device must assign the request to a specific server according to a procedure that must be fast, easy to implement and scalable. This chapter considers the problem of appropriately splitting this load to improve the system performance. After an analysis of the video packet generation, we point out the similarity between this problem and that of optimally routing packets in data networks. With this similarity in mind, a new mechanism to select the appropriate video server is proposed. The purpose of this mechanism is to minimize the average packet transfer time (waiting time plus transmission time) at the video server cluster. In this way, we are able to obtain a dynamic load balancing policy that performs satisfactorily and that is very easy to implement in real environments. The results of several experiments run with real data are shown and commented to substantiate our claims. A description of a practical implementation of the system is also included.

3.1 Introduction

Among all the emerging services offered by the new communication networks, video on demand (VoD) is one of the most appreciated by customers and has very promising market expectations. Nowadays, many companies are already offering this service, and the number of subscribers is growing steadily. As this number increases, the capacity of the VoD service must also expand in order to provide customers with the expected quality of service (QoS). This increase in service capacity can be basically obtained following two different and complementary approaches [62, 63]. On one hand, contents can be replicated (mirroring) or temporarily stored (caching) in different points in the network. This way, several content distribution points (DP) are available (see Fig. 3.1). On the other hand, in any of these distribution points, there are usually several video servers to satisfy the different incoming demands (video servers cluster).

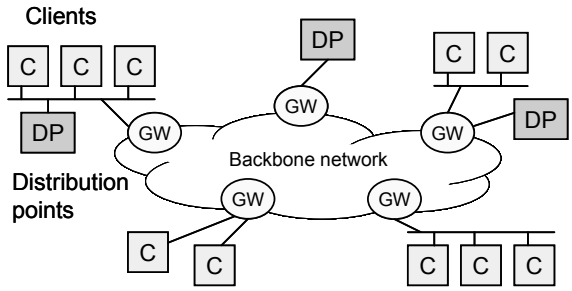


Figure 3.1: Generic architecture of a content distribution network.

If several distribution points are available, every client request must be routed to one of them. The problem of finding the best distribution point fits very well with recent works based on cloud computing. In this framework, infrastructure, software, hardware, platform, and storage, are offered as a service using a pay-per-use services model. Users access the cloud computing infrastructure that consists of data centers in the form of geographically distributed servers with different levels of virtualized technologies. In such highly variable and heterogeneous systems, where the virtualized resources are dynamically shared out by the users, traditional load balancing schemes that considers homogeneous nodes are totally unrealistic. For instance, the load balancing technique proposed by authors in [64] aims to reduce the execution time for each multimedia request in a hierarchical three layer cloud architecture. Their CMLB (Cloud-based Multimedia Load Balancing) approach uses the network proximity between a server and a client and the server loads as evaluation parameters for server selection. Namely, the selected server should be the closest to the client and less loaded. One remarkable feature of this approach is the fact that it is location aware. Client localization is achieved by a simple landmark method that measures network link latency from each client to each landmark. Other possible location alternatives could be IP geolocation systems. For instance, [65] presents a multi-step approach utilizing measurement based geolocation techniques along with a semantic based scheme. Another possibility is the Internet location service developed by the IETF [66, 67] and described in [68, 69]. As another example, [70] presents a comparative study of three methods for distributed load balancing in large scale cloud computing systems. The first one is a biologically inspired honeybee foraging based load balancing technique that works over a set of servers clustered into virtual servers. In this case, each server collaborates to construct an advert board that reflects the notion of how effectively resources are being used. Secondly, a balanced and self-organized network is achieved using a biased random sampling scheme adding location information to reduce the effects of communication latency. Finally, an algorithm that groups similar services to improve the load balancing scheme is assessed.

Once the distribution point is selected, a new choice must be made among the different available servers (physical or virtualized) in that point. Each time a movie demand arrives to the video servers cluster, a load balancing device must assign the request to a specific server according to a procedure that must be fast, easy to implement and scalable. This device must also take into account that the performance of the various video servers in the cluster can be different due to different processing speeds and also to different network boards and output channels capacities. Although when a video server cluster is first put into operation its components are rather homogeneous, this does not hold true as the cluster grows, and more

advanced equipment is brought in to jointly operate with the old one.

Another possible architecture, suggested by several researchers, is based on the idea of parallel video servers [71]. Theses architectures propose the stripping of video movies into several video servers (the whole movie is divided into segments, and each segment is placed in a different server), instead of partitioning them (some movies in a server and other movies in other servers) or replicate them (all the movies in every server). The idea behind stripping is to avoid increasing excessively the system storage needs (as occurs with replication) and to avoid that some servers are accessed more frequently than others (as occurs with partitioning). In this type of architecture a new element (the proxy) is needed, which takes responsibility for mixing and sending sequences in the correct order from servers to clients.

In any case, as the users demands increase, the need to replicate some kind of information in different servers is again obvious, mainly when the behavior of the final users (who demand movies based on popularity) is taken into consideration. Also, it must be kept in mind that nowadays technology allows for huge amounts of storage at a very reasonable cost and in a reduced physical space [72, 73]. This renders possible to obtain the added advantages (reliability, ease of implementation and quality of service offered to users) provided by the replication of information, whatever it may be (whole or stripped movies). On the other hand, it must be also taken into account that the video stripping technique increments the system complexity and, what is more, does not perform well in systems with high degree of interactivity, such as True-Video on Demand systems [74]. These reasons have led some researchers not to use video stripping in their proposals [75].

Fig. 3.2 shows a generic connection scheme between a content distribution point (a server cluster in this case) and a customer (C) that requested the transmission of a movie. Also, the various delays that a packet will experience have been indicated: packet generation time, network transit time and presentation time. The packet generation time has itself several components: disk access time, waiting time in the server queue, transmission to the switch and retransmission to the gateway. Similarly, the network transit time is also composed by all the waiting and retransmission times the packet experiences as it hops from node to node in the network. Finally, the application requires an additional time to decode the video information and to present it on the screen. It also introduces (on purpose, by means of buffering), an extra time to reduce the variability of packet delay times (jitter).

In order to minimize the end-to-end time it is necessary to minimize each of those components. With respect to the network transit time, a bunch of different techniques have been proposed that aim to somehow group the user requests. Among them are batching, piggybacking, patching, hierarchical multicast stream merging, periodic broadcasting and scheduling [72, 73, 76]. Also, adaptive multimedia streaming mechanisms to increase bandwidth efficiency while maintaining user's quality perception have been proposed and evaluated [77]. Besides, much effort has been done recently to improve the performance of video services from any kind of access network, including wireless ad hoc networks [78, 79, 80], WiMAX broadband wireless networks [81], and WMN [15, 17, 16, 82].

Regarding the packet generation time, the disk access can be reduced using redundant access architectures and internal buses whose speeds are much larger than those of the network access cards. The other components of this time are the waiting and transmission times at the server. The sum of these two times will be called transfer time in this chapter, which introduces an analytic procedure to optimize load balancing among the different servers, i.e., to minimize video packet transfer times.

Finally, to minimize presentation times, the compression algorithms are much simpler at

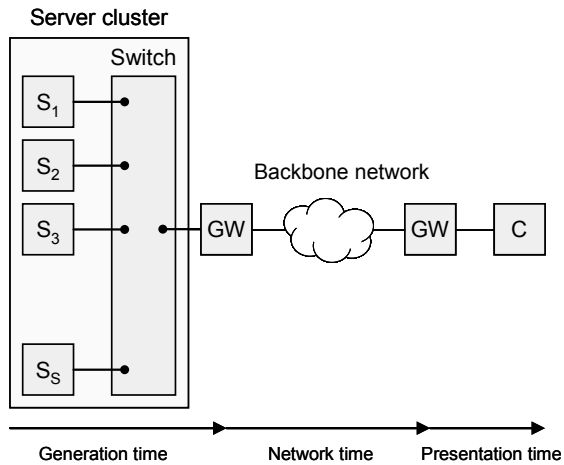


Figure 3.2: Delays experienced by packets in the video sequence.

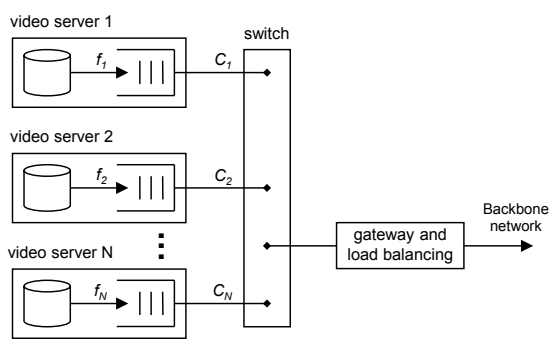


Figure 3.3: Flows generation in the clusters of video servers.

the decoding end, which allows very short processing times. The video servers in Fig. 3.3 store the movies to be transmitted at the request of users. Prior to their storage, the movies have been compressed using an appropriate coding standard, like MPEG-2 [83], MPEG-4 [84] or H.264 [85]. In these coding schemes, successive frames of the sequence are coded in different modes (I, P, B), and different scenes need different amounts of bits to be coded, which implies a variable bit rate (VBR) at the coder output. The servers and the gateway are connected to one or various switches. Each movie to be transmitted represents a certain flow for the cluster of servers. In this work, whenever a user requests a movie, a load balancing device selects one of the servers to carry out the transmission. This device can be incorporated into the gateway or could be a standalone device (see Fig. 3.3). The video server selection will be done with the objective of minimizing the video packet transfer time. In this architecture, all the video sequences are available to all the video servers (for instance, by means of replication). Calling f the aggregated bit rate in bits per second (bps) and $f_i, i = 1 \dots N$, the flow corresponding to the ith server, the problem is to split f into the different flow streams ($f = f_1 + f_2 + \cdots + f_N$), taking into account that the server channels may have different capacities ($C_1 \geq C_2 \geq \cdots \geq C_N$) due to several reasons: different network interface cards, bandwidth sharing with other devices or processes, etc.

To explain the presented approach to the solution of this problem, the body of this chapter is structured as follows. Next section deals with related work. In Section 3.3 a brief study of the video traffic generation is presented. In Section 3.4 the packet transmission and transfer times are analyzed. Section 3.5 establishes the similarity between the problem of selecting the correct video server and a splitting scheme that can be used in data networks to route packets between two nodes connected by several links. After elaborating the solution, in Section 3.6 simulation results are shown, carried out with real movies, to validate the proposed splitting algorithm. Section 3.7 proposes a practical implementation of the load balancing device. Finally, in Section 3.8 the conclusions of this contribution are detailed.

3.2 Related Work

The management of clusters of video servers is a subject that has been addressed in previous works. Several of those works are focused on the problem of distributing the movies among the available servers. In a nutshell, movies can be stored only in one server or they can be replicated in several servers. The idea behind this is to reduce the blocking probability, i.e. the probability that a movie cannot be served because the server (or servers) that contains that movie is (are) completely busy. This way, a full replication of all movies would be the best solution to decrease the blocking probability, although the amount of storage space needed would also grow. Therefore, a trade-off exists between the blocking probability and the amount of storage space. Authors in [86] present a video stream replication scheme (MMPacking) that achieves load and storage balancing by replicating a small number of video streams. They first place all the movies in a round-robin fashion, and then replicate them with a strategy based on the probability with which every movie is requested by the clients (in fact, based on the cumulative probability of all the movies stored in every server). Once the system is working with the replicated movies a new problem arises: every time a user requests a new movie the transmission must be assigned to a specific server. Here, the authors propose a weighted scheme, according to the probabilities assigned by the algorithm to each stream, achieving a uniform probability of requests assigned to each workstation.

The same problems are addressed in [87]. With the goal of minimizing the blocking probability, the authors study four resource selections schemes: Single Random Trial (SRT), Repeated Random Trials (RRT), Global Random Trial (GRT) and Least Busy Fit (LBF). With SRT, when a new request arrives to the system a server is randomly chosen. If this server is completely busy, the request is blocked. However, if RRT was used the system would continue with repeated random trials until all the servers were attempted. GRT is an efficient extension of RRT in which the system takes profit of the fact that it is possible to know a priori whether a server is completely busy or not. Finally, if the servers load can be monitored, with LBF the request is assigned to the least busy server. The authors provide an analytical framework for analysis of a large scale VoD system using those schemes. They demonstrate the inefficiency of RRT in handling requests and compare the signaling overheads of GRT and LBF. The same authors extend their work in [88], where they point out that the performance of the LBF system is determined by a good file allocation of the movie files in the available servers. They show that a good file allocation for SRT is not necessarily the best for LBF. They focus on LBF, but the algorithms and analytical methodologies proposed also apply for the RRT system. They propose a conjecture on how the movie traffic load should be ideally distributed in the LBF to achieve combination load balancing (CLB) and to minimize the blocking probability. They introduce a resource sharing index (RSI) to measure how evenly a file allocation instance distributes the multi-copy movie traffic load. Besides, they provide a file allocation method to obtain a uniform resource sharing of multi-copy movie traffic and a uniform distribution of single-copy movie traffic.

Two more load-balancing strategies are designed and presented in [89]. In this work, authors assume that movies are fully replicated, so any server in a cluster can satisfy any request. To design their proposals they focus on two facts. First, a significant portion of media content is represented by short and medium length videos encoded at low bit rates. Thus, a large fraction of these accesses could be served from memory. Second, a significant amount of clients do not finish playing an entire media file. Therefore, the distribution of durations of media sessions and the distribution of original media file durations can be vastly different. With these two facts in mind, they propose, design, and evaluate two new strategies. These strategies (FlexSplit and FlexSplitLard), which are compared to previous solutions, show better performance in terms of blocking probability, memory hit ratios and stable performance.

The contribution presented in this chapter differs from the previous in some aspects. It is focused on the video-on-demand service, where customers are supposed to watch the whole movie. On the other hand, it focuses on minimizing the transfer time instead of the blocking probability, because we are considering the statistical multiplexing of the movies at the output of every video server. Besides, the possibility of having heterogeneous servers inside the cluster is taken into account.

3.3 Video traffic generation

Each time a user demands the transmission of a movie, the load balancing device assigns the service to one of the video servers. The selected server needs a certain time until each packet of the video sequence is completely transmitted. This time includes several components: the time needed to access the information container (storage media), the time to build up the headers of the involved protocols, the time the packet has to wait in the output queue and, finally, the transmission time. Among those times, the last two are very relevant and, as said before, their

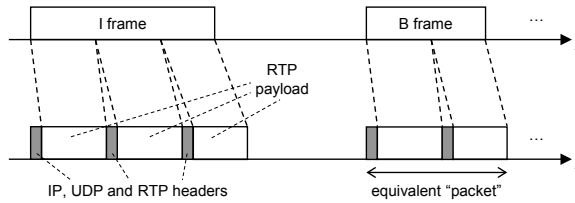


Figure 3.4: Video frames encapsulation and transmission.

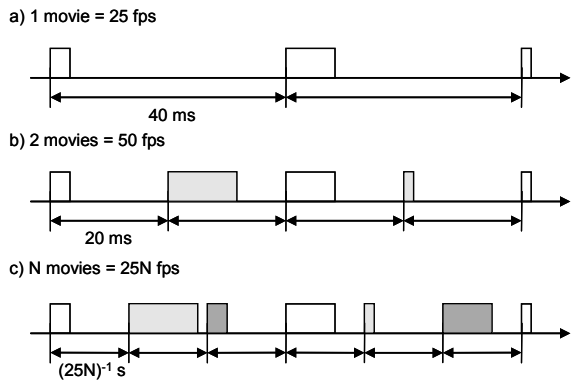


Figure 3.5: Interleaving of video sequences by the video server.

sum is called transfer time.

Picture frames are fragmented and encapsulated in several packets of the transport protocol used. For instance, in IP networks, the protocol most widely used is the Real-time Transport Protocol (RTP) [76]. Every I, P or B frame of the VBR coded sequence is fragmented (if necessary), the IP, UDP and RTP headers are added and all the packets containing information from the same picture frame are transmitted in succession (Fig. 3.4). As far as the transfer times are concerned, we can assimilate the transmission of all the packets that constitute a frame to the transmission of a single packet whose length is the sum of the length of all the transport packets. Note that, due to the large size of the packets that carry video information with the high quality required for VoD, the header lengths are very small in comparison to the total packet length.

When the server is idle and it is assigned the transmission of a movie, it usually transmits 25 or 30 frames per second (fps). In this work we focus, without loss of generality, on the PAL system which works at 25 fps. This implies that the time between packets is 40 ms. (see Fig. 3.5a). Remember that, for the purposes of this work, a packet is the sum of all packets containing information from the same video frame. If another movie is requested from the same server, the new sequence of packets must be interleaved with the previous one and so on (Fig. 3.5b and c). The more appropriate way for the server to interleave movies is to maintain constant the time between packets. The performance of the output buffer is also better since the minimum time between packets is maximized. Thus, for N movies, the time between packets will be $(40/N)$ ms.

Let us now focus on the packet length and its variability. In Table 3.1 we have selected

some relevant statistical parameters (mean, variance and coefficient of variation) related to the packet lengths for ten different movies. Taking into account that final users call for excellent performance in video on demand services, we have chosen high quality MPEG-4 coding [90]. Observe how the coefficient of variation (CV) varies from movie to movie, ranging from 0.55 to 0.79.

Table 3.1: Statistical parameters of some MPEG-4 VBR sequences.

Name	**Mean (bits)**	**Variance**	**CV**
Die Hard III	27894	3.1123E08	0.63
Star Wars IV	11011	5.2242E07	0.66
From Dusk Till Down	27243	2.5394E08	0.58
First Contact	13158	9.3090E07	0.73
Starship Troopers	24105	1.7662E08	0.55
The Firm	11728	8.4924E07	0.78
Jurassic Park	30625	3.2659E08	0.58
Mr. Bean	23294	2.0527E08	0.61
Silence of the Lambs	23011	3.3593E08	0.79
Aladdin	17415	1.9014E08	0.79

In this work, we have carried out a series of simulations on a 20 Mbps channel, varying the number of transmitted sequences in order to have a channel load between 0 and 100%. To instrument the simulations, the traffic traces detailed in Table 3.1 have been used. With these traces, and with an appropriate mixing strategy, a set of 80 sequences, corresponding to the multiplexing of 1, 2, 3, ..., and up to 80 video movies, has been built. We will refer to this set of sequences as the *multiplexed sequences*. The preparation and mixing of these traces to produce high loads is a non trivial process that requires the consideration of many factors [90]. To obtain the best network performance, it is of paramount importance to avoid sequence synchronization, at both short and long term. Simultaneous transmission of I frames from different sequences should be avoided since the size of these frames is the largest. This task can be achieved by the video server appropriately delaying the beginning of the transmission. Another possible solution is sequence smoothing prior to transmission [78, 79]. In addition, and for the same reasons, the avoidance of synchronization of high activity scenes is also advisable. However, for the servers, it is rather difficult to tackle this second problem, since this requires knowing and managing not only all the sequences being transmitted at present time, but also those that will begin in a predetermined time span. It is important to notice that the video cluster performance, in terms of transfer times, will be worse if these pieces of advice are not followed. Thus, in that case, the benefits obtained by load balancing would even be more apparent. In this work, however, the mixing of sequences has been done abiding by the above recommendations. Therefore, the improvement in real system operation would even be higher than the one shown here.

Let us study some statistical parameters of the multiplexed sequences. Fig. 3.6 shows the average (m) and standard deviation (σ) of the packet length. As it can be seen, as the number of multiplexed movies grows, both m and σ tend to a value. Observe that the average tends to approximately 21000 bits, which is of course the average of the means in Table 3.1. This implies that every video sequence, when multiplexed with others, will need in the average a channel capacity of:

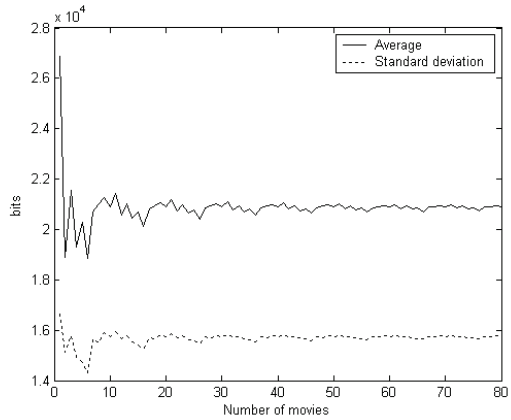


Figure 3.6: Average and standard deviation of the multiplexed sequences set.

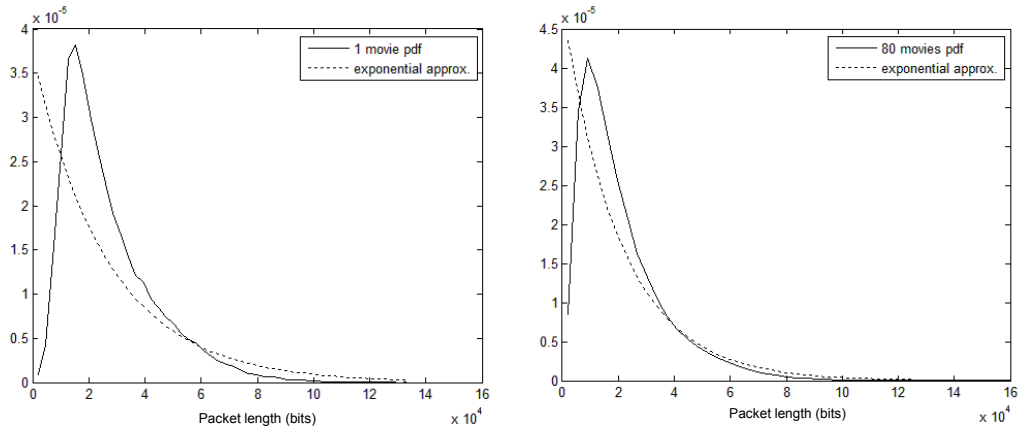
$$21000 \frac{\text{bits}}{\text{frame}} \cdot 25 \frac{\text{frame}}{\text{s}} = 525 \text{ Kbps} \tag{3.1}$$

On the other hand, the standard deviation tends to approximately 16000 bits, which implies a value of the coefficient of variation equal to:

$$CV = \frac{\sigma}{m} = \frac{16000}{21000} = 0.762 \tag{3.2}$$

The probability density function (pdf) of the packet length has been traditionally modeled by means of several functions, such as the Beta function [91], or a combination of the Gamma and Pareto functions [92, 93]. However, in order to obtain an analytical expression for the load balancing mechanism searched in this work, which could be used in practice, a very much simple model is needed. Of course, this model will be less precise, but, as it will be shown in Section 3.6, the results obtained by means of simulations justify its use. Specifically, the chosen model has been the exponential. The approximation is better when the number of multiplexed movies grows. For instance, Fig. 3.7 shows this approximation for the sequence with 1 and 80 movies, which has been obtained with the Maximum Likelihood Estimate (MLE) procedure. In this case, MLE provides an exponential function with the same mean parameter than the real sequence mean value. For the 80 movies case, the root mean square error for this approximation is $5.7463 \cdot 10^{-6}$.

Another important aspect that must be taken into account regarding the packet length, is the high autocorrelation presented. The autocorrelation function decreases slowly, showing the long range dependencies (LRD) that are present in video traffic. This fact has led to its modeling as a self-similar process [91, 92, 93, 94]. However, this fact is also reduced when the number of multiplexed video movies grows, as it can be seen in Fig. 3.8, where the autocorrelation functions for 1 and 10 video movies are shown. Therefore, even though the effect of the LRD exists, it is not a first order factor here, so that we can work with simpler models.



(a) 1 movie. (b) 80 movies.

Figure 3.7: Probability density function of the multiplexed movies.

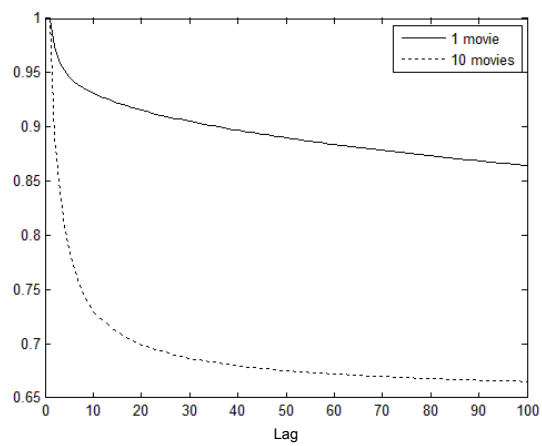


Figure 3.8: Autocorrelation function.

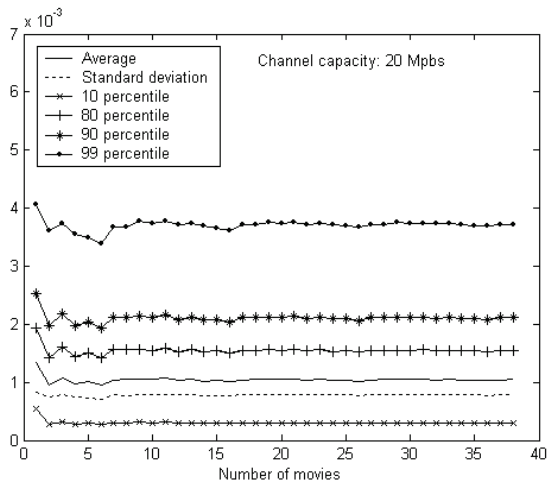


Figure 3.9: Transmission time statistical parameters.

3.4 Transmission and transfer times analysis

Transmission time is directly related to packet length. Fig. 3.9 shows the average, the standard deviation and four percentiles of the transmission time (namely the 10, 80, 90 and 99 percentiles), obtained from simulations over a 20 Mbps channel. Again, to drive the simulation, the set of multiplexed sequences has been used. The figure has been limited to 38 video movies because the values keep approximately constant for higher numbers of movies. It can be checked how the average value of the transmission time tends to approximately 1.05 ms, corresponding to the average value of the packet length in the previous section (21000 bits) divided by the channel capacity.

The reason of introducing the percentile is to provide another justification for the use of the exponential approximation. The r–percentile, $\Pi(r)$, of the transmission time can be expressed in function of the average transmission time (T_T) and standard deviation (σ_T) using the formula:

$$\Pi\left(r\right) = T_T + F\left(r\right)\sigma_T \tag{3.3}$$

where $F(r)$ is a factor that depends upon the distribution of the transmission time. This factor has been computed for the multiplexed sequences, and it is shown in Fig. 3.10 for those four values of r. Besides, the values of the factor $F(r)$ for the exponential distribution are also shown. These values can be easily computed as follows. Denoting by:

$$f_{t_T}\left(t\right) = \frac{1}{T_T} e^{-\frac{t}{T_T}}, \qquad t \geq 0 \tag{3.4}$$

the probability density function of the transmission time, by definition of $\Pi(r)$ we can write:

$$\int_{\Pi(r)}^{\infty} \frac{1}{T_T} e^{-\frac{t}{T_T}}\, dt = \frac{100 - r}{100} \tag{3.5}$$

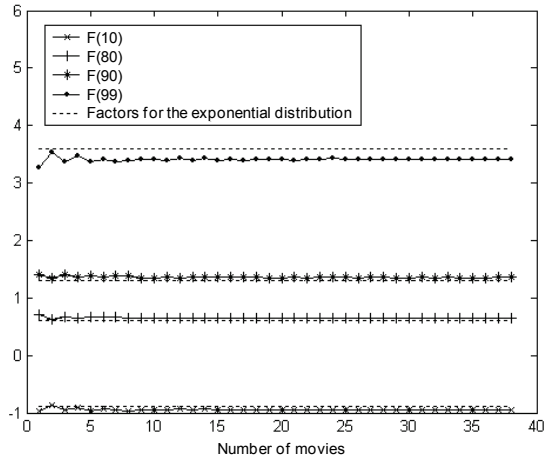


Figure 3.10: Comparison of factors corresponding to simulation and the exponential distribution for transmission time.

Integrating and solving for $\Pi(r)$, we obtain:

$$\Pi\left(r\right) = T_T \ln \frac{100}{100 - r} \tag{3.6}$$

Remembering now that, for the exponential distribution, mean and standard deviation are equal ($T_T = \sigma_T$), we have:

$$\Pi\left(r\right) = T_T + \sigma_T \left(\ln \frac{100}{100 - r} - 1 \right) \tag{3.7}$$

and, therefore, the factor $F(r)$ is given by:

$$F\left(r\right) = \ln \left(\frac{100}{100 - r} \right) - 1 \tag{3.8}$$

The value of the above expression for r =10, 80, 90 and 99 is -0.89, 0.6, 1.3 and 3.6 respectively, in good agreement with the measurements, as it can be seen in Fig. 3.10.

In view of the above measurements and computation, it is justifiable to approximate the transmission time by an exponential random variable, when several video movies are multiplexed. This approximation can be further validated computing the same factors for the transfer times. In fact, for any kind of interarrival time distribution, if the transmission time is exponentially distributed so is the transfer time [95].

Fig. 3.11 shows the measurements of the same parameters computed now for the transfer time and again for a 20 Mbps capacity channel. In this figure we have plotted only the region corresponding to a non saturated channel (before the "elbow" of the transfer time is reached) because this is the operating region of interest. The factors $F(r)$ computed for those measurements for the same four different percentiles, together with the value given by the exponential distribution, are shown in Fig. 3.12. Again, the agreement can be considered as good.

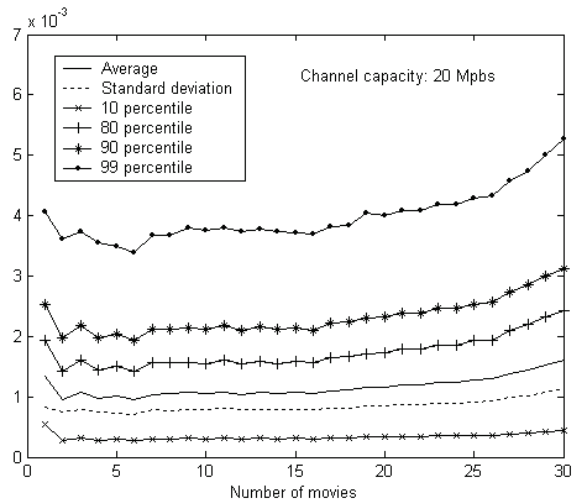


Figure 3.11: Transfer time statistical parameters.

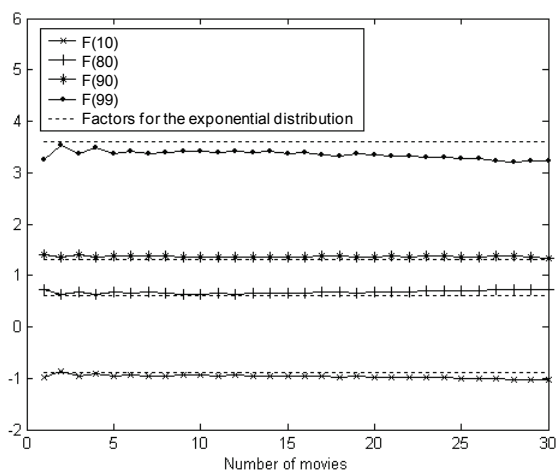


Figure 3.12: Comparison of factors corresponding to simulation and the exponential distribution for transfer time.

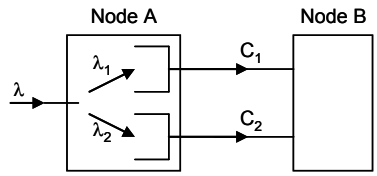


Figure 3.13: Packet splitting scheme.

All these facts allow us to assume exponentially distributed transmission times. With respect to interarrival times, in this approach we will consider and compare two cases: deterministic and exponential cases.

3.5 An analytical model for load balancing computation

As said before, the aim of the load balancing algorithm introduced in this approach is to optimally select a server each time a movie is requested. To simplify the exposition, all the computations will be carried out for the case of two servers, with $C_1 \geq C_2$. However, the formulas corresponding to the analytic models will also be given for the general case of N servers.

It is very apparent that at very low loads all the requests should be addressed to server 1 (the one with the largest available bandwidth), and server 2 will only come into play as the load increases and exceeds a certain threshold f_t. Therefore, the algorithm has to determine the splitting threshold and the amount of traffic through each server when the load exceeds that threshold.

Calling T_i the (average) time it takes to transfer a packet from server i to the gateway, since the objective is to reduce delays as much as possible, we have to minimize the expression:

$$T = \frac{1}{\lambda}\left(\lambda_1 T_1 + \lambda_2 T_2\right) \tag{3.9}$$

where, $\lambda = \lambda_1 + \lambda_2$ and λ_i, $i = 1, 2$, is the number of packets per second through video server i.

Observe that, once the problem of selecting a video server in a cluster has been stated in the previous way, it is conceptually similar to another problem: the problem of optimally routing packets between two network nodes connected by two links, as shown in Fig. 3.13 [96].

As it has been said in Section 3.3, the better way to multiplex the video movies is keeping constant the time between packets. This way, and having justified also that the transmission time can be approximated with an exponential pdf, the model that will be taken into account is D/M/1. The solution for finding the optimal splitting policy is presented in the following. Besides, if a lower degree of optimization was enough, the well-known M/M/1 model provides a simpler solution, which is also included.

3.5.1 D/M/1 model

For the D/M/1 queuing system, the transfer time is given by this expression [95]:

$$T_i = \frac{L}{C_i\,(1-\rho_{ai})} \tag{3.10}$$

where L and ρ_{ai} are the average packet length in bits and the probability that an arriving packet sees the server in channel i busy respectively.

Substituting Eq. (3.10) in Eq. (3.9), we obtain:

$$T = \frac{1}{\lambda}\left(\frac{f_1}{C_1\,(1-\rho_{a1})} + \frac{f_2}{C_2\,(1-\rho_{a2})}\right) \tag{3.11}$$

where $f_i = \lambda_i L$ is the flow in bits per second through channel i. To minimize T, the term between brackets in Eq. (3.11), which will be called $\Phi(f_1, f_2)$, must be minimized:

$$\Phi\,(f_1, f_2) = \left(\frac{f_1}{C_1\,(1-\rho_{a1})} + \frac{f_2}{C_2\,(1-\rho_{a2})}\right) \tag{3.12}$$

Calling ρ_i the average utilization of channel i, we have:

$$\rho_i = \frac{\lambda_i L}{C_i} = \frac{f_i}{C_i} < 1 \tag{3.13}$$

To continue, it must be remembered that ρ_{ai} is related to ρ_i, and therefore to f_i. The relation is [95]:

$$\rho_{ai} = F^*_{t_{ai}}\,(S)\,|_{\frac{1-\rho_{ai}}{T_{Si}}} \tag{3.14}$$

where $F^*_{t_{ai}}$ is the Laplace transform of the pdf of packet interarrival times, i.e.:

$$F^*_{t_{ai}}\,(S) = \int_0^\infty f_{t_{ai}}\,(t)\,e^{-st}dt \tag{3.15}$$

Since t_{ai} is deterministic, with mean T_{ai}, we have:

$$f_{t_{ai}}\,(t) = \delta\,(t - T_{ai}) \tag{3.16}$$

therefore:

$$F^*_{t_{ai}}\,(S) = \int_0^\infty \delta\,(t - T_{ai})\,e^{-st}dt = e^{-sT_{ai}} = e^{-(s/\lambda_i)} \tag{3.17}$$

and, finally, we obtain:

$$\rho_{ai} = e^{\left(-\frac{S}{\lambda_i}\right)\Big|_{\frac{1-\rho_{ai}}{T_{Si}}}} = e^{\left(-\frac{1-\rho_{ai}}{\rho_i}\right)} \tag{3.18}$$

This transcendental equation does not yield easily to manipulation. For our purposes, we have approximated it by a minimum square quadratic fit:

$$\rho_{ai} \simeq A\rho_i^2 + B\rho_i + C \tag{3.19}$$

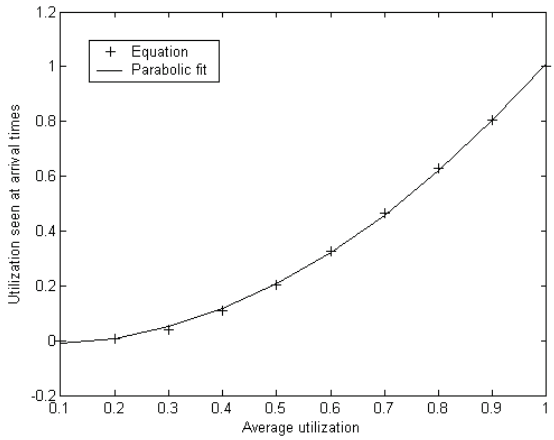


Figure 3.14: Parabolic fitting.

with $A = 1.184$, $B = -0.167$ and $C = -0.006$. In Fig. 3.14 a discrete set of points corresponding to Eq. (3.18) is plotted together with the parabolic fit. As it can be seen, the match is very good. This fact allows computation to proceed with ease.

To begin with, since:

$$\rho_i = \frac{f_i}{C_i} \tag{3.20}$$

the Eq. (3.19) can be written as:

$$\rho_{ai} = A\left(\frac{f_i}{C_i}\right)^2 + B\left(\frac{f_i}{C_i}\right) + C \tag{3.21}$$

From Eq.(3.12), the problem to solve now is to find the minimum of:

$$\Phi(f_1, f_2) = \frac{f_1}{C_1 - A\frac{f_1^2}{C_1} - Bf_1 - C_1C} + \frac{f_2}{C_2 - A\frac{f_2^2}{C_2} - Bf_2 - C_2C} \tag{3.22}$$

subject to:

$$\begin{cases} 0 \le f_1 < C_1 \\ 0 \le f_2 < C_2 \\ f_1 + f_2 = f = \lambda L \end{cases}$$

This is the classical problem of finding the minimum of a function subject to certain constraints, which can be solved using the Lagrange multipliers. In the following, a brief and graphical description of the method, particularized for this case of study, is included.

Due to the constraint $f_1 + f_2 = f$, the value of $\Phi(f_1, f_2)$ must be a point in the straight line $f_1 + f_2 = f$. This line is drawn for two cases in Fig. 3.15 ($f > f_t$ and $f = f_t$). The situation when both channels are utilized ($f_1 > 0, f_2 > 0$), is depicted in Fig. 3.15a. For the minimum value of $\Phi(f_1, f_2)$, its gradient vector, $grad\Phi(f_1, f_2)$, must be perpendicular to the line, because otherwise a lower value would exist going up or down over the line. That is, $grad\Phi(f_1, f_2)$ must

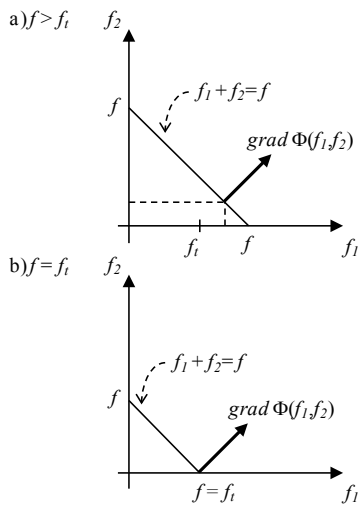


Figure 3.15: Optimization conditions.

be perpendicular to any vector parallel to the line, for instance the vector $(-1, 1)$. Thus, the following dot product is zero:

$$(-1 \qquad 1) \begin{pmatrix} \frac{\partial \Phi(f_1,f_2)}{\partial f_1} \\ \\ \frac{\partial \Phi(f_1,f_2)}{\partial f_2} \end{pmatrix} = 0 \tag{3.23}$$

and therefore:

$$\left.\begin{array}{l} \dfrac{\partial \Phi(f_1,f_2)}{\partial f_1} = \dfrac{\partial \Phi(f_1,f_2)}{\partial f_2} \\ f_1 + f_2 = f \end{array}\right\} \tag{3.24}$$

Computing the derivatives, one obtains:

$$\left.\begin{array}{l} \dfrac{C_1(1-C)+\frac{A}{C_1}f_1^2}{\left(C_1 - A\frac{f_1^2}{C_1} - Bf_1 - C_1C\right)^2} = \dfrac{C_2(1-C)+\frac{A}{C_2}f_2^2}{\left(C_2 - A\frac{f_2^2}{C_1} - Bf_2 - C_2C\right)^2} \\ f_1 + f_2 = f \end{array}\right\} \tag{3.25}$$

By continuity, at the threshold value, Eq. (3.24) also holds with $f_1 = f_t$ and $f_2 = 0$ (Fig. 3.15b). Thus, the threshold value, f_t, can be found setting $f_1 = f_t$ and $f_2 = 0$ in Eq. (3.25):

$$\frac{C_1(1-C)+\frac{A}{C_1}f_t^2}{\left(C_1 - A\frac{f_t^2}{C_1} - Bf_t - C_1C\right)^2} = \frac{1}{C_2(1-C)} \tag{3.26}$$

In conclusion, Eq. (3.26) provides the value of f_t, and Eq. (3.25) provides de amount of traffic demanded to each server (f_1 and f_2) when the load exceeds the threshold. Although closed-form expressions cannot be obtained, the two equations can be solved numerically using any of the standard techniques. For instance, if $C_1 = 20$ Mbps and $C_2 = 15$ Mbps, the value

of f_t is 6.74 Mbps. That value, translated into number of movies, corresponds to (remember from Section 3.3 that every movie needs 525 Kbps):

$$\frac{6.74 \text{ Mbps}}{525 \text{ Kbps/movie}} = 12.84 \approx 13 \text{ movies} \tag{3.27}$$

If, at a given state of its operation, the cluster of servers has to deliver, for instance, 30 movies, the flow f would be:

$$f = 30 \times 525 \text{ Kbps} = 15.75 \text{ Mbps} \tag{3.28}$$

Eq. (3.25) then yields $f_1 = 9.84$ Mbps and $f_2 = 5.91$ Mbps. These flow values, translated into number of movies, represent:

$$\begin{aligned} \frac{9.84 \text{ Mbps}}{525 \text{ Kbps}} &= 18.74 \approx 19 \text{ movies} \\ \frac{5.91 \text{ Mbps}}{525 \text{ Kbps}} &= 11.25 \approx 11 \text{ movies} \end{aligned} \tag{3.29}$$

That is, 19 movies must be transmitted by video server 1 and 11 movies by video server 2.

The previous procedure can be easily generalized to obtain the solution when N video servers are available. In this case, $N-1$ thresholds must be found. The second server is used only if the value of f is higher than the first threshold $f_{t,1}$, the third server is used when f is higher than the second threshold $f_{t,2}$, and so on. The values of the thresholds are given by:

$$f_{t,j} = f_{tj,1} + f_{tj,2} + \cdots + f_{tj,k} + \cdots + f_{tj,j} \tag{3.30}$$

where $f_{tj,k}$ is the solution of the following equation:

$$\frac{C_k(1-C) + \frac{A}{C_k} f_{tj,k}^2}{\left(C_k - A\frac{f_{tj,k}^2}{C_k} - B f_{tj,h} - C_k C\right)^2} = \frac{1}{C_{j+1}(1-C)} \tag{3.31}$$

Similarly, when the flow splits into j channels, the values of $f_1, f_2, \ldots, f_j$, are given by solving numerically the following set of j equations:

$$\left.\begin{aligned} &\frac{C_1(1-C) + \frac{A}{C_1} f_1^2}{\left(C_1 - A\frac{f_1^2}{C_1} - B f_1 - C_1 C\right)^2} = \cdots = \frac{C_j(1-C) + \frac{A}{C_j} f_j^2}{\left(C_j - A\frac{f_j^2}{C_j} - B f_j - C_j C\right)^2} \\ &f_1 + f_2 + \cdots + f_j = f \end{aligned}\right\} \tag{3.32}$$

3.5.2 M/M/1 model

As said before, the simple M/M/1 model is also introduced here just to have a simpler solution that could be used if a lower degree of optimization was enough. In this case, it is possible to obtain closed-form expressions, as shown in [96]. It is well known that for the M/M/1 queuing system, the transfer time is given by the expression [95]:

$$T_i = \frac{L}{C_i - \lambda_i L} \tag{3.33}$$

where L is the average packet length in bits.

Substituting in Eq. (3.9) we obtain:

$$T = \frac{1}{\lambda}\left(\frac{f_1}{C_1 - f_1} + \frac{f_2}{C_2 - f_2}\right) \tag{3.34}$$

where $f_i = \lambda_i L$ is the flow in bits per second through channel i. The problem now reduces to find the minimum of:

$$\Phi(f_1, f_2) = \frac{f_1}{C_1 - f_1} + \frac{f_2}{C_2 - f_2} \tag{3.35}$$

subject to:

$$\begin{cases} 0 \le f_1 < C_1 \\ 0 \le f_2 < C_2 \\ f_1 + f_2 = f = \lambda L \end{cases}$$

When both channels are utilized ($f_1 > 0, f_2 > 0$), one must have again, as in the D/M/1 case previously studied:

$$\left.\begin{array}{l} \dfrac{\partial\Phi(f_1, f_2)}{\partial f_1} = \dfrac{\partial\Phi(f_1, f_2)}{\partial f_2} \\ f_1 + f_2 = f \end{array}\right\} \tag{3.36}$$

By continuity, at the threshold value, the above equation also holds with $f_1 = f_t$ and $f_2 = 0$.

Since:

$$\frac{\partial\Phi(f_1, f_2)}{\partial f_i} = \frac{C_i}{(C_i - f_i)^2}, \qquad i = 1, 2 \tag{3.37}$$

the threshold can be found as:

$$\left.\frac{\partial\Phi}{\partial f_1}\right|_{f_1 = f_t} = \left.\frac{\partial\Phi}{\partial f_2}\right|_{f_2 = 0} \Rightarrow \frac{C_1}{(C_1 - f_t)^2} = \frac{1}{C_2}$$
$$f_t = C_1 - \sqrt{C_1 C_2} \tag{3.38}$$

Similarly, the equations to obtain the optimum splitting are:

$$\begin{cases} \dfrac{\sqrt{C_1}}{C_1 - f_1} = \dfrac{\sqrt{C_2}}{C_2 - f_2} \\ f_1 + f_2 = f \end{cases} \tag{3.39}$$

Solving for f_1 and f_2, we finally obtain:

$$f_1 = C_1 - (C_1 + C_2 - f)\frac{\sqrt{C_1}}{\sqrt{C_1} + \sqrt{C_2}}$$
$$f_2 = C_2 - (C_1 + C_2 - f)\frac{\sqrt{C_2}}{\sqrt{C_1} + \sqrt{C_2}} \tag{3.40}$$

Generalizing the previous procedure it is easy to obtain the solution when N video servers are available. The values of the thresholds are given by:

$$f_{tj} = \sum_{i=1}^{j} C_i - \sqrt{C_{j+1}} \sum_{i=1}^{j} \sqrt{C_j}, \qquad j = 1 \ldots N - 1 \tag{3.41}$$

and the splitting policy is:

$$
\begin{aligned}
&f < f_{t,1}: \\
&\quad \begin{cases} f_1 = f \\ f_k = 0, \qquad k = 2 \dots N \end{cases} \\
&f_{t(j-1)} < f < f_{tj}: \\
&\quad \begin{cases} f_k = C_k - \left(\sum_{i=1}^{j} C_i - f \right) \dfrac{\sqrt{C_k}}{\sum_{i=1}^{j} \sqrt{C_i}}, \qquad k = 1 \dots j \\ \\ f_k = 0, \qquad k = j \dots N \end{cases} \\
&f > f_{t(N-1)}: \\
&\quad f_k = C_k - \left(\sum_{i=1}^{N} C_i - f \right) \dfrac{\sqrt{C_k}}{\sum_{i=1}^{N} \sqrt{C_i}}, \qquad k = 1 \dots N
\end{aligned}
\tag{3.42}
$$

Therefore, and as it has already been said, using the simpler M/M/1 model close form expressions are available. Thus, the load balancing device does not have to deal with numerical computations. However, the obtained values for the thresholds f_{tj} and for the number of movies per server are less accurate. In next section this trade-off will be seen in more detail.

3.6 Validation of the model

In order to validate the methodology proposed in this contribution, several simulations with real traffic traces have been carried out. Firstly, in Fig. 3.16, the theoretical results corresponding to both D/M/1 and M/M/1 models are plotted. Two available video servers have been taken into account, with capacities $C_1 = 20$ Mbps and $C_2 = 15$ Mbps. The figure shows the number of movies that must be requested to every server (N_1 and N_2) in front of the total number of movies requested to the cluster. Of course, the addition of the two values, $N_1 + N_2$, must be the total number of movies requested. It can be observed how, if the M/M/1 approach is used, when the number of requested movies is lower than 6, all the movies are served by the video server 1. Beyond that threshold, the video server 2 starts sharing the load. For instance, when the number of requested movies is 10, 8 of them are assigned to server 1 and 2 to server 2. On the other hand, when the D/M/1 approach is used, the value of the threshold grows to 13 movies. As it will be shown later, this value adjusts better the experimental results.

Another important result is that both approaches provide very similar values when the load increases. As it can be seen, beyond 20 movies the number of movies requested to every server using both models is very similar and, what is more, beyond 31 movies the values are identical.

In the sequel, the results obtained by means of simulations are presented. Remember that the main objective of the proposed splitting mechanism is to minimize the transfer time. In Fig. 3.17 several transfer times are plotted in front of the number of movies being transmitted. Three cases are shown: using only video server 1 (high capacity channel), using only video server 2 (low capacity channel), and balancing the load between the two servers with the proposed algorithm. Observe that the transfer time through the low capacity channel begins to grow sharply above 23 movies. A similar effect appears for the high capacity channel beyond

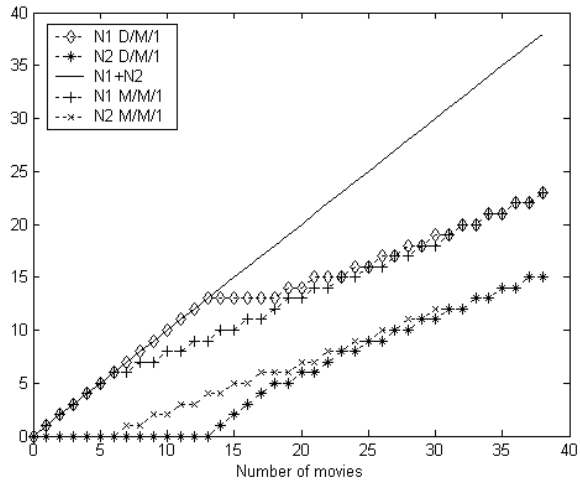


Figure 3.16: Comparison between D/M/1 and M/M/1 models.

32 movies. However, due to the benefits of using two video servers and the splitting algorithm, the average transfer time is almost constant, as it can be seen in the same figure.

Let us call $T_1(N)$ and $T_2(N)$ the two previous curves in Fig. 3.17 in which only one video server is used. The minimum transfer time T that could be achieved using the two video servers can be computed as follows. Let N_1 and N_2 denote the number of movies requested to video server 1 and 2 respectively, and let N be the total number of movies, $N = N_1 + N_2$. The value of T for any possible combination of N_1 and N_2 is:

$$T(N_1, N_2) = \frac{N_1}{N} T_1(N_1) + \frac{N_2}{N} T_2(N_2) \tag{3.43}$$

Then, the minimum value that can be achieved is:

$$\begin{aligned} &T = \min T(N_1, N_2) \\ &N_1 > N_2 \\ &N_1 + N_2 = N \end{aligned} \tag{3.44}$$

Therefore, for any value of N, once the minimum value of T has been obtained following Eqs. (3.43) and (3.44), we have also got the optimal splitting between N_1 and N_2. Obviously, if N is less than a certain value (the threshold), $N_1 = N$ and $N_2 = 0$. Beyond that threshold, the traffic is split. The results are shown in Fig. 3.18.

The fluctuations in the values of N_1 and N_2 are different when other sets of multiplexed sequences are used to drive the simulations. So, in reality, that set of points is a cloud of points representing the intrinsic randomness in the selection of movies. Therefore, to obtain a clearer picture of the trend, Fig. 3.19 also includes a linear fit to the experimentally obtained set of points. This linear fit, converted to an integer number of movies, is plotted in Fig. 3.20 together with the values provided by the D/M/1 model. As it can be seen, the model follows well the values of the simulation, although there are slight differences due to the simplifications done to keep a very simple model. Therefore, we can conclude that the proposed mechanism provides a

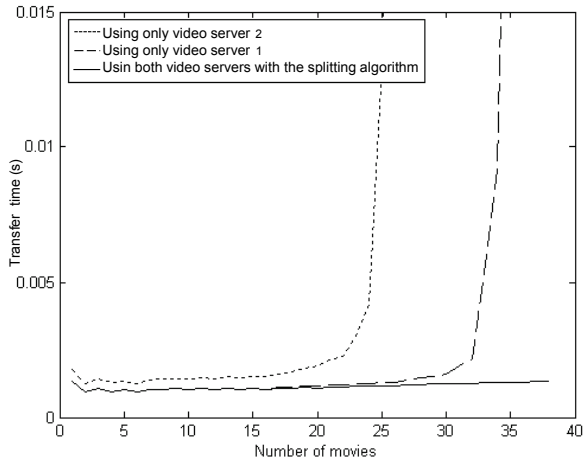


Figure 3.17: Transfer time.

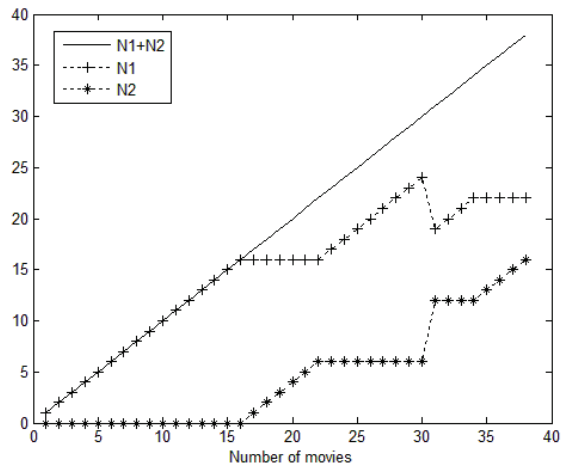


Figure 3.18: Movie splitting over two channels.

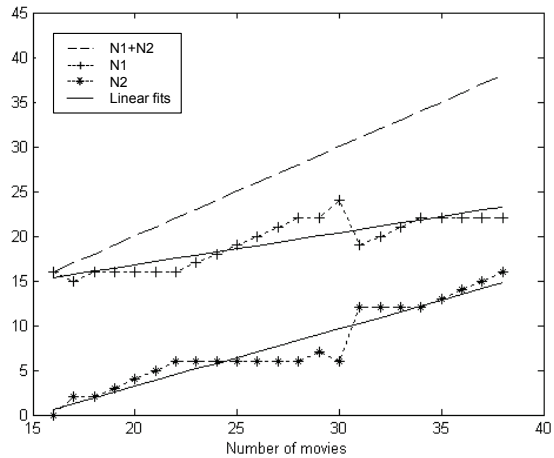


Figure 3.19: Movie splitting and liner fit.

useful tool to compute the thresholds and splitting proportions required in the cluster of video servers, without the need of using more complicated traffic models.

3.7 Practical implementation

The load balancing system operates according to the block diagram shown in Fig. 3.21. The device keeps a table with the average flow at the output channel of each server. Each time a movie is requested, the request is assigned to one of the servers taking into account the total flow and the flow through each of the servers at that moment. Subsequently, the entries in the table are updated, and the system remains on hold, waiting for new demands. During this waiting time, the table has to be periodically updated since it is possible that a movie has finished and therefore, the utilization of some of the servers would be decreased.

In summary, the load balancing device has two tasks:

1. To receive the transmission demands from customers and to assign each of them to a specific server. To this end, the load balancing expressions presented in Eqs. (3.25) and (3.26) will be used.

2. To update the utilization entries in the table due to:

   (a) New demands that increase the utilization of the serves to which they are assigned. The load balancing device will add the appropriate flow to the chosen server.

   (b) End of transmission that decreases the utilization of the corresponding server. In order to detect such finalizations, the load balancing device monitors and periodically averages the output flow through all the servers.

It should be stated that the system operation, once it has been started, is also extremely simple. This, in turn, implies that the device will not be overloaded even when the number of demands on part of customers is large. The total capacity of the server cluster to transmit

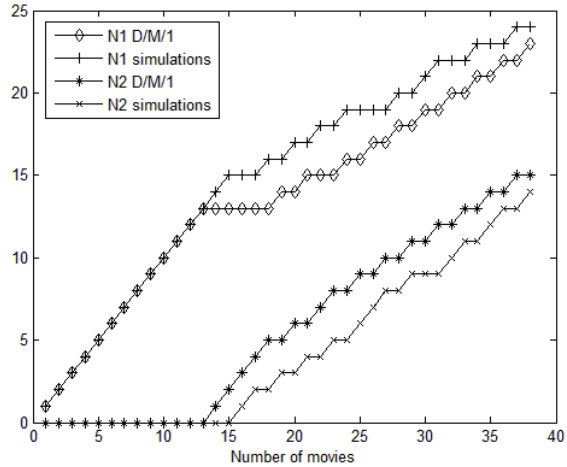


Figure 3.20: Comparison with D/M/1 approach.

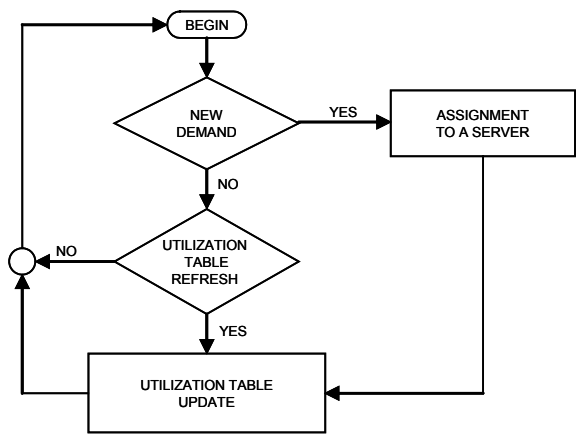


Figure 3.21: Balancing device block diagram.

movies simultaneously will be the limiting factor, and not the processing power of the balancing device. Therefore, the device scales well and does not constitute a bottleneck for the whole system.

Finally, it is interesting to distinguish two operating modes depending upon the system ability to dynamically reassign the transmission of movies among the various servers. For instance, if the fastest (or the one with more output capacity) server ends the transmission of a movie, it could be possible to assign to this server the transmission of another movie previously ascribed to another server of lesser performance. This reassignment would again be done according to the load splitting equations given before. However, it should be noted that this feature complicates the practical implementation of the system since the balancing device should be able to indicate to the new server at which point of the movie it has to start the transmission. This implies that the balancing device software has to understand the information transmitted by the servers (i.e., the coded video sequence) and that a somewhat more complicated communication protocol between the balancing device and servers should be available.

Conversely, if this dynamic reassignment is not implemented, it may happen that at times the minimum transfer time will not be reached. However, the device implementation and start-up are much easier and they do not call for any additional equipment or protocol.

3.8 Conclusions

This chapter addresses the problem of how to increase the service capacity of VoD systems. The service capacity of these systems can be increased using several content distribution points in the communication network. Moreover, in each of these points several video servers must be available to properly handle customer demands. The selection of a server to transmit the requested movie must be done quickly, be easily realizable and scalable.

This chapter has pointed out the similarity between the load balancing problem when there are several video servers available, and that of routing packets with multiple transmission channels between two nodes. Once this similarity is stated, we propose to employ the same type of method to balance the load among the servers. This method is compatible with the use of any replication, partitioning or stripping technique, and also with unicast, multicast, broadcast or other petitions groupings techniques (batching, piggybacking, ...) approaches that are currently proposed.

Solutions using D/M/1 and M/M/1 models have been obtained. The results provided by both models are very similar for very low and very high load conditions, but they differ substantially in the intermediate region. We have also compared these solutions with the results obtained by simulations. The study shows that the D/M/1 queuing formulas approximate rather well the splitting threshold and proportions. On the other hand, if a lower degree of optimization was enough and a minimum number of computations was preferred, the M/M/1 model could also be used.

Finally, we have outlined the details of a possible implementation, showing the ease of the start-up procedure. The system simplicity avoids failures caused by overloads when demand increases, which implies that the system does not present scalability problems.

Chapter 4

Dynamic Buffer Sizing via Maximum Entropy

Buffer overflow is an important phenomenon in data networks that has much bearing on the overall network performance. Such overflow critically depends on the amount of storage space allotted to the transmission channels. To properly dimension this buffering capacity a detailed knowledge of some set of probabilities is needed. In real practice, however, that information is seldom available, and only a few average values are at the analyst disposal. In this chapter, the use of a solution to this quandary based on maximum entropy is proposed. On the other hand, when wireless devices are taken into account, the transmission over a shared medium imposes additional restrictions. This chapter also presents an extension of the maximum entropy approach for this kind of devices. The main purpose is that wireless nodes become able to dynamically self-configure their buffer sizes to achieve more efficient memory utilization while keeping bounded the packet loss probability. Simulation results using different network settings and traffic load conditions demonstrate meaningful improvement in memory utilization efficiency. This could potentially benefit devices of different wireless network technologies like mesh routers with multiple interfaces, or memory constraint sensor nodes. Additionally, when the available memory resources are not a problem, the buffer memory reduction also contributes to prevent the high latency and network performance degradation due to overbuffering. Besides, it also facilitates the design and manufacturing of devices with faster memory technologies and future all-optical routers.

4.1 Introduction

Buffers are crucial elements of all kind of routers. They have a great impact in many performance evaluation parameters like packet loss probability, end-to-end delay, delay jitter, link utilization, and throughput. This impact is especially significant during congestion times. The prevention of packet losses has motivated the spread of excessively large buffering across a wide range of network devices and technologies, from Internet core routers to access devices on the edge networks. This phenomenon of buffer oversizing, known as *bufferbloat* [97], has been accelerated in recent years by the reduction in the memory cost. The bufferbloat brings as a direct consequence that end users experience excessive high latencies in their communications,

independently of their access technology and bandwidth [98]. In such environment, the quality of service provided to real-time applications, which are very sensitive to delays, could be very low. Thus, new buffer sizing schemes can provide some benefits to modern network devices. For instance, a small buffer size is extremely valued in all-optical packet switching routers design and construction [99, 100, 101]. Nevertheless, too small buffers increase the packet losses and reduce the link utilization when TCP-alike protocols are used [102, 103]. Dimensioning routers buffer size is therefore not an easy task and is an active research topic mainly for wired routers [102, 104, 105]. On the other hand, the growth in the use of mobile devices and their increasing computational capacity, together with the user ubiquitous access expectations, is causing a constant increase in the demand of wireless networking technologies (like wireless local area networks, mobile ad-hoc networks, wireless mesh networks, etc.). However, less attention has been provided to the buffer sizing in wireless devices, where new challenging issues arise [106].

In the study of queuing theory, it is customary to begin assuming knowledge of the distributions of service and inter-arrival times, and from there the theory is constructed using Markov (and embedded Markov) chains, Laplace and Z transforms and other mathematical techniques [107]. In real practice, however, that detailed information is seldom available and, in fact, in most instances, the only information at our disposal is a few average values from which other parameter of interest, related to the system performance, must be provided. Obviously, this process involves a risk since in so doing we are giving more information than we have. To be more specific, suppose that the average value of packets in a transmission system (buffer and server) and the server occupancy have been measured, using, for instance, moving averages. How could we, then, obtain, solely from those two averages, the percentiles of packets in the system? To produce these percentiles, the distribution of packets must be acquired, which is far beyond the available knowledge. The solution to this dilemma can be stated in general terms as follows: when in the course of a system evaluation, the data required to assess the system performance exceeds the available data, the extra information needed should be generated maximizing its entropy in a way compatible with the obtained measurements. This approach has been already presented by some researchers in a rigorous mathematical way [108]. This contribution includes a new derivation of the method from an engineering point of view.

Besides, when dealing with the task of buffer sizing for wireless devices new problems arise. As it will be described, this is mainly due to the fact that the node transmission state does not depend only on itself, but also on the state of the other nodes inside the same collision domain. This work also extends the maximum entropy method for wireless devices working over shared channels. We provide to these devices the capability of dynamically self-configure its buffer sizes according to the traffic load variation and keeping bounded the packet loss probability. Extensive simulations have been done to verify the proper performance of the proposal. We evaluate different scenarios varying the network topology and load conditions. The analysis with classical WLAN interfaces is done for two kinds of load variations. In the first scenario the variations are due to changes in the traffic generated precisely by the node whose buffer is being sized. In the second case the load fluctuation is due to the activation or deactivation of other nodes in the same collision domain. Finally, when the mechanism is evaluated in multi-hop mesh interfaces, the load variation comes from real movie traces and the multi-hop effect.

In summary, the purpose of this contribution is to provide to devices which transmit over shared channels a straightforward method, based on easily measured parameters, to self-configure and efficiently manage their available memory. This is achieved by dynamically adapting their buffer sizes according to the traffic load variation during the network operation.

The method is based on the application of the maximum entropy principle.

The rest of this chapter is organized as follows. Section 4.2 reports and analyzes the related work. In Section 4.3 we present, in a practical way, our approximation to the solution of the G/G/1 and G/G/1/K queues via maximum entropy. Here, we start presenting some known expression relating the distribution of packets in the system to the distribution of packets found in the system by an arrival. We continue with the maximum entropy approach for the computation of the state probabilities. This section ends with the application that motivated this study, namely: buffer sizing for the G/G/1/K queuing system. The numerical evaluation of the model is presented in Section 4.4. The application and evaluation of the method for dedicated channels is presented in Section 4.5. In Section 4.6 we extend the method for wireless devices transmitting over shared channels and Section 4.7 presents the results obtained from different wireless scenarios. Finally, conclusions are drawn in Section 4.8.

4.2 Related Work

The task of dimensioning buffer sizes can be developed in two ways: once at the design stage or dynamically in order to adapt that size to the variability of the network and traffic conditions. A recent approach of the first type is presented in [109] where authors propose a large deviations framework to dimension the buffer size of delay-tolerant network nodes, as, for instance, VANETs (Vehicular Ad-hoc NETworks) or ICMNs (Intermittently Connected Mobile Networks) nodes. Here, each node is modeled as an M/M/1/B queue and large deviations theory is used to study the queue buffer sizes in terms of buffer overflow. Authors consider a buffer loss probability exponent as the configuration parameter and evaluate the performance of their approach in terms of delivery ratio, delivery delay and message loss ratio. They state that their sized buffer model, with the adequate configuration parameter, offers a performance statistically equivalent to the infinite buffer model. Another example of the importance of buffers in such wireless networks is given in [110]. Here, authors show the impact of the buffer size on packet loss probability, throughput and delay of IEEE 802.16 networks. They also observe that beyond a certain threshold, larger buffers do not improve none of these performance parameters.

On the other hand, several authors propose dynamic buffer sizing mechanisms. For instance, authors in [106] remark the necessity of such a dynamic mechanism for wireless local area networks in order to guarantee high throughput efficiency and reasonable low end-to-end delays. They analyze a simple adaptation of the classic bandwidth-delay product (BDP) rule [102]. This well-known rule, based on the dynamics of the TCP's congestion control mechanism, states that an Internet router requires a buffer size B given by $B = C \times RTT$ to achieve a hundred percent utilization at the bottleneck links. Here, C represents the link data rate and RTT is the average round-trip time of a TCP flow passing through that link. Following this rule, and taking into account the currently high possible values of C, impractically large buffers may be obtained. This was first observed by [104] where authors showed that a link with n long-lived or short-lived TCP flows requires only $B = (C \times RTT)/\sqrt{n}$ buffers, and further analyzed in [111] for congested links with different TCP flow types. Some open issues are presented by the same authors in [112]. The adaptation proposed by [106] consists of an on-line measurement and actualization of the mean packet service time values. This is required since, in contrast to wired networks in which this value is constant, for 802.11-based wireless networks the service time depends on the number of active stations that contend for the channel

(CSMA/CA mechanism stochastic effect) and on the varying modulation and coding scheme chosen by the physical layer (which in turn depends on the radio channel conditions). For this first approach, they use a maximum queuing delay as the configuration parameter. In second place they propose the Adaptive Limit Tuning (ALT) algorithm which main idea is to decrease the buffer size when it has been busy for a long time and increase the buffer size when it has been idle for a long period. The aim is to take advantage of the statistical multiplexing of TCP congestion window backoffs when multiple flows share the same link.

In the same line, authors in [113] propose a buffer sizing mechanism in order to reduce the queuing delays of TCP multi-hop flows while maintaining high network utilization inside 802.11-based WMNs. Their main idea is to consider a joint neighborhood buffer distributed over a set of nodes that contend for channel access within a collision domain. The cumulative buffer for the collision domain is also determined using the classical bandwidth-delay product. To distribute the collective buffer amongst neighborhood nodes, they establish a simple cost function that takes into account the fact that a queue drop close to the source node wastes fewer network resources than a queue drop in a node closer to destination.

Different approaches based on modern control theory are presented in [114] and [115]. In [114], authors show the monotonic relationship between the buffer size and packet loss rate and utilization. This relationship states that if the buffer size increases then the loss rate monotonically decreases, while the utilization monotonically increases. Based on these monotonic relationships the authors propose the so called Adaptive Buffer Sizing (ABS) mechanism. ABS consists of two integral sub-controllers that dynamically adapt to variations of input traffic by regulating the buffer size. This regulation is based on the error between the measured and target values of loss rate and utilization. The controller under the utilization constraint requires an additional treatment to avoid the buffer size goes to infinity in non-bottleneck routers, since they are always under-utilized regardless of its buffer size. Finding the optimal values for integral gains, which control the convergence speed and stability of the system, is not an easy task due to the unknown underlying model for Internet traffic and its high time-variability. To solve this, authors combine a controller output error method with the gradient descent technique. As a final result they associate each sub-controller with a gradient-based parameter training component able to find optimal integral gain values. As a consequence, the ABS routers can adapt their buffer size to meet the required loss rate and utilization under variable traffic conditions. In the same line, in [115] a control-theoretic approach to analyze the network stability is provided. Authors show that desynchronized flows improve network stability and require smaller buffers, which in turn promote desynchronization.

When the research focuses on inelastic real-time offered services, UDP flows should be taken into account. Although TCP dominates the traffic carried by the Internet backbone links, recent works [116, 117] demonstrate a significantly growth of UDP traffic (up to 46-fold increase in volume in the past four years). Besides real time applications, some recent peer-to-peer data transfers contribute to this growth. Authors in [117] actually show that UDP represents the largest fraction of flows on a given link. Further, a self-contained mesh network (MBSS) [53] allows that real time applications can dispense with not only TCP but even with IP. These considerations have motivated to focus our work on inelastic real time traffic which is usually encapsulated over UDP. Note that not all the classical real-time services fall into this category. For instance, in most video streaming systems, the video sequence is previously coded and stored. This fact, together with the increasing available bandwidth and a big amount of memory in the receiver, allows sending the video sequence faster than needed and storing it at the receiver side. In summary, video streaming service has become a kind of file transfer

service. The strictest real-time services are those on which the information is consumed at the same time that it is generated. For these flows, one of the most important aspects to study is the packet loss rate, since it is directly related to the quality of service that the end user will experience. Moreover, there are two main possibilities for the traffic engineering in a queue system. Different flows from different service classes (with different traffic characteristics and requirements) could coexist in the same queue, or be allocated to independent queues. The increasing tendency of service differentiation allows that a variety of applications and services meet their specific QoS needs. Therefore, in this work we have considered the presence of a traffic classifier that separates at least TCP and UDP flows and assigns them to different queues.

Traditionally, the way to analytically study the loss probability is through the transmission system modeling, especially the G/G/1/K queue. The loss probability can be determined by obtaining the state probabilities in these queues. However, to find these set of probabilities, it is required a detailed knowledge of the traffic to be transmitted which is seldom available. This work adopts an approximation in which just a few average values are available to carry out the buffer dimensioning. This approximation is based on the idea of maximum entropy, which was presented in an exhaustive way for the G/G/1 queue in [108]. The same author studies the G/G/1/K queue in [118], and had previously presented the solution for the M/G/1 and G/M/1 queues in [119]. For their part, authors in [120] use the maximum entropy analysis in their study of a single removable server queueing system operating under the N policy (that is, the server is turned on whenever N or more customers are present, and turned off when there are no customers). They perform a comparative analysis between the approximate results obtained through the maximum entropy principle and the exact results obtained in previous works. The main conclusion is that error percentages are very small and therefore the use of the maximum entropy principle is accurate enough for practical purposes. The same authors extend in [121] their work to a M/G/1 queuing system with unreliable server and general startup times. The conclusions are very similar to the previous ones, confirming the utility of the maximum entropy principle. This principle has also been used in some related fields as, for instance, the detection of anomalies in the network traffic [122]. Here, the current network traffic distribution is compared against a baseline traffic distribution which is estimated by means of the maximum entropy technique.

This chapter includes the solution for the G/G/1 and G/G/1/K queues via maximum entropy, obtained in a practice-oriented way. The derivation is rather elementary and intuitive, but we think this approach will appeal to engineers and computer scientists, more familiar with the simulation aspects of a performance evaluation problem than with the intricacies of mathematical arguments.

In summary, the contributions of this work are the following:

- The development of an alternative and easy to follow explanation for the solution of the G/G/1 and G/G/1/K queues using a maximum entropy approach and its application to the dynamic buffer sizing problem.

- The verification of the proper functioning of the resulting analytical queue models through simulation.

- The adaptation of the proposed dynamic buffer sizing mechanism in order to be applied on devices working over shared channels.

- The implementation of the buffer sizing algorithm in ns-2 [123] and ns-3 [59] simulators and the verification of its performance under different scenarios and traffic conditions.
- And finally the definition and evaluation of some metrics that allows the quantification of the system performance in terms of efficiency in memory utilization.

As it will be verified, the obtained results could be valuable in network devices with constrained resources such as sensor nodes, or for an efficient memory resource management in wireless mesh routers with multiple interfaces. On the other hand, when the available memory resources are not a problem, the dynamic buffer sizing also contributes to prevent the high latency and network performance degradation due to the overbuffering issue.

4.3 Approximation to the Solution of the G/G/1 and G/G/1/K Queues Via Maximum Entropy: a Practical Approach

This section presents the method for the solution of the G/G/1 and G/G/1/K queues via maximum entropy in an easy and comprehensive way. As said previously, the derivation is rather elementary and intuitive. First of all, a vision of the system dynamics is presented. We follow with the computation of state probabilities via maximum entropy in both the infinite and finite buffer size cases. Finally, we present different methods to find the buffer size for a given loss probability target and show some experimental results.

4.3.1 Packets in the System and Packets Seen by Arrivals and Departures

Consider a transmission system in which packets arrive and depart one at a time. In steady state, such a system evolves through periods of activity and idleness as represented in Fig. 4.1. When the system is continuously observed over a long enough time span comprising many of those cycles, t_{obs}, the probability of having i packets in the system, $p(i)$, can be computed as:

$$p(i) = \frac{t(i)}{t_{\text{obs}}} \tag{4.1}$$

where $t(i)$ is the total amount of time the system has sojourned in state i (during t_{obs}). Similarly, the probability that the system is busy, ρ, is given by:

$$\rho = \frac{t_B}{t_{\text{obs}}} \tag{4.2}$$

where t_B is the total duration of the activity periods, i.e.:

$$t_B = \sum_{i=1}^{\infty} t(i) \tag{4.3}$$

To evaluate the probability of packets found in the system by an arrival, $a(i)$, one has to proceed differently, since now measurements are only taken at precise time instants. If there

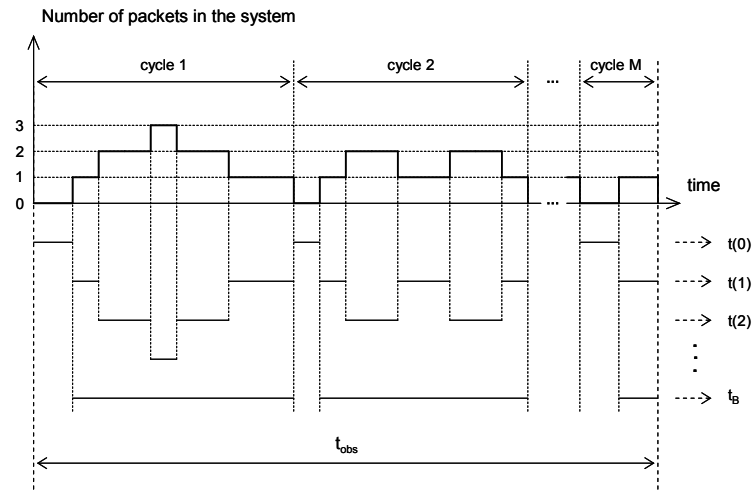


Figure 4.1: Pictorial definition of $t(i)$ and t_B.

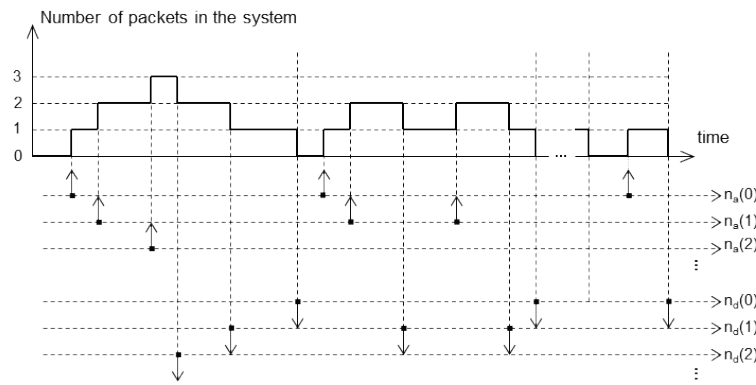


Figure 4.2: Pictorial definition of $n_a(i)$ and $n_d(i)$.

are n_a arrivals in t_{obs}, and $n_a(i)$ of those see the system in state i, $a(i)$ can be estimated as (see Fig. 4.2):

$$a(i) = \frac{n_a(i)}{n_a} \tag{4.4}$$

Note that now we are dealing with discrete quantities, as opposed to the continuous values used to estimate $p(i)$. Therefore, it should not be surprising that, in general, $p(i)$ and $a(i)$ differ.

Analogously, denoting by $d(i)$ the probability of leaving behind i packets at service completion, we can write:

$$d(i) = \frac{n_d(i)}{n_d} \tag{4.5}$$

where n_d and $n_d(i)$ are, respectively, the total number of departures and the number of those departures that leave i packets in the system. From Fig. 2, it is apparent that $n_d = n_a$ and $n_d(i) = n_a(i)$. (Observe that these equalities hold also true for every cycle). Therefore, we also have $d(i) = a(i)$.

The probabilities $p(i)$ and $a(i)$ are related through arrival and departure rates, defined respectively as follows:

$$\begin{aligned} \lambda(i) &= \frac{n_a(i)}{t(i)} \\ \mu(i) &= \frac{n_d(i)}{t(i+1)} \end{aligned} \tag{4.6}$$

The global arrival rate, λ, is:

$$\lambda = \frac{n_a(0) + n_a(1) + \cdots}{t(0) + t(1) + \cdots} = \frac{n_a}{t_{\text{obs}}} \tag{4.7}$$

Similarly, the global departure ratio, μ, is:

$$\mu = \frac{n_d(0) + n_d(1) + \cdots}{t(1) + t(2) + \cdots} = \frac{n_d}{t_B} \tag{4.8}$$

From all the above, we readily obtain the following relationships which will be useful in the sequel:

$$a(i) = \frac{n_a(i)}{n_a} = \frac{\lambda(i)t(i)}{\lambda t_{\text{obs}}} = \frac{\lambda(i)}{\lambda} p(i), \qquad i = 0, 1, 2, \ldots \tag{4.9}$$

$$p(i+1) = \frac{t(i+1)}{t_{\text{obs}}} = \frac{n_d(i)\lambda}{n_a \mu(i)} = \frac{\lambda}{\mu(i)} a(i), \qquad i = 0, 1, 2, \ldots \tag{4.10}$$

and from both, we finally have:

$$p(i+1) = \frac{\lambda}{\mu(i)} \frac{\lambda(i)}{\lambda} p(i) = \frac{\lambda(i)}{\mu(i)} p(i), \qquad i = 0, 1, 2, \ldots \tag{4.11}$$

4.3.2 The computation of probabilities: a maximum entropy approach

We will distinguish two cases, corresponding to infinite or finite buffer size.

4.3.2.1 Infinite Buffer Size

As anticipated in the introduction, the problem we address in this section is the following. Suppose that the average number of packets in a transmission system (buffer and server) and the server occupancy have both been measured. As explained before, such measurements may yield different values depending on the way they are taken: at arrival times or by continuously monitoring the system. Both ways of proceeding are used in engineering practice. Time averages are very useful for operators, for instance, since they clearly indicate the utilization of network resources. The knowledge of these values is pertinent to many aspects of paramount importance like economic revenues, equipment heating and, in general, subject matters related to network exploitation. On the other hand, users are more concerned with a different set of parameters, like packet losses, delays, etc., that, in the usual parlance of traffic engineering are grouped under the heading of Quality of Service. This work focuses on buffer dimensioning to achieve a given packet loss probability, P_L. Obviously, a packet is lost when, upon its arrival to the transmission system, no storage space is left in the buffering element. Therefore, to evaluate P_L, measurements should be taken at arrival times. Since P_L is normally a very small number, its reliable estimation is a lengthy process that needs many packets. To avoid excessive burden on the measuring mechanism, and to make the adjustments more dynamic, our model requires the collection of only two values, namely: the server occupancy, ρ_a, and the average number of packets in the system, N_a, both of them seen by arrivals. Note that ρ_a and N_a are several orders of magnitude larger than P_L, and therefore they can be estimated much faster, using for instance moving averages. Then, the problem is to evaluate the probabilities of packets seen by an arrival, $a(i)$, having at our disposal only the knowledge of those two quantities. If the $a(i)$ were given, ρ_a and N_a would follow readily. Our aim is just the opposite, i.e.: to obtain the $a(i)$ from the sole knowledge of ρ_a and N_a. Observe that by producing $a(i)$ we are providing more information than we have available. We should, therefore, be as "ambiguous" as possible, but always in compliance with the observed data. Since the pioneering work of C. Shannon [124], the uncertainty of a random variable (rv) is measured by a quantity known as entropy. Focusing in our case of interest, if a rv takes on the values $x_1, x_2, x_3, \ldots$ with probabilities $p_1, p_2, p_3, \ldots$ the entropy of this rv is given by the expression [125]:

$$\sum_{i=1}^{\infty} p_i \ln \frac{1}{p_i} \tag{4.12}$$

The base of logarithms can be arbitrary and, by convenience, in the above expression, we have chosen natural logarithms.

Going back to our case, since $a(0) = 1 - \rho_a$ is known, the problem can be stated as follows: Compute $a(1), a(2), a(3), \ldots$ to maximize:

$$\sum_{i=1}^{\infty} a(i) \ln \frac{1}{a(i)} \tag{4.13}$$

subject to conditions:

$$\begin{aligned} \sum_{i=1}^{\infty} a(i) &= \rho_a \\ \sum_{i=1}^{\infty} i a(i) &= N_a \end{aligned} \tag{4.14}$$

Using the approach routinely employed to solve constrained optimization problems of this sort, we form the Lagrangean function [126]:

$$F\left(a(1), a(2), \ldots\right) = -\sum_{i=1}^{\infty} a(i) \ln a(i) + A\left(\sum_{i=1}^{\infty} a(i) - \rho_a\right) + B\left(\sum_{i=1}^{\infty} i a(i) - N_a\right) \tag{4.15}$$

and equal its partial derivations to 0, as follows:

$$\frac{\partial F}{\partial a(i)} = -\left(\ln a(i) + 1\right) + A + Bi = 0, \qquad i = 1, 2, \ldots \tag{4.16}$$

Therefore

$$a(i) = e^{A-1+Bi} = \alpha\beta^i, \qquad i = 1, 2, \ldots \tag{4.17}$$

where

$$\alpha = e^{A-1}, \qquad \beta = e^B \tag{4.18}$$

To compute α and β we use the known conditions:

$$\rho_a = \sum_{i=1}^{\infty} a(i) = \sum_{i=1}^{\infty} \alpha\beta^i = \frac{\alpha\beta}{1-\beta} \tag{4.19}$$

which implies

$$\alpha = \rho_a \frac{1-\beta}{\beta} \tag{4.20}$$

Also

$$N_a = \sum_{i=1}^{\infty} i a(i) = \sum_{i=1}^{\infty} i\alpha\beta^i = \frac{\alpha\beta}{(1-\beta)^2} = \frac{\rho_a}{1-\beta} \tag{4.21}$$

Then,

$$\beta = 1 - \frac{\rho_a}{N_a} \tag{4.22}$$

and therefore from Eq. 4.20:

$$\alpha = \frac{\rho_a^2}{N_a} \frac{1}{1 - \frac{\rho_a}{N_a}} \tag{4.23}$$

Once α and β have been determined, the values of $a(i)$ are given by the expressions:

$$\begin{aligned} a(0) &= 1 - \rho_a \\ a(i) &= \frac{\rho_a^2}{N_a}\left(1 - \frac{\rho_a}{N_a}\right)^{i-1}, \qquad i = 0, 1, 2, \ldots \end{aligned} \tag{4.24}$$

Proceeding similarly with $p(i)$, we can write:

$$\begin{aligned} p(0) &= 1 - \rho \\ p(i) &= \frac{\rho^2}{N}\left(1 - \frac{\rho}{N}\right)^{i-1}, \qquad i = 0, 1, 2, \ldots \end{aligned} \tag{4.25}$$

where ρ and N are, respectively, the server occupancy and the average number of packets obtained by continuously monitoring the system.

To further illustrate the interest of the procedure, we will now apply the method to four different cases depending on the constancy or not of the arrival and departure rates.

Case 1: $\lambda(i)$ variable and $\mu(i)$ variable.

This is the general case. There are no relations among ρ, ρ_a, N, N_a, and nothing else can be said.

Case 2: $\lambda(i)$ constant $(= \lambda)$ and $\mu(i)$ variable.

Since

$$a(i) = \frac{\lambda(i)}{\lambda} p(i) \tag{4.26}$$

in this case we have

$$a(i) = p(i) \tag{4.27}$$

and therefore

$$\rho_a = \rho, \qquad N_a = N \tag{4.28}$$

Case 3: $\lambda(i)$ variable and $\mu(i)$ constant $(= \mu)$.

In this case

$$\rho_a \neq \rho, \qquad N_a \neq N \tag{4.29}$$

but nonetheless

$$\frac{\rho_a}{\rho} = \frac{N_a}{N} \tag{4.30}$$

To see this, observe first from Eqs.(4.2, 4.7, and 4.8) we have

$$\frac{\lambda}{\mu} = \frac{t_B}{t_{\text{obs}}} = \rho \tag{4.31}$$

Now

$$p(i+1) = \frac{\lambda}{\mu(i)} a(i) = \rho a(i) \tag{4.32}$$

From the above and Eq. 4.25 we get

$$\rho_a = 1 - a(0) = 1 - \frac{1}{\rho} p(1) = 1 - \frac{\rho}{N} \tag{4.33}$$

and

$$N_a = \sum_{i=1}^{\infty} i a(i) = \sum_{i=1}^{\infty} i \frac{1}{\rho} p(i+1) = \sum_{i=1}^{\infty} i \frac{1}{\rho} \left(1 - \frac{\rho}{N}\right) p(i) = \frac{N}{\rho} - 1 \tag{4.34}$$

From Eqs.(4.33 and 4.34) we obtain

$$N\rho_a = N - \rho, \qquad N_a \rho = N - \rho \tag{4.35}$$

and therefore

$$N\rho_a = N_a \rho \tag{4.36}$$

as claimed

Case 4: $\lambda(i)$ constant and $\mu(i)$ constant.

In this case, not only $\rho = \rho_a$ and $N = N_a$, but also N and ρ are related. In fact, from

$$\rho_a = 1 - \frac{\rho}{N} = \rho \tag{4.37}$$

we get

$$N = \frac{\rho}{1-\rho} \tag{4.38}$$

Observe that some of the equations presented in cases 2, 3 and 4 correspond, respectively, to results derived when the usual procedure to studying the M/M/1, M/G/1, and G/M/1 queuing systems is followed. The formulation is, however, different, and the expressions obtained for the general case (from which all other stem) are new, and have no equivalent in the traditional approach to the analysis of the G/G/1 queuing system.

4.3.2.2 Finite Buffer Size

Let Q be the buffer size. Similarly to before, we have:

$$\begin{aligned} a_Q(0) &= 1 - \rho_a \\ a_Q(i) &= \alpha\beta^i, \qquad i = 0, 1, 2, \ldots, Q+1 \end{aligned} \tag{4.39}$$

where $a_Q(i)$ denotes the probability seen by an arrival when the buffer size is Q. The parameters α and β are now obtained from ρ_a and N_a as indicated below.

From

$$\rho_a = \sum_{i=1}^{Q+1} a_Q(i) = \sum_{i=1}^{Q+1} \alpha\beta^i \tag{4.40}$$

we obtain:

$$\alpha = \rho_a \frac{1-\beta}{\beta} \frac{1}{1-\beta^{Q+1}} \tag{4.41}$$

The computation of N_a yields:

$$N_a = \sum_{i=1}^{Q+1} i a_Q(i) = \sum_{i=1}^{Q+1} i\alpha\beta^i = \frac{\rho_a}{1-\beta} \frac{1 - [(Q+2) - (Q+1)\beta]\,\beta^{Q+1}}{1-\beta^{Q+1}} \tag{4.42}$$

Observe that when Q increases to infinity we re-obtain Eqs.(4.20 and 4.21).

Eq.(4.42) can be rewritten as:

$$\frac{N_a}{\rho_a} = f_Q(\beta) \tag{4.43}$$

where

$$f_Q(\beta) = \frac{1}{1-\beta} \frac{1 - [(Q+2) - (Q+1)\beta]\,\beta^{Q+1}}{1-\beta^{Q+1}} \tag{4.44}$$

is an increasing function of β and $f_Q(0) = 1$ (see Fig. 4.3). Therefore, since $(N_a/\rho_a) > 1$, Eq.(4.44) can be easily solved for β, and α readily follows from Eq.(4.41).

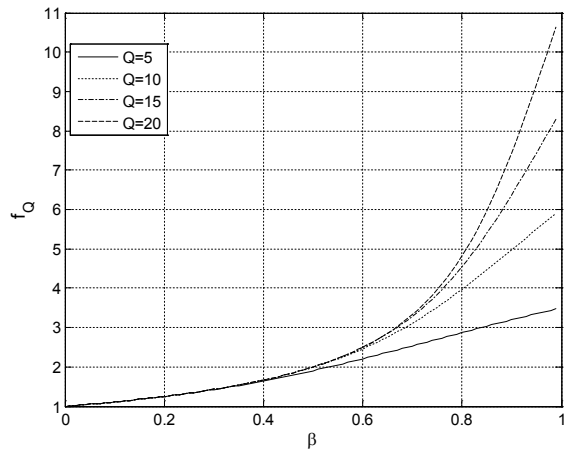


Figure 4.3: Calculation of β.

4.3.3 Application to the Buffer Sizing for the G/G/1/K Queuing System

Buffer overflow in data networks causes packet losses and, consequently, it should be evaluated and properly controlled to guarantee the desired level of service performance. Obviously, a packet is lost when, upon its arrival to the transmission system, no storage space is left in the buffering element. Therefore, as said before, to evaluate this parameter, the use of the probabilities of packets found in the system by an arrival (and not the probability of packets in the system) is in order. Based on the previous formulas, we propose and compare three different methods. For the first method we use the finite buffer model presented in Section 4.3.2.2, whereas for methods 2 and 3 we employ Eq.(4.24) derived in Section 4.3.2.1.

Method 1: The probability of packet loss, P_L, can be evaluated as the probability of being in the state K, that is, $Q+1$.

$$P_L = a_Q(Q+1) = \alpha\beta^{Q+1} \tag{4.45}$$

where α and β are computed as indicated in Section 4.3.2.2. Then, for a given P_L, the buffer size Q can be computed as:

$$Q = \log_\beta\left(\frac{P_L}{\alpha}\right) - 1 \tag{4.46}$$

On the other hand, the packet loss probability can also be calculated using the probabilities $a(i)$ obtained in Section 4.3.2.1 for the infinite buffer size. Here, we present two methods, akin to the ones used to compute blocking probabilities for voice circuits.

Method 2: Define:

$$\tilde{a}(i) = \begin{cases} \dfrac{a(i)}{\sum_{n=0}^{Q+1} a(n)}, & i \leq Q+1 \\ 0, & i > Q+1 \end{cases} \tag{4.47}$$

Then, P_L can be computed as follows:

$$P_L = \tilde{a}(Q+1) = \frac{a(Q+1)}{1 - \sum_{n=Q+2}^{\infty} a(n)} = \frac{\rho_a^2}{N_a} \cdot \frac{\left(1 - \frac{\rho_a}{N_a}\right)^Q}{1 - \rho_a \left(1 - \frac{\rho_a}{N_a}\right)^{Q+1}} \tag{4.48}$$

Observe that this method of truncation is similar to the Erlang_B approach to compute blocking probabilities for voice circuits (truncating, in that case, a Poisson distribution).

Method 3: We now parallel Molina's [127] way of computing blocking probabilities. Therefore, we set:

$$P_L = \sum_{i=Q+1}^{\infty} a(i) = \sum_{i=Q+1}^{\infty} \frac{\rho_a^2}{N_a} \left(1 - \frac{\rho_a}{N_a}\right)^{i-1} = \rho_a \left(1 - \frac{\rho_a}{N_a}\right)^Q \tag{4.49}$$

With this method Q can be expressed in closed-form as:

$$Q = \text{int}\left(\frac{\ln\left(\frac{P_L}{\rho_a}\right)}{\ln\left(1 - \frac{\rho_a}{N_a}\right)} \right) \tag{4.50}$$

where $\text{int}(\cdot)$ is the integer function.

As shown later, Eqs.(4.45) and (4.48) give similar results for the value of the needed buffer size. The value of Q provided by Eq.(4.50) is however larger than necessary, but the advantage is that Q can be expressed in closed-form.

In the next section we comment on the numerical results obtained using different traffic aggregations.

4.4 Numerical and Simulation Results

This section gives a few representative figures obtained by simulation and using the procedure and methods described in sections 4.3.2 and 4.3.3. Although the main interest of this contribution is the computation of buffer overflow using to that end the probabilities seen by arrivals, we have also indicated how to evaluate the probabilities of packets in the system, and pointed out the differences between the two of them. To more visually highlight those differences, in Fig. 4.4 we have shown the values of $p(i)$ and $a(i)$ for infinite buffers and two specific distributions of interarrival and service times. More specifically, we have chosen a Gamma(α, ν) distribution for the interarrival times and a Pareto(k, x_m) for packet lengths.

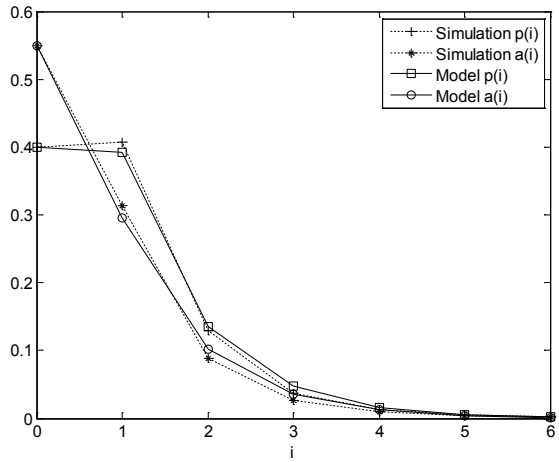


Figure 4.4: Probability of packets in the system, $p(i)$, and probabilities seen by arrivals $a(i)$, with infinite queue size.

The probability density function for the Gamma(α, ν) is given by:

$$f(x) = \frac{x^{\nu-1}}{\alpha^{\nu}\Gamma(\nu)} e^{-\frac{x}{\alpha}}, \qquad x > 0 \tag{4.51}$$

where $\alpha > 0, \quad \nu > 0$ and $\Gamma(\nu)$ is the gamma function. Its average value is $\nu\alpha$.

The density function for the Pareto(k, x_m) is:

$$f(x) = k\frac{x_m^k}{x^{k+1}}, \qquad x > 0 \tag{4.52}$$

where $x_m > 0$ and $k > 1$. Its average value is $kx_m/(k-1)$.

Observe that one of the distributions considered for the packet lengths is the Pareto distribution, which takes into account the "heavy-tailed" effect present in the self-similar traffic [128, 129]. Anyway, although self-similarity property appears in many real traffics [130, 94], it should be noted that its influence in some environments or with certain services (for instance, real time services with small buffers) is not determinant [131, 132].

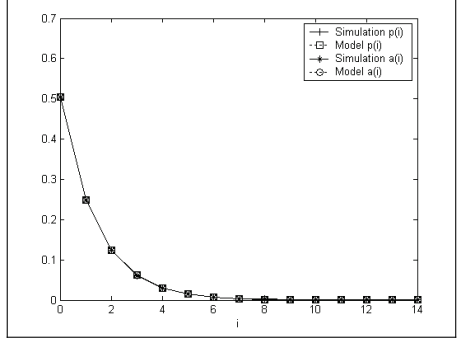
The parameters chosen to compute the model and run the simulations are shown in Table 4.1. For a 1 Mbps channel, this produces a load of 0.6. Obviously, many other parameters and distributions could have been chosen and the results were similar. In fact, eight additional cases are shown in Figs. 4.5–4.12: the results for infinite buffer size with different interarrival and service time distributions (M/M/1, M/G/1, G/M/1, G/G/1), and for different offered traffic intensities ($\rho = 0.5$ and $\rho = 0.8$) are respectively shown in Figs. 4.5–4.8, while the results for the finite buffer cases ($N = 4$ and $N = 8$) with the same time distributions and a traffic intensity of $\rho = 0.8$ are shown in Figs. 4.9–4.12. Observe how, as expected, the probabilities $p(i)$ and $a(i)$ differ, but the values provided by the model accurately agree with the results of the simulation runs.

Since we are interested in buffer overflow, for the rest of the computations and figures we are only concerned with the value of the probabilities seen by arrivals. To more realistically mimic

Table 4.1: Traffic configuration parameters.

Function	**Parameters**	
Interarrival time (gamma)	$\alpha = 0.001$ s	$\nu = 2$
Packet length (Pareto)	$x_m = 800$ bits	$k = 3$

Interarrival time [s]	Service time [s]	ρ
Exp(0.2)	Exp(0.1)	0.5



Interarrival time [s]	Service time [s]	ρ
Exp(0.2)	Exp(0.16)	0.8

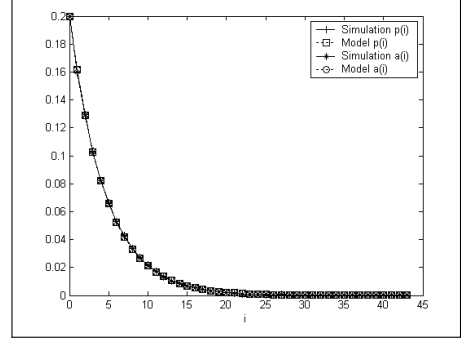
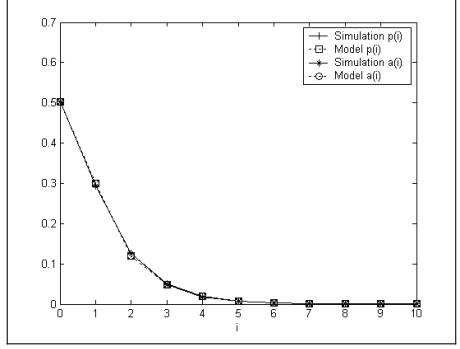


Figure 4.5: $p(i)$ and $a(i)$ for infinite buffer size and time distribution M/M/1.

Interarrival time [s]	Service time [s]	ρ
Exp(0.2)	$\text{Erlang}_3(0.1)$	0.5



Interarrival time [s]	Service time [s]	ρ
Exp(0.2)	$\text{Erlang}_3(0.16)$	0.8

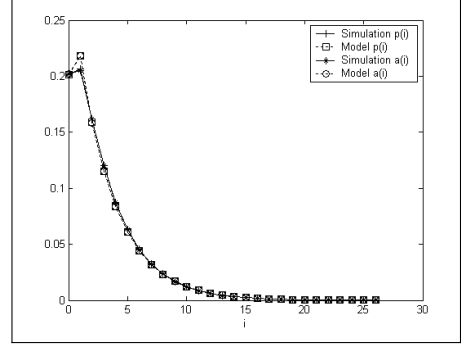
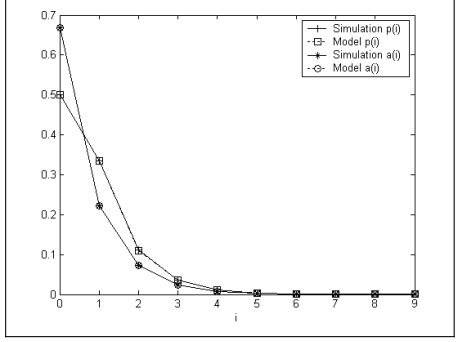


Figure 4.6: $p(i)$ and $a(i)$ for infinite buffer size and time distribution M/G/1.

Interarrival time [s]	Service time [s]	ρ
$\text{Erlang}_3(0.2)$	Exp(0.1)	0.5



Interarrival time [s]	Service time [s]	ρ
$\text{Erlang}_3(0.2)$	Exp(0.16)	0.8

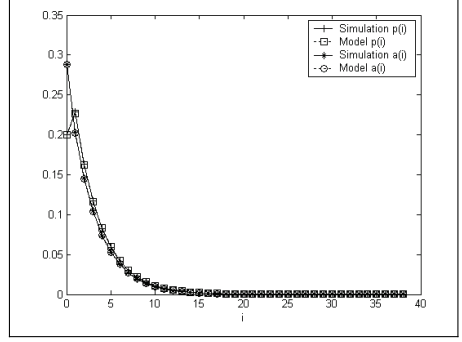
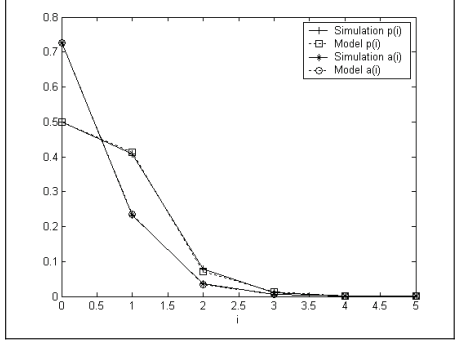


Figure 4.7: $p(i)$ and $a(i)$ for infinite buffer size and time distribution G/M/1.

Interarrival time [s]	Service time [s]	ρ
$\text{Erlang}_3(0.2)$	$\text{Erlang}_3(0.1)$	0.5



Interarrival time [s]	Service time [s]	ρ
$\text{Erlang}_3(0.2)$	$\text{Erlang}_3(0.16)$	0.8

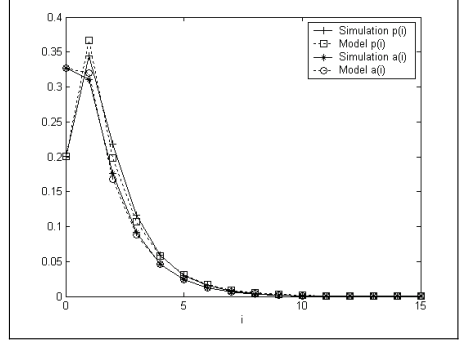
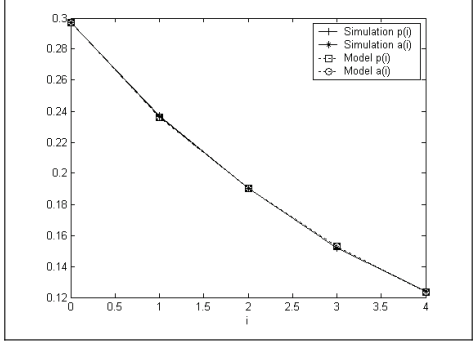


Figure 4.8: $p(i)$ and $a(i)$ for infinite buffer size and time distribution G/G/1.

Interarrival t [s]	Service time [s]	ρ	N
Exp(0.2)	Exp(0.16)	0.8	4



Interarrival t [s]	Service time [s]	ρ	N
Exp(0.2)	Exp(0.16)	0.8	8

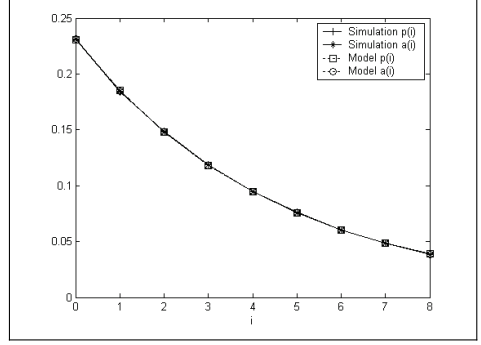
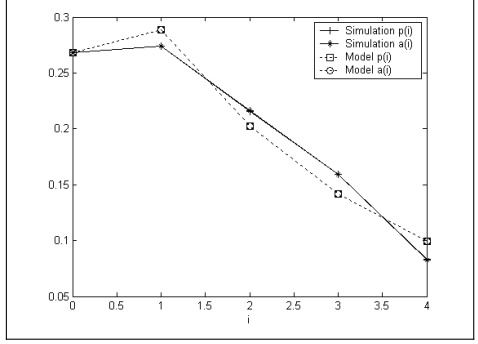


Figure 4.9: $p(i)$ and $a(i)$ for finite buffer $N = 4$ (left) and $N = 8$ (right) and time distribution M/M/1/N.

Interarrival t [s]	Service time [s]	ρ	N
Exp(0.2)	$\text{Erlang}_3(0.16)$	0.8	4



Interarrival t [s]	Service time [s]	ρ	N
Exp(0.2)	$\text{Erlang}_3(0.16)$	0.8	8

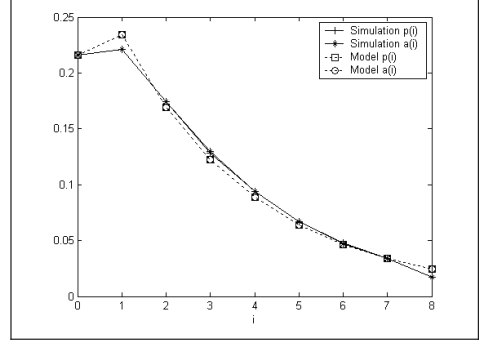
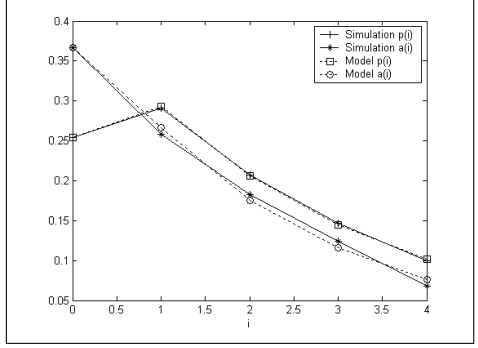


Figure 4.10: $p(i)$ and $a(i)$ for finite buffer $N = 4$ (left) and $N = 8$ (right) and time distribution M/G/1/N.

Interarrival t [s]	Service time [s]	ρ	N
$\text{Erlang}_3(0.2)$	Exp(0.16)	0.8	4



Interarrival t [s]	Service time [s]	ρ	N
$\text{Erlang}_3(0.2)$	Exp(0.16)	0.8	8

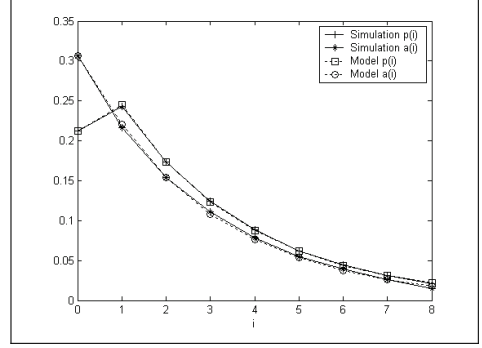


Figure 4.11: $p(i)$ and $a(i)$ for finite buffer $N = 4$ (left) and $N = 8$ (right) and time distribution G/M/1/N.

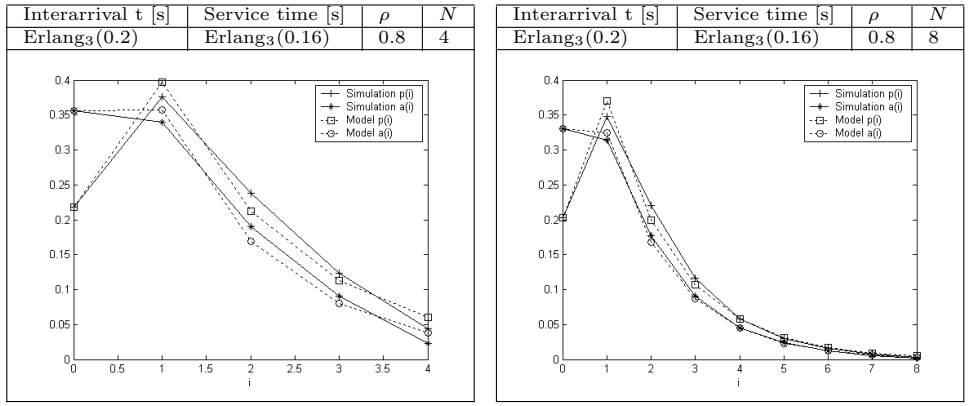


Figure 4.12: $p(i)$ and $a(i)$ for finite buffer $N = 4$ (left) and $N = 8$ (right) and time distribution G/G/1/N.

the actual traffic in any network, we have considered a traffic resulting from the aggregation of four different mixes with quite different interarrival times and packet lengths. Besides the gamma distribution, already mentioned, the other distributions that have been used are the following:

- Truncated exponential:

$$f(x) = \frac{\lambda}{1 - e^{-\lambda x_{\max}}} e^{-\lambda x}, \qquad 0 \leq x \leq x_{\max} \tag{4.53}$$

- Truncated Pareto:

$$f(x) = \frac{k}{1 - \left(\frac{x_{\min}}{x_{\max}}\right)^k} \frac{x_{\min}^k}{x^{k+1}}, \qquad x_{\min} \leq x \leq x_{\max} \tag{4.54}$$

- Uniform:

$$f(x) = \frac{1}{x_{\max} - x_{\min}}, \qquad x_{\min} \leq x \leq x_{\max} \tag{4.55}$$

- Deterministic:

$$f(x) = \delta(x - x_0) \tag{4.56}$$

  where $\delta(\cdot)$ is the Dirac's delta function.

Moreover, to fully explore the behavior under different conditions, we have compared three loads: $\rho = 0.4$, $\rho = 0.6$ and $\rho = 0.8$. Table 4.2 shows the packet length distributions and Table 4.3 the distributions of the interarrival times for the three different loads. The channel capacity is set to 1 Gbps. The analytic computations are carried out using method 1 presented in Section 4.3.3.

Fig. 4.13 represents the values (model and simulations) of $a_Q(i)$ for a buffer size of 11. It can be observed a good agreement for the whole spectrum of loads. Fig. 4.14 shows the packet

Table 4.2: Traffic streams (packet length).

Traffic No.	**Packet length**
1	Truncated Pareto $k = 1.5$ $x_{\min} = 368$ bits $x_{\max} = 12000$ bits
2	Gamma $\alpha = 500$ bits $\nu = 3$
3	Uniform $x_{\min} = 1000$ bits $x_{\max} = 9000$ bits
4	Truncated Pareto $k = 1.5$ $x_{\min} = 368$ bits $x_{\max} = 18000$ bits

Table 4.3: Traffic streams (interarrival times).

Traffic number	**Interarrival time** ($\rho = 0.4$)	**Interarrival time** ($\rho = 0.6$)	**Interarrival time** ($\rho = 0.8$)
1	Trunc. exponential $\lambda^{-1} = 9 \cdot 10^{-6}$ s $x_{\max} = 1.2 \cdot 10^{-4}$ s	Trunc. exponential $\lambda^{-1} = 6 \cdot 10^{-6}$ s $x_{\max} = 1.2 \cdot 10^{-4}$ s	Trunc. exponential $\lambda^{-1} = 4.5 \cdot 10^{-6}$ s $x_{\max} = 1.2 \cdot 10^{-4}$ s
2	Uniform $x_{\min} = 0$ s $x_{\max} = 3 \cdot 10^{-5}$ s	Uniform $x_{\min} = 0$ s $x_{\max} = 2 \cdot 10^{-5}$ s	Uniform $x_{\min} = 0$ s $x_{\max} = 1.5 \cdot 10^{-5}$ s
3	Deterministic $x_0 = 5 \cdot 10^{-5}$ s	Deterministic $x_0 = 3.333 \cdot 10^{-5}$ s	Deterministic $x_0 = 2.5 \cdot 10^{-5}$ s
4	Gamma $\alpha = 2.37 \cdot 10^{-6}$ s $\nu = 4$	Gamma $\alpha = 1.58 \cdot 10^{-6}$ s $\nu = 4$	Gamma $\alpha = 1.18 \cdot 10^{-6}$ s $\nu = 4$

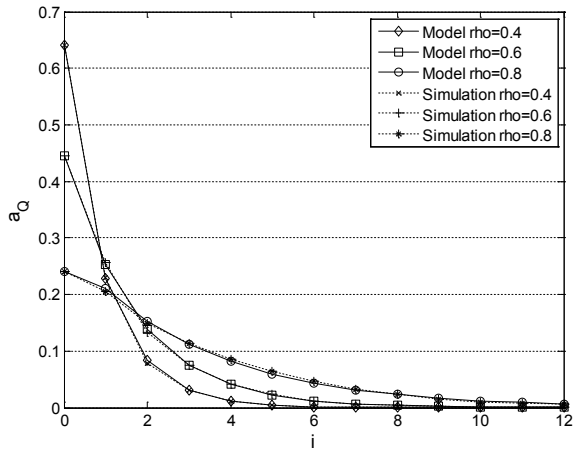


Figure 4.13: Probabilities seen by arrivals, $a_Q(i)$, with finite queue size ($Q = 11$).

loss probability as a function of the buffer size and for the three different load conditions. As usual, the values obtained with the model and by means of simulation are compared. Fig. 4.15, although similar to Fig. 4.14, shows more conveniently for a practical implementation the Q needed for a given P_L for the three loads. As can be seen, the agreement is again very good.

Finally, Figs. 4.16 and 4.17 show the results obtained using the three methods mentioned in Section 4.3.3. Not to clutter the figure, we have selected only one load: $\rho = 0.6$. From those graphs, we see the agreement of methods 1 and 2 and that, as expected, method 3 allocates more buffer than necessary.

4.5 Application to the Dynamic Buffer Sizing for Dedicated Channels

In this section we apply the maximum entropy method to dynamically adapt the buffer size of network devices in a way that the following requirements are fulfilled:

- The mechanism must assign always the smallest possible buffer size. Thus, bufferbloat effect is avoided and free memory is available to be allocated to other channels. In this way, a network device with several transmission channels is able to more efficiently distribute the available memory depending on the varying load conditions in each of their channels.

- A maximum loss probability, which can be configurable, must be guaranteed. This loss probability is determined by the nature of the traffic flow and the quality of service that the end user expects.

- The algorithm must be able to run independently on each device. Devices must be able to self-configure their buffer sizes without intervention of any other network equipment. Thus, the proposed mechanism can be used in any environment, not being required a central equipment with special features and functionalities. Consider for example the case

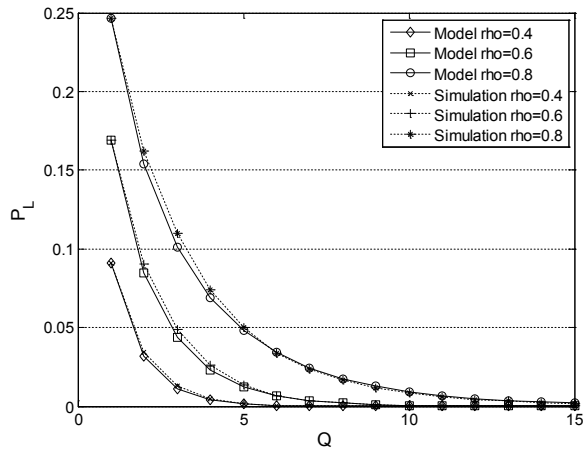


Figure 4.14: Packet loss probability vs. buffer size.

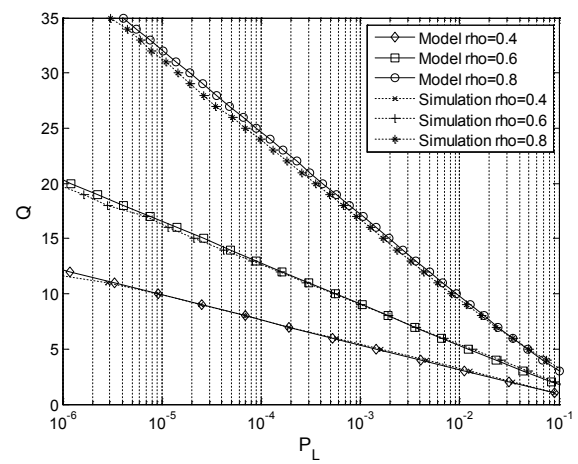


Figure 4.15: Buffer size vs. packet loss probability.

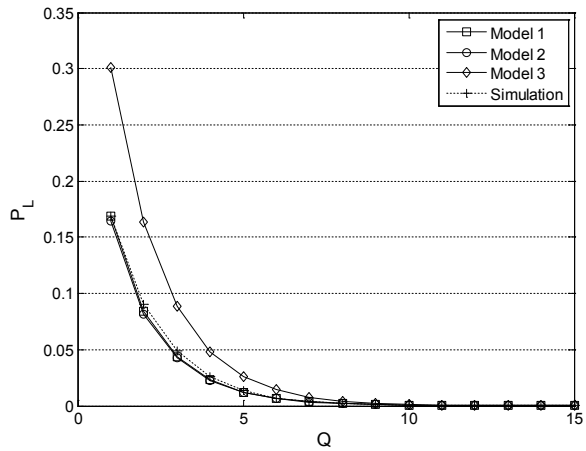


Figure 4.16: Packet loss probability vs. buffer size using the three models $\rho = 0.6$.

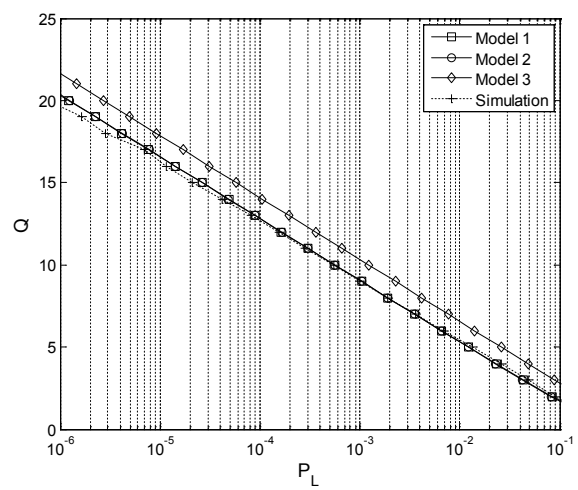


Figure 4.17: Buffer size vs. packet loss probability using the three models $\rho = 0.6$.

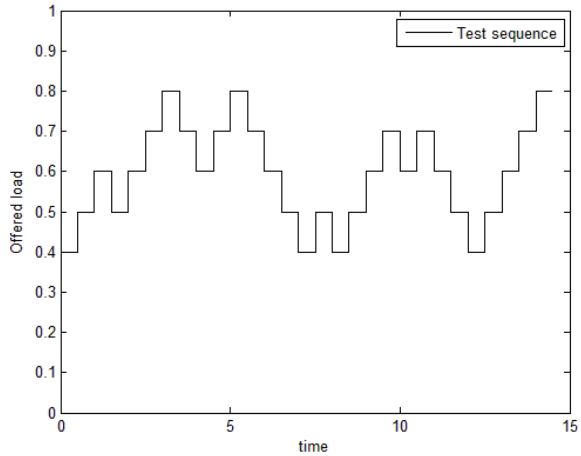


Figure 4.18: Test sequence for the simulations.

Table 4.4: Random variables used to generate the traffic flows.

Flow	Interarrival time	Packet lengths
1	Truncated exponential	Truncated Pareto
2	Uniform	Gamma
3	Deterministic	Uniform
4	Gamma	Truncated Pareto

of wide-area networks consisting of heterogeneous devices from different manufacturers and using wired and wireless channels. The requirement of a central equipment along with new protocols for information exchange relating to the dynamic buffer sizing, implies serious challenges from a practical implementation point of view. In addition, we must consider the traffic overhead that will be introduced by the mechanism to adapt to the dynamic network conditions.

The simulation results for different network scenarios and conditions are shown in the following. An adaptation of the Scalev transmission system simulator [133, 134] has been used to obtain the results reported in this section.

4.5.1 Test Sequence

Different test sequences were generated for the simulations. The idea is to vary the offered load over time. The profile of one of these sequences is shown in Fig. 4.18, where variations of the channel utilization between 0.4 and 0.8 can be observed. Four different traffic flows have been mixed to obtain these utilization levels and to simulate network conditions as realistic as possible. Each traffic flow has different distributions for the interarrival times and the packet lengths (Table 4.4). We vary the specific parameters of each distribution to obtain the required load level.

Table 4.5: Resulting P_L as function of w and the required P_L.

	Required P_L		
w	10^{-3}	10^{-4}	10^{-5}
10^{-3}	$5.66 \cdot 10^{-4}$	$2.62 \cdot 10^{-5}$	$2.38 \cdot 10^{-7}$
10^{-4}	$8.07 \cdot 10^{-4}$	$7.96 \cdot 10^{-5}$	$8.58 \cdot 10^{-6}$
10^{-5}	$9.27 \cdot 10^{-4}$	$1.18 \cdot 10^{-4}$	$1.57 \cdot 10^{-5}$
10^{-6}	$1.35 \cdot 10^{-3}$	$2.34 \cdot 10^{-4}$	$4.36 \cdot 10^{-5}$

4.5.2 Input Parameters Considerations

As stated in Section 4.3.3 the buffer sizing mechanism is based on the values of the channel occupancy (ρ_a) and the number of packets in the system (N_a). Both parameters must be measured by the device itself. Instead of using instantaneous measurements, in a practical implementation, these values must be filtered (smoothed) to avoid undesired oscillations. For this, we use an exponentially weighted moving average (EWMA):

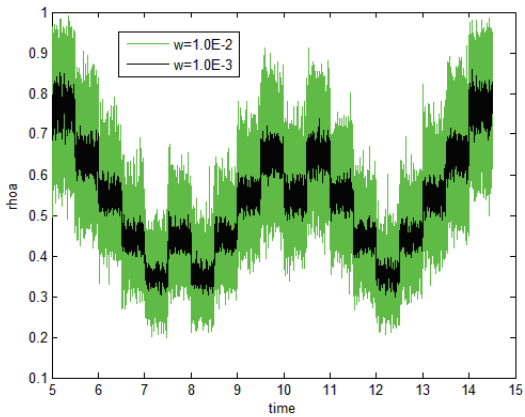
$$\rho_a = w \cdot s_\rho + (1 - w)\rho'_a \tag{4.57}$$

$$N_a = w \cdot s_N + (1 - w)N'_a \tag{4.58}$$

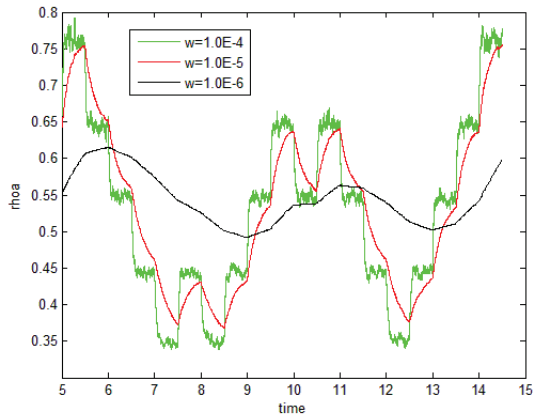
where ρ'_a and N'_a are the previous values, s_ρ and s_N are the current samples, and ρ_a and N_a are the new averages that will be used as inputs of the Eqs.(4.41) and (4.43). On the other hand, the value of $w(w \in [0, 1])$ defines the relative importance of the current sample with respect to the past history. The greater its value is, the lesser smoothing and faster following of the evolution of the parameter. For this work the objective is to achieve the maximum smoothing but without losing the evolution of the input parameters, which could lead to not accomplish the required packet loss probability.

For the test sequence under study, several simulations were performed in order to find the minimum w value which can meet the requirement of the target loss probability. Table 4.5 summarizes the resulting values for different required P_L and for different values of w. As can be observed, for $w = 10^{-4}$ the required P_L is still met. Nevertheless, for $w = 10^{-5}$ the resulting losses exceed the target in most of the cases. This effect is also evident in Fig. 4.19 which shows the time evolution of the filtered ρ_a for $P_L = 10^{-3}$ and for different values of w. For a sake of clarity, it only shows the last 10 s of such evolution. As can be observed in Fig. 4.19(a), for values of w of 10^{-2} and 10^{-3} the variations of ρ_a are very high. On the other side, for values of 10^{-5} and 10^{-6} (Fig. 4.19(b)) the system does not properly follow the evolution of the load. The value $w = 10^{-4}$ is confirmed as the most suitable since it presents the maximum smoothing that still follows such variation.

Fig. 4.20 shows the evolution of the buffer size Q (in packets) obtained from Eq. 4.46 for different values of w and $P_L = 10^{-3}$. As expected, for greater values of w the buffer oscillations increase. However, for the lowest value, the required P_L is not accomplished (as shown in Table 4.5). Finally, once the value of $w = 10^{-4}$ has been selected, Fig. 4.21 shows the buffer size evolution for different requirements of loss probability. As can be verified, the required buffer is greater when the target P_L is more demanding.



(a) $w = 10^{-2}$ and 10^{-3}.



(b) $w = 10^{-4}$, $w = 10^{-5}$ and 10^{-6}.

Figure 4.19: Smoothing of the parameter ρ_a for a required $P_L = 10^{-3}$.

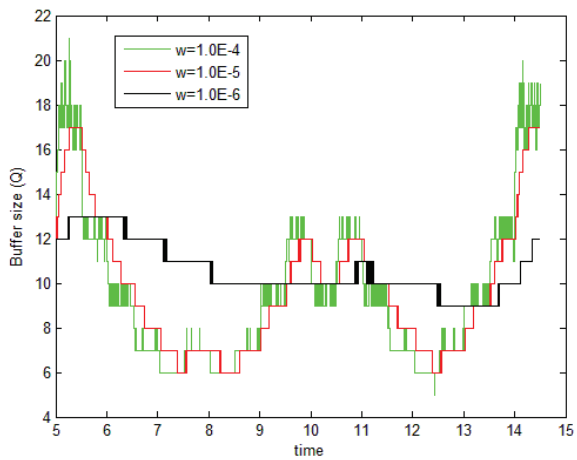


Figure 4.20: Evolution of the buffer size for different values of w and $P_L = 10^{-3}$.

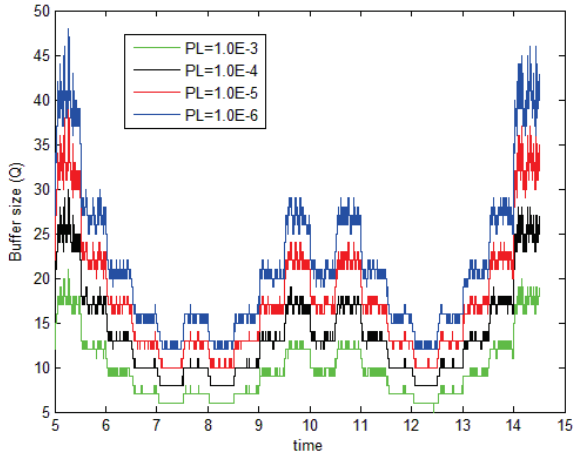


Figure 4.21: Evolution of the buffer size for different values of target P_L and $w = 10^{-4}$.

4.5.3 Minimization of the Mechanism Overload

As it can be seen in the previous simulations, the network device under study is able to adapt the amount of memory that it allocates to each transmission channel according to the load and the loss probability requirement. For this, at each new packet arrival it does all the required computations and adapts the buffer size accordingly. Observe that in order to apply the expression of Eq. 4.46 it is necessary the value of α, which in turn is obtained from the value of β. For this last, the application of numerical methods is required. Undoubtedly, this way of proceeding introduces a high overload in the device that would result in an undesired additional delay in the packet retransmission.

With the aim of reducing at maximum this overload, this section presents the results obtained after the establishment of acting thresholds. Two different thresholds have been considered, one for decreasing Th_{DW} and other for increasing Th_{UP} the buffer size. That is, for each received packet the device updates the average value of ρ_a and N_a (this does not introduce a significant overload), but the updated value of ρ_a must differ more than the established threshold with respect to its value at the time instant that the last adaptation was done. This way, the numerical computations required according to Eq.4.44 must be done only if the load variation is really significant. This will drastically reduce the algorithm execution rate, allowing its implementation in real devices.

Many simulations have been done to adjust the value of the thresholds. In addition, both absolute and relative threshold values have been tested. For instance, an increment of 0.1 in the utilization factor with respect to the previous reference (absolute value) before increasing the buffer size. Or to do so if the difference is around 5% (relative value). From all the experiments, we present the results that lead to the final selection of both thresholds. In both cases were absolute values. For the decreasing threshold a value of 0.075 was chosen in order to not to easily reduce the buffer size and to keep bounded the loss probability. Once this value has been adjusted, new simulations were done to adjust the increasing threshold. Table 4.6 summarizes the resulting values. As can be observed in this table, increasing the threshold value make more difficult that the system adapts to a load increase. And therefore the loss probability

Table 4.6: Resulting P_L as function of the increasing threshold Th_{UP} and the required P_L.

	Required P_L			
Th_{UP}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
0.01	$6.47 \cdot 10^{-4}$	$7.44 \cdot 10^{-5}$	$8.35 \cdot 10^{-6}$	$4.47 \cdot 10^{-7}$
0.05	$8.47 \cdot 10^{-4}$	$9.87 \cdot 10^{-5}$	$1.07 \cdot 10^{-5}$	$1.19 \cdot 10^{-6}$
0.1	$1.11 \cdot 10^{-3}$	$1.48 \cdot 10^{-4}$	$2.22 \cdot 10^{-5}$	$3.34 \cdot 10^{-6}$

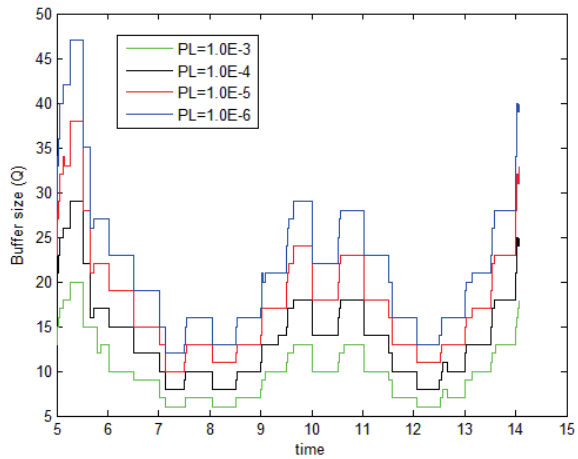


Figure 4.22: Evolution of the buffer size after the application of the increasing and decreasing thresholds, for different values of target P_L.

also increases. For all the cases, the maximum value for this threshold that allows to keep the loss probability bounded below the required one is 0.01. Using this value we obtain the buffer size evolution shown in Fig. 4.22 for different loss probability requirements. As can be observed, comparing with Fig. 4.21, all the unnecessary executions of the algorithm have been eliminated. This way, the undesired overload of the network device is avoided.

Specifically, for the reported simulations, the number of algorithm executions decreases from 6499628 for the case without thresholds (i.e. one execution per packet arrival) to 213 after thresholds application. Besides, this value was the same for the different cases of loss probability under study. Using a temporal averaging, this results are equivalent to pass from a rate of 448225 to 15 executions per second. Note that this significant drop is mainly due to the initial situation in which the number of executions is excessively high.

On the other hand, we analyze the number of algorithm executions that really causes a change of the buffer size. Namely, sometimes the algorithm execution could lead to obtain a value for the buffer size that is the same that its previous one. This fact gives us information about whether the thresholds were properly chosen. If the executions rate is much higher than the rate of real size changes, the value of the thresholds could be made larger to reduce the number of executions, and vice versa. Nevertheless, we must take into account that the values can not be so similar, because in that case required updates could be lost and the target loss probability not achieved. Fig. 4.23 shows the mean values of the rate of executions and the rate of buffer size changes for different values of required P_L. As can be observed, the rate of

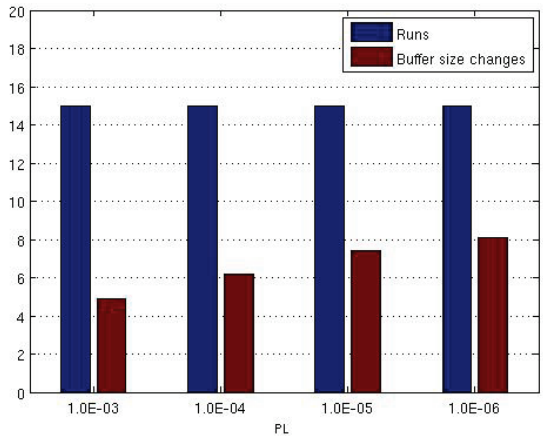


Figure 4.23: Rate of algorithm executions and rate of effective buffer size changes, for different values of required P_L.

effective buffer size changes grows when the requirement of loss probability is more demanding.

4.5.4 Traffic Differentiation

In the following we present the results obtained when the system enables the possibility of configure different service levels to different traffic flows. In particular, one of the flows is the correspondent to the test sequence used in the previous sections, and three new flows are added with different load variations between them. Each of these flows are scaled to produce a quarter of the total load offered to the system. As a whole, the temporal evolution of the channel utilization is as shown in Fig. 4.24. Note that this global utilization is equally perceived by the four flows since they share the channel.

In a first simulation round, all the flows request the same loss probability. Table 4.7 summarizes the resulting P_L for each flow and for different values of target P_L. As can be observed, the obtained loss probability for all the cases is below the configured target, which shows that the mechanism continues providing correct results when different flows are separately treated. Fig. 4.25 shows the average number of units in the system observed by each flow during the simulation, for the case in which the requested target is $P_L = 10^{-6}$. These values are different for each traffic, because they are independently measured on each queue. Then, they show the different load evolution caused by each flow. By its part, Fig. 4.26 presents the evolution of the buffer size allocated for each flow. As expected, this evolution is also different for each traffic, because it adapts to the conditions of each one.

The following round of simulations aims to verify that the mechanism works properly when different service levels (different loss probabilities) are required by each traffic flow. Specifically, loss probabilities from 10^{-3} to 10^{-6} have been requested. Table 4.8 shows the resulting values, verifying again the proper system operation. In this case, the value of the transfer time T (waiting plus transmission) for each flow has been included. This time is very similar for all the traffics independently of the requested loss probability. This is because the loss probability

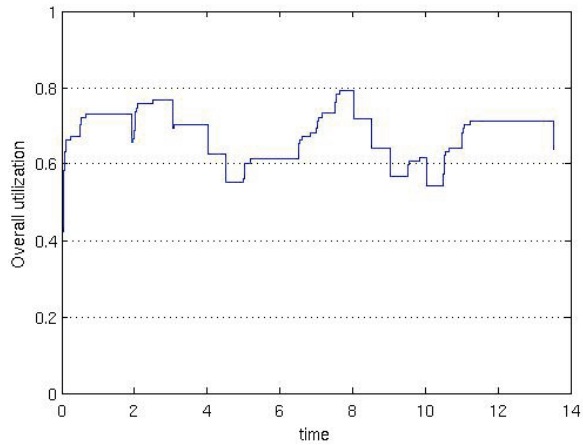


Figure 4.24: Global channel utilization produced by the aggregation of four different traffic flows.

Table 4.7: Resulting P_L for differentiated traffic flows as a function of the required P_L (equal for all the flows).

	Required P_L			
Flow	10^{-3}	10^{-4}	10^{-5}	10^{-6}
1	$3.86 \cdot 10^{-4}$	$2.78 \cdot 10^{-5}$	$2.31 \cdot 10^{-6}$	$1.54 \cdot 10^{-7}$
2	$5.21 \cdot 10^{-4}$	$4.51 \cdot 10^{-5}$	$5.54 \cdot 10^{-6}$	$4.62 \cdot 10^{-6}$
3	$3.71 \cdot 10^{-4}$	$2.69 \cdot 10^{-5}$	$3.85 \cdot 10^{-6}$	$1.54 \cdot 10^{-6}$
4	$5.00 \cdot 10^{-4}$	$2.49 \cdot 10^{-5}$	$2.46 \cdot 10^{-6}$	$6.15 \cdot 10^{-6}$

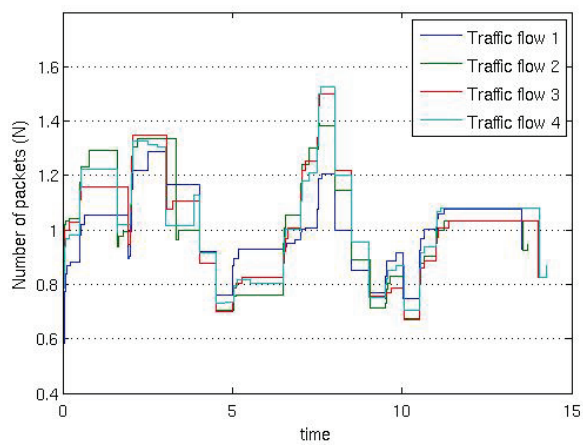


Figure 4.25: Average number of packets in the system.

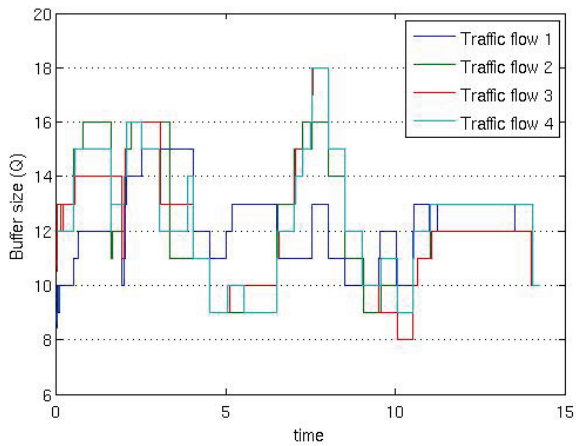


Figure 4.26: Buffer size for each traffic flow.

Table 4.8: Resulting P_L and T for differentiated traffic flows as a function of the required P_L (different for each flow).

	Required P_L			
	Flow 1 10^{-3}	Flow 2 10^{-4}	Flow 3 10^{-5}	Flow 4 10^{-6}
P_L	$3.86 \cdot 10^{-4}$	$4.43 \cdot 10^{-5}$	$3.84 \cdot 10^{-6}$	$6.15 \cdot 10^{-7}$
T	$1.17 \cdot 10^{-6}$	$1.21 \cdot 10^{-6}$	$1.22 \cdot 10^{-6}$	$1.22 \cdot 10^{-6}$

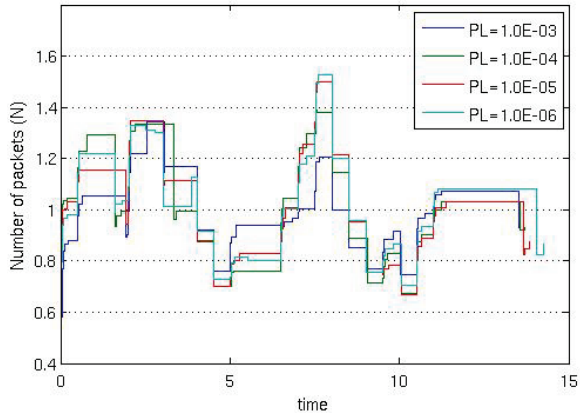


Figure 4.27: Average number of packets in the system for each traffic flow and different loss probabilities.

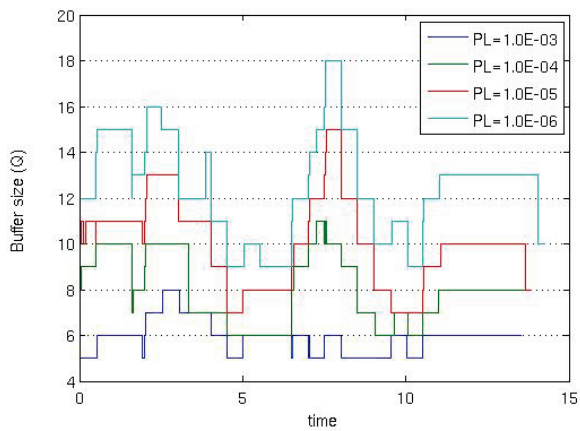


Figure 4.28: Buffer size evolution for each traffic flow and different loss probabilities.

values are small, and therefore almost all the packets are accepted in the buffers for all the cases without being excessively significant the effect of the finite buffers. Observe that for the biggest loss probability value, 10^{-3}, a smaller value of T is appreciated as it corresponds to a smaller buffer.

On the other hand, Fig. 4.27 and Fig. 4.28 show the evolution of the average number of packets in the system and the assigned buffer for each traffic respectively. As in the previous simulation round, it must be taken into account that the variation of the number of units in the system is different for each flow and therefore the increments and decrements of the allocated buffer sizes are not synchronized. Besides, we verify that the traffics which request smaller loss probabilities are the ones with larger allocated buffers.

4.6 Method Extension for Wireless Devices

To extend our proposal to wireless networks, where the stations contend inside a shared medium, we need to do some additional considerations. Mainly, we must take into account the fact that in contrast to a wired network where the node server occupancy is exclusively due to its own packet transmissions, in wireless networks a node should hold the packets in its queue if the radio channel is being used by another station inside its carrier sensing range. Actually, if we consider the CSMA/CA mechanism of IEEE 802.11-based networks, we must combine both the physical and virtual carrier sensing status information in order to determine the node server occupancy.

The physical carrier sensing mechanism reports the current state of the medium to the local MAC entity. It reports the channel as busy whenever the perceived signal strength exceeds the carrier sensing threshold or when the physical layer is in transmission state. On the other hand, the virtual carrier sensing is provided by the MAC by means of the network allocation vector (NAV). 802.11-based nodes (stations or access points) use the NAV to know how long they must defer from accessing the medium because another node is using it. The duration information required to set the NAV is carried in RTS/CTS frames and in all the data and control frames interchanged during a contention period. This virtual carrier sensing reports the medium as idle when the NAV is zero and as busy otherwise. In summary, the medium will be considered idle only when the physical and the virtual carrier sensing mechanism simultaneously report it as idle.

Additionally, we must take into account the inter frame space time (IFS) that each node has to delay between the transmission of two consecutive frames (as stated by the IEEE 802.11 standard [53].)

With all this in mind, to measure the server occupancy we consider that a server is idle only in the case that the physical and virtual carrier sensing mechanisms report an idle state, and none IFS waiting time is being carried out by the node. In all other cases the node server is seen as busy.

It is important to note that, besides buffer overflow, the proposed mechanism indirectly considers the other sources of packet losses experimented by wireless devices. These losses can be produced both by collisions in the shared medium or by physical channel impairments (noise, interference, fading, shadowing, etc.). As stated in [22], frames that are not correctly received at the MAC layer (of the next-hop destination nodes) are retransmitted by the source. These retransmissions imply higher channel utilization. Therefore, the value of the server occupancy measured by our mechanism will be increased accordingly, and so the buffer size will also be modified depending on the channel conditions.

4.7 Simulation of Wireless Scenarios

In order to validate and assess the proposed buffer sizing mechanism in wireless devices working over shared channels, it has been verified by means of extensive simulations. To be more confident with the obtained results it has been fully implemented on ns-2 and ns-3 simulators.

4.7.1 Simple Two-nodes Topology

First, a simple two-node topology has been chosen to determine the adequate configuration parameters and to illustrate its effects over the system's performance. In this case, the

Table 4.9: ns-2 MAC/PHY parameters used in simulations.

Parameter	Value
$CW_{\min}$	15
$CW_{\max}$	23
SIFS	16 µs
DIFS	34 µs
Slot duration	9 µs
Header duration	20 µs
Short retry limit	7
Long retry limit	4
CS threshold	6.31 pW
P_t	0.2 mW
Frequency	5.18 GHz
Noise floor	0.25 pW
Power monitor threshold	2.1 pW
SINR preamble capture	2.5118
SINR data capture	100
Propagation loss model	Two ray ground
Data/control OFDM rate	6 Mbps

simulations have been carried out on ns-2 version 2.34 using the overhauled IEEE 802.11 MAC and PHY modules [135]. The organized and modular design of these MAC and PHY modules allows clearer and easier channel occupancy measurements and also improves the simulations accuracy in comparison with the native IEEE 802.11 ns-2 model. The MAC/PHY parameters used in these simulations are shown in Table 4.9.

4.7.1.1 Method validation under variable traffic load conditions

Following the traffic criteria applied in Section 4.4, it has been considered that Node 1 transmits to Node 2 a UDP traffic flow resulting from the aggregation of four different flows with different interarrival times and packet lengths. The idea behind this is to obtain a complicated mix of traffics that do not follow a simple pattern or distribution. To explore the behavior of the proposed buffer sizing algorithm under diverse traffic loads, seven different loads have been generated (from $\rho_a = 0.1$ to $\rho_a = 0.7$) and combined for the tests. This way, increases and reductions of the system loads over time have been simulated. For these experiments, the time between significant variations of the load has been considered as 50 s. All the simulation rounds presented in this section were 7500 s long with the purpose of dispose of enough transmitted packets for a reliable estimation of the packet loss probability. Table 4.10 shows the packet length distributions and Table 4.11 the interarrival times distributions used for generating these seven loads. The load variation is shown in Fig. 4.29. For the sake of clarity, only a part of the simulation (between 3000 and 3500 seconds) is presented. Other traffic mixes and other load variations have been used to drive simulations with very similar results. Fig. 4.30 shows the resulting buffer size Q evolution for three different target values of the packet loss probability. Two cases are included, the one before the application of the acting thresholds (Fig. 4.30(a)) and also the one with the thresholds $Th_{UP} = 0.05$ and $Th_{DW} = 0.075$ applied (Fig. 4.30(b)). As it was expected, for both cases, the required buffer is greater when the target P_L is more

Table 4.10: Packet length description for UDP traffic load composition.

Traffic No.	Packet length
1	Truncated Pareto $k = 1.5$ $x_{\min} = 368$ bits $x_{\max} = 12000$ bits
2	Gamma $\alpha = 1500$ bits $\nu = 3$
3	Uniform $x_{\min} = 1000$ bits $x_{\max} = 9000$ bits
4	Truncated Pareto $k = 1.5$ $x_{\min} = 368$ bits $x_{\max} = 18000$ bits

Table 4.11: Interarrival time description for UDP traffic load composition.

	Interarrival time [ms] for ($\rho = 0.1$)	($\rho = 0.2$)	($\rho = 0.3$)	($\rho = 0.4$)	($\rho = 0.5$)	($\rho = 0.6$)	($\rho = 0.7$)
1	Truncated exponential $\lambda^{-1} = 6.93$ $x_{\min} = 1.17$ $x_{\max} = 39$	$\lambda^{-1} = 3.61$ $x_{\min} = 1.17$ $x_{\max} = 13$	$\lambda^{-1} = 2.3$ $x_{\min} = 0.6$ $x_{\max} = 14$	$\lambda^{-1} = 1.54$ $x_{\min} = 0.6$ $x_{\max} = 14$	$\lambda^{-1} = 1.34$ $x_{\min} = 0.44$ $x_{\max} = 14.5$	$\lambda^{-1} = 1.1$ $x_{\min} = 0.46$ $x_{\max} = 15.3$	$\lambda^{-1} = 0.92$ $x_{\min} = 0.48$ $x_{\max} = 16$
2	Uniform $x_{\min} = 0$ $x_{\max} = 26$	$x_{\min} = 0$ $x_{\max} = 13$	$x_{\min} = 0$ $x_{\max} = 9.33$	$x_{\min} = 0$ $x_{\max} = 7$	$x_{\min} = 0$ $x_{\max} = 5.8$	$x_{\min} = 0$ $x_{\max} = 5.1$	$x_{\min} = 0$ $x_{\max} = 4.6$
3	Deterministic $x_0 = 43.33$	$x_0 = 21.67$	$x_0 = 15.56$	$x_0 = 11.67$	$x_0 = 9.67$	$x_0 = 8.5$	$x_0 = 7.62$
4	Gamma $\alpha = 8.22$ $\nu = 4$	$\alpha = 4.1$ $\nu = 4$	$\alpha = 2.95$ $\nu = 4$	$\alpha = 2.21$ $\nu = 4$	$\alpha = 1.83$ $\nu = 4$	$\alpha = 1.61$ $\nu = 4$	$\alpha = 1.45$ $\nu = 4$

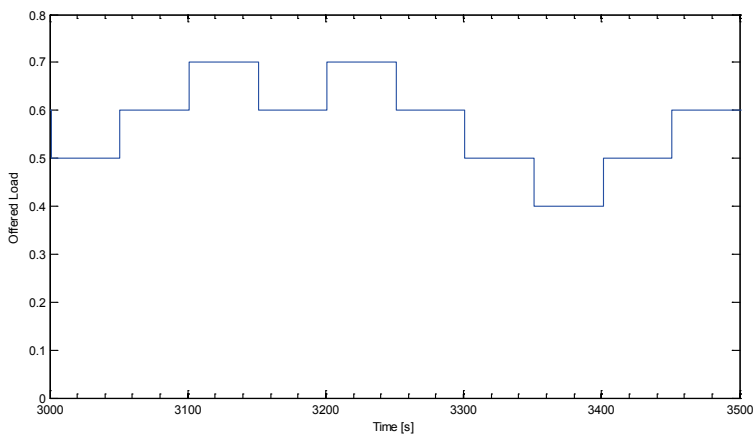
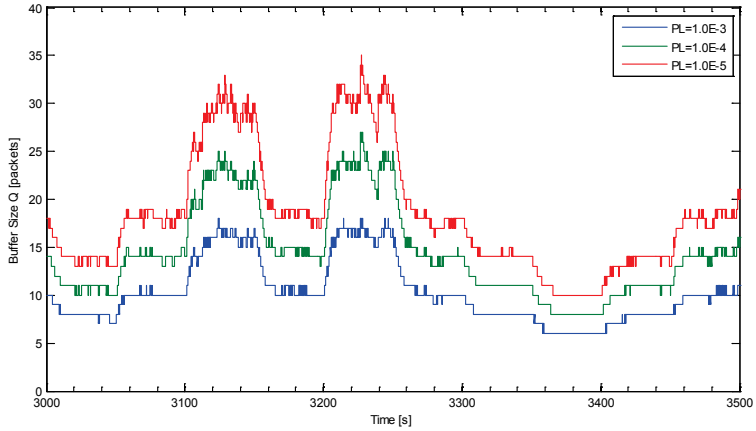
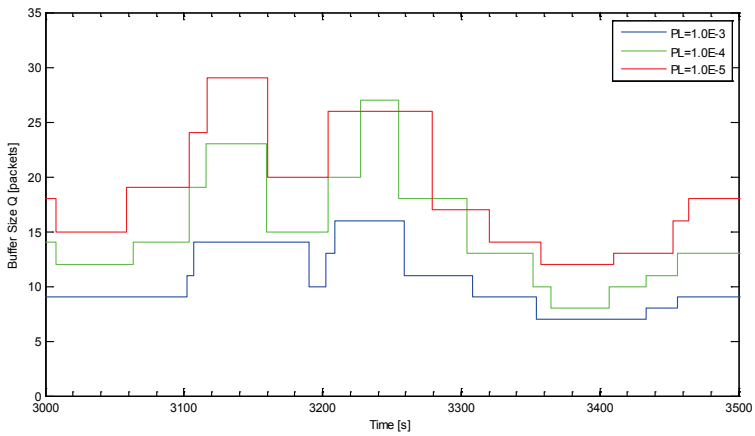


Figure 4.29: Offered traffic load variation for the wireless two-nodes topology.



(a) Without acting thresholds



(b) After acting thresholds application ($Th_{UP} = 0.05$ and $Th_{DW} = 0.075$)

Figure 4.30: Buffer size evolution for the wireless two-nodes topology with different packet loss probability target values.

demanding and it continuously adapts to the traffic load variation. The effectiveness of the thresholds is also evident. For instance, for this particular load variation, the algorithm was executed on average once every 39.4 s instead of every packet arrival and on average 91% of the executions produced real buffer size changes. This way, the buffer size presents a much more stable behavior and the device processor overload is substantially reduced. Additionally, the measured packet loss probability values for all the requested targets are shown in Table 4.12. It can be observed how the measured P_L is lesser than the target P_L for both cases. Although the application of the thresholds increases the resulting loss probability, this is still below the required target. Thus, the correct operation of the proposed mechanism in wireless devices has also been confirmed.

Table 4.12: Measured packet loss probability values for the wireless two-nodes topology.

Target P_L	**Measured P_L without acting thresholds**	**Measured P_L with acting thresholds**
1.0E−3	8.79E−4	9.69E−4
1.0E−4	5.30E−5	8.60E−5
1.0E−5	3.00E−6	8.00E−6

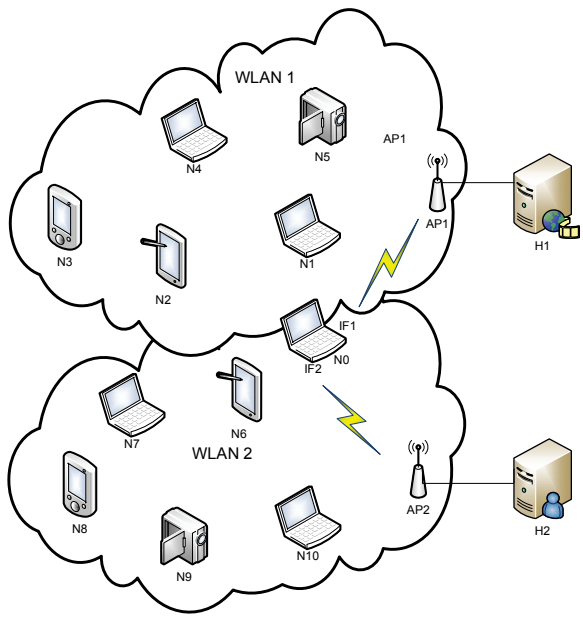


Figure 4.31: Multiple WLANs ns-3 simulation topology.

4.7.2 Multiple WLANs Topology

For this scenario, the ns-3 simulation topology shown in Fig. 4.31 has been utilized. It consists of two infrastructure-based WiFi networks with five stations associated with each AP. Additionally, a special node (N0) equipped with two network interface cards (IF1 and IF2) and associated to both WiFi networks has been considered and subjected to deeper analysis. The APs are connected to the wired hosts (H1 and H2) through 500 Mbps links. Static routing has been configured in the nodes for connectivity purposes. Wireless nodes in WLAN1 (WLAN2) transmit UDP packets to H1 (H2) through AP1 (AP2). Node N0 belongs to both networks. For all the simulations under this scenario it transmits UDP packets through its two interfaces during the entire simulation time. The other nodes transmit UDP packets according to the activation pattern shown in Fig. 4.32. The ON state means that a station is transmitting UDP traffic to its respective wired host at that time instant. During the transmission periods, we use truncated exponential distributions for the packet length and the interarrival time of such UDP packets. Fig. 4.32 also shows the total number of active nodes in each WLAN at any time. As it can be seen, the length of the simulations in this scenario is 5000 s. The MAC/PHY configuration parameters used in these simulations are shown in Table 4.13.

Table 4.13: ns-3 MAC/PHY parameters used in simulations.

Parameter	**Value**
AP beacon interval	10 s
STA probe request timeout	50 ms
STA association request timeout	500 ms
STA max missed beacons	10
STA active probing	No
CTS timeout	75 µs
ACK timeout	75 µs
Basic block ACK timeout	281 µs
Compressed block ACK timeout	99 µs
SIFS	16 µs
DIFS	34 µs
Slot duration	9 µs
Max. propagation delay	3.333 µs
Propagation loss model	Log distance Exponent: 3 Reference distance: 1 m Reference loss: 46.677 dB
Data/control OFDM rate	6 Mbps
Max No. of retransmission for RTS/data pkts	7
RTS/CTS/fragmentation threshold	2346
Random number generator	MRG32k3a

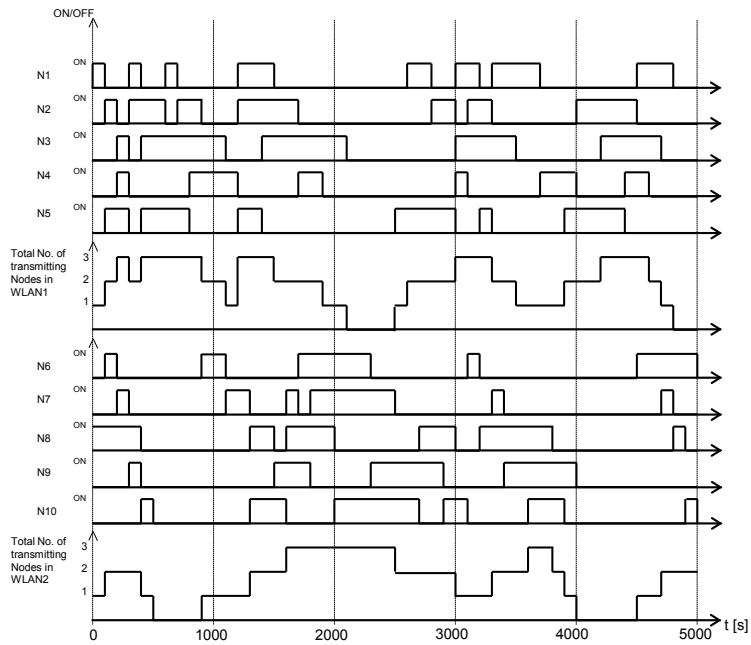


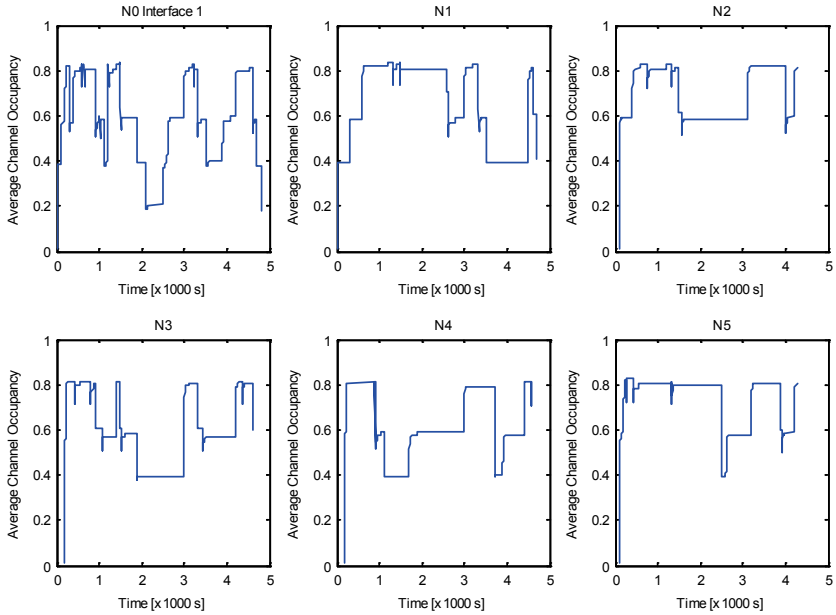
Figure 4.32: Activation pattern for multiple WLANs topology.

4.7.2.1 Method validation under variable number of active nodes

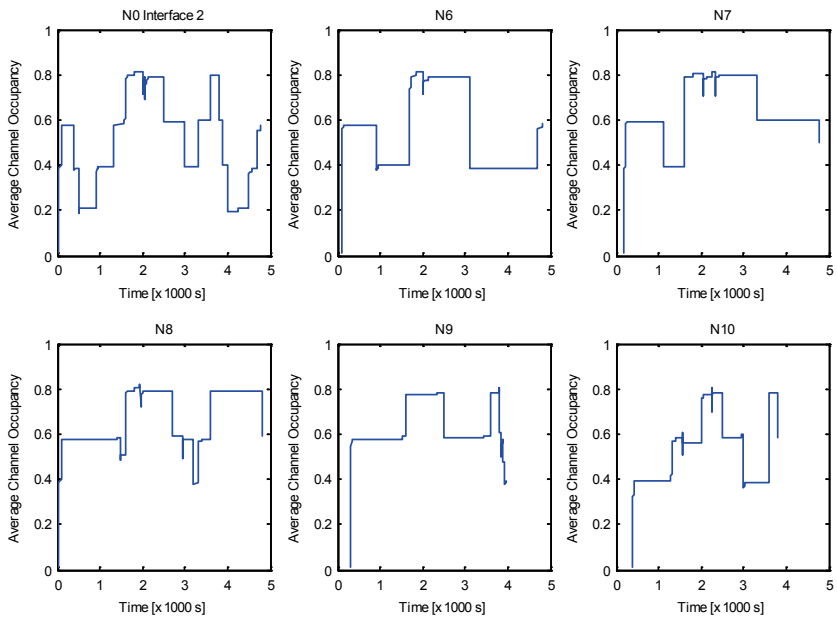
Section 4.7.1.1 demonstrated the fact that the proposed mechanism works properly for a simple wireless node adapting its buffer size according to its own traffic load variation. This section starts demonstrating that the mechanism works also for larger topologies. More specifically, it is shown that the mechanism works for the case that the channel occupancy and the traffic variations are due to the activation or deactivation of different network nodes. Fig. 4.33 shows the average channel occupancy measured in both WLANs. Here it can be observed that each node in the network is able to correctly capture the average channel occupancy during its corresponding activity periods. Due to the fact that the N0 node is transmitting by its two interfaces during all the simulation time, the first subgraph in Figs. 4.33(a, and b), which corresponds to each interface of the N0 node, shows the complete evolution of the average channel occupancy for each of the two WLANs (Observe the correspondence with the total number of active nodes for each WLAN in Fig. 4.32). With this information, every node is able to autonomously and dynamically adapt its buffer size according to the traffic and channel states. As always, this is done with the constraint of a given packet loss probability target. Fig. 4.34 shows the buffer size evolution of each N0 interface for three different packet loss probability targets. This confirms the proper functioning of the algorithm for bigger topologies where different nodes are contending for the channel.

4.7.2.2 Impact of channel errors

As previously said, lost packets due to collisions or channel errors are retransmitted by the source. These retransmissions cause the increasing of the channel occupancy and therefore the



(a) WLAN1 nodes



(b) WLAN2 nodes

Figure 4.33: Average channel occupancy for the multiple WLANs topology.

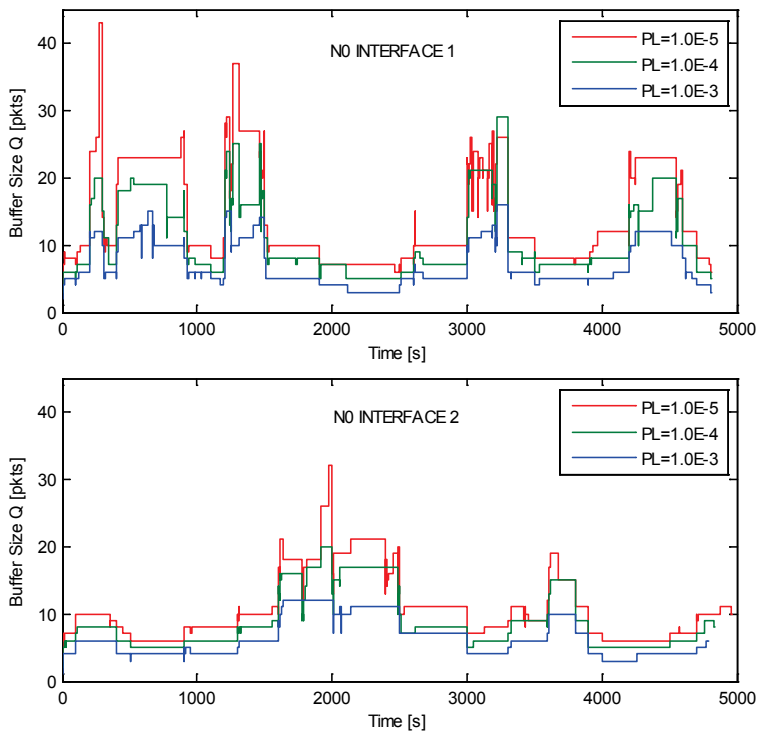


Figure 4.34: Buffer size for the N0 node configured with three different P_L target values.

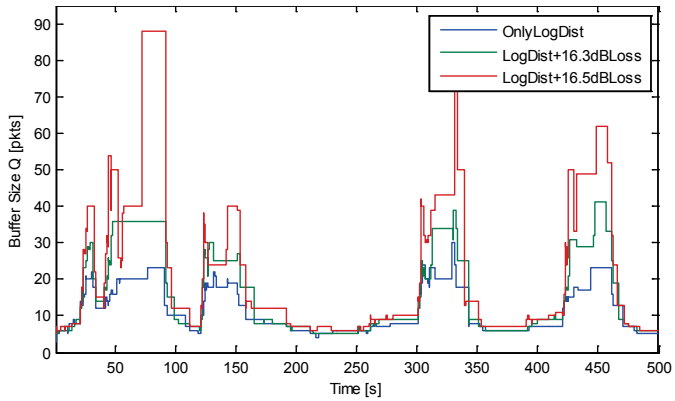


Figure 4.35: Buffer size evolution for a target $P_L = 1.0\text{E}{-4}$ and three different WLAN1 channel configurations: (i) Only Log Distance Propagation Loss Model; (ii) Log Distance plus 16.3 dB propagation loss; and (iii) Log Distance plus 16.5 propagation loss.

buffer size must be increased accordingly to still accomplish the target P_L. This behavior has been verified with another round of simulations described in the following.

For the same topology and traffic pattern describe in Section 4.7.2, we evaluate the performance of the system when the channel of the WLAN1 has an additional propagation loss. Fig. 4.35 compares the buffer size evolution for the interface IF1 of the node N0 for two general cases:

1. WLAN1 with the default ns-3 channel configuration that considers only Log Distance propagation loss model; and
2. WLAN1 with a channel with Log distance plus an additional propagation loss.

As expected, the increase in channel errors causes more retransmissions which impact on the channel occupancy of WLAN1. The average channel occupancy metric correctly captures this increase. Then, for the same traffic load, the increase in channel errors results in a larger buffer required to keep the loss rate close to the configured target P_L (Fig. 4.35). It is worth to mention that our mechanism does not require any additional configuration to adapt to different channel error conditions.

4.7.2.3 Memory utilization efficiency

To illustrate and quantify the improvements achieved with the dynamic buffer sizing mechanism, the concept of memory utilization efficiency η has been defined as the ratio between the area under the buffer occupancy curve (current number of packets waiting in the buffer) and the area under the buffer size curve (maximum number of packets allowed in the buffer). Fig. 4.36 shows these two curves, for the interface 2 of the N0 node, and for a target P_L of 1E−3. For the sake of clarity, a 'zoomed' version is also provided. It can be seen that this memory utilization efficiency is expected to be very low due to the fact that the nodes, working under normal traffic load conditions, expend long time with empty buffers.

A round of simulations has been carried out to compare the value of η with the dynamic buffer sizing algorithm against the resulting η when the nodes are configured with a static

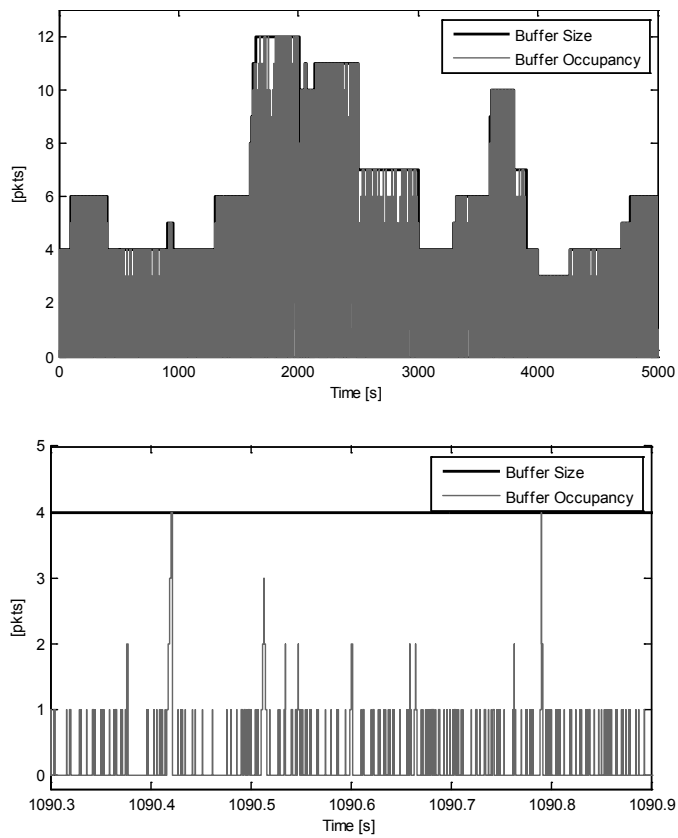


Figure 4.36: Buffer Size vs. Buffer Occupancy for the Interface 2 of the N0 node with target $P_L = 1\text{E}{-3}$.

Table 4.14: Memory Efficiency comparison for the two interfaces-node configured with static vs. dynamic buffer size.

	Q for static [pkts]	$\eta_{\mathbf{STATIC\_Q}}$	$\eta_{\mathbf{DYANMIC\_Q}}$	**% Improve**
Interface 1	10	0.03984	0.05728	43.77
	16	0.02669	0.03961	48.41
	22	0.01976	0.03114	57.59
Interface 2	10	0.02772	0.04424	59.56
	16	0.01780	0.03234	81.69
	22	0.01313	0.02649	101.85

buffer size. To do this comparison in the correct terms, the buffer size selected with the static allocation has been the one that produces the same loss probability than the dynamic allocation. That is, if we run the dynamic algorithm with a target P_L equal to 1E−3, we must compare it with a static buffer size that results in the same P_L. For the scenario under study, the correspondences are: 10 packets for $P_L = 1\text{E}{-3}$, 16 for $P_L = 1\text{E}{-4}$ and 22 for $P_L = 1\text{E}{-5}$. These static buffer size values have also been determined by simulation.

Table 4.14 summarizes the simulation results for the two interfaces-node. From here, it can be verified that the proposed mechanism achieves more efficient memory utilization for all the simulated conditions and for both node interfaces. The efficiency improvement achieved is around the 50% for the interface 1 and around 81% for the interface 2. It can be also noticed that in both, static and dynamic buffer sizing schemes, the memory utilization efficiency is lesser when the target P_L value is more stringent. This is because greater buffer sizes are required to reach this value and so the probability of underutilization increases.

Memory utilization efficiency provides a global vision of the performance achieved with the dynamic buffer sizing mechanism. On the other hand, it is interesting the study of a more instantaneous measure, such us the difference d between the allocated buffer size Q and the current buffer occupancy n_w (that is, the number of packets waiting in the buffer), both computed at packet transition times. This measure will be another systems discrete random variable defined by

$$d = Q - n_w \tag{4.59}$$

The behavior of this random variable has also been studied in the previous simulations. Its *pmf* is shown in Fig. 4.37 for the static buffer allocation, and in Fig. 4.38 for the dynamic one. As it can be observed, for the static buffer configuration the random variable d is almost all the time with a value closer to the maximum assigned buffer. This means that the buffer is almost all the time at the maximum underutilization level. Meanwhile, with the dynamic buffer sizing this variable is distributed among lesser values, and hence the underutilization level is considerably lesser.

The mean value (D) of the random variable d is also important since it gives a global idea of the buffer overprovisioning. Table 4.15 summarizes the obtained values. Once again, a better performance is obtained with the dynamic buffer sizing algorithm, since this measured memory waste is lesser than the values produced by a static buffer assignment under equivalent P_L values. Specifically, for the two interfaces-node a reduction of around 35% for the interface 1 and 47% for the interface 2 has been achieved. This proves again that to reach an equivalent packet loss probability the node configured with the dynamic buffer sizing algorithm requires less amount of memory.

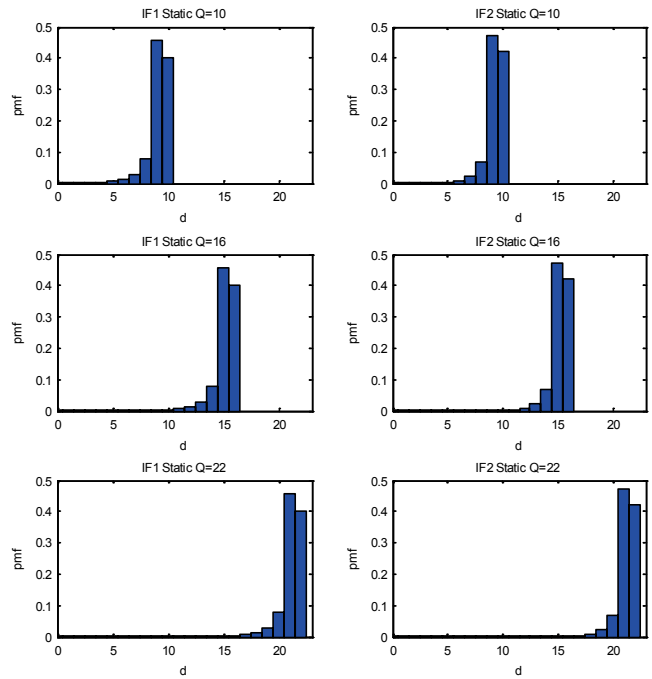


Figure 4.37: *pmf* for random variable d computed in the N0 node with static buffer allocation.

Table 4.15: Mean Difference between the buffer size and the buffer occupancy.

	Q for static [pkts]	***D* for static**	***D* for dynamic**	**Achieved Reduction [%]**
Interface 1	10	9.1122	6.1063	32.99
	16	15.0777	9.8235	34.85
	22	21.0668	12.9595	38.48
Interface 2	10	9.2252	5.5350	40.00
	16	15.2156	8.0546	47.06
	22	21.2115	10.0159	52.78

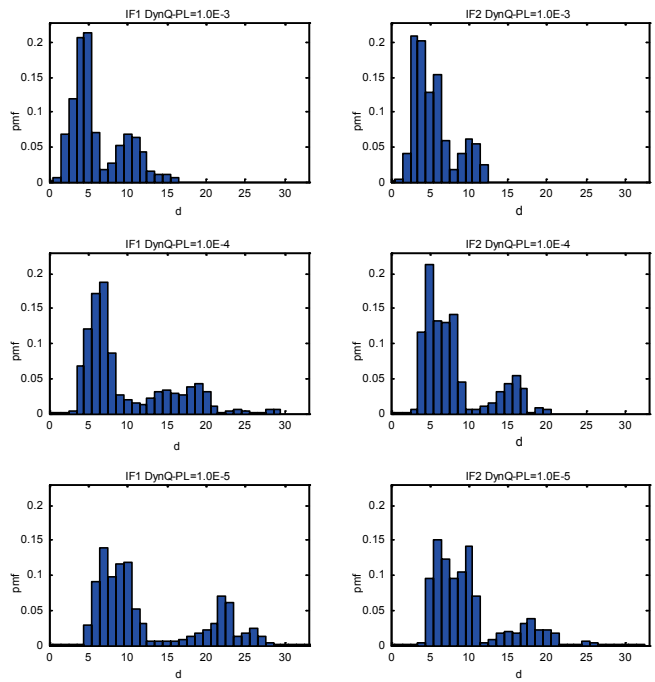


Figure 4.38: *pmf* for random variable d computed in the N0 node with dynamic buffer allocation.

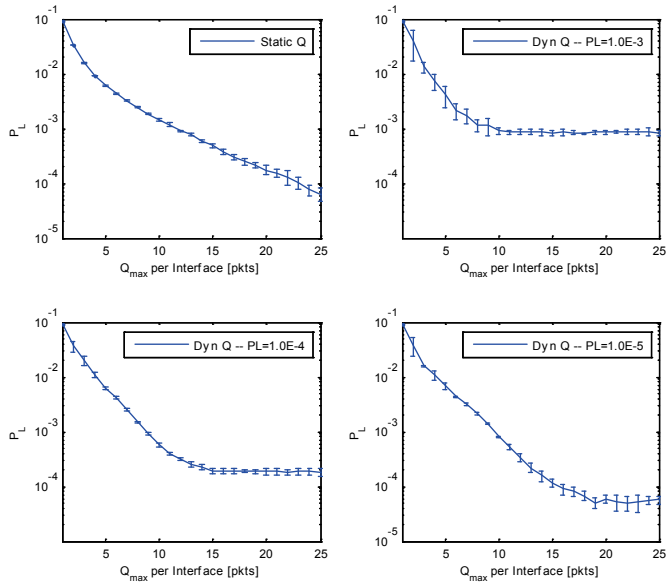


Figure 4.39: Measured P_L (for 95% confidence interval) for the N0 node with limited amount of available memory and configured with static vs. dynamic buffer sizing.

4.7.2.4 Memory share and dynamic self-configuration for nodes with multiple interfaces

An important observation is the fact that the proposed dynamic buffer sizing mechanism can be applied to improve the efficiency of memory constraint devices equipped with multiple interfaces. More efficient memory utilization can be achieved if the device is able to dynamically and autonomously release and dispose the assigned buffer that is not being used by an interface. So that, the unused memory can be harnessed by any other interface that really requires it at such time instant. This way the nodes, instead of assign a fixed amount of buffer for each interface, consider a global amount of memory and share out it among all the demanding interfaces based on current traffic load conditions.

Using the simulation topology shown in Fig. 4.31, a new round of simulations has been carried out to evaluate this fact. Firstly, all the nodes have been configured with static buffer sizes ranging from 1 to 25 packets for each node interface in each simulation round. Secondly, the nodes have been configured with the dynamic buffer sizing algorithm with an available amount of memory ranging from 1 to 25 packets for each node interface and for three different values of target P_L: 1.0E−3, 1.0E−4, and 1.0E−5. For instance, if N0 is configured with a static buffer of 10 packets, each N0 interface preserves this buffer size during the complete simulation. On the other hand, with dynamic buffer, a total amount of 20 packets is shared between the two interfaces. This way, if one interface requires less than 10 packets buffer at a given time instant, the other one can take the surplus until a maximum value of 20 packets.

Each simulation round, for each buffer size, was 5000 s long. Besides, at least ten repetitions (with different seeds) were done to present more reliable results. For the N0 node, Fig. 4.39 shows the evolution of the packet loss probability (with a 95% confidence interval)

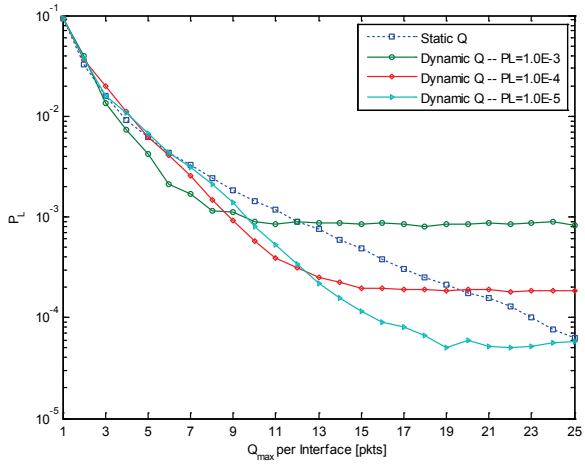


Figure 4.40: Average P_L values comparison for the N0 node with limited amount of available memory and configured with static vs. dynamic buffer sizing.

as the available memory increases. For convenience, Fig. 4.40 compares all the cases without confidence intervals. From these figures, it is possible to recognize three intervals in the behavior of the proposed buffer sizing mechanism.

- In first instance, when the memory is too much limited, (lesser than 3 packets per interface) both static and dynamic configurations exhibit the same results. In this case, under the present traffic load conditions, the available amount of memory is so small that both interfaces require much more, and then no improvement is materially feasible with the dynamic allocation.
- Secondly, the useful region appears as a result of the dynamic buffer size adaptation according to the traffic load variation inside each interface. A better memory management is possible since one interface is allowed to use the remaining memory of the other. This improvement is more significant when simultaneously both interfaces achieve a lesser P_L than the static buffer assignment due to the fact that each interface benefits the time instants in which the other frees its memory resources. This is achieved between 3 and 12 packets per interface for $P_L = 1.0\text{E}{-3}$; between 6 and 19 packets for $P_L = 1.0\text{E}{-4}$; and between 6 and 25 packets for $P_L = 1.0\text{E}{-5}$ (Fig. 4.40). In these intervals, the P_L is reduced on average around 36%, 47%, and 53% respectively (in comparison to the static allocation).
- Finally, since one of the input parameters for the dynamic buffer sizing algorithm is the target packet loss probability, when this target P_L value has been reached, it is possible that not all the available memory is used because it is not needed. Therefore, in the last interval the amount of buffer and its correspondent P_L keep almost constant. Obviously, if the memory excess is distributed among all the interfaces (as the static assignment does), the obtained P_L will be lower.

Another conclusion of these simulations is the fact that, the best performance could be achieved if the node was allowed to dynamically adapt also its target P_L value. It is not advisable to be

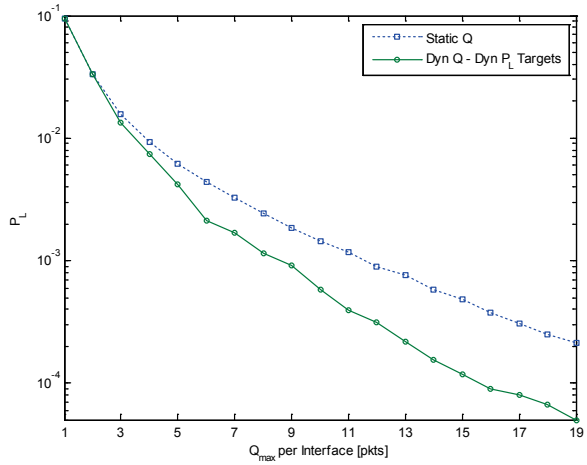


Figure 4.41: Measured P_L for the N0 node with constrained memory and configured with static vs. dynamic buffer sizing and dynamic P_L target values.

too demanding (with respect to the target P_L) when the available amount of memory is scarce. Based on Fig. 4.40, the best choice for the N0 node would be: keep target $P_L = 1.0\text{E}{-3}$ if the available memory is lesser or equal than 9 packets per interface; switch to $P_L = 1.0\text{E}{-4}$ if the available memory is greater than 9 and lesser than 13 packets per interface; and finally switch to $P_L = 1.0\text{E}{-5}$ if the available memory is greater or equal than 13 and lesser than 19 packets per interface. Fig. 4.41 shows the evolution of the P_L if the N0 node is configured in this way. A practical situation in which the available memory varies on time would appear on devices using multiple queues to offer differential treatment to different type of traffic flows. In this case, a global amount of memory would be dynamically distributed among the queues. These queues can vary their demanded buffer according to the requirement of each flow and its priority. Then, one or more queues could dispose of only few packets buffer at a particular time instant. So, under this scenario, adapting the target P_L further improves the system performance.

4.7.3 Mesh Network Topology

For this scenario we have in mind a VoD service provided through an infrastructure WMN. We assume that the data traffic is generated by a cluster of video servers like the one studied in Chapter 3 and then movies are distributed by a gateway mesh station. Fig. 4.42 illustrates the network topology subject to evaluation. All the mesh stations (M0 to M8) are based on the IEEE 802.11s amendment which is currently incorporated in the IEEE 802.11-2012 standard [53]. A detailed description of the ns-3 mesh networking implementation is available in [60]. All the mesh configurable parameters are set to the standard default values. The Log Distance propagation loss model has been considered for the radio channel.

Real video traces of MPEG-4 encoded movies are used for the evaluation. These traces, available in [136], include the frame index, the type of frame (I, P, or B), the time on which the frame was generated by the encoder and the frame size in bytes. For the transmission, the M0-Gateway station encapsulates all the video frames into UDP packets. The wireless interface of

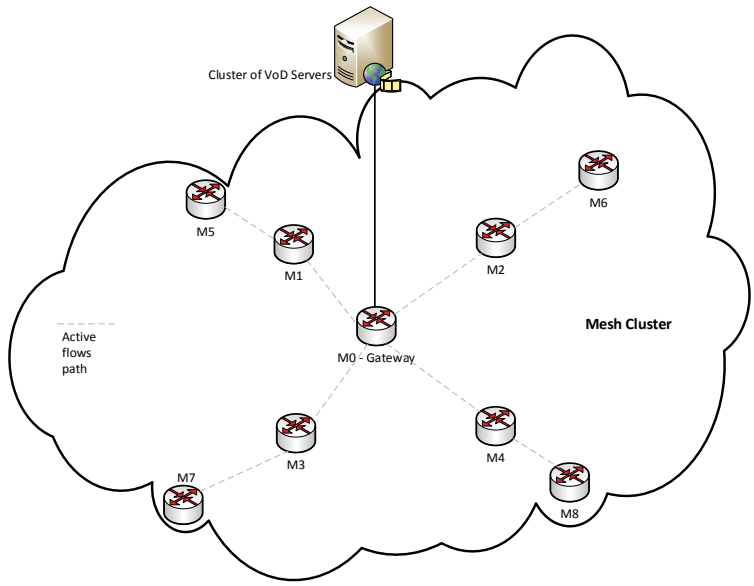


Figure 4.42: Mesh network topology.

this gateway station is the target of the dynamic buffer sizing adaptation. Fourteen concurrent VoD sessions are transmitted during the 36000 s long simulation. Table 4.16 summarizes the specific video traces used for these simulations.

As can be verified in Fig. 4.43 this amount of concurrent VoD sessions produces a high level of channel occupancy. For the sake of clarity only the interval between 11000 and 15000 s is presented in this and the following figures. Fig. 4.44 demonstrates that, even with this high level of traffic load, the gateway is able to properly adapt its buffer size when two different P_L targets are requested. Table 4.17 presents the resulting P_L. As it can be seen, in both cases the resulting value is below the target.

The resulting memory utilization efficiency comparing with the static buffer allocation that produces approximately the same loss probability is shown in Table 4.18. The efficiency improvement is more significant when the target loss probability is more demanding. The *pmf* of the difference between the allocated buffer and the current queue occupancy for both static and dynamic buffer allocations is shown in Fig. 4.45. Similar to the behavior observed in the multiple WLAN topology (Section 4.7.2.3), for the static allocation the difference is much closer to the assigned buffer size while in the dynamic case it is distributed among lower values. The comparison of the difference mean values for the both cases is summarized in Table. 4.19. This verifies the better performance of the dynamic allocation in terms of lower level of buffer overprovisioning and memory misuse.

This more efficient memory resource utilization, could facilitate scaling the mesh cluster by the addition of new mesh interfaces to the gateway in order to increase its capacity of concurrent VoD sessions and thus serve a greater number of clients.

Table 4.16: Video traces used for the evaluation of VoD services over an infrastructure WMN with dynamic buffer sizing adaptation.

Flow No.	Movie	Destination STA
1	Die Hard III	M1
2	Aladdin	M2
3	Jurassic Park I	M3
4	Star Trek - First Contact	M4
5	Robin Hood	M5
6	Simpsons	M6
7	Star Wars IV	M7
8	Lady and the Tramp	M8
9	Starship Troopers	M1
10	Alpin skiing sport event	M2
11	VIVA - Video Clips	M3
12	Mr. Bean	M4
13	South Park	M5
14	Silence Of The Lambs	M6

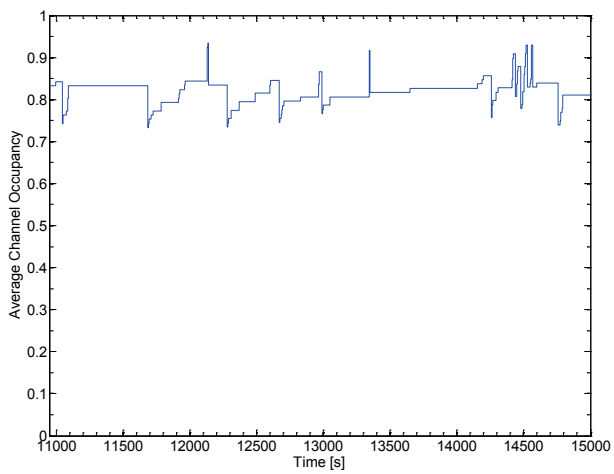


Figure 4.43: Average channel occupancy for the WMN topology.

Table 4.17: Measured packet loss probability values for the WMN topology.

Target P_L	Measured P_L
1.0E−3	8.797E−5
1.0E−4	3.748E−6

Table 4.18: Memory Efficiency comparison for the M0-Gateway configured with static vs. dynamic buffer size.

Target P_L	Q for static [pkts]	$\eta_{\text{STATIC\_Q}}$	$\eta_{\text{DYANMIC\_Q}}$	% Improve
$\leq$ 1.0E−3	27	0.0423	0.0605	43.03
$\leq$ 1.0E−4	55	0.0209	0.0425	103.37

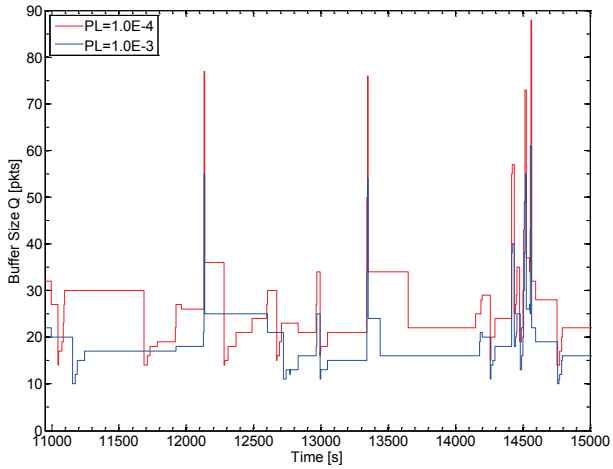


Figure 4.44: Buffer size evolution of the M0-Gateway station for different P_L targets.

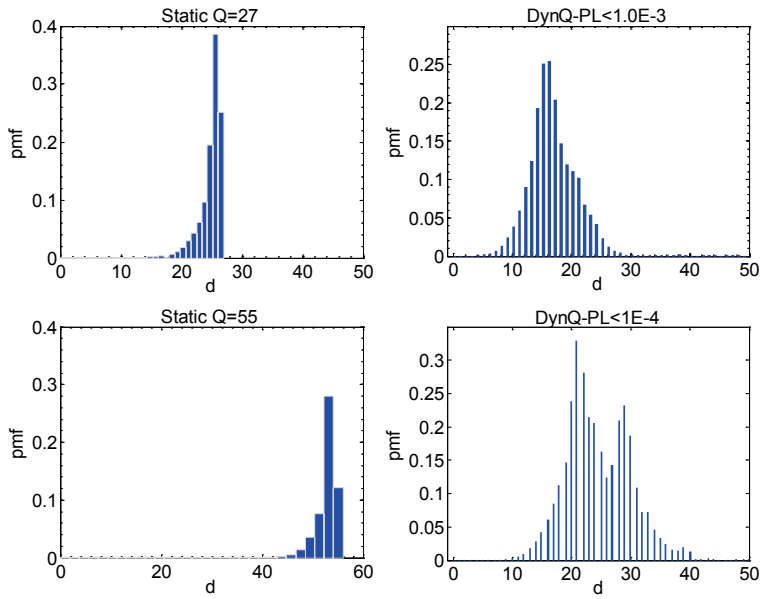


Figure 4.45: *pmf* of the difference between the allocated buffer and the queue occupancy for the M0-Gateway configured with static (left) vs. dynamic (right) buffer sizing.

Table 4.19: Mean Difference between the buffer size and the buffer occupancy for the mesh gateway.

Target P_L	**Q for static [pkts]**	***D* for static**	***D* for dynamic**	**Achieved Reduction [%]**
$\leq$ 1.0E−3	27	25.0687	17.1710	31.50
$\leq$ 1.0E−4	55	53.0575	25.2766	52.36

4.8 Conclusions

Buffer sizing of networks devices is a critical subject due to its immediate impact on the quality of service perceived by end users. Even though the memory cost is decreasing and its integration is easier, it is demonstrated that excessively large buffer could be detrimental because of the delay that it could aggregate. Besides, allocating just the required amount of memory to a transmission channel allows to utilize the remaining in other channel that requires a greater amount in a specific time instant.

This chapter proposes a buffer sizing mechanism that dynamically allocates a minimum buffer size for traffic flows which must be transmitted and consumed in real time. Such minimum buffer must guarantee not to exceed a maximum loss probability, which can be independently allocated to each traffic flow depending on its requirements. It is worth to mention that the evaluation of operating parameters (buffer occupancy, waiting times, loss probability, etc.) for the G/G/1 and G/G/1/K queues requires detailed knowledge of interarrival and service times. However, in real practice that knowledge is seldom obtainable and, therefore, the engineer has to make do with the limited amount of data about the system available to him.

To deal with this problem, this proposal uses a maximum entropy approach and verifies that it offers promising possibilities. In fact, we have shown, from a practical point of view, that the problem of buffer sizing can be solved with very good accuracy with only two measurements, namely: server occupancy and the average number of packets in the transmission system (both seen by an arriving packet). On this basis, we have extended and implemented the dynamic buffer sizing algorithm for wireless devices working over shared mediums.

The proper performance of the proposed mechanism has been verified by means of extensive simulations using different simulators, network scenarios, and traffic load conditions. Both type of network devices have been considered, those working with dedicated channels and those over shared channels. For all the cases, the obtained results demonstrate that such devices were provided with the capability of self-configure their buffer sizes, and efficiently self-manage their memory resources while keeping bounded their packet loss probability. We also show that a proper setting of the configuration parameters can considerably reduce the processor overload required by our mechanism. Results have confirmed that the system exhibits a proper convergence rate and stable behavior for all the analyzed scenarios. In addition, we evaluate the possibility of offering different loss probabilities to different traffic flows which share the same transmission channel. As in all the cases the results were also satisfactory.

On the other hand, it is important to remark that most of the current literature associates the buffer sizing problem with the dynamics of TCP congestion control mechanism. Then, the performance evaluation parameters are strongly related with this type of traffic (throughput, delay). Since this contribution presents an alternative buffer sizing method and it is focused on real time traffic, new performance evaluation metrics were also introduced in this chapter. Results demonstrate significant improvement in memory utilization efficiency achieved with the proposed mechanism, in comparison with a static buffer allocation. It is also shown that a lower amount of memory is required by nodes configured with dynamic buffers. The efficiency improvement achieved with the proposed mechanism benefits the self-organization capabilities of our target multiple-interface wireless mesh routers. Besides, it could also be valuable for resource constraint devices like wireless sensor nodes. The better memory utilization could also stimulate and facilitate the design and use of all-optical routers where small buffers are appreciated.

Chapter 5

Topology Control for Client WMNs

This chapter is devoted to make a contribution to client WMNs which main feature is that every node has the possibility of acting as a router, forwarding the received packets when they are not the final destination of the carried data. Due to the routing protocol procedures, when the network is quite dense the overload added by the routing management packets could be very high. To reduce the effects of this overload a topology control mechanism can be used, which can be implemented using different techniques. One of these techniques consists of enabling or disabling the routing functionality in every node. Many advantages result from selecting just a subset of nodes for routing tasks: reduction of collisions, protocol overhead, interference and energy consumption, better network organization and scalability. In this chapter, a new protocol for topology control in WMNs is proposed. The protocol is based on the centrality metrics developed by social network analysts. Our target is a client WMN created by user hand-held devices. For this kind of networks, we aim to construct a connected dominating set that includes the most central nodes. The resulting performance using the three most common centrality measures (degree, closeness and betweenness) is evaluated. As we are working with dynamic and decentralized networks, a distributed implementation is also proposed and evaluated. Some simulations have been carried out to analyze the benefits of the proposed mechanism when reactive or proactive routing protocols are used. The results confirm that the use of the topology control contributes to a better network performance.

5.1 Introduction

Among the three typical architectures of WMNs [1], this contribution focuses on the client meshing one. In this case, the end-user devices are able to simultaneously provide application interface, routing and network configuration capabilities. Nowadays, the most common hand-held device used by an increasing number of people is the smart phone. The evolution of such mobile devices with their variety of embedded sensors results now in a not only communication equipment, but a complete sensing system [137]. Numerous applications are emerging in many fields like health, traffic, human behavior, environment monitoring, social networking and commerce [137]. With this perspective it is entirely feasible to consider a WMN composed by mobile phone users moving around a city as a cost-effective complement to commercial cellular networks. One of the major concerns in this kind of devices is related to their energy consumption. Therefore, optimization techniques that aim to reduce it are always required [138].

Topology control techniques have been developed to improve the energy efficiency and the battery lifetime of a variety of networks. It also aims to reduce collisions, protocol overhead, and interference by means of a better control over the network connections and redundancy [139]. In general, there are three main types of topology control approaches [139]. First, power control techniques [140, 141], adjust the communication range of the wireless nodes by means of the transmission power of their transceivers. This way, nodes are able to better manage their neighborhood size, interference level, power consumption and connectivity. Secondly, power mode mechanisms [142, 143] control the active or sleep operation modes of the nodes to dispense with redundant stations and still achieve the desired connectivity. Finally, hierarchical clustering approaches [144, 145, 146] aim to construct an efficient virtual backbone for data forwarding by the selection of a connected dominating set (CDS). From graph theory, a CDS of a graph is a connected subset in which all the nodes that does not pertain at that subset have at least one adjacent neighbor inside the subset. Due to the reduced number of nodes developing

routing task, the main advantages of this CDS-based topology control are: collisions, protocol overhead and energy consumption reduction, efficient network organization and scalability improvement. In this chapter we present and evaluate an alternative method for this last category of topology control based on social network analysis metrics.

In this context, thanks to the increasing availability of network maps which depicts the behavior of complex systems and the universality of their characteristics [147], network science appears as the renewed study of the structure and the dynamic behavior of a variety of networked systems [148]. Accordingly, social network analysts have developed an important set of measures and metrics which allow understanding the behavior and quantify the topology features of a diversity of networks [149]. Specifically, in this work we focus on centrality metrics developed to identify the most important actors (nodes) in a network by means of graph theory definitions and concepts.

In summary, the purpose of this chapter is to present and evaluate an alternative topology control mechanism based on centrality measures borrowed from social network analysis. This topology optimization has been applied to a client WMN built by user hand-held devices.

The rest of this chapter is organized as follows. In Section 5.2 we report and analyze the related work. Section 5.3 provides a background on centrality metrics. Section 5.4 presents and evaluates the proposed topology control mechanism. A performance evaluation has been carried out by means of simulations, taking into account both reactive and proactive routing protocols, the results are presented in Section 5.5. The evaluation of the topology control for networks with a community structure is presented in Section 5.6. Finally, the conclusions are summarized in Section 5.7.

5.2 Related Work

Nowadays, the application of complex networks techniques and social network analysis concepts to improve the performance of wireless ad hoc networks is growing as a fertile research area [150]. Some recent works are summarized in the following. First of all, authors in [151, 152, 153] apply the small world phenomena, re-popularized by [154], to reduce the average path length of the network. The basic idea of these proposals is to modify the physical topology of the network based on the social features of the underlying graph. The small world property (or low average path length) could be achieved either by the aggregation of long-ranged links [152] or by a combination of rewiring, deletion and/or addition of links/nodes [151]. Authors in [153] combine centrality measures with directional beamforming to create long-range links between more central nodes. The same authors extend their study to sparse highly partitioned networks in [155].

SimBet [156] is a routing protocol designed for delay-tolerant MANETs. It uses two social network analysis metrics (centrality and similarity) for message forwarding decisions. Betweenness centrality is selected to identify more suitable bridge nodes, and the similarity measure is used to find nodes that are closer to the destination neighborhood. A utility function combines the similarity and the betweenness utilities and allows adjusting the relative importance of them. For performance evaluation both utilities has been assigned the same importance. For their part, authors in [157] apply social network analysis metrics to detect critical nodes in a WMN. They show how network reliability substantially degrades when coordinated attacks are directed to highest centrality nodes. Simulations evince that nodes with high betweenness centrality exhibit a greater impact than nodes with high degree or

closeness centrality. Authors also propose a socially-aware TDMA channel access scheduling algorithm. The main idea is to give higher priority (assigning more time slots) to nodes with high closeness centrality values. Simulations show important throughput improvements at the expense of increased delay.

The time-evolution of the topological characteristics of vehicular networks from the perspective of graph theory and social network analysis is the subject addressed in [158]. It is confirmed that relevant and useful information about the behavior of the VANETs could be inferred from the centrality metrics. The importance of nodes with high centrality values on the design of more efficient VANET protocols is also discussed.

A comparison between the bridging centrality defined in [159] and its distributed version is the subject of the work presented in [160]. Authors use synthetic networks and a small real WMN to prove the validity of the distributed version to identify critical nodes for network management tasks. CSAR (Connection Strength Aware Routing), a recently published social-based routing protocol for delay tolerant networks [161], uses a weighted version of this metric for bridge nodes identification. The routing decision is based on a social-tie (community structure and bridging nodes) detection stage. Each node constructs a social graph following a density-based aggregation approach which uses frequency and duration of contacts information. Then, communities and bridges are distributively detected in the resulting social graphs. As in this last paper, an increasing number of researchers consider fundamental the incorporation of social behavior to reproduce human trajectories and, as consequence, generate mobile social networks with community structures which exhibit statistical properties similar to the ones observed in real experiments. Examples of this new tendency can be found in [162] (which will be described in Section 5.6.2) and in SWIM (Small World In Motion) [163]. This last, in addition of reproducing social community structures, focuses on scaling to large scenarios. The performance evaluation of social-aware routing protocols on such large-scale networks is also included. N-body [164], recently extended in [165], is another work that aims to reproduce the community structure for large networks. Instead of using the underlying social interaction information as in [162], this model extracts the features from real sample traces of small networks.

On the other hand, a topology control algorithm for WSN based in edge betweenness centrality [166] is proposed in [167]. This metric is used to identify most relevant edges or links between nodes, regarding energy consumption. For each node, the aim of the proposal is to select a set of logical neighbors that minimize energy consumption and fulfill QoS requirements. Simulation results show better performance of this proposal in comparison with traditional topology control methods in terms of number of logical neighbors, energy consumption, latency and hit-ratio (percentage of served queries).

Authors in [168] propose a routing protocol based on a connected dominating set (CDS). A simple marking process is used to establish the initial CDS: a node is designed as gateway if it has at least two unconnected neighbors. This initial CDS is reduced by the application of rules based on the node IDs. An extension of the selective removal rules are presented in [144]. In this case the degree and the energy level of the nodes are considered to reduce the CDS and to achieve balanced energy consumption. In [169] Connected Dominating Sets with bounded diameters are taken into account. To construct these DS with the smallest size, authors propose and evaluate two centralized algorithms and one distributed version. On the other hand, given the fact that the transmission ranges of all network nodes are not necessarily equal, authors in [170] model the network by means of a Directed Graph and propose two different solutions for the case in which all the network links are bidirectional. A recent and extensive survey on

energy-aware distributed topology control algorithms is presented in [139].

Many topology control mechanisms are intended to sensor networks in which the dominant data flow traditionally goes from the sensor nodes to the sink. In this work we focus on mesh networks in which it is more common that every node may be origin or destination of the data flow. On the other hand, we evaluate the use of centrality metrics which are calculated in a centralized way (but they have an equivalent distributed version). Besides, we propose and evaluate a distributed implementation of the router selection mechanism, as well as different possibilities to achieve total network connectivity. The community structure of the target mobile social networks is a key aspect that has also been taken into account.

5.3 Background on Centrality

Centrality is one of the most useful mathematical measures developed by social network analysts to capture the structural properties of social relationships. It aims to identify the most important actors/vertices within a graph that represents any physical network. Centrality metrics could be based mainly on the degree of a vertex (number of edges connected to it [149]) or on the geodesic distances between them [150]. In the following we present a summary of the three most useful centrality metrics.

Degree Centrality is the simplest centrality metric. It is defined as the number of edges (links) attached to a vertex (node) [171]. In a WMN the degree centrality of a mesh station can be viewed as the number of one-hop neighbors with which it has been established a peer link. This centrality is generally scaled by the number of nodes N in order to be a measure independent of the network size. Then, the degree centrality Dc_i for the node i is computed as:

$$Dc_i = \frac{\sum_{j=1}^{N} x_{ij}}{N-1} \tag{5.1}$$

where $x_{ij} = 1$ if there is a link between node i and node j and $x_{ij} = 0$ otherwise. In some networks, it could result logical to think that a node which has links to many others (high degree centrality) has greater impact in the network than nodes with few links.

Closeness Centrality aims to identify nodes that spread messages in shorter time [171]. For that, this metric uses the concept of geodesic path, which is the path with the shortest distance between two nodes. The closeness centrality Cc_i for node i is computed as:

$$Cc_i = \frac{N-1}{\sum_{j=1}^{N} d_{ij}} \tag{5.2}$$

where $i \neq j$ and d_{ij} is the geodesic distance between nodes i and j. In social networks, actors with high closeness centrality can communicate their ideas faster than actors with lower closeness centrality [149].

Betweenness Centrality measures the proportion of shortest paths between any pair of nodes passing through a specific node [172]. The control over communications, connections and information flows could be dominated by nodes with high betweenness centrality. The

betweenness centrality Bc_i for node i is computed as:

$$Bc_i = \sum_{j=1}^{N} \sum_{k=1}^{j-1} \frac{g_{jk(i)}}{g_{jk}} \tag{5.3}$$

where $i \neq j \neq k$, g_{jk} is the total number of geodesic paths between node j and k, and $g_{jk(i)}$ is the number of geodesic paths between node j and k that pass through node i.

Two variants of this betweenness centrality will be introduced in Section 5.6.1 since they are close related with community structured networks.

5.4 Topology Control Mechanism and Centrality Evaluation

In this section we propose and evaluate a topology control mechanism based on centrality metrics. We start with a brief description of the dynamic wireless mesh network subject to analysis. Then, a centralized implementation and evaluation is presented in order to identify the most appropriate centrality metric for our purpose. At this point, different considerations are done for a practical distributed implementation. Finally, the practical implementation of the protocol is presented.

5.4.1 Scenario under Consideration

For the experimental evaluation we have considered a dynamic wireless mesh network generated by the ns-3 network simulator. As in Section 4.7.3, the mesh stations are IEEE 802.11s capable nodes. The simulation scenario consists of 100 mesh stations uniformly distributed in a 1040 x 520 m area. According to [173] these values guarantee minimal quality criteria for stringent protocol evaluation. Specifically, the average shortest-path hop count is greater than four hops (to avoid that most of the data packets be interchanged just among one or two-hop neighbors) and the average network partitioning is lesser than 5% (to avoid an excessive number of isolated nodes). The nodes move according to a random walk 2D mobility model inside the rectangular bounds. Each node moves with a speed chosen randomly between 2 and 4 m/s. The direction and speed of the nodes are updated after they have moved 100 m. These values are selected with the assumption that the mobile nodes are transported by people moving around a segment of a city. The initial node positions and their trajectories are shown in Fig. 5.1. For the wireless channel, the log-distance propagation loss model has been considered.

Remember that according to [53], before the transmission of data frames, mesh stations must create and maintain a logical topology using the mesh peering management protocol. Every mesh station discovers its mesh neighbors by means of periodically sent beacon frames. When a new neighbor has been discovered, the mesh station starts a peer link open handshake (Section 2.2.3). A peer link is established between two mesh stations, only if the complete handshake procedure is successfully executed. Standard does not specify why or when a peer link must be closed. The ns-3 mesh networking implementation [60] triggers a close peer link procedure when the number of successive lost beacons achieves a maximum configurable value. By default this threshold is set to 5. If there is an active data flow, the ns-3 mesh model also executes a close procedure when a station is unable to transmit to a peer a number of successive frames. This value is also set by default to 5.

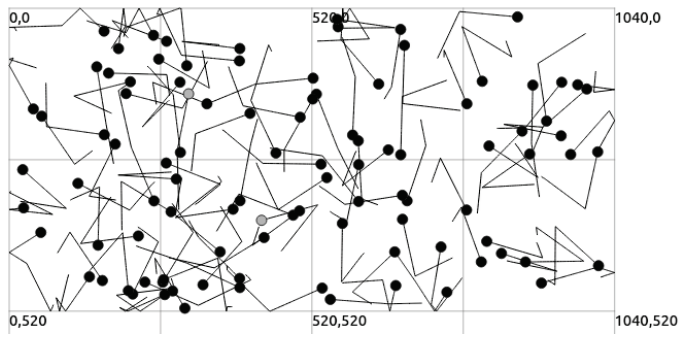


Figure 5.1: Client WMN. Initial node positions and their trajectories.

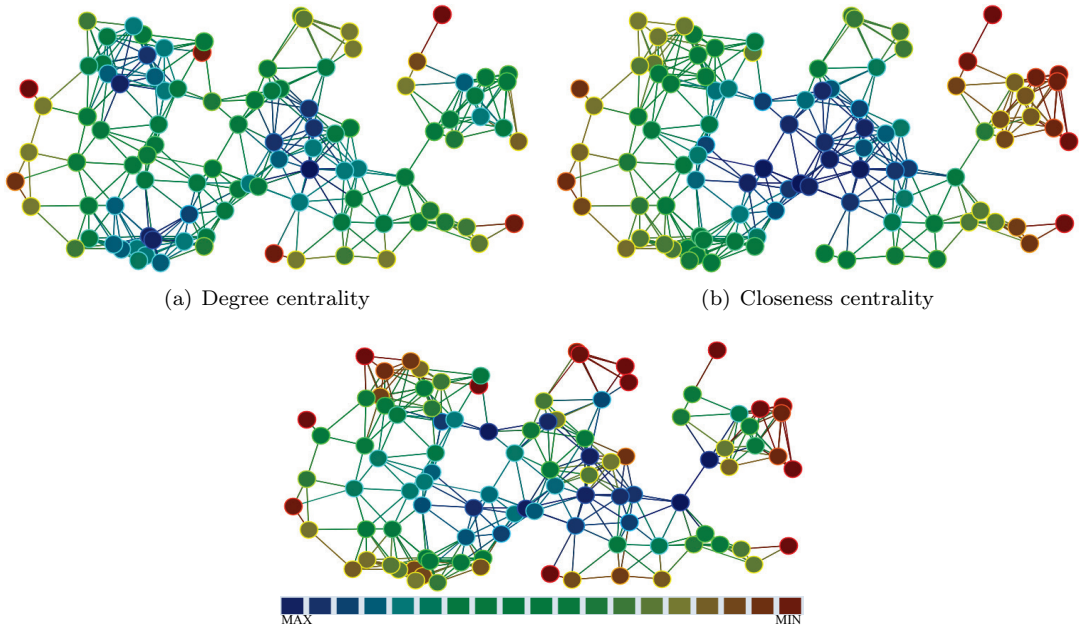
5.4.2 Centralized Implementation

Based on the presented scenario, first of all we use the list of established peer links of each node to periodically take a snapshot of the network. In this way we obtain the set of graphs which represent the network under evaluation for the corresponding time instants. We have analyzed several snapshots of the mesh network dataset taken every 5 s. For each snapshot and for each node inside the network we compute its degree, closeness and betweenness centrality metrics. ORA [174] and UCINET [175] software have been used for centrality calculations, while network graphics are done with ORA. With the centrality values, we are able to rank and identify the most central nodes from those three points of view. Fig. 5.2 shows this ranking for one specific snapshot (at time=5 s) and for the three metrics. Blue nodes represent the most central stations and the red ones the least. As expected, Fig. 5.2 confirms that different nodes are selected as most central ones for each different metric. This is because each metric evaluates a very different parameter and there is no relationship among them.

As previously said, to reduce the complexity, communication overhead and energy consumption of the network, we want to find a subset of mesh stations that forms a connected dominating set. So, instead of allowing each node to perform routing tasks, we choose only the most central stations to do it. In a roughly first approach, we limit the number of routers to the forty (40%) most central nodes for each network dataset and for each centrality metric. We virtually remove all the links between non-router stations and keep all the others. If a non-router station has a link with more than one router, we keep all these links in order to preserve network resilience and to enable future load balancing and fair energy consumption improvements. Fig. 5.3 shows the resulting virtual topology for each centrality metric and for the same snapshot considered in Fig. 5.2. Blue nodes represent the selected routers, the green ones are the mesh stations that have a link with at least one router and the red ones represents the isolated nodes.

To determine which of the centrality metrics is the most suitable for our topology control application, we compare (for all the network datasets and for the three centrality metrics) the resulting network fragmentation F, that is, the proportion of nodes in a network that are disconnected from each other. It can be computed in an efficient way through the following equation [176]:

$$F = 1 - \frac{\sum_k s_k \left(s_k - 1\right)}{N \left(N - 1\right)} \tag{5.4}$$

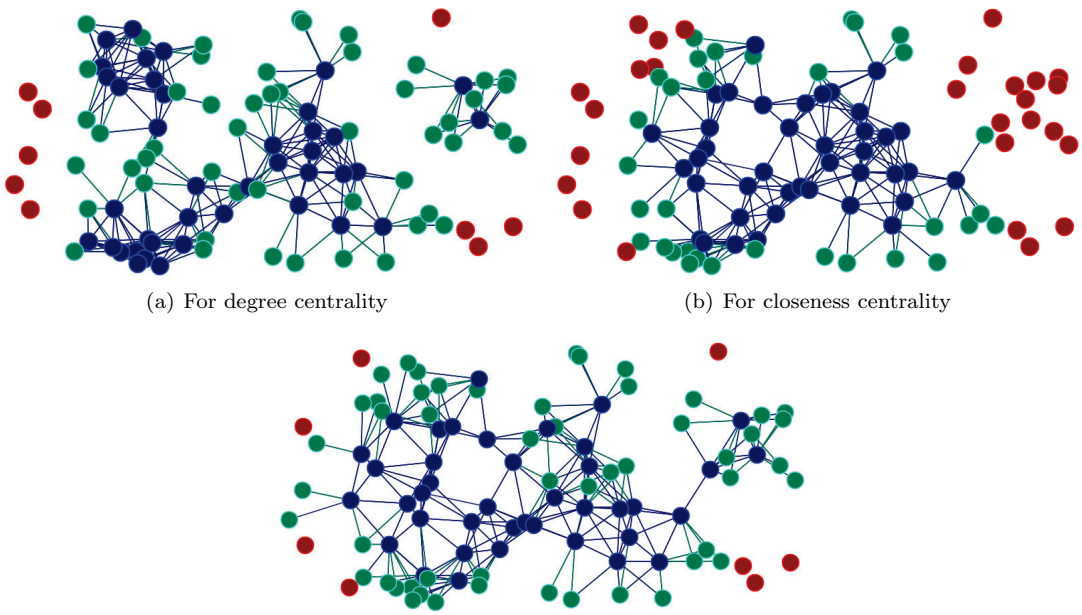


(a) Degree centrality

(b) Closeness centrality

(c) Betweenness centrality

Figure 5.2: Centrality metrics for one of the WMN snapshots.



(a) For degree centrality

(b) For closeness centrality

(c) For betweenness centrality

Figure 5.3: Resulting topology with the 40% most central nodes as routers.

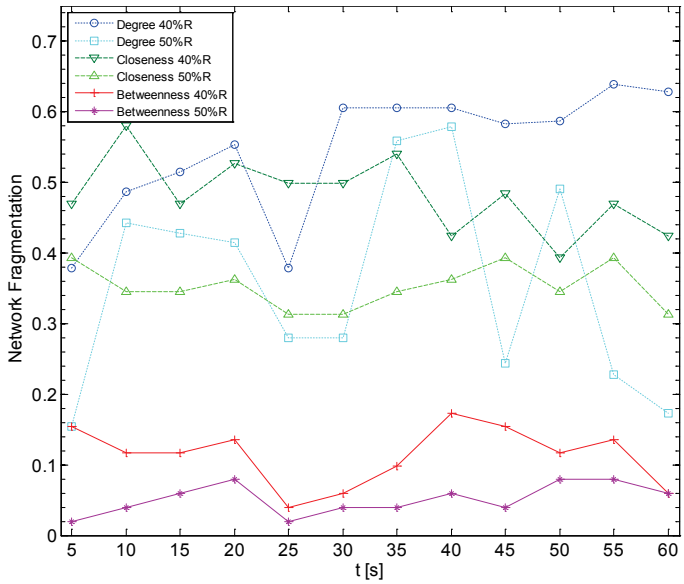


Figure 5.4: Time evolution of the network fragmentation for the three centrality metrics and for 40 and 50% selected routers.

where s_k is the number of nodes of the k^{th} component of the graph that represents the network.

Fig. 5.4 shows the time-evolution of the network fragmentation for the three centralities and for two different percentages of selected routers. As can be seen, the values for the betweenness centrality are considerably lesser than for the other two centrality metrics. And as expected, for all the centralities the fragmentation decreases when a greater percentage of nodes are selected as routers. Besides, Table 5.1 summarizes the average, the standard deviation and the maximum values for this measure. As can be observed, the mean and maximum fragmentation using the betweenness centrality are much lesser than with the other two options. Furthermore, the betweenness-based selection exhibits a more stable (confident) behavior since its standard deviation is the least among the three options. Another metric that we use to compare the three different centralities is the average number of links which remain active after router selection per connected node. Again, betweenness centrality exhibits better performance since on average it requires lesser number of links for each node that remains connected. The maximum and the standard deviation of this metric are also the least among the analyzed options. Other centrality metrics like hub, and authority, have also been evaluated and for all the cases the betweenness centrality exhibits better outcomes. Based on this, we can conclude that by far the betweenness centrality is the metric that better identifies the subset of nodes that should be considered as routers.

5.4.3 Distributed Implementation

In this work, we are dealing with a dynamic network, in which the links among mesh stations continuously change due to their movements and channel conditions. This is done in an autonomous and distributed way through the mesh peering management protocol. Accordingly,

Table 5.1: Summary of the resulting network fragmentation and the number of links per connected node values.

Centrality	%R	Network Fragmentation			Links per connected node		
		Mean	SD	Max	Mean	SD	Max
Degree	40	0.547	0.090	0.639	7.405	0.839	8.635
	50	0.356	0.148	0.579	7.226	0.424	8.158
Closeness	40	0.481	0.053	0.580	6.511	0.371	7.205
	50	0.352	0.030	0.393	6.796	0.362	7.395
Betweenness	40	0.113	0.042	0.173	5.430	0.212	5.691
	50	0.051	0.021	0.079	6.123	0.299	6.573

our topology control algorithm must be implemented in a fully distributed manner. Mesh stations must be able to decide which nodes act as routers based only on their local information, and without the need of any central controller. In the previous section we determined that betweenness centrality is the best choice for our purpose. Therefore, we must use the egocentric version of betweenness centrality [177]. This version requires only the first order (one-hop) neighborhood information of each node instead of the complete network topology. Moreover, this information is available in each mesh station since they keep an updated record of their peer links.

An efficient algorithm to compute the egocentric betweenness is presented in [178]. Each node represents its ego network (one-hop neighbors) by means of the adjacency matrix $\mathbf{A}$. The elements of this matrix are given by:

$$A_{ij} = \begin{cases} 1 & \text{if there is a peer link between nodes } i \text{ and } j \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

As an example, Fig. 5.5 shows the egocentric network structure from the perspective of the node N_{18} at a given time instant.

The adjacency matrix for this ego network is given by:

	N_{18}	N_{16}	N_{27}	N_{44}	N_{45}	N_{61}	N_{64}	N_{73}	N_{87}
N_{18}	0	1	1	1	1	1	1	1	1
N_{16}	1	0	0	0	1	0	0	0	1
N_{27}	1	0	0	1	0	1	1	0	0
N_{44}	1	0	1	0	0	1	1	0	0
N_{45}	1	1	0	0	0	0	0	1	1
N_{61}	1	0	1	1	0	0	1	0	0
N_{64}	1	0	1	1	0	1	0	0	0
N_{73}	1	0	0	0	1	0	0	0	1
N_{87}	1	1	0	0	1	0	0	1	0

Observe in Fig. 5.5 that all the shortest paths between non-adjacent stations that pass through the ego (the station who is calculating its egocentric betweenness) have a length of two hops. Therefore, when $i \neq j$, and being $\mathbf{1}$ the matrix with all its elements equal to 1, the expression $\mathbf{A}^2[\mathbf{1} - \mathbf{A}]_{i,j}$ gives the number of shortest paths with length 2 that links stations i and j. Thus, the egocentric betweenness is the sum of the reciprocal of the resulting non-cero elements [178].

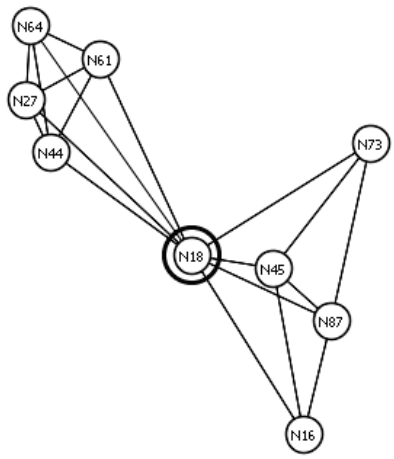


Figure 5.5: Egocentric network structure from the perspective of the node N_{18}.

Note that the peering management protocol guarantees that all the links are bidirectional, and then the matrices are symmetric and only the elements above the main diagonal must be taken into account. For the node N_{18} this results in:

	N_{18}	N_{16}	N_{27}	N_{44}	N_{45}	N_{61}	N_{64}	N_{73}	N_{87}
N_{18}	$*$	0	0	0	0	0	0	0	0
N_{16}	$*$	$*$	1	1	0	1	1	3	0
N_{27}	$*$	$*$	$*$	0	1	0	0	1	1
N_{44}	$*$	$*$	$*$	$*$	1	0	0	1	1
N_{45}	$*$	$*$	$*$	$*$	$*$	1	1	0	0
N_{61}	$*$	$*$	$*$	$*$	$*$	$*$	0	1	1
N_{64}	$*$	$*$	$*$	$*$	$*$	$*$	$*$	1	1
N_{73}	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	0
N_{87}	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$

Therefore, the egocentric betweenness for the node N_{18} is equal to 16.333. This way, whit just local information, every node is able to calculate its egocentric betweenness value. This value should be updated every time a station establishes a new peer link or it closes a previously established one. In a practical implementation, the nodes will refresh this value at predetermined time intervals.

Once selected a distributed way to obtain the centrality values, we also need a distributed way to select the nodes that will act as routers. Note that in this case there is not a central control with all the information needed to select the nodes with higher centralities. Our proposal here is that every node selects as router the node (or nodes) with higher egocentric betweenness in its neighborhood. Note that with this way of proceeding, the same node can be selected as the router for different neighborhoods. This way, the total number of routers is reduced, and this number is determined by the procedure itself. Of course, this simple method guaranties that every station is connected to a router, but it does not assure that all the routers are interconnected between them. Therefore, some further actions will be needed. On the other hand, the main advantage is that it is a very simple and implementable method.

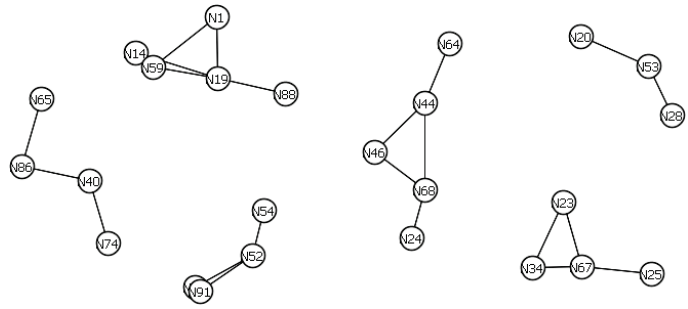


Figure 5.6: Routers selection using 1-hop egocentric betweenness.

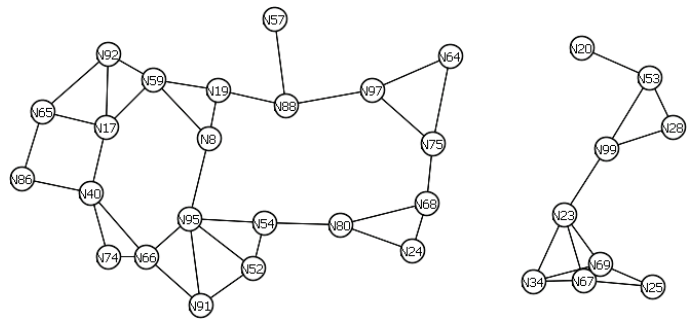


Figure 5.7: Routers selection using 2-hop egocentric betweenness.

With this background, the procedure to select the mesh stations that will develop routing tasks can be summarized as follows: first, each station calculates and broadcasts its egocentric betweenness centrality value, and second, in its respective ego network, each station selects the neighbor (or neighbors) with the highest betweenness to be used as routers. The details and practical implementation of the protocol are exposed in Section 5.4.4.

In the following we present the evaluation of this topology control algorithm for the network dataset previously described. We analyze different alternatives to create a connected dominating set in a distributed way. In first place, we consider that each station computes its egocentric betweenness with the first-order (one-hop) neighborhood information and selects as router just the one with the highest egocentric betweenness. As expected due to the distributed router selection, instead of a connected backbone, this procedure generates a set of isolated clusters that have no common routers among them. Fig. 5.6 presents the worst case obtained in the simulations, that is, the time instant with the higher number of isolated clusters. For this specific case, 25 percent of stations have been selected as routers. Although this is the worst case, it is worth to mention that the 8.33 percent of the network snapshots analyzed construct a fully connected backbone with this most simple procedure, and with 30 percent of stations selected as routers on average.

To improve the connectivity among the selected routers, in the following some different approaches are proposed and evaluated.

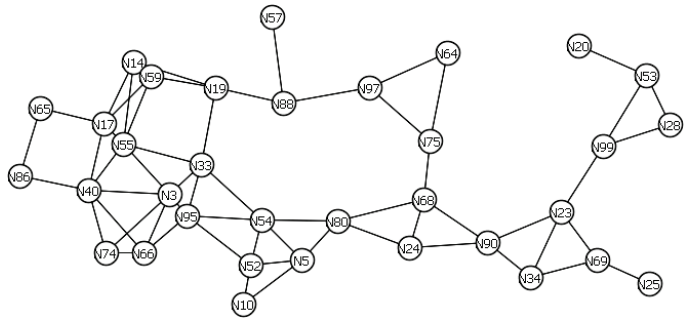


Figure 5.8: Routers selection using the sociocentric betweenness.

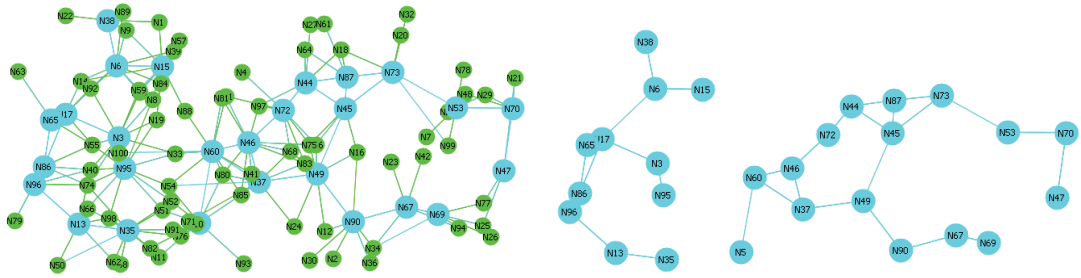
5.4.3.1 K-hops Egocentric Betweenness Centrality

The first possibility is to consider a higher number of hops when computing the egocentric centrality. For instance, we can assess the case in which the stations compute it taking into consideration the two-hop neighborhood information. Again, only the most central station is selected as router in every neighborhood. Fig. 5.7 shows a significant connectivity improvement for the same network snapshot shown in Fig. 5.6. Nevertheless, there are still two unconnected clusters. In this case, 31 percent of stations have been selected as routers.

The theoretical limit for this approximation is the case in which the stations compute the sociocentric instead of the egocentric betweenness centrality. Namely, each station needs to know the complete network topology information, which is not a practical consideration for dynamic WMNs. Fig. 5.8 shows that this procedure creates a complete backbone with 34 percent of stations selected as routers. However this total connectivity among all the selected routers is not achieved for all the network snapshots that have been evaluated. Then, this procedure does not guarantee the creation of a CDS.

5.4.3.2 Probe Packets

In the following, we describe another possible solution which guaranties that all the selected routers create a single connected backbone at the expense of increasing the signaling overhead. After a first router selection round, based on egocentric betweenness, the non-router stations virtually linked to at least two routers verify the connectivity between them. If there is no path between any two routers connected to a non-router station, that station must be marked as router. To this end, the station transmits a probe message through one (and only one) of its links. This special message must be retransmitted by the routers and, if connectivity exists, it will be received by the sending station through the other links. If the message is not received before a predetermined time, the station marks itself as router and communicates this decision to its neighborhood. To illustrate this, Fig. 5.9(a) shows a different network snapshot. The cyan bigger nodes represent the stations that have been selected as routers based on one-hop egocentric betweenness and a selection with just one router per station. As it can be seen, selected routers forms two components (Fig. 5.9(b)) that can be linked through either of stations N_{88}, N_{33}, N_{54}, N_{52}, N_{51}, N_{91}, N_{82} and N_{11}. These nodes will be marked as routers according to the proposed procedure.



(a) Complete topology (b) Selected routers

Figure 5.9: Connectivity achieved with the probe packets approach.

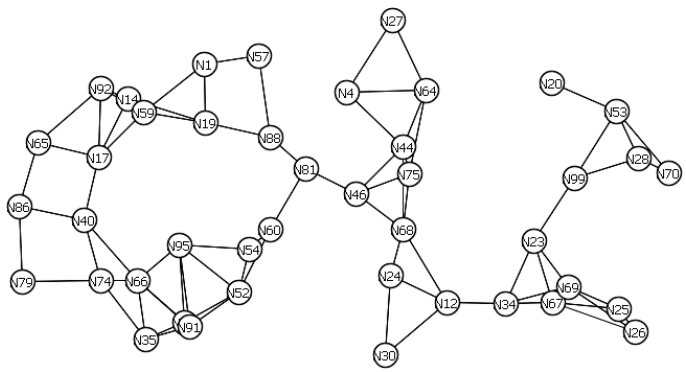


Figure 5.10: Resulting backbone selecting two routers per neighborhood.

5.4.3.3 Increasing the Number of Routers per Neighborhood

We have evaluated another procedure in which the stations, instead of selecting only the most central node, mark as routers the two nodes with the higher one-hop ego betweenness values in each neighborhood. Fig. 5.10 shows the resultant backbone for the same mesh network dataset. In this case, 43 percent of stations have been selected as routers. This process generates a fully connected backbone for the 83.33 percent of the network snapshots tested. The increase in the number of redundant routers can be exploited by load balancing techniques, and it is also valuable from a network resilience perspective. Due to the fact that this approach is the one with a lower amount of overload, we have chosen it to propose a practical implementation in the next section. Obviously, incrementing the number of routers selected by each station will ensure a fully complete backbone for all the scenarios at the expense of a higher router redundancy. As a future line of work we will study the minimum number of routers per neighborhood needed to guaranty a desired connectivity (expressed as a percentage of the total connectivity).

5.4.4 Practical Implementation

Energy efficiency and fair energy consumption are consider for the practical implementation of the protocol. It is carried out in three steps:

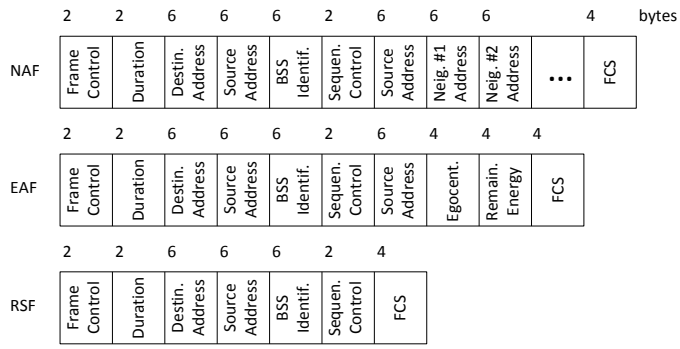


Figure 5.11: Frames format.

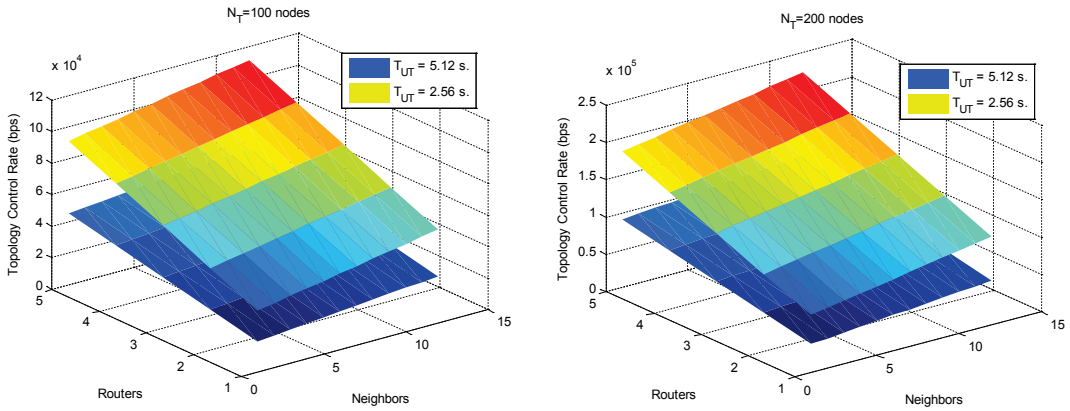
- **Step 1:** Every node broadcasts its own neighbor's list by means of a Neighborhood Advertising Frame (NAF). This is a special management frame where every neighbor is identified by its MAC address.
- **Step 2:** Every node receives the NAFs from its neighbors. With this information, they are able to compute their own egocentric value. Once computed, they broadcast that value in the Egocentric Advertising Frame (EAF). In order to allow fair energy consumption in the network, the node's remaining energy is also included in the frame.
- **Step 3:** Every node receives the egocentric value of all of its neighbors and selects the one with the higher value as its router. If different options are available (i.e., different neighbors with the same centrality value), the node with the higher value of remaining energy is chosen. A multi-metric protocol, assigning different weights to the centrality and energy remaining values, is one of the targets of a future work. The decision is communicated to the new router with the Router Selection Frame (RSF). When the new router receives the RSF, it switches to the router mode (or router state) and activates its routing functionality. As said in the previous section, in order to guarantee obtaining a connected topology, a bigger number of routers can be chosen by every node.

Due to the network variability, this three steps procedure must be repeated every certain interval time, called the Topology Updating Time (TUT). Besides, the router state must be seen as a soft state. That is, to keep the routing functionality activated, a node must receive at least a RSF every TUT seconds. Otherwise the node understands that its routing functionality is not needed anymore and switches it off.

Fig. 5.11 shows the specific frame format for the new three management frames needed by this protocol. The NAF and EAF frames can be included into the mesh-specific management frames. They are broadcast frames and therefore they do not need to be acknowledged by the receiver. The RSF frame is also a mesh-specific management frame but in this case it is a unicast frame and therefore it must be acknowledged by the receiver. The three frames can be identified by the sub-type code in the Frame Control field.

Calling N_T the total number of nodes in the network and N_R the number of routers that every node selects, the total number of bytes transmitted by this protocol procedure is:

$$N_T \left[\text{size}\left(NAF\right) + \text{size}\left(EAF\right) + N_R \left[\text{size}\left(RSF\right) + \text{size}\left(ACK\right)\right]\right] \quad \text{bytes} \tag{5.6}$$



(a) 100-nodes mesh network

(b) 200-nodes mesh network

Figure 5.12: Topology control protocol bit rate for different N_T and TUT values.

where size(X) stands for the size in bytes of the frame X.

As it can be seen in the Fig. 5.11, the size of the NAF depends on the number of neighbors. Calling N_N to the average value of this number, the average size of these frames is $34 + 6N_N$ bytes. The size of EAF and RSF frames are 42 and 28 bytes respectively. Substituting these values in Eq. 5.6 we obtain:

$$N_T\,(34 + 6N_N + 42 + 42N_R) = N_T\,(76 + 6N_N + 42N_R) \quad \text{bytes} \tag{5.7}$$

This amount of bytes is sent by the protocol every TUT seconds. Therefore, the bit rate added by the topology control protocol to the network, R_{TC}, is:

$$R_{TC} = \frac{8N_T\,(76 + 6N_N + 42N_R)}{TUT} \quad \text{bps} \tag{5.8}$$

Obviously, R_{TC} grows linearly with N_T, N_N, N_R, and $(1/TUT)$. These dependencies are shown in Fig. 5.12 for several values of the involved parameters. It is important to remark that the lifetime of a network topology must be greater than or at least equal to the lifetime of a routing path. In the figure two typical values have been considered (one of them, 5.12 s, is the standard default value). The resulting protocol bit rate will affect the node energy consumption and the network load, and will be taken into account later in the performance evaluation.

5.5 Performance Evaluation

In this section, extensive simulations have been carried out to evaluate the performance of WMNs with the proposed centrality based topology control mechanism. For the evaluation, we consider the following metrics and parameters:

– The rate of routing management messages.

– The total number of forwardings per successfully received packet.

- The network efficiency in terms of packet delivery ratio.
- The energy consumed on a node by node basis.
- The end-to-end packet delay.
- The transmission bit-rate savings taking into account the overload added by the topology control protocol.

We take into account the two routing modes of the hybrid wireless mesh protocol (HWMP). Remember, from Section 2.2.5, that in the reactive on-demand mode, a source mesh station starts a path discovery procedure only when it has a data packet to send to any destination. This operation mode is more suitable for peer to peer communication inside the mesh network. On the other hand, proactive mode is more appropriate when a single or a few mesh stations are configured as gateways for external communications and therefore they are the targets of most of data connections. In this mode, a single or few mesh stations are configured to be path tree roots, and they periodically initiate path selection procedures.

All simulations are done with the ns-3 network simulator. To model the mesh network, we have used one of the network snapshots described in Section 5.4.2. It consists of 100 IEEE 802.11s-based mesh stations randomly distributed in a 1040 x 520 m rectangular area. For all the experiments, we compare the performance of the mesh network when all the station are configured as routers with the case that only 40 most central nodes are selected as routers (from now they will be called the All R and Top 40R topologies for convenience). All the simulations are 500 s long and we have carried out 100 statistically independent runs for every experiment. The average of these runs with 95% confidence interval is shown for all the results. In the following, we present and discuss the simulation results for different mesh network conditions and configurations.

5.5.1 Reactive Mode. Impact of Traffic Load

The aim of this part is to evaluate the impact of the traffic load on the network performance when peer-to-peer communications are more relevant and then, mesh stations are operating in the reactive mode. We vary the number of simultaneously active UDP data flows from 5 to 40. The source and the destination nodes are randomly selected for each independent run and for each flow. All the UDP flows are configured with an exponential distribution for the interarrival time with mean value of 20 ms. For the packet length we use an exponential distribution with mean of 512 bytes that has been truncated to a minimum of 12 bytes (header size), and a maximum of 1500 bytes. Mesh stations are configured with the default value of 5.12 s for the lifetime of reactive routing information.

First of all, Fig. 5.13 shows the evolution of the mean rate of routing management messages as a function of the number of data flows. Evidently, since mesh stations are using the on-demand path selection mode, a greater number of sources demand a greater number of path selection procedures. Therefore, the rate of routing management messages increases when the number of data flows increases. As expected because of the lower number of mesh stations with path selection functionality, a significant reduction occurs (76.14% on average) when the mesh network operates with the topology control mechanism. Observe that this reduction directly impacts on the total energy consumed by the network.

The next parameter that we consider is the total number of unicast data packets forwarded by all the stations in the network (Fig. 5.14). We can clearly identify two different trends of this

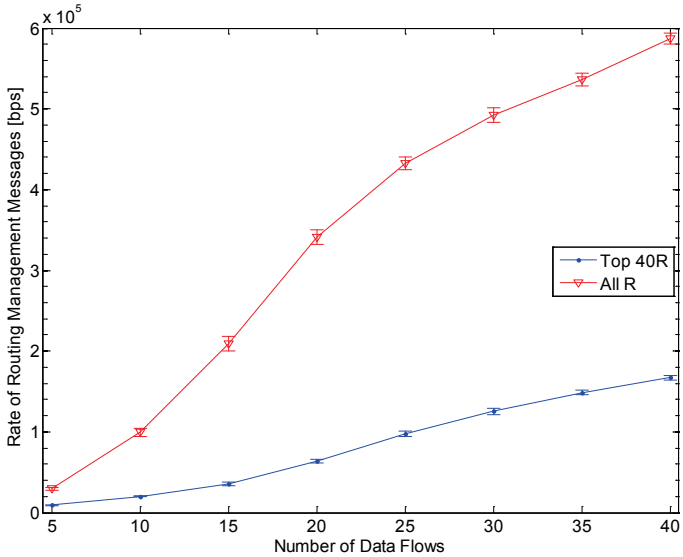


Figure 5.13: Rate of routing management messages as a function of the number of active flows.

parameter. For low traffic loads, up to 15 concurrent flows for the network without topology control and up to 20 flows when topology control is applied, the data forwardings growth near linearly with the number of flows. Beyond these values the network starts to exhibit a saturated behavior and then the growth becomes less significant. This is also verified when we study the total number of successfully received packets (Fig. 5.15). This figure confirms that the network is saturated beyond 15 data flows for the All R topology and beyond 20 flows for the Top 40R case. Over these values, although there are more sources generating data packets, there is practically no increase in the number of successfully received data packets. The topology control mechanism produces a savings of the total data forwardings of 32.77% on the average. This is another important reduction of the total amount of energy consumed by the network. It is important to remark that this is not in detriment of the number of received packets. In fact, when the topology control is applied, the stations get on average 2.3% more data packets than without topology control. Precisely, to better analyze this relation we compute the total number of data forwardings per successfully received packet. Fig. 5.16 shows the evolution of this measure and it can be observed that for all the cases the network with topology control is more efficient. Namely, the number of forwardings is reduced a 33.43% on average. We also verify that an increase of traffic sources causes more collisions and therefore a greater number of retransmissions and forwardings are required to deliver each data packet. This growing trend continues until the network reaches the saturated state.

The next parameter that has been evaluated is the network efficiency in terms of packet delivery ratio. Fig. 5.17 shows that the network with the topology control becomes more efficient when the number of sources is equal or greater than 20. This is because the impact of the signaling required for path selection is not so important when a few number of stations are generating data traffic. Nevertheless, in more practical situations, the routing overhead is more relevant and then the advantage of reducing the number of stations involved in routing is evident. Within the interval of this experiment, although this is not our main objective,

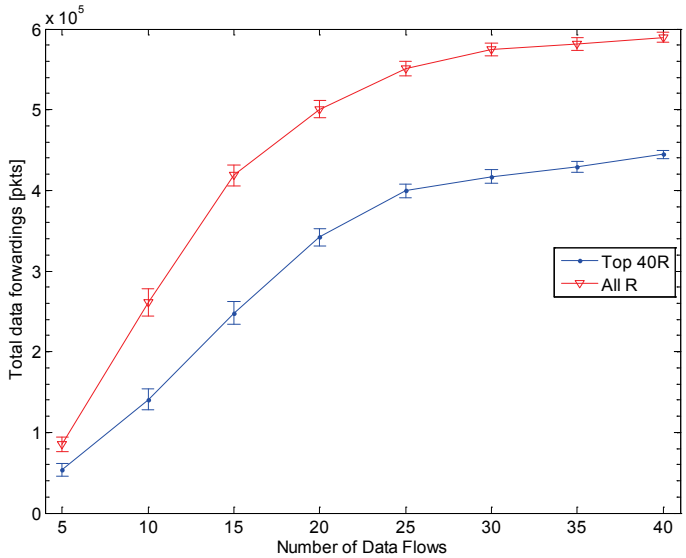


Figure 5.14: Total data forwarding as a function of the number of active flows.

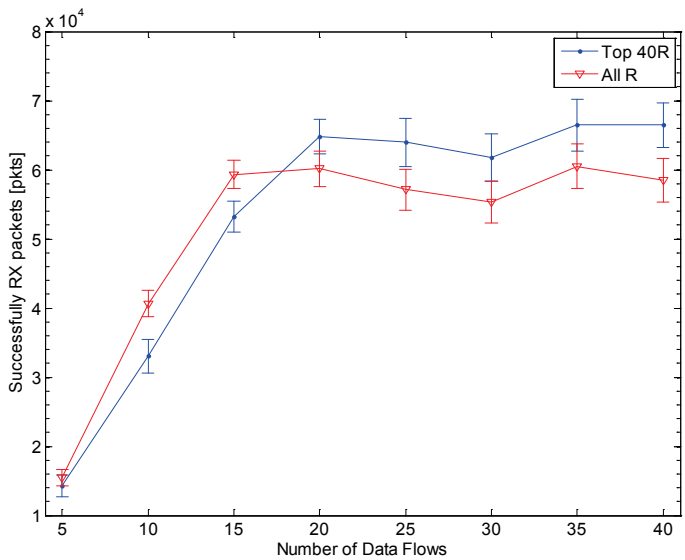


Figure 5.15: Total number of successfully received data packets.

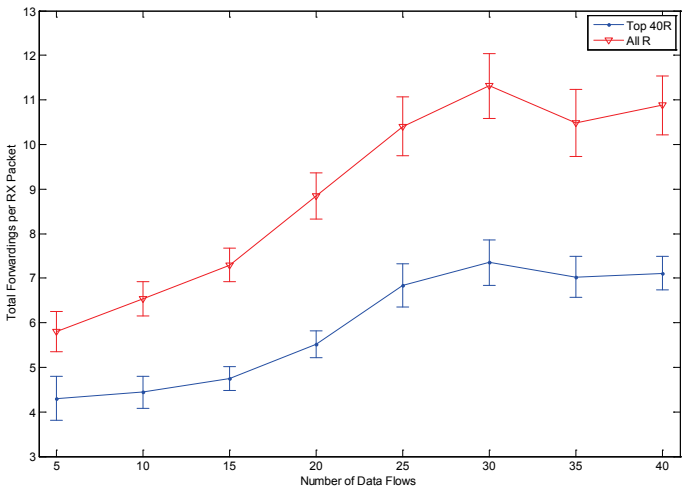


Figure 5.16: Total data forwardings per successfully received packet.

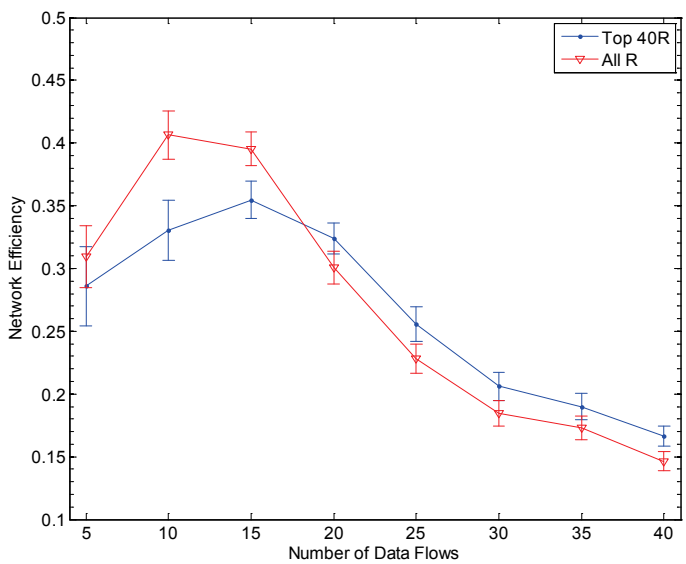


Figure 5.17: Network efficiency in terms of packet delivery ratio as a function of the traffic load.

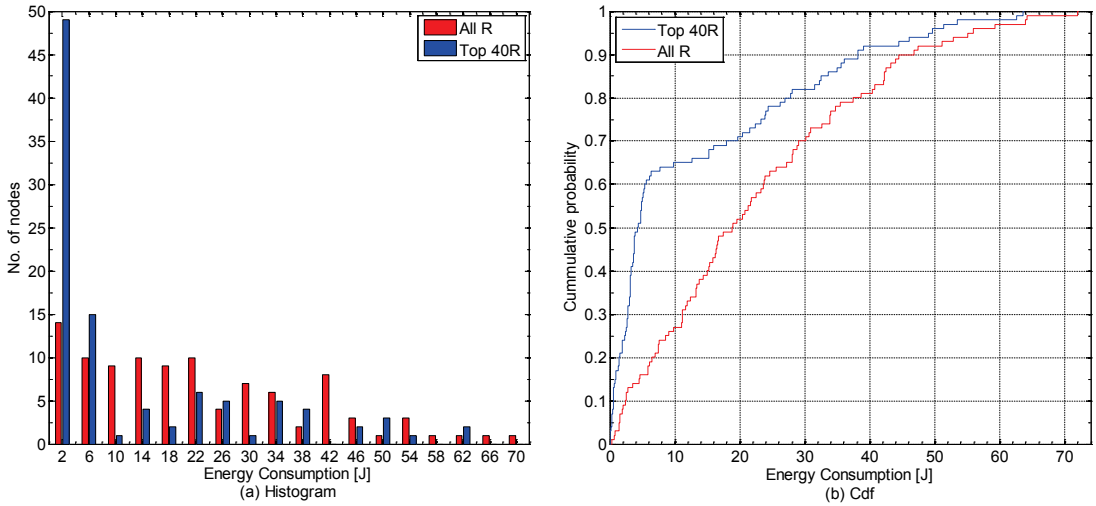


Figure 5.18: Energy consumption statistics for the 15-flows reactive mode case.

the average improvement of the network efficiency is about 3.7% when the topology control is applied.

This small improvement in the efficiency comes together with a significant reduction of the global energy consumption inside the network, which is in fact the main goal of this proposal. To evaluate this, we consider the total amount of energy dissipated by each node for all the transmissions and receptions during the entire simulation time. The energy model presented in [179] has been considered for all the reported cases. This model assumes an r^2 energy loss due to channel transmission. Then, to transmit a k-bit message over a distance d, the radio expends [179]:

$$E_{Tx}(k,d) = E_{elec} \cdot k + \epsilon_{amp} \cdot k \cdot d^2 \tag{5.9}$$

and to receive the message, the radio expends:

$$E_{Rx}(k) = E_{elec} \cdot k \tag{5.10}$$

where, it is considered that the transmitter or receiver electronics dissipate $E_{elec} = 50$ nJ/bit and the transmitter amplifier dissipates $\epsilon_{amp} = 100$ pJ/bit/m^2.

Fig. 5.18(a) shows the histogram of the energy consumed by the nodes for the 15-active flows case (which corresponds to a non-saturated situation). As can be observed, the maximum energy dissipated by a node has been reduced from 72 to 63.6 J when topology control is applied. The total energy consumed by the network has been reduced around a 41% (from 2247.4 to 1324.4 J). Besides, the cumulative distribution function of the energy consumption (Fig. 5.18(b)) shows also the better behavior achieved with the topology control. For instance, it can be observed that around 27% of nodes dissipate less than 10 J for the All R case, and that this figure is increased to 65% for the Top 40R case.

Finally, Fig. 5.19 shows the cumulative distribution function of data packets end-to-end delay. As expected, a greater number of data sources increases the contention for channel access and also the path selection messages overhead. This implies that path selection requests require more time to be resolved and then the end-to-end delay of data packets grows with the

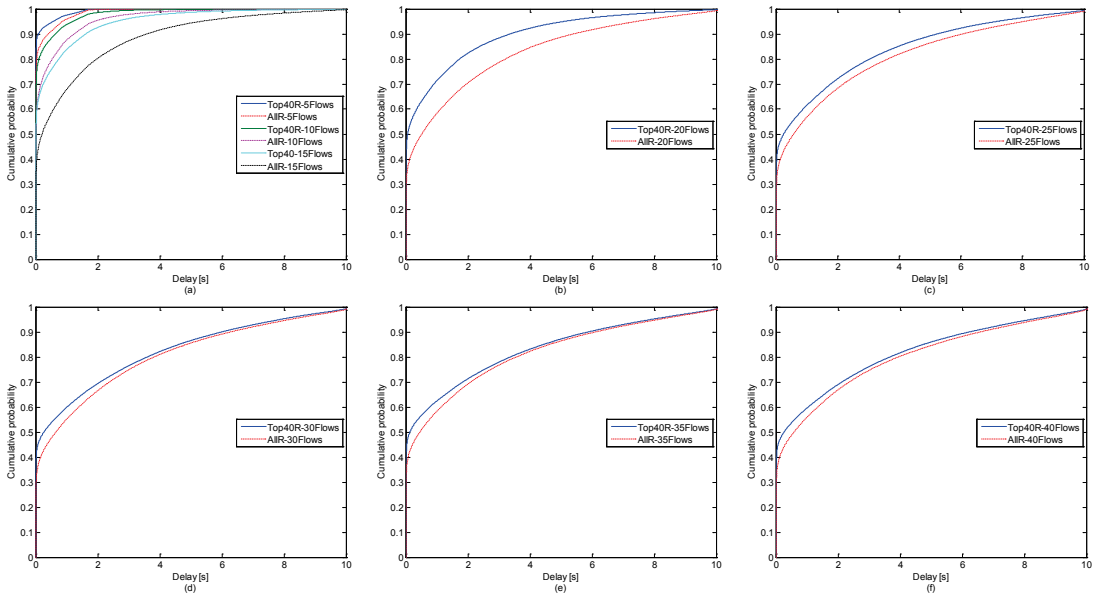


Figure 5.19: Cumulative distribution function of data packets end-to-end delay.

number of traffic sources. For all the cases the delay whit the Top 40R is lower than with the All R topology. The difference is more significant for low traffic loads (Figs. 5.19a–b) and is reduced when the network is saturated (Figs. 5.19c–f).

5.5.2 Reactive Mode. Impact of Reactive Routing Information Lifetime

The aim of this set of experiments is to analyze how dynamic a network could be. Mesh stations continue working in reactive mode and 20 simultaneous peer-to-peer data flows are established between randomly selected sources and destinations. A more dynamic and variable network requires a more frequent activation of the mesh path selection mechanism in order to keep the paths updated. As stated in the IEEE 802.11-2012 standard [53], we can control the time for which mesh stations consider the forwarding information to be valid by means of the *dot11MeshHWMPactivePathTimeout* control variable. Therefore, in this section we evaluate the performance of the network when this lifetime varies from 5.12 to 1 second. The evolution of the mean rate of routing management messages is shown in Fig. 5.20. Of course we can observe that, although the traffic load is equivalent for all the cases, the rate of path selection related messages grows as the lifetime of reactive routing information decreases. As expected, topology control produces an important reduction of around 83.43% because of the reduced number of mesh stations involved in routing. This directly corresponds to the same significantly energy consumption savings due to the lower amount of routing messages exchanged by the mesh stations. We verify that the limitation in the number of routers has no negative impact in the total number of successfully received packets. In fact, as can be seen in Fig. 5.21, topology control produces an average increase of around 8.75% in the number of successfully received packets with regard to the All R configuration. The decreasing behavior is explained as follows:

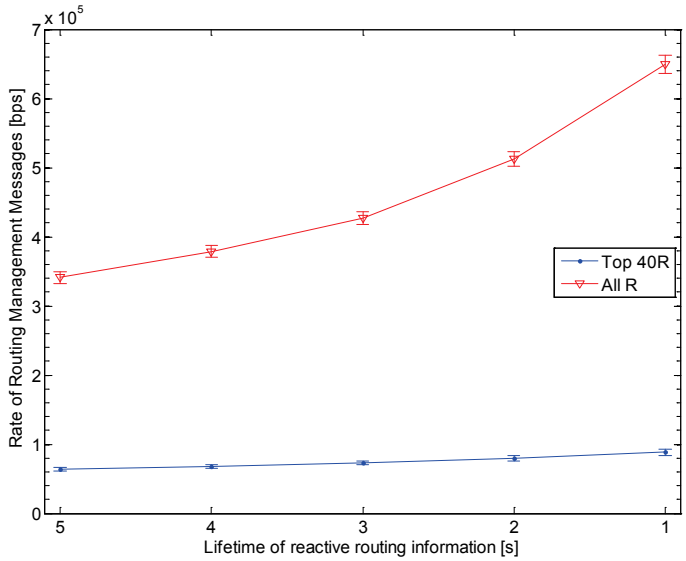


Figure 5.20: Rate of routing management messages as a function of the reactive path lifetime.

lower duration of established paths implies a greater number of path discovery messages which in turn increases the collisions and the contention inside the network. Thus, data packets have less transmissions opportunities and then there are less successfully received packets.

This behavior is also confirmed when we analyze the total number of data forwardings per successfully received packet (Fig. 5.22). We can observe that, although the traffic load remains equivalent for all the cases and even the increasing number of collisions requires more data packet retransmissions, the high network contention caused by the routing messages overhead reduces the transmission opportunities of data packets and then the number of forwardings decreases instead of remain constant or grow due to collisions. Nevertheless, it is important to remark that the topology control contributes with an average reduction in the number of data forwardings of about 42.19%. This represents again a considerably energy consumption savings.

The network efficiency in terms of packet delivery ratio also exhibits a decreasing tendency when the lifetime or reactive routing information decreases (Fig. 5.23). As previously said, this is due to the big amount of routing management messages overhead required for a faster update of the established paths. An important observation is that, for a given network efficiency value, the topology control allows almost twice routing information updates compared with the network operating with all routers. For example, the efficiency reached by the All R network with the default active path timeout value of 5.12 s could be achieved with the Top 40R topology even if the path updates are done approximately every 2.7 s. In summary, topology control also improves the scalability when more dynamism and variability is required.

With the same criteria of the previous section, Fig. 5.24 shows the energy consumption statistics for the network configured with a reactive path lifetime of 3 s. The histogram (Fig. 5.24(a)) shows that the maximum energy dissipated by a node in the network reduces from 79.3 to 71.3 J when topology control is applied. The total energy dissipated in the network reduces around a 38.44% (From 2655.8 to 1635 J). The cdf (Fig. 5.24(b)) shows that while only

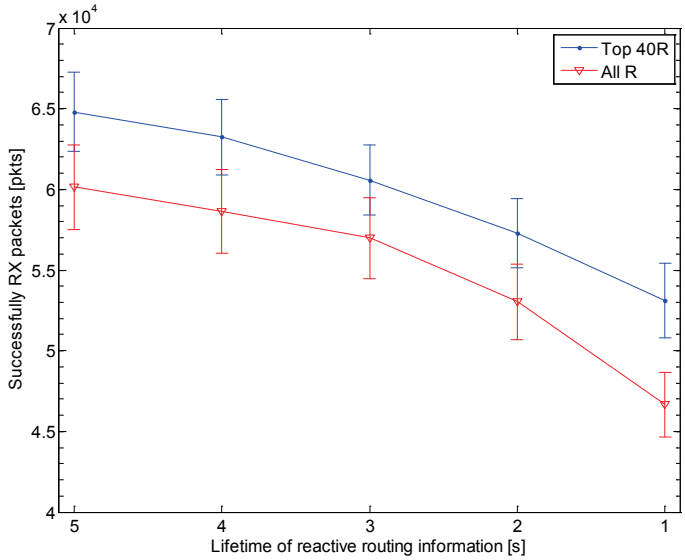


Figure 5.21: Total number of successfully received packets as a function of reactive path lifetime.

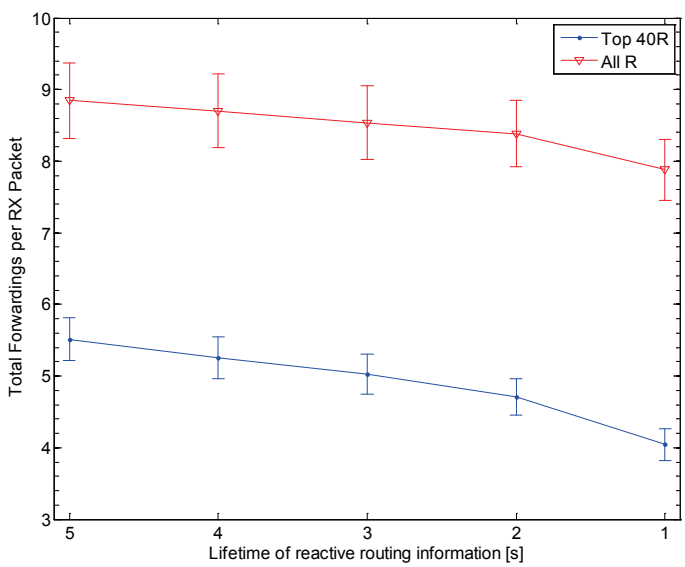


Figure 5.22: Total data forwardings per successfully received packet as a function of reactive path lifetime.

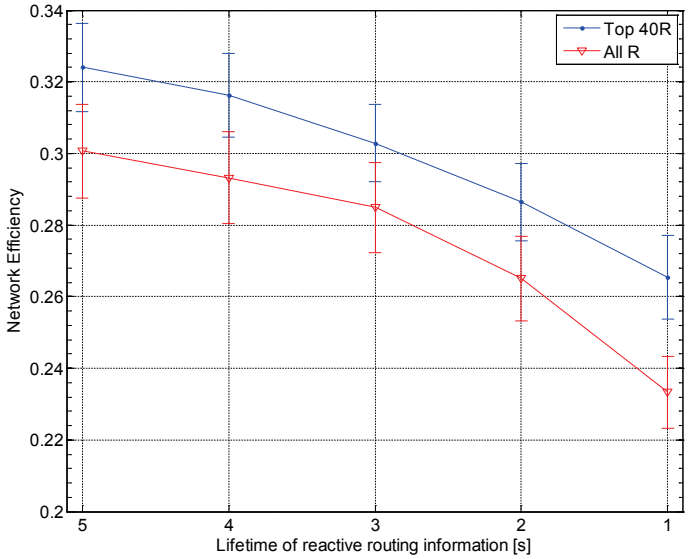


Figure 5.23: Network efficiency as a function of reactive path lifetime.

23% of nodes consumes less than 10 J in the All R configuration, this percentage increases to 67 in the Top 40R network.

The data packets end-to-end delay analysis (Fig. 5.25) reveals that the delay decreases when the path updates are more frequents. This is due to the fact that the network reacts earlier to link failures and to paths that should no longer be valid for data forwarding. But it is also because there is a lesser number of data packets forwarded by intermediate stations and a lesser number of successfully received packets. When the level of routing overhead increases, most of the packets that finally arrives to the destination are those that traverse a few number of intermediate nodes. Therefore, those packets arrives in shorter time. On the other hand, it is important to remark that the application of topology control improves the performance of the network regarding end-to-end delay for all the simulation cases. Even the worst case results for the network with topology control (Lifetime = 5.12 s) are better than the results for network with all routers and lifetime greater than or equal to 2 s. The reason for this is that the paths are established faster when a lesser number of stations are involved in the path discovery procedure.

5.5.3 Proactive Mode. Impact of Proactive Routing Information Lifetime

Proactive mode of HWMP protocol is more suitable when the target of data communications is mainly outside of the mesh basic service set. The most common use cases are mesh stations accessing to Internet or the internetworking with other network technologies. In these cases, one or more mesh stations are configured to be the root mesh station(s) and they play the role of gateways. Each root periodically sends path request (PREQ) messages in order to build a proactive tree, and thus, create paths from all the mesh stations to the root.

As in the previous section, we begin evaluating how fast proactive data forwarding

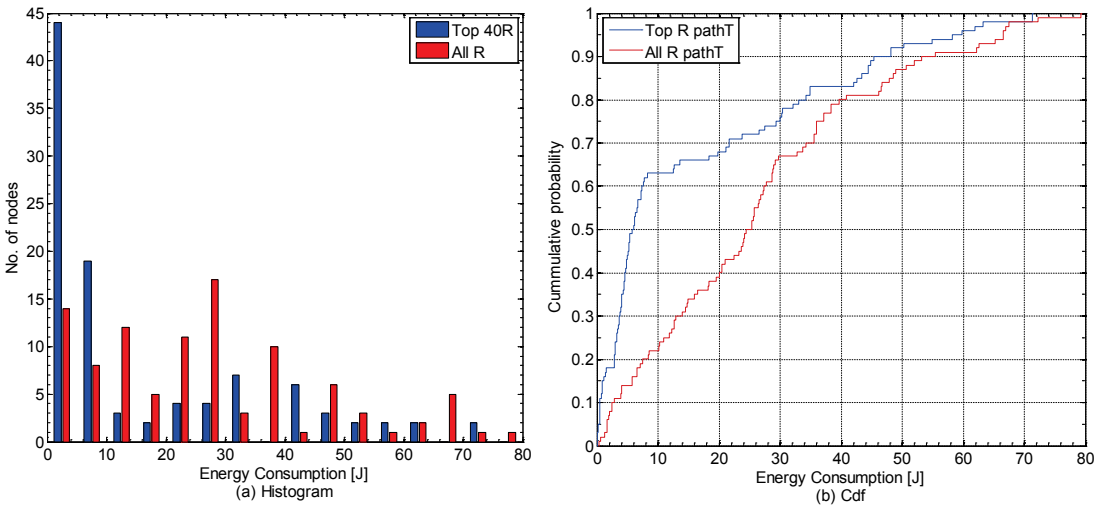


Figure 5.24: Energy consumption statistics for the nodes working with the reactive mode, 20 peer-to-peer data flows and reactive routing information lifetime of 3 s.

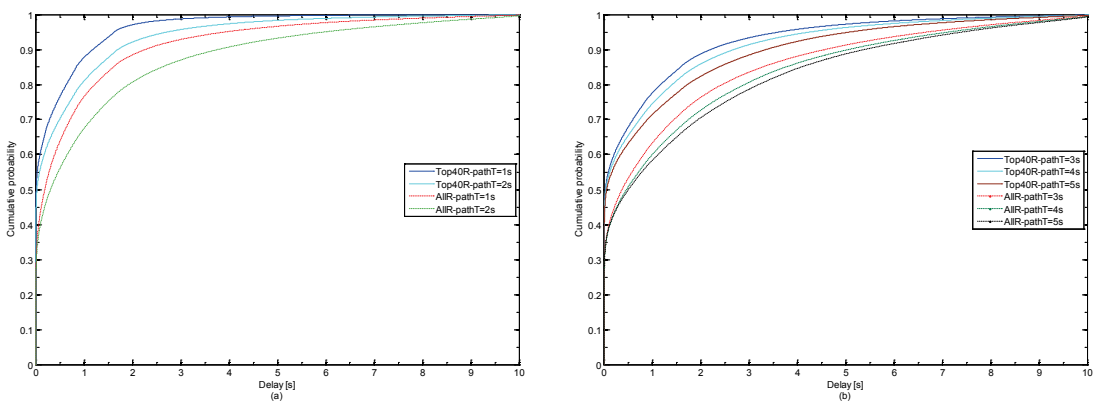


Figure 5.25: Cumulative distribution function of data packets end-to-end delay for different reactive path lifetimes.

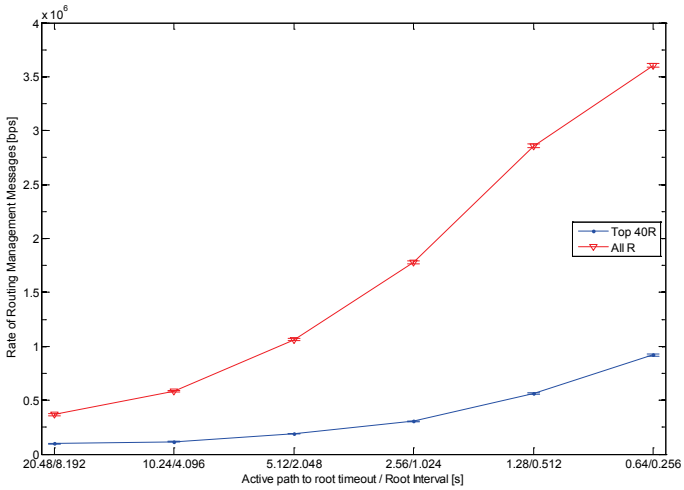


Figure 5.26: Rate of proactive routing management messages for different update times.

paths could be updated. A faster route update is required by more dynamic networks in which stations continuously vary their established peer links. Obviously, this implies a greater path selection management messages overhead. As stated in the IEEE 802.11-2012 standard [53], the lifetime of the proactive routing information is configured by the *dot11MeshHWMPactivePathToRootTimeout* control variable. This variable should be greater than *dot11MeshHWMProotInterval* which in turn controls the frequency of proactive PREQ frames sent by the root. For the evaluation we configure one station as the root and then we vary the values of these pair of variables from 20.48 to 0.64 s for the active path to root timeout and from 8.192 to 0.256 s for the root interval. As it can be seen, we maintain constant the default ratio between these variables in all the reported cases. For each independent run, the traffic load consists of 25 simultaneously active data flows, five uploads from randomly selected mesh stations to the root and 20 downloads from the root to randomly selected stations. The expected growth of routing management messages rate as the lifetime of routing information decreases is verified in Fig. 5.26. We also confirm that proactive routing overhead is greater than the reactive one. For instance, when the default routing lifetime values are used and for the 25 flows case, the routing management rate increases from around 432 Kbps (Fig. 5.13) to 1.1 Mbps for the All R configuration, and from around 97 Kbps to 190 Kbps for the network with topology control. The application of topology control produces an average reduction of around 78.87% in the routing management messages overhead.

The increase of routing overhead due to faster path updates causes more collisions and then data packets need to be retransmitted a greater number of times. This is verified in Fig. 5.27 which shows the evolution of total data forwardings per successfully received packet as a function of routing information lifetime. It can be seen that, although the traffic load remains equivalent for all the cases, the average number of forwandings required to successfully transmit each data packet increases as the lifetime decreases. The average reduction achieved with the application of topology control for this parameter is around 51.75%.

The growing routing overhead makes increasingly difficult to successfully transmit data packets and then the network efficiency gradually reduces (Fig. 5.28). As previously said,

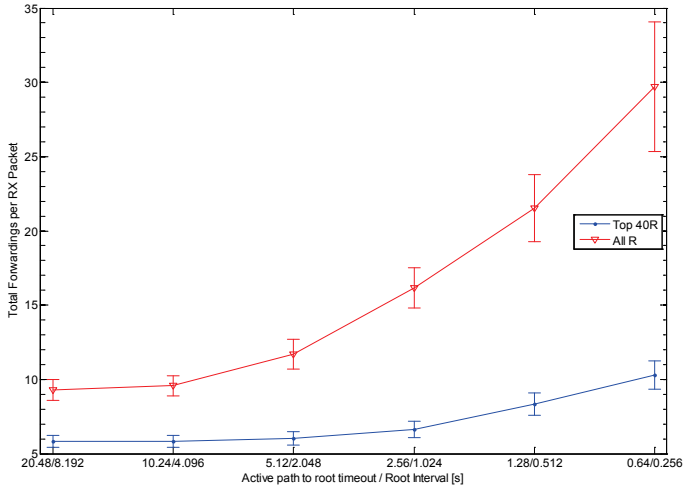


Figure 5.27: Total data forwardings per successfully received packet for different proactive path lifetimes.

although this is not the main objective of the topology control mechanism, the network efficiency significantly improves when only the most central stations are involved in the path selection algorithm. The average improvement achieved in the presented range is about 58.17%. This improvement is much more significant than in the reactive mode case in which the path selection overhead level is lower. Fig. 5.28 also shows that, for a given network efficiency, the mesh network with topology control supports almost four times faster path selection updates.

The energy consumption statistics for all the nodes in the network, including the root mesh station configured with the default values for the routing information lifetime (i.e. the active path to root is 5.12 s and the root interval is 2.048 s) are shown in Fig. 5.29. As expected, the station configured as root dissipates significantly more energy than non-root stations since all the traffic (data and routing) starts and ends in that node. For the root station, the energy dissipation reduction achieved with the topology control is around 14.79% (from 775.1 to 660.5 J). The total energy dissipated in the network reduces around 30.4% (from 3233.1 to 2250.2 J). These higher energy dissipation values also confirm the greater overhead of the proactive mode regarding to the reactive one. The cdf of Fig. 5.29(b) shows that 52% of nodes dissipates less than 20 J for the All R configuration and this value increases to 73% in the Top 40R network. On the other hand, since the root is a special type of station, Fig. 5.30 shows the energy consumption statistics for only non-root stations. The histogram and the cdf in Fig. 5.30 identifies a drawback of the proposed topology control mechanism: three nodes selected as routers are overloaded regarding the All R configuration. This fact also confirms that the remaining energy of nodes must be included in the router selection process as stated in Section 5.4.4.

The end-to-end delay cdf (Fig. 5.31) also confirms that each successfully received data packet had to be retransmitted a greater number of times. It can be observed that faster proactive path updates increases the data packet delays. This behavior holds for both network configurations, with or without topology control. Nevertheless, delays are smaller when topology control is applied due to the lower channel access contention.

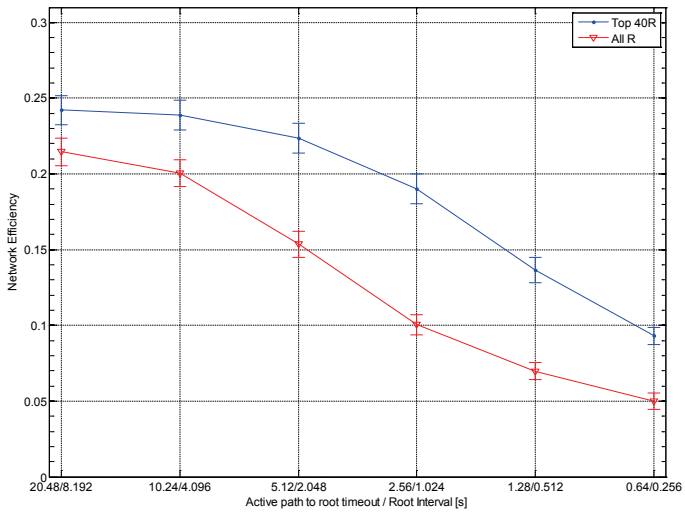


Figure 5.28: Network efficiency for different proactive path lifetimes.

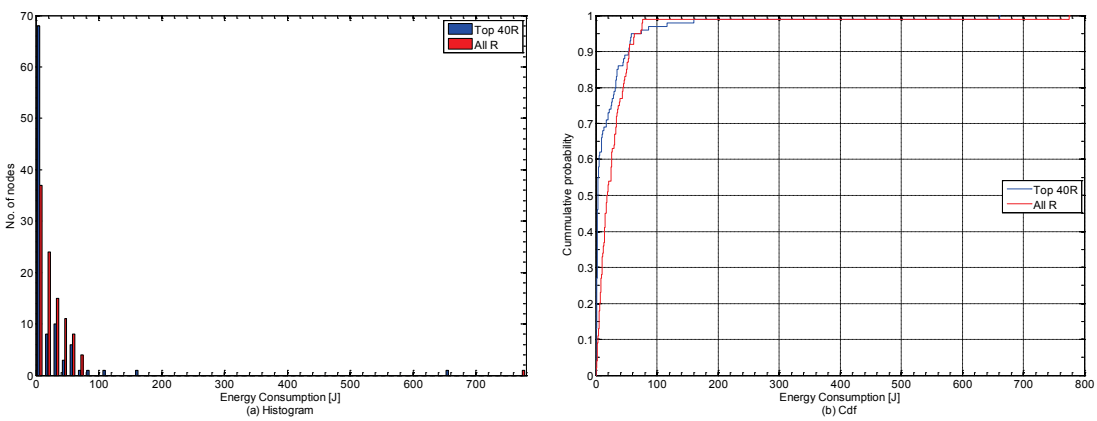


Figure 5.29: Energy consumption statistics for the network working in the proactive mode with one root station. The root node is included and the proactive routing information lifetime is set to the default value of 5.12 s.

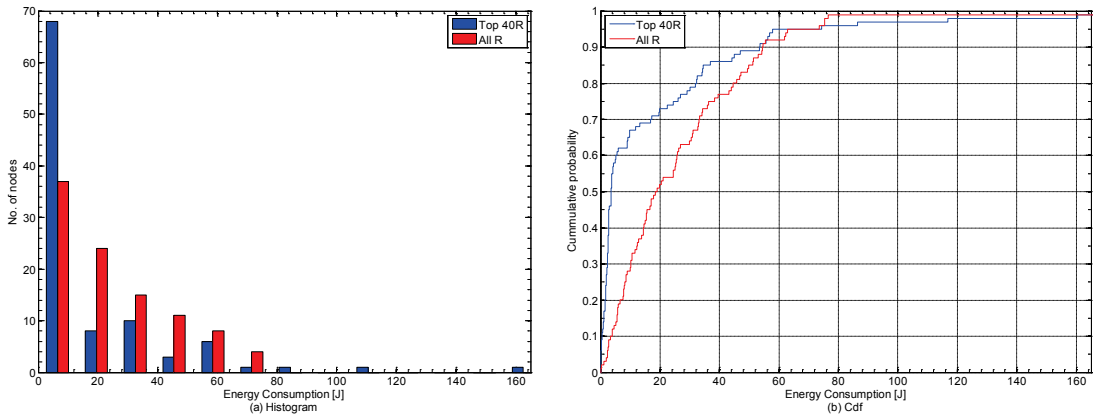


Figure 5.30: Energy consumption statistics for the network working in the proactive mode with one root station. Only common mesh nodes are included and the proactive routing information lifetime is set to the default value of 5.12 s.

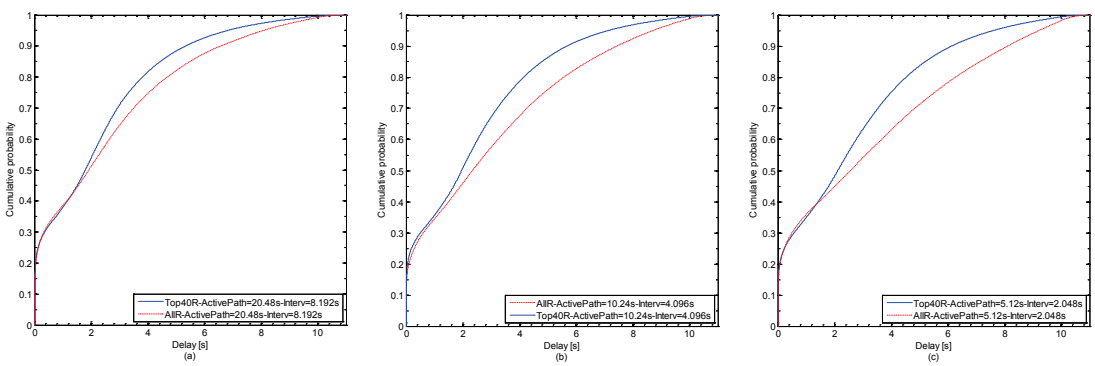


Figure 5.31: Cumulative distribution function of data packets end-to-end delay for different proactive path lifetimes.

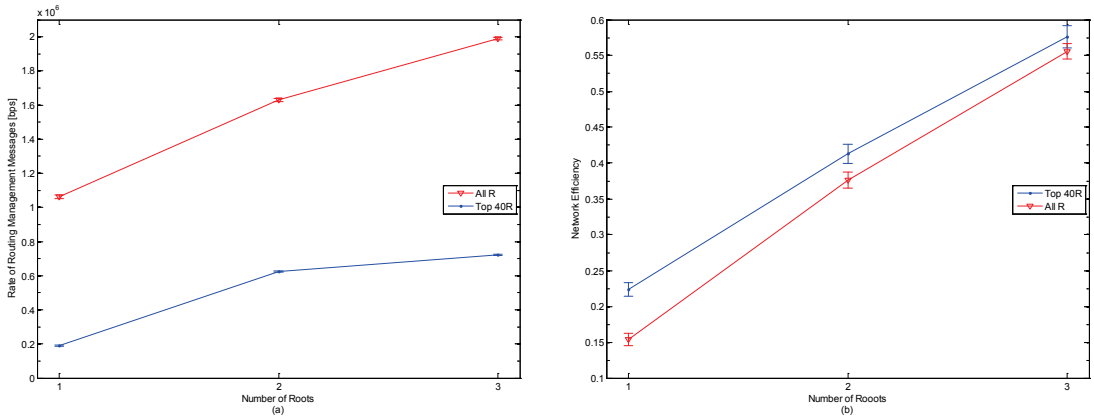


Figure 5.32: Proactive mode, variable number of root stations and constant traffic load; (a) rate of routing management messages; (b) network efficiency.

5.5.4 Proactive Mode With More Than One Root Station

A way to improve the network efficiency is to distribute the root or gateway functionality to a greater number of mesh stations. This section evaluates the effect of adding one or two roots to the previous configuration. For the reported simulations, the proactive path lifetime remains constant in its default values ($activePathToRootTimeout$ = 5.12 s and $rootInterval$ = 2.048 s). In the first set of experiments, the traffic load also remains constant (for each independent run there are 25 continuously active data flows: five uploads from randomly selected mesh stations to the roots and 20 downloads from the roots to randomly selected stations). Fig. 5.32(a) shows the rate of routing management messages as a function of the number of root stations. Obviously, since proactive mode is used, this rate increases when there are a greater number of roots independently of the traffic load. It is important to observe that the growth is not linear because of, while each root sends periodic PREQs, the non-root stations just response with a PREP to the root with the best path metric. The average path selection overhead reduction achieved with the application of the topology control in this case is around 69.14%.

The evident improvement of the network efficiency in terms of packet delivery ratio due to the distribution of the load in a greater number of roots can be verified in Fig. 5.32(b). On the other hand, the reduction of path selection overhead decreases the collisions probability and thus a greater number of data packets successfully arrives to the target stations when topology control is applied. The average value of the improvement due to topology control is around 19.71% for the reported experiments. The improvement is more significant when the level of congestion around the root(s) is greater (i.e. the one root case).

Finally, we evaluate the situation in which the number of roots and the traffic load increases at the same time. Namely, for the single root network data traffic consist of 25 flows, for the network with two roots there are 50 flows, and for three roots there are 75 active flows. For all the cases, downloads represents the 80% traffic and uploads the remaining 20%. The results for this set of simulations are shown in Fig. 5.33. As expected, since proactive mode is used, the rate of routing management messages (Fig. 5.33(a)) is pretty similar to the constant traffic results. There is a slight difference, a bit more noticeable for the network with tree roots and all stations as routers (around 145 Kbps), due to the higher level of channel contention produced

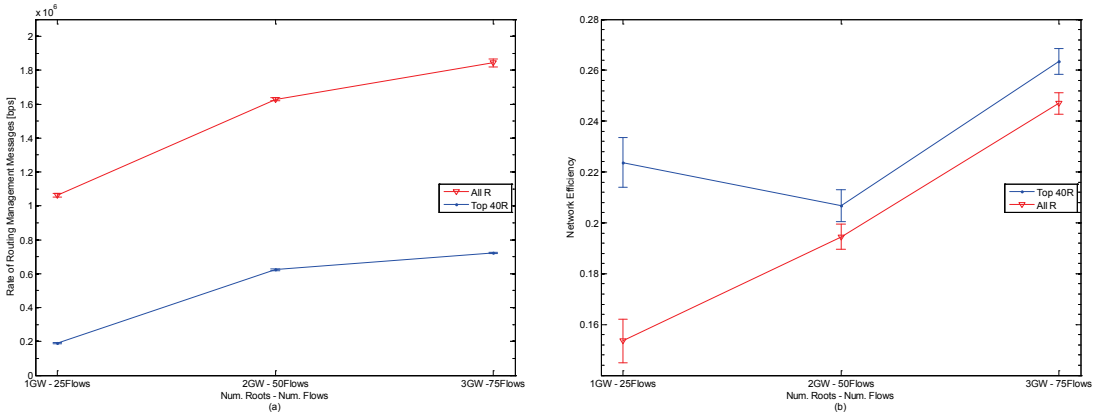


Figure 5.33: Proactive mode, variable number of root stations and traffic load; (a) rate of routing management messages; (b) network efficiency.

by the 75 data flows. This higher contention reduces the transmission opportunities of routing messages and therefore its rate is lower than in the 25 flows case. This does not happen when the topology control is applied since the path selection overhead has been reduced a 68.2% on average.

Although the traffic load is increased, the distribution of root functionality to a greater number of roots still improves the network efficiency (Fig. 5.33(b)) of the network configured with all stations as routers. The reduction of the number of mesh stations involved in routing implies the availability of fewer alternative paths, and causes a decrease by 2% of the network efficiency with topology control and two roots with respect to the one root case. This does not hold for the three roots case, which achieves around a 6% of improvement in comparison with the one root network because of a better distribution of the traffic load. Nevertheless, for all the cases, the efficiency of the network with topology control is better than the all routers network even though consuming less energy. The average improvement is about 19.54%.

The energy consumption statistics for the three roots and 75 active flows configuration are shown in Fig. 5.34 for all the nodes in the network and in Fig. 5.35 for only non-root stations. Again, the three root stations can be easily identified. Although the global energy dissipation is reduced around a 10.25% (from 5782.9 to 5190.3 J) with the topology control, we can recognize, as in the previous section, four overloaded non-root stations regarding the All R network. However, the nodes that dissipate less than 20 J increases from 30 to 64% in the Top 40R case.

5.5.5 Total Bit Rate Savings for the Distributed Implementation

This section includes a first approximation to the evaluation of the proposed distributed implementation. We analyze the total bit-rate savings achieved with the application of the topology control mechanism. The computations include the reduction of data forwardings, the reduction of path selection related messages and also the cost of the router selection protocol (Eq. 5.8). For all the reported cases, in order to compute the bit rate added by the topology control we make the following considerations: $N_T = 100$ nodes, $N_N = 8$ (i.e. N_N equal to the

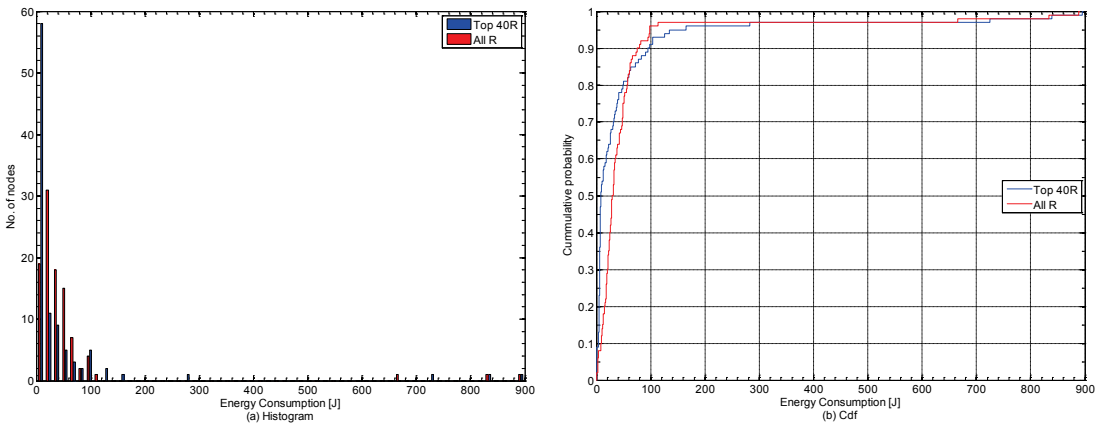


Figure 5.34: Energy consumption statistics for the network working in the proactive mode with three root stations and 75 active flows. The root nodes are included and the proactive routing information lifetime is set to the default value of 5.12 s.

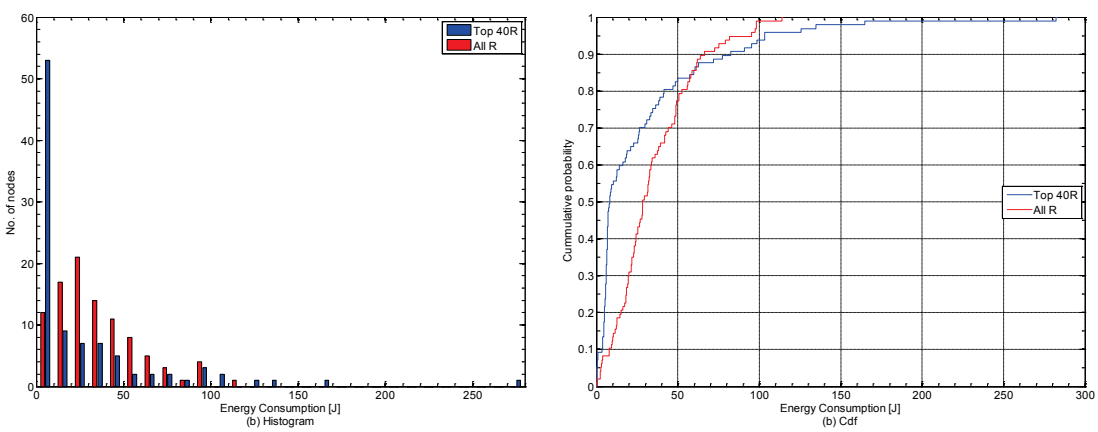


Figure 5.35: Energy consumption statistics for the network working in the proactive mode with three roots and 75 active flows. Only the non-root stations are included and the proactive routing information lifetime is set to the default value of 5.12 s.

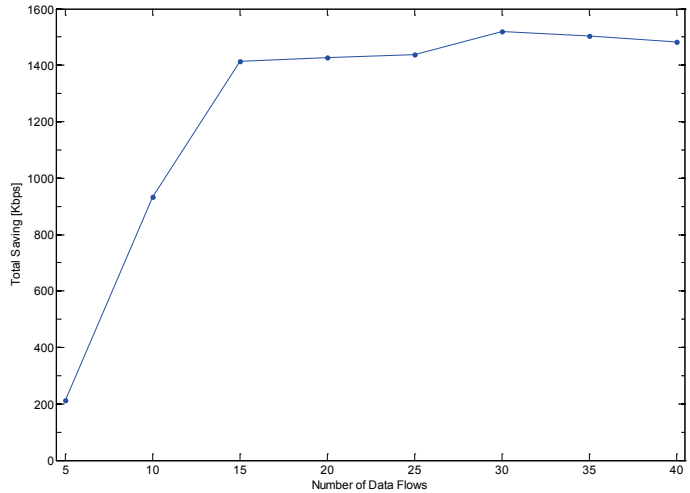


Figure 5.36: Global bit-rate savings as a function of traffic load.

average degree of all the nodes in the network), $N_R = 2$ which results on average in around 40% of nodes selected as routers, and the periodicity (TUT) of the required topology control related messages is the same that the routing information lifetime. Remember from Section 5.4.4 that TUT must be greater than or equal to *dot11MeshHWMPactivePathTimeout*.

For the first set of experiments, reported in Section 5.5.1, Fig. 5.36 shows the total bit-rate savings achieved with the topology control application as a function of the number of data flows. In this case TUT= 5.12 s and remains constant for all the different traffic loads. An average savings of 1.24 Mbps is reached with the topology control and the previously mentioned network saturation is also verified in this figure. It is important to remark that the transmission/reception energy savings of the mesh stations is directly proportional to this reported quantity.

On the other hand, Fig. 5.37 shows the evolution of this global traffic reduction as a function of the lifetime of the reactive routing information (experiments reported in Section 5.5.2). As previously justified, in these experiments the periodicity of topology control updates is again the same than the path lifetime. The average reduction for the reported cases is around 1.46 Mbps.

Finally, the global bit-rate reduction for the proactive one-root case (experiments reported in Section 5.5.3) is shown in Fig. 5.38. The average value of the savings is around 1.86 Mbps and as it was expected this is greater than in the reactive mode configuration.

5.6 Client WMNs with a Community Structure

In Section 5.4.2 we identified betweenness centrality as the more suitable metric for virtual backbone construction of randomly deployed WMNs. In this section we aim to explore the behavior of this metric, together with some variants, over more realistic networks that exhibit a community structure and work with socially-aware mobility models.

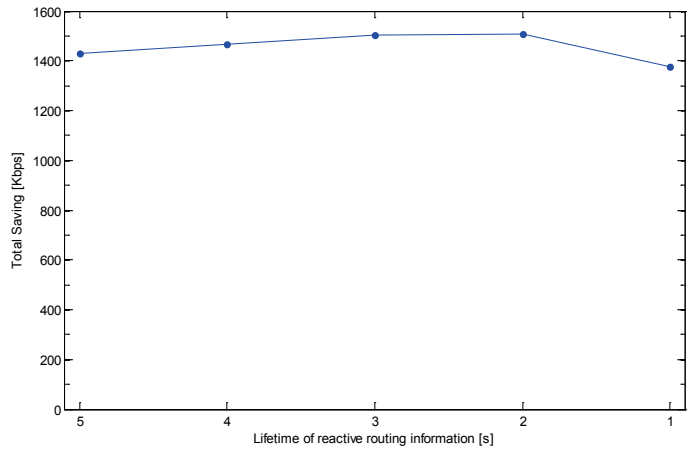


Figure 5.37: Global bit-rate savings as a function of reactive path lifetime.

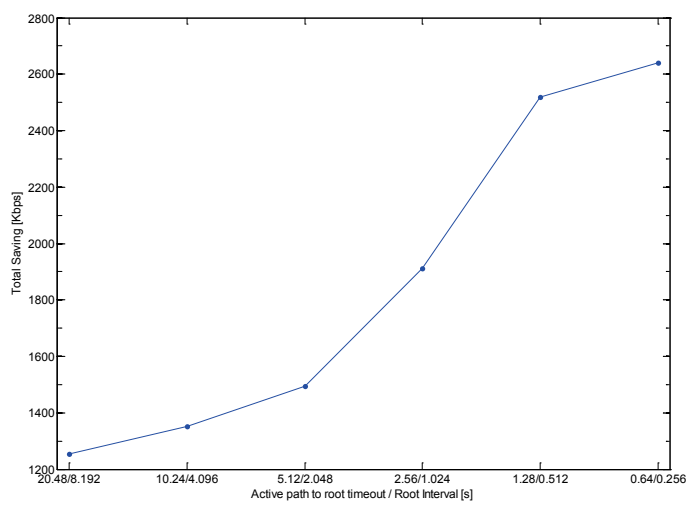


Figure 5.38: Global bit-rate savings as a function of proactive path lifetimes.

5.6.1 Betweenness Centrality Variants. Bridging Centrality

Recall that, from a topology control point of view, the betweenness metric produces better results than the other classical metrics like degree or closeness centrality in terms of network fragmentation, stability and required link density. On the other hand, some variants of the betweenness centrality have been proposed in order to better identify nodes that link the different communities that may appear in a network. This new metric, called *Bridging* centrality, has the objective of distinguish between the nodes that are locally important inside each community from those that have a global significance since they connect different components of the network. To this purpose, authors in [180] define a measure of bridgeness that is independent of the community structure and can be efficiently computed. The approach consist in not to consider, in the summation of the geodesic paths used in the computation of the classical betweenness, paths that start or end at the 1-hop neighborhood of the node. The main idea behind this is to overcome with the high correlation that may exhibits the betweenness centrality with the degree of the nodes. This way, the importance of a node for long paths is still considered but the effect of locally high connections is discarded. In the same line, authors in [159] introduce the concept of bridging coefficient to identify nodes that are located between highly connected neighbors. The bridging coefficient β_i of a node i is defined as:

$$\beta_i = \frac{\frac{1}{d(i)}}{\sum_{j \in N(i)} \frac{1}{d(j)}} \tag{5.11}$$

where $d(i)$ is the degree of the node i, and $N(i)$ is the set of neighbors of node i. Then, according to [159] the bridging centrality Br_i for node i is defined as the product of its traditional betweenness centrality Bc_i and its bridging coefficient β_i:

$$Br_i = Bc_i \times \beta_i \tag{5.12}$$

These both bridging metrics have been considered in this section because of the clear community structure present in the networks under study.

5.6.2 Socially-aware Mobility Models

As previously said, for the network subject of this work, the node movements are completely dependent on the human mobility patterns since the hand held devices are carried by people. Research studies in this field show that in order to understand human mobility, there are three important properties that must be taken into account: the spatial, the temporal, and the social ones [181]. The spatial property is related with the travel distance. The temporal property considers the frequency of visit to most common locations, the time that a user spends at one specific location, and the probability of returning to frequently visited places. One of the works with the largest data sets to study these statistical properties can be found in [182]. Finally, social is the most relevant property for this work because it defines the connectivity features between different users. It is characterized by the *inter-contact time* (frequency) and by the *contact time* (duration).

Based on these statistical properties of human mobility patterns, different models have been proposed in the literature. According to [181] these models can be classified in one of the following categories. Mobility models based on *location preference* exploits the idea that

each node has a set of highly visited places that conditions the chosen destinations of a node. On the other hand, *Agenda-based* models prioritize temporal patterns reproduction through the knowledge of people daily activities and schedules. Finally, the new trend in modeling human mobility is to exploit the knowledge of human relationships and interactions. Through complex network theory, the social behavior is represented by time-varying graphs which in turn are the inputs of the *social graph-based* mobility models. The main idea is to capture the fact that people movements are strongly dependent on their social ties and interactions, which vary during a day. Since this last category seems to be the more promising one, because it includes the most of the statistical properties of human mobility, we choose for this part one available approach of this type. Specifically, the social-based group mobility model presented in [162] has been used. In the following we summarize the main features and the configuration parameters of this model.

The input of the model is the social network that captures people interactions and it is mathematically represented by the Interaction Matrix M. Each element of this matrix is the weight that measures the strength of social tie between two specific individuals and it will have an impact on the probability of geographic collocation of them. These weights could vary with time to mimic the fact that a person interacts with different groups (family, work colleagues, friends) depending on the period (working hours, evening) and on the type of day (working days, weekends). According to the practical implementation of the movement generator, this reconfiguration period can be set through the *reconfigurationInterval* input parameter. This matrix is binarized through the interaction threshold (*connectionThreshold* configuration parameter) in order to obtain a Connectivity Matrix C. This matrix represents the fact that interactions occurs only between people with strong relationships. By default, this mobility model generates a synthetic network that fulfill the main properties of real social networks in terms of high clustering coefficient and low average path length. The procedure employed for this uses the combination of order and randomness introduced by Watts in [183]. The social network starts from a caveman graph, a set of isolates cliques which initial number is configurable through the parameter *numberOfGroups*. With certain rewiring probability, configurable through the *rewiringProbability* parameter, each initial edge is rewired to a neighboring clique resulting in a connected-caveman graph [183]. This way, as wanted, people inside their community are densely connected meanwhile connections between communities are sparse.

Once the required social network is available in the form of the connectivity matrix, the next step is to detect the communities inside this network. For this, the well-known edge betweenness-based method of Newman and Girvan [184, 185] is used. This algorithm consist on the following steps: compute the betweenness centrality for all the edges, remove the edge with the highest value, recompute the betweenness of the surviving edges, and repeat the procedure from the second step. The finishing criteria of the algorithm is based on the maximization of the modularity, a measure also defined in [185]. This metric aims to evaluate the quality of certain network division. The main idea is to quantify how different is the structure of the resulting network after the division in comparison with an equivalent random case. A greater modularity value, implies a stronger community structure of the network. According to [185] the typical values for real networks are between 0.3 to 0.7.

By its part, the simulation space, defined through the parameters *sideLengthXcoordinates* and *sideLengthYcoordinates*, is divided in *numberOfRows* rows and *numberOfColumns* columns. Then, each community found in the previous step is randomly located to one of this sub-areas. Starting from these initial positions, nodes move toward a goal inside their sub-area. The

Table 5.2: Mobility model input parameters for the sports monitoring network.

Input parameter	Value
numberOfNodes	100
totalSimulationTime	500 s
numberOfGroups	16
connectionThreshold	0.1
rewiringProbability	0.1
reconfigurationInterval	250 s
sideLengthXcoordinates	1040 m
sideLengthYcoordinates	520 m
numberOfRows	4
numberOfColumns	2
lowerBoundSpeedHost	3 m/s
upperBoundSpeedHost	9 m/s

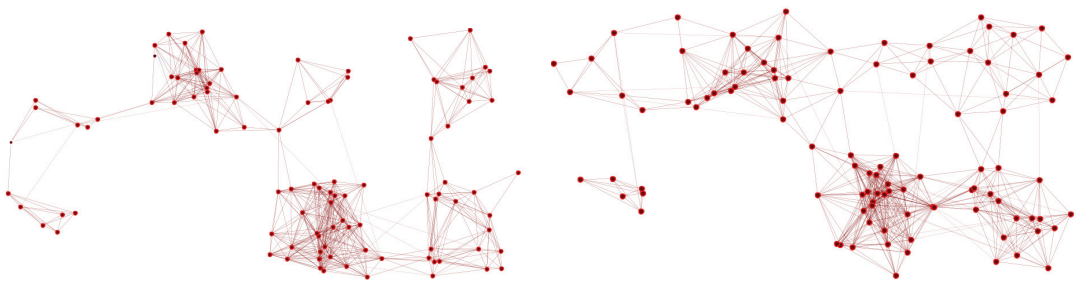
subsequent destination goals of a node are chosen in the sub-area that exerts the highest social attractivity to it. This attractivity quantifies the strength of the social relationships between the node under analysis with the nodes that are already in this area or are moving toward it [162]. The implementation of the model permits to choose between a deterministic selection of the destination goal to the sub-area with the highest attractivity or a probabilistic selection proportional to the attractivity. This last option could be useful if a greater level of randomness is required. The speed of the nodes is randomly selected from a uniform distribution in the configurable range [*lowerBoundSpeedHost*, *upperBoundSpeedHost*].

5.6.3 Scenario under Consideration

The increasing availability of smart gadgets like watches, headphones or headsets [186], shoes [187], and multi-sport sensors [188], allows us to have in mind a sports monitoring system. We consider a hundred of athletes initially grouped in 16 teams and distributed in a 1040×520 m rectangular field, which is divided in 8 courts. Many possible applications arise from a system like this mainly intended for improving the performance and training experience and avoiding injuries of sportspeople. Some examples are, the prevention of over-training and its effects studied in [189], the real-time golf swing motion capturing system described in [190], a cycling monitoring system [191], among others. For the system under evaluation, we assume that all the sensors of each athlete are managed by a IEEE 802.11s-capable hand-held device. The scenario is modeled as a dynamic WMN generated by the ns-3 network simulator. The mesh stations move according to the social based mobility model configured with the parameters summarized in Table 5.2. The C++ implementation of this model generates mobility traces according to the widely used ns-2 style [123]. This traces can be read by ns-3 through the available ns-2 Mobility Helper.

5.6.4 Simulation Results

As in Section 5.4.2, once the previously described dynamic mesh network is running with all these parameters, we use the list of established peer links of each node to periodically take a snapshot of the network. In this way we obtain the set of graphs which represent the network



(a) Sports network snapshot at 100 s (b) Sports network snapshot at 240 s

Figure 5.39: Sport network snapshots at two different time instants.

under evaluation for the corresponding time instants. Fig. 5.39 shows an example of such graphs for two different time instants. From this figure we can verify the clear community structure presented in this networks as consequence of the athletes grouped in different teams.

For all the nodes in each available graph, we compute the three centrality metrics: classical Betweenness, bridging according [159] (BRI-1) and bridging according [180] (BRI-2). With these values and in a distribute manner, each node run the topology control algorithm described in Section 5.4.4 in order to construct the required virtual backbone. The first simulation round aims to evaluate the minimum size of the resulting backbone that keeps the network with the same fragmentation level than the original graph. Starting from one, we use if necessary an increasing number of selected routers per 1-hop neighborhood to fulfill this objective. The resulting details for a subset of 10 snapshots from the sports monitoring network is shown in Table 5.3. As can be seen, different alternatives have been evaluated, including a backbone built

Table 5.3: Resulting backbone size for different number of selected routers per neighborhood and different centrality metrics.

Snap	No. of Selected Routers						
	1-BET	2-BET	Min-BET	1-BRI-1	2-BRI-1	Min-BRI-1	1-BET+1-BRI-2
1	22	29	22	21	33	21	24
2	NA[1]	33	33	31	43	31	28
3	NA	32	32	24	37	24	27
4	NA	30	30	NA	34	34	28
5	20	30	20	28	39	28	27
6	NA	35	35	33	44	33	27
7	NA	26	26	29	38	29	22
8	NA	28	28	23	34	23	22
9	18	25	18	NA	40	40	24
10	NA	27	27	33	45	33	25

[1]NA stands for "Not Accomplish" with the fragmentation level requirement. i.e., the same as the original graph

using the nodes with the highest classical betweenness value in each ego-network (x-BET, with x equals to the number of neighbors selected as routers) and also a backbone built according to the bridging centrality defined in [159] (x-BRI-1). It is not possible to built a backbone based on the bridging definition of [180] (BRI-2) without an additional consideration because of the way that it is computed: we observe that in most of the network graphs there is at least one ego-network with all its nodes with a bridging value of 0. Therefore, an additional consideration must be done for these cases. This is the reason why these results are not included in Table 5.3.

Table 5.4: Resulting statistics for the minimum backbone size built based on different centrality metrics.

	CDS size			
Used Centrality	**Min**	**Average**	**Max**	**SD**
Classical Betweenness	15	26	38	6.236
BRI-1	18	29	44	6.635
1 BET + 1 BRI-2	18	25	32	2.911

On the other hand, following the natural intuition that for each 1-hop neighborhood the local centers and the global bridges are both important for the construction of the virtual backbone, we include the evaluation of a backbone built selecting in each ego-network the node with the highest betweenness and also the node with the highest bridging centrality (1-BET + 1-BRI-2).

For all the cases, when the fragmentation level of the resulting network is greater than the value already presented in the original one, each neighborhood adds a new high centrality node to the backbone. Fig. 5.40 shows the resulting backbone for each procedure of the third snapshot in Table 5.3. As expected, different number of routers have been required by each approach. In this specific case, two routers per neighborhood (32 in total) are required using betweenness centrality while one router per neighborhood (24 in total) is enough using the BRI-1 metric. The combination of betweenness and bridging requires 27 to construct the backbone.

Finally, for the complete set of 50 snapshots, Table 5.4 summarizes the global statistics of the minimum size of the resulting backbone. From these results we can observe that in terms of backbone size, the classical betweenness is still the better metric to built the virtual backbone in comparison with the bridging alternative. Nevertheless, the combination of betweenness and bridging, although on average produce similar results, considerably reduces the variability of the minimum backbone size. Therefore, a more stable network operation is achieved with this approach.

5.7 Conclusions

In this chapter we have evaluated the feasibility of using the centrality metrics from social network analysis to create a topology control mechanism based on a connected dominating set. Specifically, for a dynamic wireless mesh network formed by user hand-held devices, we have proposed and evaluated some mechanisms to select which of those devices must act as routers, forwarding the packets received from other hand-held to their destination. The mechanism implementation has been proposed in two ways, centralized and distributed. For the centralized approach, we have evaluated the three most common centrality measures: degree, closeness and betweenness centrality. In all those cases, the more central nodes have been selected to form the backbone. The experimental analysis has shown that the network fragmentation using the betweenness centrality is considerably lesser than the resultant with degree or closeness centralities. Thus, it is possible to conclude that by far the betweenness centrality is the metric that better identifies the stations that should be considered for routing tasks. In the second place, taking into consideration that a centralized implementation could be unfeasible in most scenarios, a distributed implementation has been proposed. In this case only the

(a) Betweenness

(b) Betweenness based backbone

(c) Bridging-1

(d) Bridging-1 based backbone

(e) Betweenness plus Bridging-2

(f) BET plus BRI-2 based backbone.

Figure 5.40: Results of the router selection process for one snapshot of the sport monitoring scenario. Blue bigger nodes represent the selected routers while the green smaller ones are the associated clients.

betweenness centrality has been considered. To distribute the responsibility, the concept of the egocentric network perspective of each station is needed. That is, every station computes its own centrality knowing only its own established links (neighbors). Besides, due to the dynamic and time evolving nature of the network with which we are dealing, we also discuss and evaluate different distributed mechanisms to select the central stations that will act as routers. First of all, every station selects the node in its neighborhood with the highest centrality. This mechanism works well in most cases if a minimum density of nodes exists. Then again, the complete connectivity is not assured. To guarantee that connectivity, some approaches have been proposed and compared in terms of overload and effectiveness. As a future line of work, the minimum number of routers per neighborhood needed to guaranty a desired connectivity in the distributed implementation will be studied.

The effectiveness of this centrality-aware topology control protocol has been verified through extensive simulations. The evaluation has been carried out for both reactive and proactive path selection modes. For all the studied cases the efficiency of the network was improved or at least remain equal but always with a considerably lower amount of transmission/reception energy consumption. Another common factor in all the simulation results is the reduction of the packet end -to-end delays. This is a direct consequence of the global bit-rate savings which implies lower channel contention. The reduction of the number of mesh stations involved in the path selection procedure also improves the scalability when the variability of the network demands faster path updates. Finally, we confirm that greater energy savings are obtained in the proactive path selection case because of the greater overhead present with this operation mode.

The second part of this chapter takes into account the importance of considering realistic mobility models when the networks under study are created by devices carried by humans. Three main properties of human mobility have been identified by researchers in the field: spatial, temporal and social. On this basis, different mobility models have been proposed to mimic real movement traces. The integration of social graph based models with temporal considerations appears as the most promising technique to generate realistic traces of human movement. The main observation is that human location and movements are mainly lead by their social relationships and interactions. In turn, these interactions depends on the specific schedules of the people involved and can be practically represented by time-varying social graphs.

The use of this type of mobility models results in networks with a clear community structure that considerably differs from commonly used random models. Networks with community structure are characterized by groups of nodes that are densely connected inside each group but with lower number of connections between different groups.

Under this scenario we consider useful, in addition to the classical betweenness centrality metric, to focus our attention on some variants of it proposed to identify bridging nodes in networks. Specifically, we study the feasibility of using the betweenness along with these bridging metrics to construct a virtual backbone for an efficient performance of networks with such community structure.

Although the first simulation results show that a combination of betweenness and bridging improves the stability of the resulting backbone size further work is required to be more confident with this observation.

Chapter 6

Conclusions and Future Work

This chapter summarizes the main conclusions and the possible future lines of work of our research.

6.1 Conclusions

Wireless Mesh Networks exhibit a set of valuable advantages that makes them a suitable and low-cost communication technology for a variety of applications and services. Such potentialities have attracted the attention of academic and industrial researchers during these last years. We have witnessed of standardization efforts, and the development of open and proprietary mesh solutions. While urban mesh deployments could be an important constitutive part for evolving towards the smart cities paradigm, the mesh deployments in under served and rural areas is socially relevant to reduce the digital divide and improve the population quality of life.

On the other hand, the success and increasing demand of real time and multimedia services in the present days is undeniable. Under this scenario, in addition to contribute for improving the performance of one application service of this type, we consider that other aspect of paramount importance of WMNs is the efficient utilization of the network resources, and therefore, this thesis also makes a contribution to encourage a better use of two specific ones: memory and energy. In the following we summarize the main results and conclusions for each corresponding part of our research.

6.1.1 Load Splitting in Clusters of Video Servers

The high popularity and bandwidth requirements of VoD services makes that any proposal to increase the system capacity is very appreciated. For this purpose, the most common approach, is to distribute the video contents between different distribution points. In turn, each distribution point usually consist of a cluster of video servers which finally satisfies the user requests. In a general case, such clusters are not homogeneous and a variety of physical or virtual devices with different capacities could coexist. For such environment we propose a server selection mechanism that optimally chooses an available video server in a way that the system performance is improved. Specifically, the proposed approach minimizes the video

packet transfer time at the server by balancing the load among the different servers. This is done, taking into account the fact that the server channels could have different capacities.

To carry out a realistic analysis of the transmission and transfer times (targets of our minimization strategy) we consider real video traces which have been properly multiplexed. It is important to remark that such multiplexed video sequences were generated avoiding short and long term sequence synchronization. For the average and standard deviation of the packet length, we observe a clear tendency to a fixed value when the amount of multiplexed movies increases. This was useful to estimate the average channel capacity required by each video sequence in the case that it is multiplexed with others.

From the perspective of the analytical model tractability and from a practical point of view, simulation results confirms that, although the exponential model for the packet length pdf is less precise than other functions, it is valid enough when the number of multiplexed movies grows. In the same line, simulations also confirm that the self-similar behavior of the packet length is not so meaningful when a large number of movies are multiplexed. The validity of this exponential modeling is furthermore reinforced with the analysis and the simulation results of the transmission and transfer times percentiles. We observe a good agreement between the resulting simulations with real traces and the exponential model for both the transmission and the transfer times.

Once the input parameters were deeply studied, the key idea for the load balancing solution was to visualize the conceptual similarity between our server selection system and the problem of optimally routing packets between two network nodes linked by two (or more) channels. We present the solutions for the D/M/1 and M/M/1 models. The simpler M/M/1 model has the advantage of producing closed form expression which reduces the algorithm overload. Nevertheless, the D/M/1 model generates more accurate results useful when a higher degree of optimization is required. Simulation results confirm that the D/M/1 approach better approximates the splitting threshold and the flow proportions, although the difference with the simpler M/M/1 model is smaller for very low and very high loads. Experimental results also confirm that there is no need to increase the traffic models complexity since the proposed solution follows well the optimum values.

Lastly, through the practical implementation analysis, we can conclude that the proposed load splitting system could easily scale thanks to its simplicity and low overload. However, a more complex server selector will be necessary if a dynamic movie reassignment is required.

6.1.2 Dynamic Buffer Sizing

Mesh routers that build the WMN backbone are generally equipped with multiple interfaces. In the simplest cases they have at least a traditional WLAN interface to provide AP functionality to classic clients and an additional radio for mesh connectivity with its peers. Further mesh radios could be necessary when the demanded capacity is greater. Especial mesh routers with gate and portal functions also includes wired interfaces for the connection with external networks. An important resource for such devices is the amount of memory intended for buffering. An efficient use of such resource not only benefits the device scalability but also prevents the well known problems associated with excessive buffers. At the end, most of this issues directly degrades the service provided to end-users.

Under this premise we propose a dynamic buffer sizing mechanism mainly designed to be used with real time flows. The approach considers the packet loss probability as the key factor to be controlled. In fact, the input of the system is the maximum value for such loss probability

that could be tolerated by each flow. As expected, for traffic differentiation matters, this value can be independently configured for each flow. Following the self-configuration capabilities criteria of WMNs, another important requirement satisfied by our approach is that devices must be able to self-configure their buffers without the intervention of any other central equipment.

The main problem we had to deal with was that, the study of queuing theory requires a detailed knowledge of the distributions of service and interarrival times, but in most practical situations such complete information is rarely available. Specifically, in the present case, we are able to dispose of just two average values, the average number of packets in the transmission system and the server occupancy, both measured at packet arrivals. With this limited information our approach uses the maximum entropy principle to approximates a solution of the G/G/1 and G/G/1/K queues. We then apply this approximation to the dynamic buffer sizing problem and verify its utility with extensive experimental analysis. To be more confident with the results, we use different simulators, networks scenarios, and traffic load conditions. Complicated traffic mixes with quite different interarrival times and packet lengths have been considered to avoid unreal traffic patterns.

The first set of simulations, presented in Section 4.4, verifies the accuracy of the analytical model. For all the cases the agreement between the model and the simulation results is very good. A slight difference is perceptible when we use the closed-form expressions for Q resulting from the simplest Molina method. In this case the model allocates a buffer larger than the necessary but the required computations are less demanding.

The next set of experiments (Section 4.5) demonstrates the utility of the method for wired interfaces working over dedicated channels. We point-out the importance of filtering the input parameters (average server occupancy and average number of packets in the transmission system) and their impact over the buffer stability. We prove that it is possible to significantly reduce the system overload by means of two acting thresholds, without compromising the achievement of the loss probability goal. This facilitates the mechanism implementation in real devices. We also observe that absolute thresholds performs better than relative ones and the rate of effective buffer size updates grows when the loss probability target is more demanding. Another important verification is the proper operation of the mechanism when traffic differentiation is considered. Two cases has been tested, one on which all the flows requires the same loss probability and also the more general case on which each flow requires a different service levels. The results were satisfactory in both cases.

On the other hand, wireless interfaces impose additional challenges mainly due to the shared channel. The server occupancy is no longer exclusively dependent on the node own transmissions. The adaptation of our mechanism to such wireless interfaces have been verified that properly performs under a variety of simulators, network scenarios, and channel conditions. Two types of traffic have been used, synthetic flows and flows based on real traces. For all the cases, the simulations have been made long enough to obtain a sufficient number of packets that allows a reliable measurement of the loss probability. The first set of wireless simulations, focused on traditional WLAN (single hop infrastructure-based) interfaces, verifies the proper operation the our dynamic buffer sizing algorithm under such environments. We also prove that aggregated channel errors induced by the radio link impairments are properly captured by the system and their effects are taken into account. Besides, a metric to quantify the memory utilization efficiency has been introduced. Its evaluation permits us to demonstrate the benefits of a dynamic buffer allocation in comparison with the fixed buffer case. It is worth to mention that the comparison were done in a fair way. This is, instead of considering buffer sizes with typical values we compare with the buffers that results in approximately the same packet loss

probability. The same criteria was used to verify the benefits of memory share and dynamic self-configuration of nodes with multiple interfaces. Finally, the buffer sizing mechanism was tested on wireless mesh interfaces working with real video traces. This simulation results ratify the proper performance of the system even under high load levels. The improved memory efficiency could facilitate the scalability of the mesh gateways with additional mesh interfaces in order to increase the overall capacity of the mesh backbone.

6.1.3 Topology Control for Client WMNs

The evolution of smart hand-held devices with increasing computation, sensing and communications capabilities along with their growing level of penetration, encourage us to consider a client WMN built by such devices for an unlimited number of applications. Despite the great technological development of this kind of hand-helds, battery lifetime remains as one of the major concerns. Communication systems are the components that demand the most significant amount of such limited energy resources.

Under this scenario, to promote a better use of energy resource of client mesh networks, we propose a topology control mechanism based on the construction of a virtual backbone. Specifically, we evaluate the use of centrality metrics from social network analysis, to select just a subset of mesh stations to carry out the routing functions. Simulation results allows to verify that we are able to reduce the routing overhead without compromising the overall network performance.

We analyze and evaluate the three most common centrality metrics and additionally, for a second step, we also include some variations of the betweenness centrality which are strong related with the resulting community structure of the networks under study.

A centralized implementation of the proposal, useful in cases in which the mesh nodes are more static, determines that, from the resulting network fragmentation point of view, betweenness centrality performs by far better than the degree or closeness centralities.

For the case of completely dynamic networks we evaluate three different alternatives for a distributed implementation: increasing the number of hops when computing the egocentric centrality, a probe packet mechanism, and increasing the number of routers per neighborhood. We consider that the selection of higher number of routers per neighborhood is the more practical alternative since it presents a lower level of overhead. Although this approach could also increase the redundancy of the backbone, this can be exploited for the network resilience and for a better load balancing among the selected mesh nodes.

Extensive simulations demonstrate the feasibility and benefits of the centrality-based topology control approach. Reactive and proactive operation modes of the HWMP protocol have been taken into account. When the level of load is meaningful, the network with topology control behaves better than the original one for a diversity of performance evaluation parameters (routing overhead, total forwardings per received packet, packet delivery ratio, delay, and energy consumption) and for all the set up conditions and operation modes. Important energy savings were achieved in all the simulated scenarios, being more significant when the network works with the proactive mode. This confirms that proactive mode produces a higher level of path selection overhead.

In a second step, we analyze the behavior of the topology control approach with another mobility model. Such model uses human sociality information to generate the user mobility traces, which in turn produce networks with a clear community structure. Although the topological features of these community networks substantially differs from networks resulting

from random movements, we observe that betweenness assisted topology control still performs properly in this kind of mesh networks. We include the analysis of the bridging centrality, computed as different variants of the betweenness metric that are strongly related with the community structure of the networks. As expected, bridging nodes are those that connect different communities presented in a network and therefore they are fundamental for the virtual backbone construction. The first simulations combining betweenness and bridging metrics for the topology control approach show promising results in terms of stability of the resulting backbone size and encourage us to continue working in this subject with a deeper level of detail.

Finally, as a general conclusion, and based on all the outcomes presented in this and precedent sections, we can state that the three main objectives of this thesis have been successfully achieved.

6.2 Future Work

This section summarizes some potential research directions and improvements for each of the proposals presented in this thesis.

6.2.1 Load Splitting in Clusters of Video Servers

For the server selection mechanism proposed in Chapter 3, one possible future research line could be devoted to quantify and evaluate the overload required to dynamically reassign the transmission of movies among the various servers. This dynamic reassignment exploits better the advantages of the proposal, but we must verify if the resulting benefits are more significant that the increased overload.

As another possibility, we can consider a similar mechanism to the one introduced here to select the distribution point (DP in Fig. 3.1) to service customer requests.

Additionally, the approach could be explored as a possible solution of the route selection problem in multi-source, multi-path VoD systems over WMNs.

6.2.2 Dynamic Buffer Sizing

Regarding the dynamic buffer sizing proposal, further work will consider the implementation and evaluation of the mechanism in real devices. Two possible alternatives would be evaluated. The first one consist of directly modify the source code of the network card drivers, in a way that the attribute *tx_queue_len* (maximum number of frames that can be admited in the internal transmission queue of the device), which is by default configured with a fixed value, could be dynamically adjusted using the proposed algorithm. One starting point to this end would be the adaptation of the Dynamic Queue Limit (DQL) library developed by researchers of Google [192, 193] and included in the Linux kernel from version 3.3. This library implements a mechanism to dynamically limit the size of the transmission queue for network interface cards. The second alternative that will be evaluated is the implementation of the proposal in routers working with a very flexible operating system which allows a full configuration and customization. For this purpose, we will use the GNU/Linux distribution for embedded devices OpenWrt [194].

On the other hand, we also plan to combine the buffer sizing proposal with the use of different schedulers. Even though the theoretical results presented in Section 4.3 have been

obtained for a First Come First Served (FCFS) service policy, it is interesting to study the behavior with any other type of scheduler that prioritizes some flows with respect to the others. The aim is not only look for diverse loss probabilities for different flows, but also different times spent in the system. A first set of simulations shows that it is possible to separately work with both parameters, which is very convenient to facilitate the task of the device self-configuration.

Another possible improvement of the mechanism consist of the automatic configuration of the acting thresholds on the basis of the traffic load intensity and the rate of effective buffer size updates.

We could also deal with the problem of multi-objective optimization, considering as another goal, in addition to the packet loss probability, to minimize the time that packets wait in the transmission queues.

Finally, another possible line of work is the performance evaluation of the mechanism for TCP and a mix of TCP and UDP traffic flows as well as the use in different network technologies as diverse as WSN or optical routers.

6.2.3 Topology Control for Client WMNs

As current and future lines of work we are carrying out performance evaluations considering realistic human mobility models. Besides, we are designing and evaluating a multi-metric protocol which assigns different weights to the egocentric betweenness centrality and the node remaining energy values.

Although the first simulation results with community structured networks show that a combination of betweenness and bridging improves the stability of the resulting backbone size, further work is required to be more confident with this observation. Specifically, the following tasks are planned in order to get more insightful and complete results:

- The evaluation of the combination of the classical betweenness with the bridging computed according to [159].
- The evaluation of different scenarios with a greater number of nodes and different input parameters of the mobility model.
- The validation with other recently published mobility models that we have already identified like SWIM [163] or N-body [165].
- The assessment with available real human mobility traces.
- The evaluation of network related parameters like packet delivery ratio, delay, routing overhead, among others.
- The comparison of the proposed topology control algorithm with alternative backbone construction algorithms.

Besides, we plan to study and evaluate two of the optional functionalities of the IEEE 802.11-2012 standard and their integration with our proposals. These are the MCCA mesh coordination that aims to optimize the channel access efficiency and the power management which allows that each mesh station manage the activity level of its peer links.

Finally, we could work with the implementation and evaluation of a global framework including the three main contributions of this dissertation.

Bibliography

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, no. 4, pp. 445 – 487, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128604003457

[2] J. Ishmael, S. Bury, D. Pezaros, and N. Race, "Deploying Rural Community Wireless Mesh Networks," *Internet Computing, IEEE*, vol. 12, no. 4, pp. 22–29, July 2008.

[3] L. Maccari and R. L. Cigno, "A week in the life of three large Wireless Community Networks ," *Ad Hoc Networks*, no. 0, pp. –, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870514001474

[4] P. Frangoudis, G. Polyzos, and V. Kemerlis, "Wireless community networks: an alternative approach for nomadic broadband network access," *Communications Magazine, IEEE*, vol. 49, no. 5, pp. 206–213, May 2011.

[5] C. Szabó, Z. Horváth, and K. Farkas, "Wireless community networks: Motivations, design and business models," in *Proceedings of the 3rd International Conference on Wireless Internet*, ser. WICON '07. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, pp. 23:1–23:10. [Online]. Available: http://dl.acm.org/citation.cfm?id=1460047.1460076

[6] S. Chakraborty and S. Nandi, "IEEE 802.11s Mesh Backbone for Vehicular Communication: Fairness and Throughput," *Vehicular Technology, IEEE Transactions on*, vol. 62, no. 5, pp. 2193–2203, Jun 2013.

[7] M. Gerla and L. Kleinrock, "Vehicular networks and the future of the mobile internet," *Computer Networks*, vol. 55, no. 2, pp. 457 – 469, 2011, wireless for the Future Internet. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128610003324

[8] J.-H. Huang, L.-C. Wang, and C.-J. Chang, "QoS Provisioning in a Scalable Wireless Mesh Network for Intelligent Transportation Systems," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 5, pp. 3121–3135, Sept 2008.

[9] Z. Fan, P. Kulkarni, S. Gormus, C. Efthymiou, G. Kalogridis, M. Sooriyabandara, Z. Zhu, S. Lambotharan, and W. H. Chin, "Smart grid communications: Overview of research challenges, solutions, and standardization activities," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 1, pp. 21–38, First 2013.

[10] J. Kim, D. Kim, K.-W. Lim, Y.-B. Ko, and S.-Y. Lee, "Improving the reliability of IEEE 802.11s based wireless mesh networks for smart grid systems," *Communications and Networks, Journal of*, vol. 14, no. 6, pp. 629–639, Dec 2012.

[11] M. Abid, A. Khallaayoun, H. Harroud, R. Lghoul, M. Boulmalf, and D. Benhaddou, "A Wireless Mesh Architecture for the Advanced Metering Infrastructure in Residential Smart Grids," in *Green Technologies Conference, 2013 IEEE*, April 2013, pp. 338–344.

[12] M. Portmann and A. Pirzada, "Wireless Mesh Networks for Public Safety and Crisis Management Applications," *Internet Computing, IEEE*, vol. 12, no. 1, pp. 18–25, Jan 2008.

[13] A. Pirzada, M. Portmann, R. Wishart, and J. Indulska, "SafeMesh: A wireless mesh network routing protocol for incident area communications," *Pervasive and Mobile Computing*, vol. 5, no. 2, pp. 201 – 221, 2009, homeland and Global Security. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119208001065

[14] R. B. Dilmaghani and R. R. Rao, "Hybrid Wireless Mesh Network Deployment: A Communication Testbed for Disaster Scenarios," in *Proceedings of the 1st International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, ser. WiNTECH '06. New York, NY, USA: ACM, 2006, pp. 90–90. [Online]. Available: http://doi.acm.org/10.1145/1160987.1161005

[15] Y. Ding, Y. Yang, and L. Xiao, "Multisource Video On-Demand Streaming in Wireless Mesh Networks," *Networking, IEEE/ACM Transactions on*, vol. 20, no. 6, pp. 1800–1813, Dec 2012.

[16] Y. Ding and L. Xiao, "Video On-Demand Streaming in Cognitive Wireless Mesh Networks," *Mobile Computing, IEEE Transactions on*, vol. 12, no. 3, pp. 412–423, March 2013.

[17] X. Du, N.-S. Vo, W. Cheng, T. Duong, and L. Shu, "Joint Replication Density and Rate Allocation Optimization for VoD Systems Over Wireless Mesh Networks," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 23, no. 7, pp. 1260–1273, July 2013.

[18] D. De Couto, D. Aguayo, B. A. Chambers, and R. Morris, "Performance of multihop wireless networks: Shortest path is not enough," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 83–88, Jan. 2003. [Online]. Available: http://doi.acm.org/10.1145/774763.774776

[19] D. Aguayo, J. Bicket, S. Biswas, and D. De Couto, "MIT roofnet: Construction of a Production Quality Ad-Hoc Network," ser. MobiCom Poster '03, 2003. [Online]. Available: http://www.sigmobile.org/mobicom/2003/posters/13-Biswas.pdf

[20] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '05. New York, NY, USA: ACM, 2005, pp. 31–42. [Online]. Available: http://doi.acm.org/10.1145/1080829.1080833

[21] N. Tsarmpopoulos, I. Kalavros, and S. Lalis, "A low-cost and simple-to-deploy peer-to-peer wireless network based on open source Linux routers," in *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2005. Tridentcom 2005. First International Conference on*, Feb 2005, pp. 92–97.

[22] H. Lundgren, K. Ramachandran, E. Belding-Royer, K. Almeroth, M. Benny, A. Hewatt, A. Touma, and A. Jardosh, "Experiences from the design, deployment, and usage of the UCSB MeshNet testbed," *Wireless Communications, IEEE*, vol. 13, no. 2, pp. 18–29, April 2006.

[23] R. Sombrutzki, A. Zubow, M. Kurth, and J.-P. Redlich, "Self-Organization in Community Mesh Networks The Berlin RoofNet," in *Operator-Assisted (Wireless Mesh) Community Networks, 2006 1st Workshop on*, Sept 2006, pp. 1–11.

[24] M. Campista, P. Esposito, I. Moraes, L. Costa, O. Duarte, D. Passos, C. de Albuquerque, D. Saade, and M. Rubinstein, "Routing Metrics and Protocols for Wireless Mesh Networks," *Network, IEEE*, vol. 22, no. 1, pp. 6–12, Jan 2008.

[25] D. Wu, D. Gupta, and P. Mohapatra, "QuRiNet: A wide-area wireless mesh testbed for research and experimental evaluations," *Ad Hoc Networks*, vol. 9, no. 7, pp. 1221 – 1237, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870511000424

[26] V. Siris, E. Tragos, and N. Petroulakis, "Experiences with a metropolitan multiradio wireless mesh network: design, performance, and application," *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 128–136, July 2012.

[27] A. R. Ulucinar, I. Korpeoglu, and E. Karasan, "Effects of physical channel separation on application flows in a multi-radio multi-hop wireless mesh network: An experimental study on BilMesh testbed," *Journal of Network and Computer Applications*, vol. 39, pp. 253 – 265, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804513001689

[28] M. Afanasyev, T. Chen, G. Voelker, and A. Snoeren, "Usage Patterns in an Urban WiFi Network," *Networking, IEEE/ACM Transactions on*, vol. 18, no. 5, pp. 1359–1372, Oct 2010.

[29] Microsoft Research. Self Organizing Wireless Mesh Networks. [Online]. Available: http://research.microsoft.com/en-us/projects/mesh/

[30] R. Draves, J. Padhye, and B. Zill, "Comparison of Routing Metrics for Static Multi-hop Wireless Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 133–144, Aug. 2004. [Online]. Available: http://doi.acm.org/10.1145/1030194.1015483

[31] CISCO. Cisco Enterprise Wireless Mesh Solution Overview. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/wireless/aironet-1240-ag-series/net_brochure0900aecd806b9180.html

[32] V. Brik, S. Rayanchu, S. Saha, S. Sen, V. Shrivastava, and S. Banerjee, "A Measurement Study of a Commercial-grade Urban Wifi Mesh," in *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC

'08. New York, NY, USA: ACM, 2008, pp. 111–124. [Online]. Available: http://doi.acm.org/10.1145/1452520.1452534

[33] ABB. (2014) Tropos mesh network architecture. Topos Networks Inc. [Online]. Available: https://www.tropos.com/products/architecture.html

[34] ——, *Tropos Mesh OS: Foundation of the Tropos mesh network architecture*, ABB Inc., 2014. [Online]. Available: https://www.tropos.com/solutions/

[35] Motorola. (2014) Mesh wide area networks solutions. Motorola Solutions, Inc. [Online]. Available: http://meshsupport.motorolasolutions.com/Documents

[36] Aruba Networks. (2014) Aruba AirMesh multiservice mesh networks. Aruba Networks, Inc. [Online]. Available: http://www.arubanetworks.com/products/mesh-routers/

[37] Guifi.net. (2014) Open, Free and Neutral Network Internet for everybody. Guifi.net Foundation. [Online]. Available: http://guifi.net

[38] M. Oliver, J. Zuidweg, and M. Batikas, "Wireless commons against the digital divide," in *Technology and Society (ISTAS), 2010 IEEE International Symposium on*, June 2010, pp. 457–465.

[39] D. Vega, L. Cerda-Alabern, L. Navarro, and R. Meseguer, "Topology patterns of a community network: Guifi.net," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on*, Oct 2012, pp. 612–619.

[40] FunkFeuer. (2014) FunkFeuer Free Net. [Online]. Available: http://www.funkfeuer.at

[41] AWMN. (2014) Athens Wireless Metropolitan Network (AWMN). [Online]. Available: http://www.awmn.net/

[42] Community-Lab. (2014) Community Networks Testbed by the CONFINE project: a global facility for experimentation with network technologies and services for community networks. [Online]. Available: http://community-lab.net/

[43] A. Neumann, I. Vilata, X. Leon, P. Garcia, L. Navarro, and E. Lopez, "Community-lab: Architecture of a community networking testbed for the future internet," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on*, Oct 2012, pp. 620–627.

[44] M. Aymerich, R. Baig, P. Escrich, I. Vilata, A. Neumann, D. Vega, E. Lopez, F. Freitag, and L. Navarro, "Deploying applications with Community-Lab in wireless community networks," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, June 2013, pp. 1–3.

[45] L. J. de la Cruz Llopis, A. Vázquez-Rodas, E. Sanvicente, and M. Aguilar, "Load splitting in clusters of video servers," *Computer Communications*, vol. 35, no. 8, pp. 993 – 1003, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366412000746

[46] A. Vázquez-Rodas, L. J. de la Cruz Llopis, M. A. Igartua, and E. S. Gargallo, "Dynamic buffer sizing for wireless devices via maximum entropy," *Computer Communications*, vol. 44, pp. 44 – 58, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366414000899

[47] A. Vázquez-Rodas and L. J. de la Cruz Llopis, "A centrality-based topology control protocol for wireless mesh networks," *Ad Hoc Networks*, vol. 24, Part B, pp. 34 – 54, 2015, modeling and Performance Evaluation of Wireless Ad-Hoc Networks. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870514001577

[48] R. Rumipamba, A. Vázquez-Rodas, L. J. de la Cruz Llopis, and E. S. Gargallo, "Dynamic Buffer Size Allocation in Wireless Mesh Networks for Non-Elastic Traffic," *Revista Politécnica*, vol. 44, no. 2, pp. 17 – 26, 2014. [Online]. Available: http://www.revistapolitecnica.epn.edu.ec/revista_archivos/edicion_34_2014/TOMO_2_FINAL.pdf

[49] A. Vázquez-Rodas and L. J. de la Cruz Llopis, "Topology control for wireless mesh networks based on centrality metrics," in *Proceedings of the 10th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, ser. PE-WASUN '13. New York, NY, USA: ACM, 2013, pp. 25–32. [Online]. Available: http://doi.acm.org/10.1145/2507248.2507257

[50] A. Vázquez Rodas, L. J. de la Cruz Llopis, M. Aguilar Igartua, and E. Sanvicente Gargallo, "Dimensionado dinámico de buffers para flujos de tráfico diferenciados no elásticos," in *Proc. XI Jornadas de Ingeniería Telemática (JITEL 2013)*, October 2013, pp. 202–208.

[51] L. Urquiza-Aguiar, A. Vázquez-Rodas, C. Tripp-Barba, M. A. Igartua, L. J. de la Cruz Llopis, and E. S. Gargallo, "Max-min based buffer allocation for vanets," in *Wireless Vehicular Communications (WiVeC), 2014 IEEE 6th International Symposium on*, Sept 2014, pp. 1–5.

[52] L. Urquiza-Aguiar, A. Vázquez-Rodas, C. Tripp-Barba, A. M. Mezher, M. Aguilar Igartua, and L. J. de la Cruz Llopis, "Efficient deployment of gateways in multi-hop ad-hoc wireless networks," in *Proceedings of the 11th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, ser. PE-WASUN '14. New York, NY, USA: ACM, 2014, pp. 93–100. [Online]. Available: http://doi.acm.org/10.1145/2653481.2653487

[53] "IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pp. 1–2793, March 2012.

[54] R. Carrano, L. Magalhães, D. Saade, and C. Albuquerque, "IEEE 802.11s Multihop MAC: A Tutorial," *Communications Surveys Tutorials, IEEE*, vol. 13, no. 1, pp. 52–67, First 2011.

[55] G. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke, "IEEE 802.11s: The WLAN Mesh Standard," *Wireless Communications, IEEE*, vol. 17, no. 1, pp. 104–111, February 2010.

[56] X. Wang and A. O. Lim, "IEEE 802.11s wireless mesh networks: Framework and challenges," *Ad Hoc Networks*, vol. 6, no. 6, pp. 970 – 984, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870507001370

[57] J. Camp and E. Knightly, "The IEEE 802.11s Extended Service Set Mesh Networking Standard," *Communications Magazine, IEEE*, vol. 46, no. 8, pp. 120–126, August 2008.

[58] G. Hiertz, Y. Zang, S. Max, T. Junge, E. Weiss, and B. Wolz, "IEEE 802.11s: WLAN mesh standardization and high performance extensions," *Network, IEEE*, vol. 22, no. 3, pp. 12–19, May 2008.

[59] ns-3 Network Simulator. [Online]. Available: http://www.nsnam.org/

[60] K. Andreev and P. Boyko, "IEEE 802.11s mesh networking ns-3 model," in *Workshop on ns3*, 2010, p. 43.

[61] C. Perkins, E. Belding-Royer, and S. Das, "IETF RFC 3561, Ad hoc On-Demand Distance Vector (AODV) Routing," *Internet RFCs*, pp. 1–38, 2003. [Online]. Available: https://www.ietf.org/rfc/rfc3561.txt

[62] G. Conklin, G. Greenbaum, K. Lillevold, A. Lippman, and Y. Reznik, "Video coding for streaming media delivery on the internet," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 3, pp. 269–281, Mar 2001.

[63] D. Wu, Y. Hou, W. Zhu, Y.-Q. Zhang, and J. Peha, "Streaming video over the internet: approaches and directions," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 3, pp. 282–300, Mar 2001.

[64] H. Wen, Z. Hai-ying, L. Chuang, and Y. Yang, "Effective load balancing for cloud-based multimedia system," in *Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on*, vol. 1, Aug 2011, pp. 165–168.

[65] A. Dahnert, "Hawkeyes: An advanced ip geolocation approach: Ip geolocation using semantic and measurement based techniques," in *Cybersecurity Summit (WCS), 2011 Second Worldwide*, June 2011, pp. 1–3.

[66] M. Barnes, "HTTP-Enabled Location Delivery (HELD)," *IETF RFC 5985*, 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5985.txt

[67] M. Thomson and J. Winterbottom, "Discovering the local location information server (lis)," *IETF RFC 5986*, 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5986.txt

[68] M. Dawson, "The internet location services model," *Computer Communications*, vol. 31, no. 6, pp. 1104 – 1113, 2008, advanced Location-Based Services. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S014036640800039X

[69] J. Winterbottom and C. Bryce, "The internet location service," in *Intelligence in Next Generation Networks (ICIN), 2010 14th International Conference on*, Oct 2010, pp. 1–7.

[70] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A comparative study into distributed load balancing algorithms for cloud computing," in *Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on*, April 2010, pp. 551–556.

[71] J. Lee, "Parallel video servers: a tutorial," *MultiMedia, IEEE*, vol. 5, no. 2, pp. 20–28, Apr 1998.

[72] K. Hua, M. Tantaoui, and W. Tavanapong, "Video delivery technologies for large-scale deployment of multimedia applications," *Proceedings of the IEEE*, vol. 92, no. 9, pp. 1439–1451, Sept 2004.

[73] N. da Fonseca and H. Rubinsztejn, "Dimensioning the capacity of true video-on-demand servers," *Multimedia, IEEE Transactions on*, vol. 7, no. 5, pp. 932–941, Oct 2005.

[74] M. Reisslein, K. Ross, and S. Shrestha, "Striping for interactive video: is it worth it?" in *Multimedia Computing and Systems, 1999. IEEE International Conference on*, vol. 2, Jul 1999, pp. 635–640 vol.2.

[75] M. Moghal and M. Mian, "Effective load balancing in distributed video-on-demand multimedia system," in *Multi Topic Conference, 2003. INMIC 2003. 7th International*, Dec 2003, pp. 164–169.

[76] M.-Y. Wu, S. Ma, and W. Shu, "Scheduled video delivery-a scalable on-demand video delivery scheme," *Multimedia, IEEE Transactions on*, vol. 8, no. 1, pp. 179–187, Feb 2006.

[77] G. Muntean, P. Perry, and L. Murphy, "A comparison-based study of quality-oriented video on demand," *Broadcasting, IEEE Transactions on*, vol. 53, no. 1, pp. 92–102, March 2007.

[78] M. Aguilar Igartua, V. Carrascal Frías, L. J. de la Cruz Llopis, and E. Sanvicente Gargallo, "Dynamic framework with adaptive contention window and multipath routing for video-streaming services over mobile ad hoc networks," *Telecommunication Systems*, vol. 49, no. 4, pp. 379–390, 2012. [Online]. Available: http://dx.doi.org/10.1007/s11235-010-9388-x

[79] M. Aguilar Igartua and V. Carrascal Frías, "Self-configured multipath routing using path lifetime for video-streaming services over ad hoc networks," *Computer Communications*, vol. 33, no. 15, pp. 1879 – 1891, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S014036641000280X

[80] M. Aguilar Igartua, L. J. de la Cruz Llopis, V. Carrascal Frías, and E. Sanvicente Gargallo, "A game-theoretic multipath routing for video-streaming services over mobile ad hoc networks," *Computer Networks*, vol. 55, no. 13, pp. 2985 – 3000, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128611002027

[81] A.-K. Al-Tamimi, C. So-In, and R. Jain, "Modeling and resource allocation for mobile video over WiMAX broadband wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 28, no. 3, pp. 354–365, April 2010.

[82] X. Du, N.-S. Vo, W. Cheng, T. Duong, and L. Shu, "Joint Replication Density and Rate Allocation Optimization for VoD Systems Over Wireless Mesh Networks," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 23, no. 7, pp. 1260–1273, July 2013.

[83] ISO/IEC, "H. 262— ISO/IEC 13818-2," *Information technology–Generic coding of moving pictures and associated audio information–Part 2: Video*, 2013.

[84] ISO/IEC., "ISO/IEC 14496-2:2004," *Information technology – Coding of audio-visual objects – Part 2: Visual*, 2004.

[85] ITU-T, "H.264 : Advanced video coding for generic audiovisual services," *ITU-T H Series*, 2005.

[86] D. Serpanos, L. Georgiadis, and T. Bouloutas, "MMPacking: a load and storage balancing algorithm for distributed multimedia servers," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 8, no. 1, pp. 13–17, Feb 1998.

[87] J. Guo, E. W. Wong, S. Chan, P. Taylor, M. Zukerman, and K.-S. Tang, "Performance analysis of resource selection schemes for a large scale video-on-demand system," *Multimedia, IEEE Transactions on*, vol. 10, no. 1, pp. 153–159, Jan 2008.

[88] J. Guo, E. Wong, S. Chan, P. Taylor, M. Zukerman, and K.-S. Tang, "Combination load balancing for video-on-demand systems," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 7, pp. 937–948, July 2008.

[89] Q. Zhang, L. Cherkasova, and E. Smirni, "Flexsplit: A workload-aware, adaptive load balancing strategy for media clusters," in *In Multimedia Computing and Networking (MMCN'06*, 2006.

[90] P. Seeling, M. Reisslein, and B. Kulapala, "Network performance evaluation using frame size and quality traces of single-layer and two-layer video: A tutorial," *Communications Surveys Tutorials, IEEE*, vol. 6, no. 3, pp. 58–78, Third 2004.

[91] N. Ansari, H. Liu, Y. Shi, and H. Zhao, "On modeling MPEG video traffics," *Broadcasting, IEEE Transactions on*, vol. 48, no. 4, pp. 337–347, Dec 2002.

[92] J.-S. Lee and H.-D. Jeong, "Practical modelling for generating self-similar VBR video traffic," in *Computational Science – ICCS 2005*, ser. Lecture Notes in Computer Science, V. Sunderam, G. van Albada, P. Sloot, and J. Dongarra, Eds. Springer Berlin Heidelberg, 2005, vol. 3514, pp. 655–663. [Online]. Available: http://dx.doi.org/10.1007/11428831_81

[93] M. W. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," *SIGCOMM Computer Communication Review*, vol. 24, no. 4, pp. 269–280, Oct. 1994. [Online]. Available: http://doi.acm.org/10.1145/190809.190339

[94] L. de la Cruz, E. Pallares, J. Alins, and J. Mata, "Self-similar traffic generation using a fractional ARIMA model. application to the VBR MPEG video traffic," in *Telecommunications Symposium, 1998. ITS '98 Proceedings. SBT/IEEE International*, Aug 1998, pp. 102–107 vol.1.

[95] L. Kleinrock, *Queueing Systems, Vol. I: Theory.* Wiley-Interscience, 1975.

[96] D. Bertsekas and R. Gallager, *Data Netwoks.* Prentice-Hall, 1992.

[97] J. Gettys and K. Nichols, "Bufferbloat: Dark buffers in the internet," *Communications ACM*, vol. 55, no. 1, pp. 57–65, Jan. 2012. [Online]. Available: http://doi.acm.org/10.1145/2063176.2063196

[98] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson, "Netalyzr: Illuminating the edge network," in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '10. New York, NY, USA: ACM, 2010, pp. 246–259. [Online]. Available: http://doi.acm.org/10.1145/1879141.1879173

[99] N. Beheshti, Y. Ganjali, R. Rajaduray, D. Blumenthal, and N. McKeown, "Buffer sizing in all-optical packet switches," in *Optical Fiber Communication Conference and Exposition and The National Fiber Optic Engineers Conference*. Optical Society of America, 2006, p. OThF8. [Online]. Available: http://www.opticsinfobase.org/abstract.cfm?URI=OFC-2006-OThF8

[100] A. Vishwanath, V. Sivaraman, and G. Rouskas, "Anomalous Loss Performance for Mixed Real-Time and TCP Traffic in Routers With Very Small Buffers," *Networking, IEEE/ACM Transactions on*, vol. 19, no. 4, pp. 933–946, Aug 2011.

[101] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden, "Part III: Routers with very small buffers," *SIGCOMM Computer Communication Review*, vol. 35, no. 3, pp. 83–90, Jul. 2005. [Online]. Available: http://doi.acm.org/10.1145/1070873.1070886

[102] C. Villamizar and C. Song, "High Performance TCP in ANSNET," *SIGCOMM Computer Communication Review*, vol. 24, no. 5, pp. 45–60, Oct. 1994. [Online]. Available: http://doi.acm.org/10.1145/205511.205520

[103] L. Andrew, T. Cui, J. Sun, M. Zukerman, K.-T. Ko, and S. Chan, "Buffer sizing for nonhomogeneous TCP sources," *Communications Letters, IEEE*, vol. 9, no. 6, pp. 567–569, Jun 2005.

[104] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," *SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 281–292, Aug. 2004. [Online]. Available: http://doi.acm.org/10.1145/1030194.1015499

[105] N. Beheshti, Y. Ganjali, M. Ghobadi, N. McKeown, and G. Salmon, "Experimental study of router buffer sizing," in *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '08. New York, NY, USA: ACM, 2008, pp. 197–210. [Online]. Available: http://doi.acm.org/10.1145/1452520.1452545

[106] T. Li, D. Leith, and D. Malone, "Buffer Sizing for 802.11-Based Networks," *Networking, IEEE/ACM Transactions on*, vol. 19, no. 1, pp. 156–169, Feb 2011.

[107] H. Akimaru and K. Kawashima, *Teletraffic: Theory and Applications*. Springer, 1999.

[108] D. D. Kouvatsos, "A Maximum Entropy Analysis of the G/G/1 Queue at Equilibrium," *The Journal of the Operational Research Society*, vol. 39, no. 2, pp. pp. 183–200, 1988. [Online]. Available: http://www.jstor.org/stable/2582381

[109] V. Mahendran, T. Praveen, and C. Murthy, "Buffer dimensioning of delay-tolerant network nodes - a large deviations approach," in *Distributed Computing and Networking*,

ser. Lecture Notes in Computer Science, L. Bononi, A. Datta, S. Devismes, and A. Misra, Eds. Springer Berlin Heidelberg, 2012, vol. 7129, pp. 502–512. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-25959-3_37

[110] J. Liu, S. Chan, and H. Vu, "Performance Modeling of Broadcast Polling in IEEE 802.16 Networks with Finite-Buffered Subscriber Stations," *Wireless Communications, IEEE Transactions on*, vol. 11, no. 12, pp. 4514–4523, December 2012.

[111] A. Dhamdhere, H. Jiang, and C. Dovrolis, "Buffer sizing for congested internet links," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 2, March 2005, pp. 1072–1083 vol. 2.

[112] A. Dhamdhere and C. Dovrolis, "Open issues in router buffer sizing," *SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 87–92, Jan. 2006. [Online]. Available: http://doi.acm.org/10.1145/1111322.1111342

[113] K. Jamshaid, B. Shihada, L. Xia, and P. Levis, "Buffer sizing in 802.11 wireless mesh networks," in *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on*, Oct 2011, pp. 272–281.

[114] Y. Zhang and D. Loguinov, "ABS: Adaptive buffer sizing for heterogeneous networks," *Computer Networks*, vol. 54, no. 14, pp. 2562 – 2574, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128610001052

[115] G. Raina, D. Towsley, and D. Wischik, "Part II: Control Theory for Buffer Sizing," *SIGCOMM Computer Communication Review*, vol. 35, no. 3, pp. 79–82, Jul. 2005. [Online]. Available: http://doi.acm.org/10.1145/1070873.1070885

[116] C. Lee, D. Lee, and S. Moon, "Unmasking the Growing UDP Traffic in a Campus Network," in *Passive and Active Measurement*, ser. Lecture Notes in Computer Science, N. Taft and F. Ricciato, Eds. Springer Berlin Heidelberg, 2012, vol. 7192, pp. 1–10. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-28537-0_1

[117] M. Zhang, M. Dusi, W. John, and C. Chen, "Analysis of UDP Traffic Usage on Internet Backbone Links," in *Applications and the Internet, 2009. SAINT '09. Ninth Annual International Symposium on*, July 2009, pp. 280–281.

[118] D. Kouvatsos, "Maximum entropy and the G/G/1/N queue," *Acta Informatica*, vol. 23, no. 5, pp. 545–565, 1986. [Online]. Available: http://dx.doi.org/10.1007/BF00288469

[119] M. El-Affendi and D. Kouvatsos, "A maximum entropy analysis of the M/G/1 and G/M/1 queueing systems at equilibrium," *Acta Informatica*, vol. 19, no. 4, pp. 339–355, 1983. [Online]. Available: http://dx.doi.org/10.1007/BF00290731

[120] K.-H. Wang, S.-L. Chuang, and W.-L. Pearn, "Maximum entropy analysis to the N policy M/G/1 queueing system with a removable server," *Applied Mathematical Modelling*, vol. 26, no. 12, pp. 1151 – 1162, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0307904X02000562

[121] K.-H. Wang, T.-Y. Wang, and W.-L. Pearn, "Maximum entropy analysis to the N policy M/G/1 queueing system with server breakdowns and general startup times ,"

Applied Mathematics and Computation, vol. 165, no. 1, pp. 45 – 61, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0096300304003613

[122] Y. Gu, A. McCallum, and D. Towsley, "Detecting anomalies in network traffic using maximum entropy estimation," in *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '05. Berkeley, CA, USA: USENIX Association, 2005, pp. 32–32. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251086.1251118

[123] The Network Simulator – ns-2. [Online]. Available: http://www.isi.edu/nsnam/ns/

[124] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication.* University of Illinois Press, 1972.

[125] T. M. Cover and J. A. Thomas, *Elements of information theory.* John Wiley & Sons, 2006.

[126] R. K. Sundaram, *A first course in optimization theory.* Cambridge University Press, 1996.

[127] E. C. Molina, "The theory of probabilities applied to telephone trunking problems," *Bell System Technical Journal*, vol. 1, no. 2, pp. 69–81, 1922. [Online]. Available: http://dx.doi.org/10.1002/j.1538-7305.1922.tb00390.x

[128] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *Networking, IEEE/ACM Transactions on*, vol. 2, no. 1, pp. 1–15, Feb 1994.

[129] M. Fras, J. Mohorko, and Z. Cucej, "Packet size process modeling of measured self-similar network traffic with defragmentation method," in *Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on*, June 2008, pp. 253–256.

[130] Z. Sahinoglu and S. Tekinay, "On multimedia networks: self-similar traffic and network performance," *Communications Magazine, IEEE*, vol. 37, no. 1, pp. 48–52, Jan 1999.

[131] B. K. Ryu and A. Elwalid, "The Importance of Long-range Dependence of VBR Video Traffic in ATM Traffic Engineering: Myths and Realities," *SIGCOMM Computer Communication Review*, vol. 26, no. 4, pp. 3–14, Aug. 1996. [Online]. Available: http://doi.acm.org/10.1145/248157.248158

[132] T. G. Robertazzi, *Computer Networks and Systems: Queueing Theory and Performance Evaluation.* Springer New York, 2000.

[133] L. J. de la Cruz Llopis and E. Sanvicente, "Scalev: Herramienta software para la evaluación de algoritmos de scheduling," in *VI Jornadas de Ingeniería Telemática*, 2007, pp. 237–244.

[134] SCALEV transmission system simulator. [Online]. Available: http://scalev.upc.es

[135] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, "Overhaul of IEEE 802.11 modeling and simulation in ns-2," in *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, ser. MSWiM '07. New York, NY, USA: ACM, 2007, pp. 159–168. [Online]. Available: http://doi.acm.org/10.1145/1298126.1298155

[136] Trace Files. [Online]. Available: http://www2.tkn.tu-berlin.de/research/trace/ltvt.html

[137] W. Khan, Y. Xiang, M. Aalsalem, and Q. Arshad, "Mobile phone sensing systems: A survey," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 1, pp. 402–427, First 2013.

[138] Y. Jiang, A. Jaiantilal, X. Pan, M. Al-Mutawa, S. Mishra, and L. Shi, "Personalized energy consumption modeling on smartphones," in *Mobile Computing, Applications, and Services*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, D. Uhler, K. Mehta, and J. Wong, Eds. Springer Berlin Heidelberg, 2013, vol. 110, pp. 343–354. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36632-1_20

[139] A. Aziz, Y. Sekercioglu, P. Fitzpatrick, and M. Ivanovich, "A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 1, pp. 121–144, First 2013.

[140] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, "Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the compow protocol," in *in European Wireless Conference*, 2002, pp. 156–162.

[141] N. Li, J. Hou, and L. Sha, "Design and analysis of an mst-based topology control algorithm," *Wireless Communications, IEEE Transactions on*, vol. 4, no. 3, pp. 1195–1206, May 2005.

[142] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '01. New York, NY, USA: ACM, 2001, pp. 70–84. [Online]. Available: http://doi.acm.org/10.1145/381677.381685

[143] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "Peas: a robust energy conserving protocol for long-lived sensor networks," in *Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on*, May 2003, pp. 28–37.

[144] J. Wu, F. Dai, M. Gao, and I. Stojmenovic, "On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks," *Communications and Networks, Journal of*, vol. 4, no. 1, pp. 59–70, March 2002.

[145] R. Misra and C. Mandal, "Minimum connected dominating set using a collaborative cover heuristic for ad hoc sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 21, no. 3, pp. 292–302, March 2010.

[146] Z. Yuanyuan, X. Jia, and H. Yanxiang, "Energy efficient distributed connected dominating sets construction in wireless sensor networks," in *Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing*, ser. IWCMC '06. New York, NY, USA: ACM, 2006, pp. 797–802. [Online]. Available: http://doi.acm.org/10.1145/1143549.1143709

[147] A.-L. Barabási, *Network Science*, 2012. [Online]. Available: http://barabasilab.neu.edu/networksciencebook/

[148] T. G. Lewis, *Network Science: Theory and Applications.* Wiley Publishing, 2009.

[149] M. Newman, *Networks: An Introduction.* New York, NY, USA: Oxford University Press, Inc., 2010.

[150] D. Katsaros, N. Dimokas, and L. Tassiulas, "Social network analysis concepts in the design of wireless ad hoc network protocols," *Network, IEEE*, vol. 24, no. 6, pp. 23–29, November 2010.

[151] E. Stai, V. Karyotis, and S. Papavassiliou, "Topology enhancements in wireless multihop networks: A top-down approach," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 7, pp. 1344–1357, July 2012.

[152] C. Verma, B. Tamma, B. S. Manoj, and R. Rao, "A realistic small-world model for wireless mesh networks," *Communications Letters, IEEE*, vol. 15, no. 4, pp. 455–457, April 2011.

[153] A. Banerjee, R. Agarwal, V. Gauthier, C. K. Yeo, H. Afifi, and F. Lee, "A self-organization framework for wireless ad hoc networks as small worlds," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 6, pp. 2659–2673, July 2012.

[154] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks." *Nature*, vol. 393, no. 6684, pp. 409–10, 1998.

[155] R. Agarwal, A. Banerjee, V. Gauthier, M. Becker, C. K. Yeo, and B. S. Lee, "Achieving small-world properties using bio-inspired techniques in wireless networks," *The Computer Journal*, 2012. [Online]. Available: http://comjnl.oxfordjournals.org/content/early/2012/03/28/comjnl.bxs024.abstract

[156] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '07. New York, NY, USA: ACM, 2007, pp. 32–40. [Online]. Available: http://doi.acm.org/10.1145/1288107.1288113

[157] M. Kas, S. Appala, C. Wang, K. Carley, L. Carley, and O. Tonguz, "What if wireless routers were social? approaching wireless mesh networks from a social networks perspective," *Wireless Communications, IEEE*, vol. 19, no. 6, pp. 36–43, December 2012.

[158] G. Pallis, D. Katsaros, M. Dikaiakos, N. Loulloudes, and L. Tassiulas, "On the structure and evolution of vehicular networks," in *Modeling, Analysis Simulation of Computer and Telecommunication Systems, 2009. MASCOTS '09. IEEE International Symposium on*, Sept 2009, pp. 1–10.

[159] W. Hwang, Y.-r. Cho, A. Zhang, and M. Ramanathan, "Bridging centrality: identifying bridging nodes in scale-free networks," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 20–23.

[160] S. Nanda and D. Kotz, "Localized bridging centrality for distributed network analysis," in *Computer Communications and Networks, 2008. ICCCN '08. Proceedings of 17th International Conference on*, Aug 2008, pp. 1–6.

[161] K. Wei, D. Zeng, S. Guo, and K. Xu, "On social delay-tolerant networking: Aggregation, tie detection, and routing," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 6, pp. 1563–1573, June 2014.

[162] M. Musolesi and C. Mascolo, "Designing mobility models based on social network theory," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 3, pp. 59–70, Jul. 2007. [Online]. Available: http://doi.acm.org/10.1145/1317425.1317433

[163] S. Kosta, A. Mei, and J. Stefa, "Large-Scale Synthetic Social Mobile Networks with SWIM," *Mobile Computing, IEEE Transactions on*, vol. 13, no. 1, pp. 116–129, Jan 2014.

[164] C. Zhao and M. Sichitiu, "N-body: Social based mobility model for wireless ad hoc network research," in *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*, June 2010, pp. 1–9.

[165] C. Zhao, M. L. Sichitiu, and I. Rhee, "N-body: A social mobility model with support for larger populations," *Ad Hoc Networks*, vol. 25, Part A, pp. 185 – 196, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870514002212

[166] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002. [Online]. Available: http://www.pnas.org/content/99/12/7821.abstract

[167] A. Cuzzocrea, A. Papadimitriou, D. Katsaros, and Y. Manolopoulos, "Edge betweenness centrality: A novel algorithm for qos-based topology control over wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 4, pp. 1210 – 1217, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804511001135

[168] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, ser. DIALM '99. New York, NY, USA: ACM, 1999, pp. 7–14. [Online]. Available: http://doi.acm.org/10.1145/313239.313261

[169] D. Kim, Y. Wu, Y. Li, F. Zou, and D.-Z. Du, "Constructing minimum connected dominating sets with bounded diameters in wireless networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 20, no. 2, pp. 147–157, Feb 2009.

[170] M. Thai, F. Wang, D. Liu, S. Zhu, and D.-Z. Du, "Connected dominating sets in wireless networks with different transmission ranges," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 7, pp. 721–730, July 2007.

[171] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215 – 239, 1978-–1979. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0378873378900217

[172] ——, "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, Mar. 1977. [Online]. Available: http://links.jstor.org/sici?sici=0038-0431%28197703%2940%3A1%3C35%3AASOMOC%3E2.0.CO%3B2-H

[173] S. Kurkowski, W. Navidi, and T. Camp, "Constructing manet simulation scenarios that meet standards," in *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, Oct 2007, pp. 1–9.

[174] K. M. Carley, J. Pfeffer, J. Reminga, J. Storrick, and D. Columbus, "ORA user's guide 2012," DTIC Document, Tech. Rep., 2012.

[175] S. P. Borgatti, M. G. Everett, and L. C. Freeman, *Ucinet for Windows: Software for Social Network Analysis*, 2002.

[176] S. P. Borgatti, "Identifying sets of key players in a social network," *Computational & Mathematical Organization Theory*, vol. 12, no. 1, pp. 21–34, 2006. [Online]. Available: http://dx.doi.org/10.1007/s10588-006-7084-x

[177] P. V. Marsden, "Egocentric and sociocentric measures of network centrality," *Social Networks*, vol. 24, no. 4, pp. 407 – 422, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378873302000163

[178] M. Everett and S. P. Borgatti, "Ego network betweenness," *Social Networks*, vol. 27, no. 1, pp. 31 – 38, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S037887330400067X

[179] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, Jan 2000, pp. 10 pp. vol.2–.

[180] P. Jensen and et al, "Detecting global bridges in networks."

[181] D. Karamshuk, C. Boldrini, M. Conti, and A. Passarella, "Human mobility models for opportunistic networks," *Communications Magazine, IEEE*, vol. 49, no. 12, pp. 157–165, December 2011.

[182] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, Jun. 2008. [Online]. Available: http://dx.doi.org/10.1038/nature06958http://www.nature.com/nature/journal/v453/n7196/suppinfo/nature06958_S1.html

[183] D. J. Watts, *Small worlds: the dynamics of networks between order and randomness*. Princeton university press, 1999.

[184] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002. [Online]. Available: http://www.pnas.org/content/99/12/7821.abstract

[185] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, p. 026113, Feb 2004. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.69.026113

[186] C. Prest and Q. Hoellwarth, "Sports monitoring system for headphones, earbuds and/or headsets," Jun. 12 2014, uS Patent App. 14/181,578. [Online]. Available: http://www.google.com/patents/US20140161300

[187] T. Blackadar and J. Darley, "Monitoring activity of a user in locomotion on foot," May 24 2005, uS Patent 6,898,550. [Online]. Available: http://www.google.com/patents/US6898550

[188] Zepp. Analyze and improve your baseball, golf and tennis game. [Online]. Available: http://www.zepp.com/

[189] M. Abdol Aziz, A. Salleh, S. Sudin, F. Aziz, A. Mohamad Shakaff, M. Salim, and N. Abdul Rahim, "Athlete overtraining monitoring system," in *Proceedings of the International Colloquium on Sports Science, Exercise, Engineering and Technology 2014 (ICoSSEET 2014)*, R. Adnan, S. I. Ismail, and N. Sulaiman, Eds. Springer Singapore, 2014, pp. 209–217.

[190] D. K. Arvind and A. Bates, "The speckled golfer," in *Proceedings of the ICST 3rd International Conference on Body Area Networks*, ser. BodyNets '08. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 28:1–28:7. [Online]. Available: http://dl.acm.org/citation.cfm?id=1460257.1460295

[191] R. Marin-Perianu, M. Marin-Perianu, P. Havinga, S. Taylor, R. Begg, M. Palaniswami, and D. Rouffet, "A performance analysis of a wireless body-area network monitoring system for professional cycling," *Personal and Ubiquitous Computing*, vol. 17, no. 1, pp. 197–209, 2013. [Online]. Available: http://dx.doi.org/10.1007/s00779-011-0486-x

[192] T. Herbert. (2011) bql: Byte Queue Limits. [Online]. Available: http://article.gmane.org/gmane.linux.network/213308/

[193] J. Corbet. (2011) Network transmit queue limits. [Online]. Available: https://lwn.net/Articles/454390/

[194] OpenWrt. Openwrt wireless freedom. [Online]. Available: https://openwrt.org/