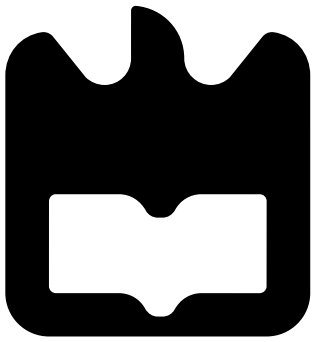
Universidade de Aveiro
Departamento de
Electrónica, Telecomunicações e Informática,
2012

Luís Manuel Cerqueira Barreto

Controlo de Congestionamento em Redes sem Fios.

Wireless Networks Congestion Control.



Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2012

Luís Manuel Cerqueira Barreto

Controlo de Congestionamento em Redes sem Fios.

Wireless Networks Congestion Control.

"Anyone who has never made a mistake has never tried anything new."

— Albert Einstein

Luís Manuel Cerqueira Barreto

Controlo de Congestionamento em Redes sem Fios.

Wireless Networks Congestion Control.

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Electrotécnica, realizada sob a orientação científica de Susana Sargento, Professora Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri / the jury

presidente / president

Carlos Borrego
Professor Catedrático da Universidade de Aveiro (por delegação do Reitor da Universidade de Aveiro)

vogais / examiners committee

Manuel Alberto Pereira Ricardo
Professor Associado da Faculdade de Engenharia da Universidade do Porto)

Alexandre Júlio Teixeira Santos
Professor Associado da Escola de Engenharia da Universidade do Minho

Amaro Fernandes de Sousa
Professor Auxiliar da Universidade de Aveiro

Susana Isabel Barreto de Miranda Sargento (orientadora / supervisor)
Professora Auxiliar da Universidade de Aveiro

agradecimentos / acknowledgements

This is the only page of my thesis that will not be judged by my excellent advisor, my great committee members and, of course, future readers. I am going to relax and acknowledge most of the people (certainly I will forget someone) that made this thesis possible.

I am deeply grateful to my advisor Professor Susana Sargento. She taught me how to do research, how to think critically, how to write papers and how to give presentations. She supplied me with her enthusiasm, support and creative insights, all of which were essential to complete this work.

I am quite fortunate to be a member of the IT NAP group. The memories of the fruitful discussions and exchange of questions helped me, indeed, with my research and kept me almost every time in the right direction. My special thanks, in particular to the NAP members that are in the "Farol" lab of IT. Your friendship and exchange of opinions surely were important for my final goal, this thesis. Thank you all for being so great colleagues. I cannot forget to thank Bruno Rés for his contributions and support.

I would like to thank Eduardo Rocha for his willingness in helping me, for his comments, suggestions and for helping me in my first days at IT. Thank you also for encouraging me, I will always be thankful for that.

My sincere thanks go to my mother, father, mother-in-law and father-in-law for always supporting me and understanding that I was not so present during these years.

Last, but not least, there are no words or actions to describe my gratefulness to my loving wife Quinhas and my beloved daughter Sara. Their love, patience, understanding and, of course, friendship kept me going all these years. I dedicate these thesis to both of them.

Resumo

O controlo de congestionamento continua a ser extremamente importante quando se investiga o desempenho das redes sem fios. Trabalhos anteriores mostram o mau desempenho do *Transport Control Protocol* (TCP) em redes sem fios. Os fatores que contribuem para um pior desempenho do TCP nesse tipo de redes são: a sua falta de capacidade para identificar/detetar e reagir adequadamente a eventos da rede; a utilização de um algoritmo de controlo de fluxo que não é adequado para o canal sem fios; e o colapso de congestionamento devido à mobilidade. Para colmatar este problemas foram propostos novos mecanismos de controlo de congestionamento baseados na taxa de transmissão. No entanto, estes mecanismos também apresentam um pior desempenho em redes sem fios, já que não utilizam mecanismos adequados para a avaliação da largura de banda disponível. Assim, é importante para melhorar o desempenho do controlo de congestionamento em redes sem fios, incluir componentes que são adequados para esse tipo de ambientes. Um esquema de controlo de congestionamento que permita uma partilha eficiente e justa da capacidade da rede e da largura de banda disponível entre múltiplas aplicações concorrentes é crucial para a definição de novos, eficientes e justos mecanismos de controlo congestionamento para as redes sem fios.

A Tese está dividida em três partes. Primeiro, apresentamos um estudo sobre a avaliação de desempenho de vários protocolos de controlo de congestionamento relativamente ao TCP, em redes sem fios em malha e *ad-hoc*. Os resultados obtidos mostram que os protocolos baseados na taxa de transmissão precisam de uma técnica de avaliação da largura de banda disponível que seja eficiente e precisa . A segunda parte da Tese apresenta um novo mecanismo de avaliação da capacidade da ligação e da largura de banda disponível, designada por *rt-Winf* (*real time wireless inference*). A avaliação é realizada em tempo real e sem a necessidade de inserir tráfego na rede. Os resultados obtidos através de simulação e emulação mostram que o *rt-Winf* obtém com precisão a largura de banda disponível e a capacidade da ligação sem sobrecarregar a rede. A terceira parte da Tese propõe novos mecanismos de controlo de congestionamento em redes sem fios. Estes mecanismos de controlo de congestionamento apresentam um conjunto de características novas para melhorar o seu desempenho, de entre as quais se destaca a utilização da informação de largura de banda disponível obtida pelo *rt-Winf*. Os resultados da avaliação destes mecanismos, utilizando o simulador ns-2, permitem concluir que a cooperação entre o *rt-Winf* e os algoritmos de controlo de congestionamento aumenta significativamente o desempenho da rede.

Abstract

Congestion control in wireless networks is an important and open issue. Previous research has proven the poor performance of the Transport Control Protocol (TCP) in such networks. The factors that contribute to the poor performance of TCP in wireless environments concern its unsuitability to identify/detect and react properly to network events, its TCP window based flow control algorithm that is not suitable for the wireless channel, and the congestion collapse due to mobility. New rate based mechanisms have been proposed to mitigate TCP performance in wired and wireless networks. However, these mechanisms also present poor performance, as they lack of suitable bandwidth estimation techniques for multi-hop wireless networks.

It is thus important to improve congestion control performance in wireless networks, incorporating components that are suitable for wireless environments. A congestion control scheme which provides an efficient and fair sharing of the underlying network capacity and available bandwidth among multiple competing applications is crucial to the definition of new efficient and fair congestion control schemes on wireless multi-hop networks.

The Thesis is divided in three parts. First, we present a performance evaluation study of several congestion control protocols against TCP, in wireless mesh and ad-hoc networks. The obtained results show that rate based congestion control protocols need an efficient and accurate underlying available bandwidth estimation technique. The second part of the Thesis presents a new link capacity and available bandwidth estimation mechanism denoted as *rt-Winf* (real time wireless inference). The estimation is performed in real-time and without the need to intrusively inject packets in the network. Simulation results show that *rt-Winf* obtains the available bandwidth and capacity estimation with accuracy and without introducing overhead traffic in the network.

The third part of the Thesis proposes the development of new congestion control mechanisms to address the congestion control problems of wireless networks. These congestion control mechanisms use cross layer information, obtained by *rt-Winf*, to accurately and efficiently estimate the available bandwidth and the path capacity over a wireless network path. Evaluation of these new proposed mechanisms, through ns-2 simulations, shows that the cooperation between *rt-Winf* and the congestion control algorithms is able to significantly increase congestion control efficiency and network performance.

Contents

List of Figures

List of Tables

Acronyms

TCP Transmission Control Protocol

XCP Explicit Control Protocol

XCP-b XCP blind

RCP Rate Control Protocol

MANET Mobile Ad-Hoc Network

WLAN Wireless Local Area Network

WMN Wireless Mesh Network

WSN Wireless Sensor Network

WCP Wireless Control Protocol

WCPCap WCP with Capacity

WXCP Wireless XCP

WRCP Wireless RCP

CNA Cooperative Neighborhood Airtime-limiting

TCP-AP TCP with Adaptive Pacing

ADR Average Dispersion Rate

AIMD Additive Increase Multiplicative Decrease

VoIP Voice over IP

NAV Network Allocation Vector

ACK Acknowledgment

RTO Retransmission Timeout

RTT Round Trip Time

ELFN Explicit Link Failure Notification

ICMP Internet Control Message Protocol

MSS Maximum Segment Size

RTS Request-To-Send

CTS Clear-To-Send

LL Logical Link

MAC Medium Access Control

PHY Physical

SNR Signal To Noise Ratio

RSSI Received Signal Strength Indication

DSDV Destination-Sequence Distance-Vector

CSMA-CA Carrier Sense Multiple Access with Collision Avoidance

MSDU MAC Service Data Unit

FTP File Transfer Protocol

CBR Constant Bit Rate

UDP User Datagram Protocol

MTU Maximum Transmission Unit

DCF Distributed Coordination Function

DIFS DCF Interframe Space

SIFS Short Interframe Space

LQ Localized Query

IPMON IP Monitoring

MRTG Multi Router Traffic Grapher

RST Reset

TCP-F TCP with Feedback

TCP-ELFN TCP with Explicit Link Failure Notification

imTCP TCP with in-line Measurement

TPA Transport Protocol for Ad-Hoc Networks

COPAS Contention-based Path Selection

ATCP TCP for mobile ad-hoc networks

RED Random Early Detection

NRED Neighborhood RED

LRED Link RED

EWCCP Explicit Wireless Congestion Control Protocol

ERDN Explicit Route Disconnection Notification

ERSN Explicit Route Successful Notification

ECN Explicit Congestion Notification

RFN Route Failure Notification

PN Pivot Node

EWMA Exponentially Weighted Moving Average

FHD Four Hop Propagation Delay

GSM Global System for Mobile

UMTS Universal Mobile Telecommunications System

OWD One Way Delay

OSI Open Systems Interconnection

Chapter 1

Introduction

1.1 Motivation

During this last decade there has been an important breakthrough regarding wireless networks technologies, having as a consequence the emergence of several types of those networks with different kinds of applications [3]. Wireless multi-hop networks can be classified into cellular network systems and wireless local networks (WLAN). Cellular systems can provide long range coverage, for example between different countries or regions. The Global System for Mobile communication (GSM) [4] or the Universal Mobile Telecommunications System (UMTS) are examples of cellular systems. WLAN use, for their communication process, the IEEE 802.11 standard [5]. WLANs provide only coverage from a few hundred meters to a few kilometers coverage range. WLANs can be further divided into Wireless Mesh Networks (WMN) and Wireless Ad-Hoc Networks.

A wireless ad-hoc network, or infrastructure-less network, is established when a collection of wireless nodes organize themselves to constitute a wireless network without the use of any infra-structure, such as access points or base stations. Wireless ad-hoc networks can also be classified, based on their application, in Mobile Ad-Hoc Networks (MANET). In a wireless ad-hoc network it is assumed that all nodes cooperate with each other forming a flexible and dynamic communication network. This type of cooperative model allows communication among mobile nodes also when no communication infrastructure exists in remote locations. Ad-hoc networks do not rely on dedicated routers for forwarding data packets. In a wireless ad-hoc network each mobile node uses routing functions to discover routes and forward packets.

WMNs have been designed to provide connectivity in sparse populated areas. WMNs are, as wireless ad-hoc networks, self-organized and self-configured networks. Nodes in a WMN automatically establish communications and maintain mesh connectivity [6]. A WMN is composed of two types of nodes: mesh routers and mesh clients. A mesh router consists of a node with gateway/bridge functions that allows the node to support mesh networking. Normally, a mesh router has no or little mobility. This allows to define a backbone for wireless mesh clients. The gateway/bridge functions of mesh routers en-

able feasible integration of WMNs with other types of networks, such as Ethernet, cellular networks and WiMAX [7] networks. WMNs, due to their overall flexibility, are commercialized and used for several applications, like broadband home networks, community networks and metropolitan area networks. For example, several wireless community networks are functional in Europe, Australia and USA [8].

In wireless networks there is no need to use cables and physical resources. Therefore, wireless networks rapidly become to be available on airports, universities, schools, hospitals and restaurants. They have revolutionized the way we communicate and have also propelled the increased use of new applications. Emerging applications in emergency search-and-rescue operations, data acquisition operations in inhospitable terrain, military control-and-command operations in a battlefield, and in meetings or conventions can be supported in wireless networks. In more recent years videoconferencing, voice communication, multiplayer gaming and streaming applications have emerged. These new emerging applications rely on reliable and efficient use of the network resources to be fully operational. In high mobility congested networks packet losses become an important issue, and such applications perform poorly being, thus, important to have effective and efficient congestion control mechanisms that maximize utilization and achieve a desired allocation of network resources.

Another critical drawback of wireless networks is the inability of nodes to detect collisions while they are transmitting. As a result, bandwidth is wasted in transmitting corrupted packets and the achieved throughput degrades. This situation is exacerbated as the number of nodes in the network increases, as the rate of collisions also increases. Channel overlapping also consists an important issue when using wireless networks: nodes might interfere with each other, resulting in packet errors and less efficient transmissions.

Congestion control is an important and an integral part of current communication networks. The Transmission Control Protocol (TCP) [9], the de-facto congestion control protocol on network communications, is known to perform well in wired networks but exhibits poor performance in wireless networks. TCP was developed taking into consideration traditional wired networks where packet losses occur mostly because of congestion [10]. However, networks with wireless and other lossy links also suffer from significant losses due to bit errors and hand-offs. TCP responds to the losses by invoking congestion control and avoidance algorithms, resulting in degraded end-to-end performance in wireless systems and networks [11]. TCP has been performing well in the wired networks, where losses are very low and mainly resulting from buffer overflowing in routers due to network congestion. The increase demand of network resources in wired communications has, also, put in evidence some TCP flaws. TCP uses a control law called Additive-Increase Multiplicative-Decrease (AIMD) to control congestion and fairness [12], thus, it does not effectively distinguish between the problem of congestion control and efficient network usage from the problem of proportionally sharing the bandwidth among individual users.

In wireless networks the most fundamental problem of TCP is the fact that TCP assumes that a packet loss is a result of congestion, thus TCP activates congestion control for every loss detected. This is not correct in wireless networks where losses can result from channel bit errors, route failures or hand-offs. As stated in [13] frequent events, other than congestive losses, in wireless networks require different control actions than

congestion control. TCP's AIMD mechanism implicitly allows the congestion window to grow beyond the optimum size overloading the channel, resulting in Medium Access Control (MAC) contention and inefficient medium usage. The AIMD mechanism probes for excess bandwidth in the network by increasing its congestion window after every successful transmission, and drastically reduces the window size after packet lost is detected. Since TCP has no mechanism to determine whether the network bandwidth is already fully utilized, it will continue to increase its window until the network is congested resulting in packet loss. This aggressive growth of congestion window beyond the network's available bandwidth causes network instability, that results in inefficient use of wireless bandwidth.

Congestion control in wireless networks is an active research challenge. In wired networks, excessive node queue utilization is an accurate indication of congestion. But this is not true in wireless networks. Congestion in a wireless channel can be the result of contention for channel access with neighboring nodes, that results in collisions prior to excessive queue build up. Congestion in wireless networks is not due to one specific node, but is a result of the combined activity of neighboring nodes of any wireless channel. Therefore the queue utilization at one specific node does not reflect the network congestion status. This is not true in wired networks, as only nodes on the same path can access that path, thus, congestion is only dependent of the traffic on that path. Mobility is another characteristic that affects congestion control in a wireless environment. In wireless networks nodes can move arbitrarily and be out of each other transmission range, causing route breakage. This can potentially lead to packet losses and to no communication between nodes. During the 'no communication' phase, packets are kept at the node's buffer resulting in buffer overflow, or congestion, if the congestion control mechanism keeps inserting packets at the source node. As previously referred a wireless channel, due to fading and shadowing effects, has high packet error rate. Thus, an efficient congestion control mechanism cannot rely on packet loss as a sign of congestion. The available bandwidth of a wireless multi-hop path is much lower than the capacity of each individual wireless link. This leads to bad channel utilization and high collision rate resulting in congestion collapses.

Many new congestion control proposals try to enhance TCP performance, and its AIMD process, over wireless networks. These proposals use a link failure notification message, that explicitly informs the sender that a link failure along the path has occurred. The sender then does not respond to this notification with a congestion control mechanism, freezing the congestion window size and the timers. Studies, like [14], show that these proposals perform worse than standard TCP in wireless ad-hoc networks.

Recent efforts to design better congestion control, in wired environments, have led to the origin of several rate based congestion control with explicit-feedback control methods. The main goal of flow/congestion control is to regulate the traffic injected by the source into the network to prevent overloading of paths, while maximizing the network utilization for good throughput performance. These methods solicit active multi-byte feedback from the routers to the end-hosts, delivering a precise and timely congestion signal that is used to accurately adjust flow sending rates, and hence, to achieve faster convergence, smaller packet loss rate, high link utilization and better fairness between flows. Examples of these congestion control mechanisms that rely on network interaction are the eXplicit Control

Protocol (XCP) [15] and the Rate Control Protocol (RCP) [16]. Their main purpose is to generalize explicit congestion notification, where nodes inform each other about the degree of congestion. XCP decouples channel utilization from fairness control, and requires the efficient estimation of the aggregate traffic behavior, i.e both available bandwidth and link capacity. RCP is a congestion control algorithm whose main key is to finish the flows as quickly as possible. RCP updates dynamically the rate assigned to the flows, approximating a processor sharing in the presence of feedback.

Since rate based congestion control can effectively decouple congestion control mechanism from the reliability mechanisms, new rate based congestion control mechanisms have been proposed for wireless networks. However, their rate estimation techniques are not very accurate introducing overhead and over-estimating link capacity. Hence, their performance is not very good.

Based on the TCP AIMD and on rate based congestion control schemes, new approaches have been proposed, like TCP with Adaptive Pacing (TCP-AP). These approaches are normally called hybrid schemes. These mechanisms try to infer link available bandwidth to control the AIMD process. However, their available bandwidth estimation is not correctly performed, overloading the network, which results in an inefficient use of network resources.

It is thus important in any type of network, but specially in wireless networks to develop a congestion control mechanism that is able to avoid network congestion and to improve network resources utilization, increasing network performance. To define a new congestion control mechanism for wireless networks, it is important to understand the main factors that degrade congestion control in wireless networks and how they can be mitigated. A congestion control mechanism for wireless networks must have the following characteristics: it needs to be aware of path breakage events that occur due to mobility; its congestion control scheme cannot rely on packet loss, thus a robust congestion detection mechanism and an efficient recovery loss mechanism are required; it must be able to detect and avoid wireless congestion, reducing the waste of wireless bandwidth; and a congestion control mechanism for wireless networks must be able to balance between achieving high throughput and low collision probability. Furthermore, due to the limited bandwidth of wireless networks, it must be understood if bandwidth estimation and control is only confined to the transport layer, or if it can be more effectively estimated and used with cooperation between different protocol layers.

1.2 Objectives

It is important that emerging wireless networks support a broad variety of applications like real-time media applications such as VoIP, multimedia applications such as video streaming, as well as data applications. Thus, one of the key issues is to provide efficient, effective and fair congestion control mechanisms. This is recognized as one of the biggest challenges for improving network performance and network resources utilization.

Congestion control main goal is to improve performance and to optimize resources usage in a communication network. This means that sources should be able to send data with

sending rates as high as possible without overloading the network, maximizing network utilization with good throughput performance. TCP has been shown to be a good choice in wired networks, where the networks are stable. However, in environments, like wireless networks, where rate is less stable and packet loss can be a consequence of bit errors due to interference, TCP has been proven to be inefficient and its rate behavior is less accurate. There has been an extensive study of the TCP AIMD congestion control algorithm, that has been proven to not be efficient in wireless networks as it tends to overload the channel, thus increasing packet losses [17].

Rate based congestion control protocols that rely on available bandwidth and link capacity for transmission control, using rate feedback from the nodes along a communication path, can prevent congestion using fast and precise rate adaptation techniques, allowing to maximize the medium usage. Rate based congestion control schemes allow a better definition of loss detection and recover mechanisms as they decouple congestion control from loss recovery. This allows to perform more efficiently in dynamic situations with losses due to congestion, channel bit error or mobility. Rate based congestion control mechanisms provide also a better use of the wireless channel as packets are sent at regular intervals, as opposed to TCP window-based flow control where packets are sent in bursts of traffic. Consequently, rate based congestion control mechanisms are able to provide an efficient and effective congestion control on wireless networks. However, rate based congestion control uses rate or available bandwidth estimation as its main component. The accuracy of available bandwidth and link capacity estimation defines the stability and the efficiency of the rate control mechanism.

Rate based control protocols such as XCP [15] and RCP [16] have been proposed, in wired networks, to solve the problems of TCP, and, due to the previous considerations, can be a good solution for wireless networks. Their sending rate is determined by the network and feedback to the sender as an explicit rate. Then, the sender transmits data at the given explicit rate. However, these protocols may underestimate or fail to correctly use the available bandwidth. This may lead to congestion collapse due to the use of incorrect rate information.

It is thus clear that available bandwidth and link capacity are main issues and major factors on wireless networking behavior. An accurate mechanism that estimates those parameters, and that is able to make them available to a rate based congestion control mechanism, like XCP and RCP, will improve network performance. Overcoming such challenges is a key requirement for wireless congestion control improvement.

Estimation of link capacity has been widely studied, and can be achieved through either active or passive measurement [18]. Active measurement works by injecting measurement probe packets into the network, while passive measurement mechanisms use existing data transmission. Active measurement has some important drawbacks, such as adding excessive overhead and not always maintaining end-to-end semantics; passive measurement can be less reliable as it cannot rely on all the data. Active measurement tools work by sending out a series of various probe packets with different sizes and, for each probe, they measure the time an error packet is received. The bandwidth of each link and its latency are obtained through statistical analysis of those measurements. Other active measurement tools rely on

self induced congestion. They use the principle of injecting probe packets in the network, with some inter-packet separation, and measuring the effects of these packets. When a probe packet arrives at the receiver, it is timestamped. The inter-packet separation is increased if the packets have suffered congestion on the path. Passive measurement tools rely on the network information and participants to obtain the desired calculations. These tools only add time processing overhead.

Due to its particularities available bandwidth and link capacity in wireless networks should be performed in real time without affecting the network dynamics. Using information from the MAC layer, it is possible to accurately characterize the busy time and the total elapsed time, hence obtaining accurate link capacity and available bandwidth estimations. Thus, a cross layer design scheme must be taken into consideration for congestion control improvement.

A main concern that it must be taken into consideration is that new congestion control mechanisms must cohabit with TCP. TCP is still the most used congestion control protocol, and many applications rely on their mechanisms for communicating. The TCP protocol is the most widely used protocol for Internet traffic including email, data, web browsing. Although, TCP was not originally designed for real time applications such as live streaming of multimedia content, various studies show that up to 80% of existing Internet multimedia services are TCP based [19]. It is also expected that, for the next decades, TCP will still be the most used congestion control protocol. A new congestion control protocol must, then, allow TCP to act normally. TCP friendliness is, thus, an important factor in congestion control development. Numerous congestion protocols have been proposed for enhancing TCP in wireless networks. TCP with Adaptive Pacing (TCP-AP) [20] is one of such proposals. It uses the AIMD process combined with rate based mechanisms.

Our focus in this Thesis is on understanding and addressing performance issues associated with congestion control in 802.11 based wireless networks. The main aim of this research is to develop a set of mechanisms that improve the performance of transport protocols (not necessary TCP), and are TCP compliant, in wireless multi-hop networks.

Our research objectives can be summarized as follows:

- Understand and explore the behavior of congestion control protocols in wireless multi-hop networks.

- Define mechanisms that can effectively and efficiently estimate, in real-time and without affecting the network, wireless link capacity and available bandwidth.

- Propose a set of improved congestion control solutions that can use the information of the estimation mechanism to improve network performance. We are interested in developing solutions that can adapt to link capacity and effectively use the medium.

In order to reach the goals stated, this work addresses several key areas as follows:

1. A survey on available bandwidth and link capacity estimation, and on congestion control mechanisms.

2. A performance evaluation of a set of congestion control protocols on wireless networks, exploring their advantages and disadvantages leading to potential new solutions.

3. The design and simulation of an experimental available bandwidth and path capacity estimation mechanism.

4. A performance evaluation of the estimation mechanism against existing estimation approaches.

5. The design and simulation of congestion control protocols using the integration, with cross layer techniques, of the estimation mechanism on XCP and RCP, thus, defining new congestion control mechanisms for wireless networks.

6. A performance comparison of the congestion control mechanisms with TCP and other existing congestion control protocols defined for wireless networks.

7. The improvement of an existing TCP based wireless congestion control protocol with the integration of the estimation mechanism and node path contention effect.

8. A performance comparison of all the congestion control proposals.

1.3 Main Contributions

Congestion control in wireless networks is an interesting research area already with important results. However, most of the research done has been confined to improving TCP behavior or to define new congestion control protocols that use a closed layer approach. It is evident, from the main results of this Thesis, that congestion control in wireless multi-hop networks has to be seen as the result of multiple layers cooperation. One of the most important contributions that arise from this Thesis is that MAC layer information, like link capacity and available bandwidth, can be directly used in the transport layer, increasing significantly network performance.

The work in this Thesis presents a performance evaluation, through network simulation, of some AIMD based congestion control protocols specifically developed for wireless networks, like WCP [21], TCP-AP [20] (TCP-AP is an hybrid protocol using both AIMD and rate based schemes), and some rate based congestion control mechanisms, like XCP [15], RCP [16] and XCP-b [22] (a rate based congestion mechanism design for wireless multi-hop networks), against TCP. To the best of our knowledge, the performance comparison between these protocols has never been done before. From the analysis of this performance study, it was possible to establish the way a new congestion control protocol should be developed, and what key metrics, namely link capacity and available bandwidth, should be used for improving network performance.

The further innovative contribution is the design of a MAC layer estimation mechanism, called *rt-Winf*, that can infer the link capacity and available bandwidth in real-time and without affecting the network dynamics. Link capacity and available bandwidth are major network performance factors that need to be accurately obtained in real-time and then actively used in network congestion control. *rt-Winf* is based on *IdleGap* [1], which

is an estimation tool designed for wireless networks. Although it uses a very accurate approach to characterize the busy time and the total elapsed time, *IdleGap* is not realistically determining the link available bandwidth. Our contribution optimizes *IdleGap* algorithm in order to obtain all the necessary times to calculate the path capacity and available bandwidth.

We contribute, then, with two innovative rate based congestion control protocols for wireless networks that offer the following properties: efficiency, fairness, scalability in network utilization, and also, feasibility in terms of implementation. These new proposals are called *XCP-Winf* and *RCP-Winf*. These proposals are among the earliest efforts to effectively incorporate XCP and RCP into wireless multi-hop networks. Our design is based on using *rt-Winf* estimations on the base XCP and RCP algorithms, using a simple cross layer communication strategy. We also introduce the collision probability factor, which is determined using MAC layer information, and taken into account by *rt-Winf*. Using network cooperation and accurate *rt-Winf* information for precise rate feedback, nodes are able to use more effectively the shared medium and converge to a transmission state that improves overall network performance. We also introduce the concept of TCP awareness into *XCP-Winf* and *RCP-Winf*, making them behave more TCP friendly. The comparison of *XCP-Winf* an *RCP-Winf*, using ns-2 [23] simulations in several wireless scenarios and topologies, show that these proposals are successful in improving overall network performance.

Finally, to develop a TCP-based approach that is able to deal with wireless multi-hop networks, we contribute with the incorporation of *rt-Winf* estimations into TCP-AP. TCP-AP uses the capacity estimations for nodes within the considered four hops propagation delay, and the standard TCP AIMD over the other hops. Using only the capacity estimations within the four hops propagation delay introduces inaccuracy, thus we further contribute with the consideration of the node path contention count effect. This new proposal is called *Wireless Enhanced TCP-AP* (*WE TCP-AP*). Evaluation of *WE TCP-AP*, with ns-2 simulations against other experimental proposals, shows that *WE TCP-AP* clearly outperforms base TCP-AP behavior.

1.4 Publications

The research performed in the scope of this Thesis resulted in a number of publications: 1 paper published in a national conference with referees, 3 papers published in international conferences with referees, and two other papers were submitted to International Journals, and are pending publication decision.

1.4.1 International Proceedings with Independent Review

- L. Barreto and S. Sargento, "TCP, XCP and RCP in wireless mesh networks: An evaluation study." *in 15th IEEE Symposium on Computers and Communications (IEEE ISCC10)*, Riccione, Italy, 6 2010.

- B. Rés, L. Barreto, and S. Sargento, "*rt-winf*: Real time wireless inference mechanism." *in IEEE Globecom 2010 Workshop on Mobile Computing and Emerging Communication Networks (MCECN 2010)*, Miami, Florida, USA, 2010.

- L. Barreto and S. Sargento, "*XCP-winf* and *RCP-winf*: Congestion control techniques for wireless mesh networks." *in IEEE International Conference on Communications (ICC 2011)*, Kyoto, Japan, 2011.

1.4.2 National Proceedings with Independent Review

- L. Barreto and S. Sargento,"How Real-time Bandwidth Inference Improves the Congestion Control in Wireless Mesh Networks." *In Actas da 10 Conferência sobre Redes de Computadores (CRC'2004)*, Braga, Portugal, November 2010.

1.4.3 Pending

- L. Barreto and S. Sargento, "*XCP-winf* and *RCP-winf*: A new approach for wireless congestion control." submitted to *Springer AD HOC NETWORKS journal*, November 2011.

- L. Barreto and S. Sargento, "*WE TCP-AP*: Wireless Enhanced TCP-AP." *submitted to Springer COMNET - COMPUTER NETWORKS journal*, January 2012.

1.5 Thesis Organization

The remainder of this Thesis is organized as follows. Chapter 2 presents the state of the art regarding the main and major available bandwidth and link capacity estimation techniques used in wired and wireless networks. This chapter also discusses and presents the related work on congestion control mechanisms, including both rate based congestion control mechanisms and Additive Increase Multiplicative Decrease (AIMD) congestion control mechanisms. Chapter 2 ends with the discussion of the principle models and research carried out in collision probability and cross layer design.

In Chapter 3 we present a performance study and analysis of congestion control protocols. We establish a set of strong evaluation parameters that will be used in all congestion control performance evaluation along this Thesis. We analyze the performance of some experimental AIMD based congestion control protocols against TCP, WCP and TCP-AP. Rate based congestion control protocols are also analyzed against TCP. The proposals chosen are XCP, RCP and XCP-b. This Chapter allows to understand that much work is still needed in the definition of efficient congestion control mechanisms for wireless multi-hop networks.

Chapter 4 introduces a new available bandwidth and link capacity estimation technique, called *rt-Winf*. We provide a detailed description, implementation, and a performance

study of *rt-Winf* against other popular estimation mechanisms. The performance study is done using the ns-2 simulator and the CMU wireless emulator [24].

In Chapter 5, we discuss the design, implementation, and evaluation of two new experimental congestion control mechanisms for wireless environments, *XCP-Winf* and *RCP-Winf*. We demonstrate that the use of MAC layer information improves congestion control performance that results in overall improved network performance. The results and conclusions obtained also motivate the need for cross layer approaches to wireless congestion control. In the comparison of *XCP-Winf* and *RCP-Winf*, we evaluate the metrics used in Chapter 3, including a set of measurements that allowed us to verify TCP friendliness.

In Chapter 6 we present a new congestion control mechanism based in TCP-AP, *WE TCP-AP*. TCP-AP is a TCP based protocol, which was developed to improve TCP performance in wireless ad-hoc networks. We provide a performance evaluation of TCP-AP against other wireless congestion control techniques. With the performance results and the identification of the factors that limit TCP-AP performance, we then, focus on proposing enhancements to be applied into the original TCP-AP design. The Chapter ends with a performance evaluation of the improved TCP-AP protocol, the *WE TCP-AP*.

In Chapter 7 we summarize the contributions of this Thesis, discuss the limitations of this work, present possible future directions of this work, and outline the concluding remarks.

Chapter 2

Related Work

2.1 Introduction

Congestion control has been a high debated and researched issue in wired and wireless networks. A large number of protocols and mechanisms have been proposed with the main objective to improve congestion control, specially in wireless environments. Some of the proposals try to enhance Transport Control Protocol (TCP) behavior, incorporating new congestion parameters and algorithms for wireless networks. These proposals rely on TCP modifications that address specific inefficiencies of TCP itself, or rely on routing or link layer improvements aimed at improving TCP performance, while keeping unchanged the Additive Increase Multiplicative Increase (AIMD) behavior of TCP. We can classify these proposals as AIMD based congestion control mechanisms. Some of its examples are TCP-Feedback (TCP-F), TCP-BuS or the Wireless Control Protocol (WCP).

Other mechanisms, however, focused in using rate control feedback between receiver and sender, to improve network performance. We can classify these proposals as rate based congestion control mechanisms. The eXplicit Control Protocol (XCP) and the Rate Control Protocol (RCP) are two examples of rate based congestion control mechanisms developed for wired networks. Some new proposals like the XCP-blind (XCP-b) and the Wireless RCP (WRCP) were defined for wireless congestion control.

Recent research on congestion control tries to use the AIMD process of TCP together with a rate based congestion control scheme. These proposals attempt to benefit from the best of the two worlds. We can classify these new approaches as hybrid congestion control mechanisms. Some examples of hybrid congestion control schemes, specially developed for wireless environments, are TCP with Adaptive Pacing (TCP-AP) and WCP Capacity based (WCPCap).

In this chapter, we first discuss the main operating principles of TCP and its problems to deal with wireless environments. Then, we describe some of the congestion control protocols specially developed for wireless networks.

When discussing wireless networks congestion control, link capacity and available bandwidth are two important characteristics that can be used for congestion control improve-

ment, allowing a better medium usage and a better congestion control. Link capacity and available bandwidth estimation have been widely studied, specially in wired networks. However, new efforts have been made in developing estimation mechanisms that can be applied in the context of wireless networks. We then discuss in this chapter some of the most used mechanisms for capacity and available bandwidth estimation.

For higher accuracy, both link capacity and available bandwidth should be inferred at the Medium Access Control (MAC) layer. Then, to optimize transport layer behavior and improve the overall network performance cross layer information, enabling layer interaction is fundamental. Other aspect that must be taken into consideration when the communicating nodes are mobile is that, due to the nature of mobile communications, collision probability is increased. This chapter then introduces some of the more relevant work that has been addressed in those fields.

As a resume, this chapter main aim is to provide the fundamental knowledge of mechanisms and protocols that will allow the definition of our new proposed estimation techniques and congestion control mechanisms.

This chapter is organized as follows. Section 2.2 describes the major available bandwidth and link capacity estimation techniques, categorized into active measurement tools (section 2.2.1) and passive measurement tools (section 2.2.2). Then, section 2.3 discusses the main congestion control schemes. Section 2.3.1 introduces the TCP, the most used congestion control mechanism. Next, section 2.3.1.1 presents a set of TCP's limitations and issues in wireless networks. Then, section 2.3.2 presents congestion control approaches that use TCP's AIMD scheme, and section 2.3.3 introduces the mechanisms that use a rate control scheme for congestion control. Section 2.3.3.1 presents the main operating principles of XCP and some XCP-based congestion control mechanisms for wireless networks. Section 2.3.3.2 introduces the main operating principles of RCP and a RCP-based congestion control mechanisms for wireless environments. Then, section 2.3.4 discusses the congestion control mechanisms that are both AIMD and rate based. Section 2.4 presents the most important research on collision probability in wireless networks dynamics, and finally, section 2.5 describes the most used models and some of the research carried out in cross layer design.

2.2 Capacity and Available Bandwidth Estimation

In a network path there is a sequence of H store-and-forward links that transfer packets from a sender to a receiver. Each link i can transmit data at a rate C_i, referred as link capacity. Then, the wireless link end-to-end capacity can be defined as $C \equiv \min_{i=1...H} C_i$. The available bandwidth can, thus, be defined as the fraction of the links capacity that has not been utilized during a period of time. If we extend this concept to the entire path, the end-to-end available bandwidth is the minimum available bandwidth among all links in the path.

Estimation of link capacity has been widely studied, and can be achieved through either active or passive measurement [25]. Active measurement works by injecting measurement

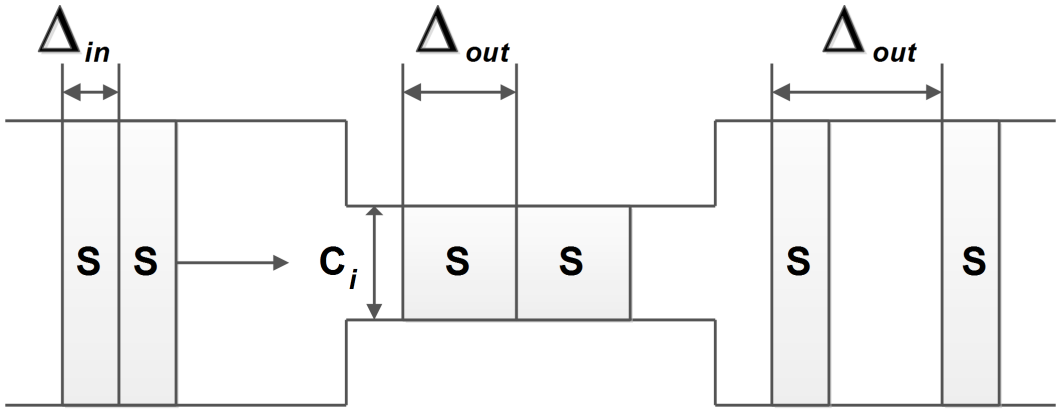


Figure 2.1: Packet Pair Dispersion.

probe packets into the network, while passive measurement tools use the trace history of existing data transmission. Active measurement has some important drawbacks, such as adding excess overhead and not always maintaining end-to-end semantics. In contrast, passive measurement can be less reliable as it cannot rely in all the data.

2.2.1 Active Measurement

Active measurement mechanisms inject probe traffic into the network at a traffic source and measure the network's influence at a probe traffic receiver. Thus, active measurement methods affect the network traffic itself, contrary to the passive measurement mechanisms. It must be noticed that active measurement methods need to access two hosts, one traffic source and one traffic receiver. Probing is the basic element of all active measurement methods, including measurements to gain information about link capacity and available bandwidth of a network path. There are some proposals of probing schemes, but the most common are the packet-pair and the packet-train probing schemes.

Packet pair and packet train probing are also known as packet dispersion techniques. The packet pair probing technique measures the bottleneck capacity of a path. Packet pair dispersion sends two packets with the same size back-to-back into the network. When two packets are sent one after the other, they will be received at the end of the path spaced in time and, if there is no cross-traffic, the spacing (or dispersion) between the packets is linearly related to the narrow link capacity.

The basic concept of packet pair dispersion is shown in Figure 2.1. The packet dispersion technique assumes that there is no cross traffic during the packet pair probing. When packets of size S with initial dispersion Δ_{in} are transmitted on the link with capacity C_i, the dispersion after the link Δ_{out} is:

$$\Delta_{out} = max(\Delta_{in}, \frac{S}{C_i}) \tag{2.1}$$

When the packets are transmitted over each link along an H hop end-to-end path, the final dispersion Δ_{final} at the receiver is:

$$\Delta_{final} = max_{i=1,...,H}(\frac{S}{C_i}) = \frac{S}{min_{i=1,...,H}C_i}) = \frac{S}{C} \tag{2.2}$$

where C is the end-to-end capacity. Thus, end-to-end capacity can be obtained from:

$$C = \frac{L}{\Delta_{final}} \tag{2.3}$$

Packet pair dispersion techniques usually have a faster measurement time, and induce less stress on the network path. However, the effects caused by cross traffic may significantly degrade the accuracy of the link capacity measurement [26].

An approach that is similar in concept to the packet pair technique is the use of packet trains. The main concept is to send L back-to-back packets of size S and to measure, as defined in [27], at the receiver the Average Dispersion Rate;

$$ADR = (L-1)\frac{S}{\Delta} \tag{2.4}$$

where the dispersion Δ is the time between the arrival of the first and the last packet of the train. Again, if no cross traffic is present, the dispersion of the train will be due solely to the bottleneck link and the ADR will be equal to the capacity.

Packet train techniques are more robust and less sensitive to errors and timestamps granularity (the dispersion is measured over more packets), when compared to packet pair techniques, but the probability that a cross traffic packet interferes with the train of probe packets is higher.

Active link capacity and available bandwidth measurement has been a research topic in wired networks, since the introduction of Cprobe [28], the first tool to attempt to measure end-to-end available bandwidth as stated in [29]. Cprobe is a method for estimating bandwidth using Internet Control Message Protocol (ICMP) packet trains. It measures the dispersion of a train of eight maximum sized segments.

Tools like IPerf [30], AbGet [31], Pathchar [32], PathChirp [33], PathLoad [29], Path-Rate [27] and CapProbe [34] are examples of mechanism specifically defined for wired networks. The most used techniques are CapProbe and Pathrate. CapProbe uses only packet pairs, while PathRate uses both packet pair and packet trains on its estimations.

IPerf [30] is a benchmarking tool designed firstly to be functional on wired networks but that can be used on wireless networks. IPerf uses large transfers of packets to measure the achievable throughout or available bandwidth in an end-to-end path. It attempts to throttle a network with TCP or UDP traffic - discovering the maximum transfer throughput (bandwidth) between two nodes in a network, without monitoring in-between nodes/routers. It can also utilize parallel-streamed transfers if the appropriate libraries are installed and have the benefit of using user-specified window size for network transfers. IPerf is nowadays a commonly used network testing tool that measures the throughput of a network. IPerf allows the user to set various parameters that can be used for testing a

network, or alternately for optimizing or tuning a network. IPerf has a client and server functionality, and can obtain the measurements between the two ends, either unidirectionally or bi-directionally.

PathRate [27] analyzes the multi-modal nature of a packet gap distribution. PathRate is a tool that estimates the capacity of a path (bottleneck bandwidth). It first uses a large number of UDP packet pair measurements to identify all clusters, which generally include the one corresponding to the real capacity. It then uses long packet trains until it gets an unimodal bandwidth distribution. In this phase, PathRate generates 500 trains of N packets and measures the packet trains dispersion. The resulting bandwidth distribution becomes unimodal and converges to the asymptotic dispersion rate. The smallest cluster that is larger than the unimodal cluster then corresponds to the real capacity value. To minimize the influence of cross traffic, the measurement is repeated multiple times and with different packet sizes. PathRate with a high-resolution clock and timestamping facility becomes very accurate, specially in wireless networks and in high bandwidth paths. However, when the path is congested or suffers of high load variations, PathRate accuracy drops significantly, since it is not able to obtain an unimodal bandwidth distribution, as almost all packet pairs encounter additional dispersion due to cross traffic.

CapProbe [34] uses one way delays (OWD) to identify the packet pairs that are still back-to-back on the bottleneck link. It is based on the observation that packet pairs that are not interfered by competing traffic will have the smallest sum of one-way delays for the two packets in each pair. The main principle of CapProbe is that at least one of the two probing packets must have queued if the dispersion at the destination has been distorted from that corresponding to the narrow link capacity. CapProbe calculates delay sums of all packet pair samples and uses the dispersion of the sample with the minimum delay to estimate the link capacity. The estimation is based in the following observation: for samples that estimate an incorrect value of capacity, the sum of the delays of the packet pair packets, which is called the delay sum, includes cross-traffic induced queuing delay; this delay sum will be larger than the minimum delay sum, which is the delay sum of a sample in which none of the packets suffer cross-traffic induced queuing. The dispersion of such a packet pair sample is not distorted by cross-traffic and will redirect the correct capacity. Then, the capacity is easily derived from:

$$C = \frac{S}{T} \tag{2.5}$$

where S is the sampling packet size and T is the interval between packets with minimum delay sum.

Thus, CapProbe combines dispersion and delay measurements of packet pair probes. Searching for the pair with the minimum delay sum implies no post processing of probing pair data, allowing lower computation costs and faster capacity estimation.

Both PathRate and Caprobe are able to measure available bandwidth and link capacity. Several studies ([35], [36]) show that these two techniques are fast and accurate in both wired and last hop wireless scenarios. However, a more recent work, [37], shows that, in multi-hop wireless networks, CapProbe and Pathrate are not accurately obtaining the link

capacity due to the dynamics of the wireless environments.

AdHoc Probe [37], SenProbe [38], WBest [39] and TSProbe [40] are active measurement tools specifically designed for wireless networks. Both AdHoc Probe and SenProbe are based on CapProbe and they estimate end-to-end path capacity. AdHoc Probe uses the packet pair technique, as CapProbe, to estimate link capacity in a wireless network, when the estimation is based on OWD measurements. The OWD is obtained at the receiver and communicated to the sender. AdHoc Probe uses fixed size packet pairs sent from the sender to the receiver, and from the receiver to the sender. The receiver calculates the OWD of each packet subtracting the time received (obtained by the receiver clock) and the time sent (stamped in the packet header). The one delay sum is also calculated by the receiver. With this information, the receiver obtains the path capacity and informs the sender using the header of the probe packets. AdHoc Probe only provides the path capacity of wireless links, being thus limited in terms of gathered information.

SenProbe [38] is a lightweight capacity estimation tool that provides the effective end-to-end capacity of a wireless link. SenProbe was designed having into consideration Wireless Sensor Networks (WSN) that use Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) MAC scheme. SenProbe uses one way measurements to determine the capacity of a link and was inspired by CapProbe. The one way measurements allow SenProbe to better reflect the true capacity of a wireless channel. SenProbe uses the time dispersion between estimation packets to obtain capacity estimation. SenProbe uses the back-to-back train to avoid the effect of the hidden terminal in the CSMA-CA [41] scheme. While using the packet pair dispersion concept, SenProbe is a packet train technique. SenProbe simply reports the capacity estimation after receiving a certain number of samples. Similar to CapProbe, the accuracy of the capacity estimation increases as the number of packet trains grows. An important aspect of SenProbe is that it can be applied to situations different than the ones that are present on a typical Internet path, namely when there is mobility and high interference. However, a large number of samples is not suitable for mobile wireless networks as it will lead to high latency in estimation and may not allow to capture the dynamic properties of the wireless network.

WBest [39] is able to provide both available bandwidth and capacity. Its algorithm is divided in two phases. In the first phase, it uses packet pair techniques in order to determine the path capacity. In the second phase, it uses packet train techniques in order to determine the available bandwidth. In the first phase, WBest sends, during the estimation time period, n packet pairs to estimate the capacity C_n, using:

$$C_n = \frac{\int_{t_0}^{t_1} \frac{L}{T(t)} dt}{t_1 - t_0} \tag{2.6}$$

where L is the packet size and $T(t)$ is the packet dispersion time. Then, the median of the n packet pair is used to estimate the effective capacity of the path:

$$C = median(C_i), i = 1, \ldots, n \tag{2.7}$$

In the second phase, a packet train of length m is sent at rate C to estimate available

bandwidth. In a wired network with constant capacity C, the available bandwidth $AB(t)$ at time t can be derived by the relationship of $AB(t) = C - S(t)$, where $S(t)$ is the amount of cross traffic at time t. In a wireless network, this relationship holds for the instantaneous capacity $C(t)$ at time t:

$$AB(t) = C(t) - S(t) \tag{2.8}$$

In fact, in this phase, packets are sent at the rate obtained in the first phase. It means that in this period of time, the WBest tool is being very intrusive, causing undesired problems in the network.

Recently a new tool, based on AdHoc Probe, called TSProbe [40] was propose. TSProbe tries to overcome the main issues of the existing methods, for time-slotted networks such as bluetooth and WIMAX. TSProbe uses the interaction between several link layer properties to deploy an adaptive probing scheme that utilizes payloads that vary in size. TSProbe estimates the path capacity analyzing the relationship between channel utilization and a set of system parameters, as the data header size of the different network layers. While accurate in time-slotted connections, it lacks efficiency in dynamic wireless environments. As AdHoc Probe, TSProbe is a two step probing method. First it sends probe traffic with a predefined probe gap, making the queue to build up and allowing to obtain C_A, the capacity of probe A. Then, after a short period of time, a new probe B is sent with the same packet size as probe A but with a different probe gap, and obtains C_B. Then, using C_A , C_B and the measured packets gaps, the receiver obtains the available bandwidth.

2.2.2 Passive Measurement

Available bandwidth and path capacity can also be measured using passive measurement methods and tools. These methods and tools do not interfere on the network nor with the existing traffic. They just observe what is happening in the network, and with the observer parameter, they estimate the intended characteristics. Passive measurement is more complex to achieve then active measurement, as it needs some control and privileges over the underlying network infrastructure. A passive measurement strategy can, if all parameters are not well defined, give a wrong knowledge about end-to-end path characteristics, thus, conducting to inaccurate measurements. However, when dealing with large scale networks, passive measurement is the most suitable option as it does not interfere in the network dynamics.

Multi Router Traffic Grapher (MRTG) [42], IP Monitoring (IPMON) [43] and PPrate [44] are examples of passive measurement tools for wired environments. MRTG [42] was developed to obtain traffic loads on both outgoing and incoming router links. MRTG determines the available bandwidth as it measures the actual load of a link and the link capacity. Based on the measured MRTG link load and the MRTG capacity, a router can decide more effectively where to route traffic. For MRTG to work, the devices must have the Simple Network Management Protocol (SNMP) [45] configured.

IPMON [43] uses packet traces to obtain the necessary network characteristics. Packet

traces are more accurate than traffic loads, as they give a more detailed observation of the traffic on a link. The packet traces include information such as type of protocol, packet size, header size, which are used by IPMON to analyze the traffic behavior and obtain passively link capacity and available bandwidth. IPMON has three main elements: a set of passive monitoring entities which collect the packet traces, where each trace is a sequence of the packet records with the first 40 bytes of each packet; a data repository that stores the traces once they have been collected; and an analysis platform which performs off-line analysis. IPMON can collect information from multiple, geographic dispersed link, simultaneously. All of the collected information is timestamped with a synchronized global clock, giving the ability to do detailed analyses of packet queuing and transmission behaviors on an Internet backbone.

The advantage of IPMON is that it provides the capability to collect traces from multiple locations in the network and correlate the traces through highly accurate timestamps. This provides the capability to study both single link characteristics, as well as characteristics which require data from multiple links (e.g. delay). The main disadvantage of the system is that the amount of data collected is very large. Data from a single 24 hour period exceeds 3.3 TB. This requires both a large amount of resources to be installed in network facilities for data collection purposes, and a large amount of resources to perform data analysis.

PPrate [44] is a new passive measurement tool that uses packet dispersion techniques, and takes as input a packet trace from a TCP connection, normally obtained by *tcpdump*. PPrate can work at the sender and at the receiver. The sender uses the arrived ACK packets to determine the existing capacity, while the receiver uses the data stream, being more accurate. As PPrate relies on TCP connection from which it has no control, it can suffer from poor accuracy. PPrate takes as input a set of inter-arrival times and automatically estimates the link capacity. A PPrate receiver defines the distribution of the packet pair dispersion. This is accomplished by grouping the inter-arrival durations, and thus, building the distribution of the packet trains. If the traffic was not captured at the receiver side, the time dispersion of the packet pair arriving at the receiver is unknown. However, it is possible to use the dispersion of the pure ACKs as an approximation of the packet pair dispersion at the receiver. The inter-arrival times of the ACKs that acknowledge two Maximum Segment Size (MSS) packets are divided by two to reduce the effect of delayed ACK. Then, when the multi-modal distribution of the path is constructed, the capacity is estimated. Without observing long enough packet trains, PPrate will have problems to correctly estimate the dispersion rate, thus, conducting to capacity values that are not very accurate.

IdleGap [1] and ProbeGap [46] are two passive measurement tools designed for wireless networks. They rely on the principles of the IEEE 802.11 [5] standard and on the CSMA-CA scheme to obtain available bandwidth and link capacity.

ProbeGap [46] is a tool that measures available bandwidth in access networks such as cable-modem and 802.11-based wireless access networks. ProbeGap aims to evaluate the capacity by determining the ratio between non delayed probe packets and probe packets delayed due to cross traffic in the bottleneck link. An estimate of the available bandwidth

is obtained by taking the non-utilized share of the "narrow link" capacity. In ProbeGap, the sender sends a series of Poisson-spaced probes. Normally, a train of 200 probe packets are sent with a size of 20 bytes each over a time interval of 50 seconds. After probing the network, an algorithm performs a search for a turning point in the OWDs. Packets with delays below the turning point are assumed to have passed an idle link. For longer OWDs it is concluded that the link is busy. If no turning point is found, the link is assumed to be 100% busy, and no bandwidth is available. ProbeGap estimates the available bandwidth (AB) by:

$$AB = f_{idle} \times C \tag{2.9}$$

where f_{idle} is the idle time fraction and C is the link capacity. It is obvious that the link capacity must be known in advance to obtain the available bandwidth. The idle time fraction is estimated by gathering the OWDs samples. The capacity has to be obtained through other estimation techniques, like, for example, PathRate [27].

IdleGap [1] is a recent mechanism that obtains available bandwidth in wireless networks. IdleGap is focused on the CSMA-CA scheme of wireless networks. It takes Network Allocation Vector (NAV) [47] into consideration, that is then used by the idle nodes which are waiting to transmit. It uses an approach to characterize the busy time and the total elapsed time, obtaining an Idle Rate. IdleGap takes into consideration the CSMA-CA scheme of wireless networks to obtain its available bandwidth. The NAV shows how long other nodes allocate the link in the IEEE 802.11 MAC protocol. The idle time in the wireless network can then be estimated from the NAV information.

The Available Bandwidth (AB) is calculated using the occupation time:

$$AB = R_{idle} \times R_{data} \tag{2.10}$$

where R_{idle} is the idle rate and R_{data} is the data rate, or transmission rate. The R_{data} value is obtained from the IEEE 802.11 *Data Rate* frame field, thus, being considered the link capacity. In IdleGap, the authors propose to consider 3 different states for a wireless node: *Sender*, *Receiver* and *Onlooker*. In the *Sender* state, the node is transmitting data; in the *Receiver* state, the node is receiving data and; finally, in the *Onlooker* state, the node is not participating in the transmission. The busy time of a wireless link can be estimated adding up all communication state transactions of the nodes.

Since all this knowledge would be difficult to obtain, as it would be necessary to increase network traffic and, thus, traffic on the network, IdleGap proposes a new method to obtain all the necessary information only from one node. The communication transaction time of a node can be obtained by the sum of the sending time (ST) when a node is in the sender state, the receiving time (RT) when the node is in the receiver state, and the on-looking time (OT) when the node is not participating in the communication. All these times are distinguished in each node's NAV. IdleGap, therefore, determines the busy time (T_{Busy}) for each communication transaction by:

$$T_{Busy} = ST + RT + OT \tag{2.11}$$

The *Idle Rate* is:

$$R_{idle} = 1 - \frac{T_{Busy}}{T_{Elapsed}} \tag{2.12}$$

where T_{Busy} is the amount of time a node is in a transaction state, and $T_{Elapsed}$ is the total elapsed time that represents the difference between the last captured ACK time and the initial time. Figure 2.2 shows a time diagram revealing how NAV, T_{Busy} and $T_{Elapsed}$ are obtained. The authors consider 10 seconds intervals for obtaining the total elapsed time.

IdleGap uses the different transaction states to obtain the T_{Busy} and to obtain the $IdleRate$ an *Idle Module*. The *Idle Module* is added between the MAC and Network layers (Figure 2.3) of a wireless node. The update process of the NAV signals the *Idle Module* to update the T_{Busy}.

IdleGap was compared with ProbeGap and the results showed that its results outperformed ProbeGap, presenting more accurate results and reflecting available bandwidth variation with short observation times, being independent of cross traffic.

The introduction of the *Idle Module* has an important disadvantage, that is the modification, by the introduction of a new sublayer, of the Open Systems Interconnection (OSI) Model [48]. The *Idle Module* is responsible for the determination of the *Idle Rate*, differentiating the existing traffic in three categories, according to its destination: to the node, from the node and between other nodes. The existing wireless network equipments cannot use this method, as they need to be modified in order to have this sublayer. This is an important issue, as it will introduce incompatibility problems.

IdleGap uses the pre-defined IEEE 802.11 header *Data Rate* value, which is not practical and real, thus leading to not very accurate and over-dimensioned estimation values. The *DataRate* value used can be considered as a maximum theoretical value. Even in ideal conditions, this maximum value will never be reached, due to overhead caused by packet headers, time and Quality of Service (QoS) constraints. This difference between the real and the used *DataRate* introduces a systematic calculation error in IdleGap, leading to an important lack of accuracy.

2.2.3 Summary

Several efforts have been made to correctly estimate and evaluate wired and wireless link capacity and available bandwidth. Table 2.1 shows the previous referred mechanisms, including some of their characteristics: their type (active/passive), the type of network they can use, measurement methodology and the metrics they estimate.

It is very important for wireless network design, management and usage, to correctly obtain link capacity and available bandwidth. Those two parameters are very important to characterize wireless network performance, and also, to minimize congestion collapses in such networks.

It is showed that most of the mechanisms presented in the previous section, specially the ones using probe packets, are very intrusive and heavily increase network load, thus,

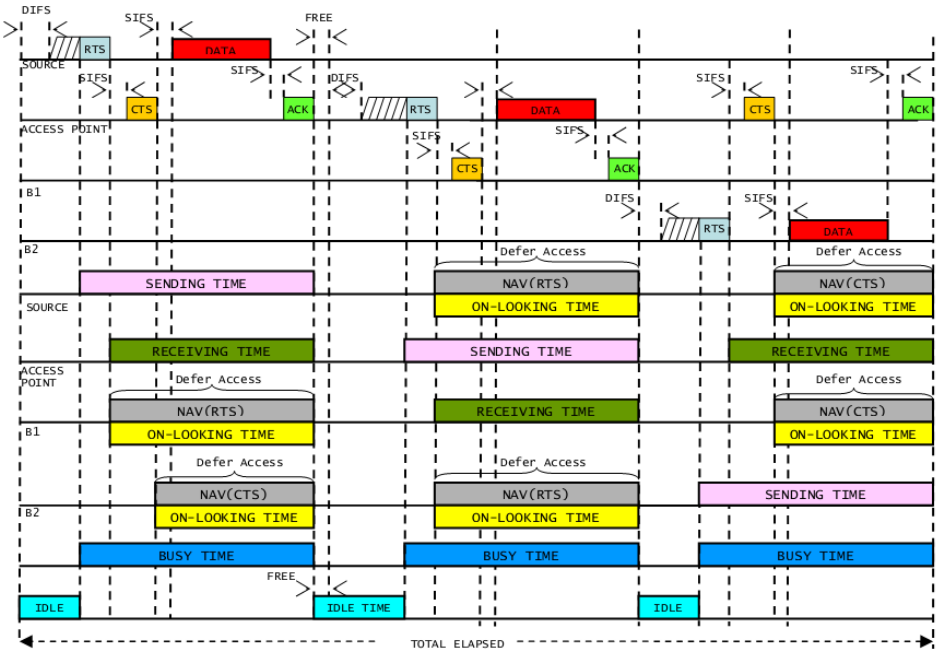


Figure 2.2: IdleGap NAV, T_{Busy} and $T_{Elapsed}$ Timing Diagram. [1]

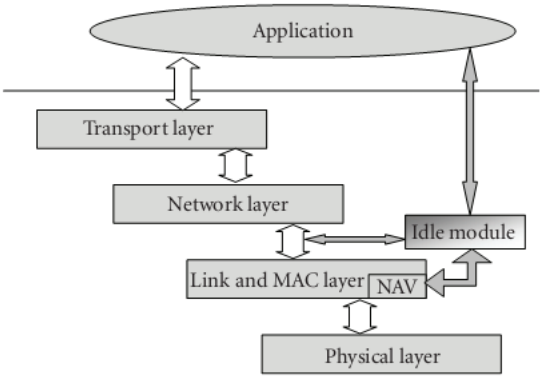


Figure 2.3: IdleGap Idle Module. [1]

obtaining inaccurate results that are under-estimated. Other mechanisms, while not intrusive, are very dependent on cross traffic, overestimating the network performance. One main requirement is the estimation in real-time, allowing the estimation results to be used by other applications or protocols.

IdleGap provides real time available bandwidth estimations and it is independent of cross traffic; it estimates the available bandwidth via the ratio of free time or idle time in

Table 2.1: Capacity and Available Bandwidth Estimation Tools.

Tool Name	**Type**	**Network**	**Methodology**	**Metrics**
IPerf	Active	Wired & Wireless	Path Flooding	Bandwidth
PathRate	Active	Wired	Pack. Pair & Gap	Band. & Cap.
CapProbe	Active	Wired	Packet Pair	Band. & Cap.
AdHoc Probe	Active	Wireless	Packet Pair	Capacity
SenProbe	Active	Wireless	Pack .Train (OWM)	Cap.
WBest	Active	Wireless	Pack. Train & Pair	Band. & Cap.
TSProbe	Active	Wireless	Adaptive Alg.	Cap.
MRTG	Passive	Wired & Wireless	Path Load	Band.
IPMON	Passive	Wired & Wireless	Packet Traces	Band.& Cap.
PPrate	Passive	Wired & Wireless	Packet Traces	Band.& Cap.
ProbeGap	Passive	Wireless & Wireless	One Way Delay	Cap.
IdleGap	Passive	Wireless	Elapsed Time Alg.	Band.& Cap.

the wireless links. IdleGap provides an efficient and fast method for available bandwidth estimation and, as it relies in NAV updates, it is also scalable. However, IdleGap available bandwidth estimation uses the IEEE 802.11 Header *DataRate*, considering it as the link capacity. This, as proved, is not very accurate introducing some lack of accuracy in IdleGap. Another important issue of IdleGap is the use of an *Idle Module* for time estimations between the MAC and the network layer.

Based on our analysis, we recommend that a next generation link capacity and available bandwidth estimation algorithm should have the following requirements: (1) perform efficient and fast estimations in real-time; (2) consider the actual link capacity, and not the one stamped in the IEEE 802.11 header; (3) be independent of cross traffic, not over estimating the results when such traffic is present; (4) not introducing modules that change the protocol stack, being transparent for the protocol stack; (5) allow network scalability; and (6) consider node and network cooperation for better accuracy.

As some of the requirements for a new estimation tool are present in IdleGap, we will base in IdleGap to propose a new link capacity and available bandwidth estimation mechanism, in Chapter 4.

2.3 Congestion Control in Wireless Networks

TCP is the most used and popular data transfer protocol. However, TCP in wireless networks is impacted by different environmental properties like radio signal related with fading, shadowing, interference, mobility and handovers, that change the network conditions dramatically. Delays and packet losses are seen by TCP as congestion, making it reacting wrongly and degrading network performance.

A significant number of works have tried to propose different strategies to optimize TCP performance in wireless networks. To provide smoother transmission rate than that

given by standard TCP, several TCP-like window-based congestion control mechanisms have been proposed. These mechanisms use a moderate window decrease parameter to reduce rate variability, meanwhile using a matching window increase parameter to satisfy TCP friendliness.

Other works defined new strategies, for both wired and wireless networks, that use network cooperation and explicit congestion notification based in rate control between a sink and a source. They define new transmission rate equation-based congestion control schemes. In these schemes, the end-systems use a TCP friendly equation to compute and use that information to control the transmission rate. Rate based congestion control schemes form an alternative option to the traditional end-to-end and window-based schemes. Rate based control schemes allow intermediate network nodes to participate in the control of congestion, by monitoring congestion and returning information on congestion to traffic sources. This feedback may be a single bit or an explicit rate value.

Another approach to provide smoother transmission control is the use of both TCP standard congestion mechanisms and rate based congestion control. These approaches are defined as hybrid congestion control protocols. In these schemes, the end-systems measure the packet loss rate and Round Trip Times (RTT), and use the TCP friendly equation to compute the transmission rate. However, they are very difficult to deploy and introduce unnecessary processing and network overload.

The next sub-sections describe the several sets of congestion control approaches: TCP and AIMD based, rate based and hybrid approaches.

2.3.1 TCP Overview

TCP [9] is the most used congestion control protocol in computer networks. TCP provides a connection-oriented, reliable data transmission between a source and a destination. Connection-oriented means that, before initiating any communication, there is a connection establishment process known as handshake negotiation.

TCP uses a three-way handshake negotiation between two communicating nodes to send data. Initially the sender sends a segment with the synchronization (SYN) bit activated and with a random sequence number to the receiver. The receiver, upon reception of the data segment, responds with other segment to the sender. If the receiver accepts the connection, the segment includes the SYN bit activated, the acknowledgement (ACK) bit activated and the sender random sequence number. If the receiver rejects the connection, it sends a segment with the reset (RST) bit activated.

Data is sent using segments that do not exceed a maximum segment size (MSS), negotiated via the three-way handshake initial connection establishment phase. Each byte (octet) of data has a sequence number assigned to it. When the receiver receives a segment, it stores the bytes of data (or sequence number range) of the segment and responds by sending back a cumulative ACK which confirms that all bytes up to the given sequence number have been successfully received. The TCP sender also maintains a retransmission timeout (RTO) timer, which on expiration indicates that a segment has been lost and has to be retransmitted. The functionality offered by cumulative ACKs, the RTO timer as well

as a checksum on the segment header and data, ensures reliability on top of IP.

TCP includes also a flow control mechanism that allows the receiver to limit the transmission rate, and a congestion control mechanism. The main idea of TCP is that the sender senses the network for available resources and increases the transmission rate until a packet loss is detected.

TCP flow congestion control is obtained through the use of a sliding window [49] that is measured in bytes. The sending rate is controlled by the congestion window maintained at the sender and the receiving window advertised by the receiver. Both windows are compared and the minimum of the two define the maximum amount of data that the TCP sender may maintain at any time in the network and along the communications path. The adjustment of the receiving window allows the receiver to set the rate of incoming segments, so that it does not become overloaded. Changing the congestion window is a mean to adapt to the varying network conditions and avoid causing congestion in the network.

TCP, as stated before, considers packet loss as network congestion indication, that triggers various congestion control mechanisms. The whole congestion control mechanism of TCP uses four distinct phases: the slow start, congestion avoidance, fast retransmit and fast recovery phases. The slow start phase is initiated after the initial handshake that establishes the connection, or following the expiration of the retransmission timer. Every time an ACK is received, the congestion window (*cwnd*) increases by one segment size, and the *cwnd* is doubled by RTT (i.e. increases exponentially). Initially, the slow start mechanism increases the *cwnd* until a congestion indication event is triggered or the maximum sending rate is reached. A congestion indication event can either be the reception of three duplicate ACKs (dupACKs) or a retransmission timeout (RTO).

TCP uses a variable that stores the value of half the sending rate, this variable is called the slow start threshold (*ssthresh*), which is used to update the *cwnd* when congestion is detected.

The congestion avoidance phase, which uses an Additive Increase Multiplicative Decrease (AIMD) process, is triggered when the *cwnd* reaches the *ssthresh* value during slow start or after the fast retransmit/fast recovery phase. During the congestion avoidance phase, *cwnd* increases linearly and up to one full sized segment per RTT. This phase tries to smoothly send segments into the network after reaching half the rate when the previous segment delivery failure occurred.

Finally, the fast retransmit/fast recovery phases occur when the sender receives three dupACKs which indicate that a TCP segment has been lost. A dupACK is sent by the receiver whenever it cannot acknowledge an arriving segment because it has not received all the segments sent prior to that one. The fast retransmit algorithm requires an immediate retransmission of the missing segment without waiting for the RTO timer to expire. Fast recovery sets

$$ssthresh = \frac{1}{2} \times cwnd \tag{2.13}$$

and sets

$$cwnd = ssthresh + (3 \times MSS) \tag{2.14}$$

Then, the *cwnd* increases for each additional dupACK received, so that it is possible to continue sending segments in an attempt to keep the network link utilized, while waiting for an ACK to acknowledge new data. When such an ACK arrives, *cwnd* decreases to *ssthresh* and TCP enters, again, the congestion avoidance phase. The linear increase of the sending rate (during the congestion avoidance phase) as well as its radical decrease (after an RTO or three dupACKs) form the Additive Increase Multiplicative Decrease (AIMD) behavior of TCP.

Due to the constant changes in the network routes and in the number of existing flows, it is important that TCP reacts to these changes and dynamically obtains both the RTT and the RTO. First, TCP must measure the RTT between sending a byte with a particular sequence number, and receiving an acknowledgment that covers that sequence number. Then, when an acknowledgment to a segment arrives at the TCP sender, the sender adjusts the RTO estimate as follows:

$$RTTVAR = (1 - \beta) \times RTTVAR + \beta \times |SRTT - R| \tag{2.15}$$

$$SRTT = (1 - \alpha) \times SRTT + \alpha \times R \tag{2.16}$$

$$RTO = max(SRTT + 4 \times RTTVAR, 1second) \tag{2.17}$$

where R is the measured round-trip time for the acknowledged segment, $RTTVAR$ is the variation of the recent round-trip times, and $SRTT$ is the smoothed mean round-trip time based on the recent measurements. α and β are constants with recommended values of $\alpha = \frac{1}{8}$ and $\beta = \frac{1}{4}$.

2.3.1.1 TCP in Wireless Mesh Networks

Due to its AIMD strategy, TCP is known to have several limitations: unstable throughput, increased queuing delay, limited fairness. It is also worth to mention that it was developed in the early 1980s, and today's application demands and network topologies differ greatly from the networks of that time. TCP and other congestion protocols assume that, in its operation and with today's network improvements, the probability of a lost packet is higher than the one of a corrupted packet [50]. It must be noticed that such a corollary is not true in wireless networks.

In a wireless network, packet loss is typically due to: wireless channel impairments causing bit errors, handoffs due to mobility and, of course, possibly congestion. TCP assumes that a packet loss is due to congestion in the network and, but not very often, packet reordering. As TCP mechanisms do not respond well to packet loss due to bit errors and handoffs, TCP-based applications suffer of poor performance. When a signal strength

weakness or noise is inferred in a wireless network, burst errors can occur. This means that more than one packet will be lost and TCP will detect it as a timeout, resuming to the slow-start strategy, and reducing significantly network performance. Usually, in a wireless network, delay is very high when compared to a wired network, which causes very long and variable RTT times making TCPs timeout mechanism leading to exacerbated link-level retransmissions. Wireless networks also have, typically, asymmetric links, where the ACK link is slower than the data transmission link. This is very important in TCP as delayed ACKs will limit throughput in the fast link.

The use of TCP in wireless networks showed that the TCP congestion mechanism is inadequate, leading to excessive false detection of congestion that results in a poor utilization of the network capacity [11], [51], and in bad allocation of bandwidth to contending flows.

The fast development and growth on wireless technology and networks, combined with the previously referred TCP issues, has been a great factor for the development of new approaches and paradigms of congestion control. New techniques for congestion detection/avoidance strategies have been proposed. Some of these proposals focus on improving TCP performance over wireless networks, while other proposals try to define new congestion control mechanisms based, not in the AIMD algorithm of TCP, but in using rate control as the main congestion control strategy. There are still others that have a hybrid approach.

2.3.2 Additive Increase Multiplicative Decrease (AIMD) Based Schemes

Several congestion control mechanisms were proposed to enhance TCP's behavior. TCP-Feedback (TCP-F) [52], TCP-Explicit Link Failure Notification (TCP-ELFN) [53], TCP-BuS [54], Wireless Control Protocol (WCP) [21], TCP with in-line measure (imTCP) [55] represent proposals for wireless networks in general. TCP for mobile ad-hoc networks (ATCP) [56], Transport Protocol for Ad-Hoc Networks (TPA) [57], Contention-based Path Selection (COPAS) [58], Link Random Early Detection (LRED) [59] and Neighborhood RED (NRED) [60] are some proposals specifically developed for ad-hoc networks. They try to enhance TCP behavior introducing new mechanisms that inform TCP of the reason for packet loss over the wireless link. The Explicit Wireless Congestion Control Protocol (EWCCP) [61] is a congestion control mechanism that relies on TCP's AIMD scheme, and is designed for wireless multi-hop networks.

TCP-F, TCP-ELFN, ATCP and TCP-BuS concentrate on improving TCP's throughput by freezing TCP's congestion control algorithm during link-failure induced losses, especially when route changes occur. These TCP developments differ in the manner in which losses are identified and notified to the sender, and in their details of freezing TCP's congestion control algorithm. Even though these schemes do not recognize the need of congestion detection and signaling over a neighborhood, their congestion metric implicitly takes some degree of neighborhood congestion into account.

In TCP-F [52] the TCP sender uses the network layer feedback mechanism, also known as route failure notification (RFN), from intermediate nodes to distinguish between a route failure and congestion in the network. A TCP sender, when receiving a RFN, stops sending packets and freezes all its parameters, such as the congestion window. This is called the *snooze* state. When, through the routing protocol, the TCP sender receives a route re-establishment notification (RRN), the sender knows that the route is re-established, resuming the transmission using the same parameters as before the *snooze* state. In order to avoid infinite wait for RRN messages, it is defined a route failure timer. When this timer is expired, the sender resumes to TCP normal congestion control mechanisms.

When the number of required re-establishments is low, [52] shows that TCP-F performs better than standard TCP in wireless networks. However, when used in a high density, high mobility wireless network, with a high number of RRN messages and re-establishments, TCP-F behaves poorly and the standard TCP is a better choice.

TCP-ELFN's [53] main objective is to provide with link and route failure notification to the TCP sender. The explicit link failure notification (ELFN) can be implemented using the Internet Control Message Protocol (ICMP) *host unreachable* or, if the routing protocol sends route failure messages, this information can be piggybacked with the information on these messages. After receiving a route failure information, the TCP sender enters the *stand-by* state, and freezes all timers. To check if the route has been re-established, the sender sends probe packets periodically. If an ACK packet is received, acknowledging the reception of one of the probe packets, the sender resumes the data transmission and the timer, leaving the *stand-by* state. The TCP-ELFN major problem in wireless environments is that as mobility increases, the performance is reduced since more probe messages need to be sent. When multiple flows co-exist in a wireless network, TCP-ELFN behavior is also considerably bad as the flooding of probe packets increases congestion in the network.

ATCP [56] introduces a new layer between the TCP and the IP layer. This layer is called Ad-Hoc TCP. ATCP uses ICMP messages to detect network division, and explicit congestion notification (ECN) messages to detect network congestion. These messages, sent by intermediate nodes, allow TCP sender to enter three states: the persist state, the congestion control state and the retransmit state. The sender enters the persist state when a route failure occurs; this is indicated by a *destination unreachable* ICMP message. This state ends when the connection is re-established. The retransmit state is entered when three duplicate ACK packets are received, indicating random errors. In this state, ATCP retransmits the lost packets from TCP buffer. After receiving a ECN message, the TCP sender enters in the congestion control state. In this state, the TCP sender uses the standard congestion control procedures.

The main problem of ATCP is the fact that it has to be associated with ICMP and ECN messages for its correct behavior. In high mobility networks the flooding of such messages will also increase congestion and reduce ATCP behavior. Moreover, if these messages are lost, the feedback gets unreliable.

TCP-BuS [54] uses feedback information similarly to TCP-F and TCP-ELFN, but needs the Associativity-Based Routing (ABR) [62] protocol to correctly work. TCP-BuS uses special messages of the ABR protocol, modified to carry a TCP connection and seg-

ment information, such as localized query (LQ) and REPLY to find a communication path. TCP-BuS uses two messages for explicit notification: the Explicit Route Disconnection Notification (ERDN) and the Explicit Route Successful Notification (ERSN). When the sender receives an ERDN message from the node that detects a route failure, called the pivot node (PN), it stops sending packets. The PN is also responsible for detecting route re-establishment using LQ messages. When a new route is activated or a route is re-established, the PN sends a ERSN message to the sender, and the sender resumes the transmission. TCP-BuS also introduces the buffering capability in mobile nodes. Packets along the path from the source to the PN are buffered, which allows a selective retransmission.

The evaluation of TCP-BuS presented in [54] shows that TCP-BuS outperforms the standard TCP. However, it must be noticed that a major issue of TCP-BuS is that it only works with the ABR protocol. ABR is not the most widely used routing protocol in wireless networks, introducing strong compatibility problems.

WCP [21] is AIMD based and, for every flow, the source maintains a rate R which represents the long term sending rate for the flow. WCP explicitly reacts to congestion within a neighborhood. WCP tries to identify the set of nodes within the vicinity of the congested node that need to reduce its rates. These nodes are marked using a lightweight congestion sharing mechanism. In WCP the source node additively increases R on every ACK packet received, and multiplicatively decreases R upon receiving a congestion notification from intermediate nodes (routers). Routers signal congestion by setting a congestion bit in the packet header of ongoing packets. The main idea of WCP is to perform congestion sharing. When a congestion link is detected, all packets forwarded on that link are marked with an explicit congestion notification. Sources receiving those packets can appropriately adapt the rates of the flows. A router detects congestion on its outgoing link using a simple thresholding scheme. It maintains an exponentially weighted moving average (EWMA) of the queue size, and when this average queue size is greater than a congestion threshold, the link is considered congested.

The main issue of WCP is that it does not correctly evaluate the available rate, making an assumption that all links have equal rate. Another important issue of WCP is that it does not take into consideration available bandwidth for its rate adaptation. WCP was developed for wireless mesh networks, but can be used in a wider scope.

ImTCP [55] introduces a new bandwidth measurement algorithm that can perform inline measurements. The available bandwidth estimation results from the arrival intervals of ACKs packets. ImTCP adjusts the interval of data packets according to the measurement algorithm, and then calculates the available bandwidth by observing the change of ACK intervals. During each measurement, ImTCP uses a search range to find the value of the available bandwidth. The search range is a range of bandwidth that is expected to include the current available bandwidth. The measurement algorithm uses a packet stream (a group of packets sent simultaneously) to initially obtain a very rough estimation of the available bandwidth and use the result to set the initial search range. ImTCP determines the available bandwidth as the largest rate of the packet pairs, for which the arrival interval is the same as the transmission interval.

A major issue of imTCP is that the arrival intervals of ACKs packets in dynamic wireless environments can suffer high variation, thus introducing lack of accuracy. Another important drawback of imTCP is that it can only estimate available bandwidth, and it does not use the link capacity in its congestion control.

TPA [57] is a new transport protocol designed to operate in mobile ad-hoc networks. TPA congestion control mechanism is inspired by the flow control window algorithm of TCP, but it is optimized to minimize the number of required packet retransmissions, transmitting blocks using a window-based scheme. The flow control mechanism in TPA is essentially TCP's flow control mechanism. TPA can also exploit, if there exists, the routing layer information of explicit link failure notification (ELFN). A TPA sender, when receiving an ELFN, enters a freeze state where the transmission window size is limited to one segment. Then, at the expiration of each retransmission timeout, TPA sends segments in the main or retransmission streams, probing the network for a new route. To limit the number of segments sent when there is no available route while in the freeze state, the value of the retransmission timer doubles after each timer expiration. The congestion control of TPA is based on ACK packet inhibition. Every time the sender detects a number of consecutive timeout expirations (this number has to be higher or equal to one), it considers an ACK inhibition and enters the congested state, freezing all important parameters such as congestion window and timers. When the sender receives a consecutive number of ACKs (defined by a threshold), it leaves the congested state. In normal operating conditions, when the path is not congested, the congestion window of TPA is set to the maximum value (2 to 3 segments). When TPA enters the congested state, the congestion window is reduced to 1 to allow congestion to disappear.

While providing reliable and connection-oriented type of service, when operating in highly dynamic wireless environments, TPA is very conservative and unfair. As TPA uses blocks on its operation, it can suffer from high performance degradation as it does not consider information on link capacity and available bandwidth measurements.

COPAS [58] is a novel congestion control algorithm that incorporates two mechanisms to improve TCP performance. COPAS uses disjoint forward paths, for TCP data, and reverse paths, for TCP ACK, trying to reduce conflicts and collisions. It also includes a dynamic contention-balancing technique that continuously monitors network contention and selects routes with minimum contention to avoid capture conditions. Specifically, when network contention on a route exceeds a certain threshold, called the backoff threshold, a new and less contended route is selected to replace the high contended route. COPAS is also able to redirect segments when a route is broken, using the second alternate route. The contention on the wireless channel is, in COPAS, a function of the number of times a node has to do backoff in each time interval. Intermediate nodes are continuously piggybacking its contention information on packets flowing through the forward and reverse paths, allowing the sender and the receiver to control the status of the reverse and forward routes, respectively.

COPAS has some important drawbacks. It is limited to static networks or limited mobility networks, as stated in [58]. Also, as the proposed route selection scheme attempts to find disjoint paths for different flows by assigning weights to links proportional to the

average number of backoffs on the link, COPAS needs to use a large amount of network information, thus being very aggressive to the network and requiring a large amount of processing, inducing excessive overhead in the network.

LRED [59] was proposed to reduce the problem of channel contention. LRED proposes an algorithm to tune wireless link drop probability, and an adaptive link layer pacing scheme to increase the spatial channel reuse. LRED achieves the contention reduction on the wireless medium by using an exponential weighted moving average of the number of retransmissions in the link layer. When the average retransmissions become larger than a given threshold, the probability of dropping/marking is computed according to the RED algorithm [63]. LRED can be coupled with ECN to notify the TCP sender about the congestion level. Adaptive pacing lets a node further backoff an additional packet transmission time when necessary, in addition to its standard backoff time. This extra backoff interval helps in reducing contention drops caused by exposed receivers, and extends the range of the link-layer coordination from one hop to two hops, along the packet forwarding path.

The LRED main problem is the fact that it calculates dropped packets based only on its own perception, which may make LRED behavior unfair, inefficient and unstable.

NRED [60] was initially proposed to enhance RED [63] to operate on the distributed neighborhood queue. This enhancement tries to make TCP behaving more fairly. As in RED, NRED nodes estimate the size of its neighborhood queue. Once the queue size exceeds a certain threshold, a drop probability is computed using the RED algorithm. As the NRED uses neighborhood queues, thus, being the aggregate of local queues at neighboring nodes, the drop probability is propagated to all neighboring nodes for cooperative packet drops. NRED identifies a subset of flows which share channel capacity with flows passing through a congested node. However, it identifies only a subset of contending flows: it misses flows that traverse two hop neighbors of a node without traversing its one hop neighbors. The local drop probability of each node is obtained using its channel bandwidth usage and, then, packets are dropped accordingly. The overall drop probability will realize the calculated drop probability on the whole neighborhood queue.

The NRED mechanism to regulate the traffic rates on the flows is complex, since it involves estimating a neighborhood queue size and uses RED-style marking on packets in the queue. NRED and LRED have an important disadvantage, since they are intimately tied to a particular queue management technique (RED) and require special hardware for channel monitoring. Both NRED and LRED are active queue management policies.

EWCCP [61] identifies the set of flows that share the channel capacity with flows passing through a congested node. EWCCP is based on a cross layer design. A EWCCP system is composed of end systems and intermediate nodes. The intermediate nodes are responsible for giving congestion feedback. The congestion feedback is based on the size of the neighborhood queue. When congestion is detected, different flows get a portion of negative feedback proportionally to their sending rate. However, if the network is not congested, every flow will get a same amount of positive feedback. This simulates the AIMD behavior of TCP. A EWCCP sender maintains a congestion window, as TCP, that controls the maximum number of packets that are allowed to be sent. Every data packet has a

special congestion header that can be used by intermediate nodes. The congestion header contains end-to-end congestion control information, used by senders and receivers, and link control information, used by intermediate nodes. Intermediate nodes detect congestion if the size of the neighborhood queue is larger than a threshold.

EWCCP, as TCP, lacks the maximization of the network utilization to achieve the highest minimum rate possible. This postulate becomes an issue when we are dealing with more dynamic wireless environments.

It must be noticed that ATCP, EWCCP and WCP are examples of clean slate design of congestion control mechanisms over wireless networks, that only rely on the AIMD process of TCP.

2.3.3 Rate Control Based Schemes

Recent efforts to design better congestion control mechanisms have led to the origin of several explicit-feedback congestion control methods. These methods solicit active multi-byte feedback from the routers to the end-hosts, delivering a precise and timely congestion signal that is used to accurately adjust flow sending rates. Therefore, they aim to achieve faster convergence, smaller packet loss rate, high link utilization and better fairness between flows. Examples of these congestion control mechanisms that rely on network interaction are the eXplicit Control Protocol (XCP) [15] and the Rate Control Protocol (RCP) [16]. Their main purpose is to generalize explicit congestion notification, where nodes inform each other about the degree of congestion, being a clear alternative to other congestion control approaches.

2.3.3.1 XCP Overview

The eXplicit Control Protocol (XCP) [15] takes a different approach than TCP to congestion control. It assumes that the network consists of routers capable of calculating the current network load, and thereby letting the sender know how much bandwidth is available to use in the network. By letting the network give more information back to the sender, XCP tries to prevent congestion and packet drops. The use of packet drops as a signal of network congestion is inaccurate and slow. The observation that a packet loss is a poor way to signal congestion is the basis for the XCP protocol.

XCP was designed to extract congestion information directly from routers. According to [15], XCP achieves fairness, maximum link utilization and efficient use of bandwidth. XCP is also scalable, as per-flow congestion state is carried in packets. However, XCP has its disadvantages: it is more difficult to deploy, since changes need to be made in all routers and end-systems in the network. A XCP network is composed of XCP sender hosts, receiver hosts and intermediate nodes where queuing from the sender to the receiver occurs. The intermediate nodes are usually routers but, with the networking equipment developments, they can also be link-layer switches containing packet queues. XCP uses a feedback mechanism to inform the sender about the best network conditions, that is, the maximum throughput. This feedback is accomplished by the use of a congestion header in

each packet sent. Along the path, intermediate nodes update the congestion header. When the packet reaches the receiver, it copies the network information obtained from the last intermediate router into outbound packets of the same flow (normally acknowledgment packets). The congestion header contains the following information: (1) senders RTT current estimation; (2) senders current throughput or sending rate; (3) delta throughput which is the network's allocated change in throughput, calculated and updated by the routers; (4) reverse feedback, which is the delta throughput of a packet that reaches the receiver - this value is returned to the sender, for example, in an acknowledgment packet. Bottleneck routers are the only ones that calculate re-allocation capacity for a specific flow.

XCP considers that it is needed a more precise feedback mechanism than the one in use by TCP, in order to get a less oscillatory protocol. As the feedback delay increases with high RTT, the protocol needs to take this feedback delay into account, by having the sender change its sending rate more sporadically. The important question is how the protocol should adapt to changing feedback delay in order to achieve stability even when the feedback delay gets very high. XCP will automatically slow down its adjustment rate of the sending speed when the feedback delays (i.e the RTT) increase. This adaptation to increased network delay prevents the protocol from becoming unstable and oscillating, in contrast to TCP [64].

Another important aspect of XCP, in contrast to TCP, is that it decouples flow control from utilization control. This decoupling has multiple benefits. First, it is possible to specify what is considered as a "fair" sharing of bandwidth between multiple flows. This allows for service differentiation using schemes that are either too aggressive or too weak to be used for controlling congestion. It also allows XCP to use a much more aggressive utilization control algorithm, such as the Multiplicative Increase, Multiplicative Decrease (MIMD) algorithm. This leads to allocating much faster the available bandwidth. Accordingly to the available bandwidth, XCP will try to proportionally allocate bandwidth (Multiplicative Increase) and will, equally, reduce proportionally if it finds that excessive bandwidth is being used (Multiplicative Decrease). XCP, as opposed to TCP, distinguishes bandwidth allocation and per flow allocation. This is possible because the way bandwidth is distributed between different flows is not dependent on how much bandwidth XCP is willing to distribute. XCP uses an AIMD strategy to achieve fairness between flows, thus, being TCP friendly.

The XCP protocol creates a new protocol layer between the IP and Transport layers; it also introduces a new header in packets. This header, the congestion header, has a size of 20 bytes and is placed between the IP header and the TCP header. The XCP routers do not keep any state information about each flow, but they obtain feedback values on a per packet basis. As the number of flows in a router is an unknown and fast changing parameter, the congestion control mechanism should not be dependent on it. This allows for quite simple implementations in routers, and makes the protocol more scalable.

Routers determine if it is allowed any change in throughput, as specified by the sender, by checking its aggregated load. If the system is not over-utilized, the router will allow the sender to use the bandwidth as requested, but taking into consideration the router's capacity. If, for any reason, the router has no available capacity, the request is reduced

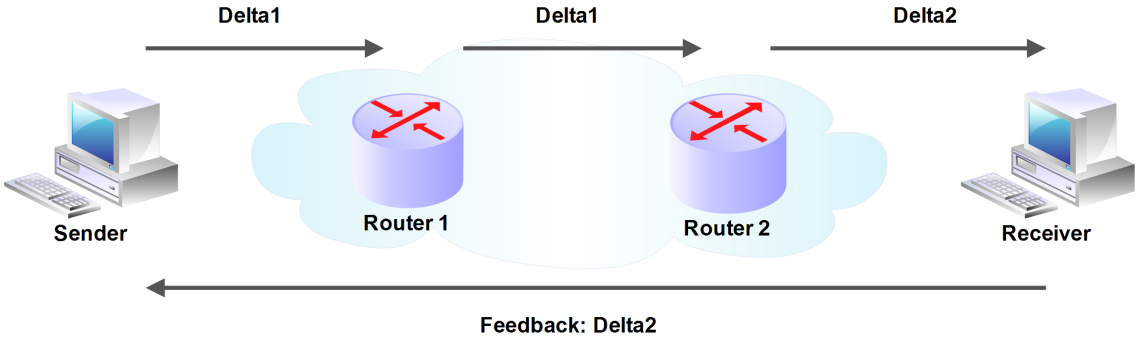


Figure 2.4: XCP operation.

to the limits of the router's available capacity. At the end of the connection, the receiver copies the allowed throughput into the specific reverse feedback header field. The sender will then know the throughput to use for the next packet. In the case that the XCP enabled router is experiencing congestion, it can reduce the allowed throughput from the sender.

Figure 2.4 shows a basic XCP operation. The sender tries to increase the current congestion window by $Delta1$, and it signals this request in the XCP congestion header. The next router in the path analyzes and forwards the packet to the other router. Since there is enough capacity to deal with the request, the router does not modify the header. The following router considers that $Delta1$ increase is excessive and modifies the congestion header, replacing $Delta1$ with $Delta2$, the maximum allowed throughput change for this particular flow, where $Delta2$ is smaller than $Delta1$. The receiver copies $Delta2$ and returns it to the sender as feedback, and then, the sender proceeds to adjust its congestion window. In this case, the second router is considered the bottleneck in the path.

To determine the $Delta1$ value, the sender needs to know its current throughput (available bandwidth) and its desired throughput. The $Delta1$ value can, thus, be obtained through the following expression:

$$Delta1 = \frac{C - TH}{TH \times \frac{T_{RTT}}{M}} \tag{2.18}$$

where C is the link capacity or the maximum interface speed, TH is the current sending throughput or link available bandwidth at the sender, T_{RTT} is the sender estimate of the RTT and M is the maximum segment size.

As $Delta2$ is returned to the sender by the receiver in the reverse feedback field of the XCP message, the sender adjusts the congestion window (CW_{Sender}) by:

$$CW_{Sender} = max(CW_{Sender} + Delta2 \times \frac{T_{RTT}}{1000}, M) \tag{2.19}$$

This will set the minimum value of CW_{Sender} to M. As stated in [65], a bad implementation of the window algorithm can lead to extremely poor overall performance, known as

silly window syndrome. The *silly window syndrome* appears when the minimum CW_{Sender} becomes too small to allow the sending of any data at all. To prevent CW_{Sender} to become 0, it is always set to be at least M bytes large.

The calculation of the bandwidth adjustment required to a certain XCP flow is performed by two algorithms on each intermediate node or router: the efficiency controller algorithm and the fairness controller algorithm. The efficiency algorithm is concerned to maximize the aggregated throughput without causing unnecessary or unexpected packet drops. This algorithm does not take into consideration the way any change in the aggregated throughput is distributed among the flows. This is the job of the fairness algorithm. The efficiency algorithm periodically (every T seconds) calculates the amount of bandwidth, that is the aggregated feedback (F) that will be distributed among all flows during the next T seconds and that maximizes the throughput:

$$F = \alpha \times (C - AB) - \beta \times \frac{q}{\bar{T_{RTT}}} \tag{2.20}$$

where C is the capacity of the link, AB is the bandwidth actually used during the last period T, and q is the persistent queue or, in other words, the minimum queue length observed during the last T seconds. $\bar{T_{RTT}}$ is usually set to be the average RTT of the flows traversing this queue; α and β are constants.

Then, the fairness algorithm makes sure that each packet in each flow traversing the node/router receives its fair share of any bandwidth feedback. The formulas used by this algorithm are not correlated to the efficiency algorithm. The algorithm evaluates F, and if $F > 0$, the node is not heavily utilized and the fairness algorithm will increase the throughput of all flows with the same amount, regardless of the previous bandwidth usage, having as result a relatively higher increase in throughput for flows running at low bandwidth. If $F < 0$, the router is being heavily used and the algorithm will decrease each flow's throughput proportionally to its current throughput. This behavior is very similar to the same principals of TCP's AIMD.

To enable XCP operation in the network, there needs to be at least one XCP aware router along the flow's path in the network. If the routers are not all XCP compliant, the protocol will not work optimally - but to some extent it will still work. If no XCP routers are located between two XCP hosts, the sender will just send data as fast as possible, since no XCP router will reduce the sender's request for bandwidth. This would be comparable to using TCP without any congestion control, as the XCP protocol overrides TCPs congestion control scheme.

XCP has three important properties that provide larger efficiency in lossy networks such as wireless networks: the explicit feedback from the network, the efficiency controller, and the feedback filter. XCP was designed to keep queue sizes to a minimum, reducing losses due to oversized queues, and thus, keeping the link utilization to a maximum. XCP is also able compute the increase or decrease in the sender's window based on several parameters, one of which is the currently available bandwidth. By making the feedback proportional to the available bandwidth, XCP can quickly adapt to changes in the environment. This would allow quick convergence to full efficiency whereas TCP exponentially converges to

efficiency during slow start, but additively elsewhere. This is made possible through the congestion header, which has a field for the sender's current rate and a field for the network to provide this feedback by increasing or decreasing the window size.

Due to the difficulty in estimating correctly link capacity and due to the presence of capacity errors in shared-access media, such as IEEE 802.11, some modifications to the calculation of F were proposed in XCP-b(blind) [22]. XCP-b is a XCP based congestion control mechanism that tries to extend XCP for shared-access, multi-rate wireless networks by calculating, using very complex heuristics, the available bandwidth of the wireless channel. XCP-b uses indirect parameters such as queue sizes and number of link layer retransmissions to obtain the desired measurements. In XCP-b, the spare bandwidth is measured from variations of the persistent queue. This, of course, is an important issue as the queue controller can only effectively measure queue variations when the medium is being fully utilized. In wireless environments with few nodes and less mobility, XCP-b outperforms other wireless transmission protocols in terms of stability, fairness and convergence.

The Wireless XCP (WXCP) [66] is also a XCP based congestion control mechanism. In WXCP, the sender regulates the transmission rate from the explicit feedback of bottleneck routers, not estimating nor probing for available bandwidth. WXCP uses calculations to implement active queue management without keeping per flow information. WXCP is also considered a segment-based congestion control protocol that operates within the wireless network, and enforces congestion control to all flows in TCP over the wireless segment. WXCP congestion metrics are available bandwidth, interface queue and average link layer retransmissions. WXCP considers that information of less available bandwidth represents congestion. Available bandwidth in WXCP is obtained by local observation. Interface queue is also used for congestion control: when the input traffic rate is greater than the output rate, packets start to be buffered in the interface queue and the length of the queue increases. When the queue is full, further packets coming to the queue are dropped. TCP uses this event to infer the existence of congestion in the network. Finally, the average link layer retransmissions are used to detect the degree of self-interference that is then used for congestion control. Self-interference exists when a flow sends too many packets in the network, resulting in an increase of the transmission delay and less throughput. By adjusting the transmission rate, a flow can change the degree of self-interference.

An important disadvantage of WXCP is the fact that it relies on TCP's information to implement active queue management. Moreover, self-interference is not the only interference that must be accounted in a wireless medium: interference due to multi-path reflection and signals from other kind of sources, attenuation, path dispersion and route failures must also be considered.

2.3.3.2 RCP Overview

The Rate Control Protocol (RCP) [16] is part of the 100x100 clean state project [67]. The mission of this project is to create blueprints for a network that goes beyond today's Internet [67]. RCP, similarly to XCP, is a congestion control algorithm. The main goal of RCP is to deliver fast flow-completion times. RCP was also designed having in mind

typical flows of typical users in today's Internet. RCP aims to improve web users flows, distributed computing and distributed file-systems, decreasing their transfer time. RCP is, as XCP, a router assisted congestion control algorithm that emulates process sharing.

RCP uses the same feedback principle of XCP and tries to emulate a processor sharing. However, it uses a different approach. Routers along the path do not determine incremental changes to the end-system's throughput, but determine the available capacity and the rate at which the end-system should operate. Unlike XCP, which signals rate changes, RCP always reports the target sending rate to the end-systems. Furthermore, the same rate is signaled to all flows bottlenecked at the respective link. With this operation mode, RCP tries to increase fairness and to simplify the underlying algorithms. The proposed packet header only contains the RTT and Feedback fields as well as a Rate field with the allowed rate at the most congested link on the path, instead of the *Delta* field in XCP.

The RCP congestion control mechanism can be summarized as follows:

1. Every router/intermediate node maintains a single fair-share of the available rate ($R(t)$). $R(t)$ is offered to all flows and is updated once every RTT;

2. Every packet header includes a rate field (R_p). If a packet is received and $R(t)$ is less than R_p, then R_p is updated with the $R(t)$ value, i.e $R_p \leftarrow R(t)$; otherwise R_p keeps its value. The receiver is just responsible to copy R_p into the ACK packets, thus, notifying the sender of the minimum possible rate;

3. The sender sends packets at rate R_p;

4. Each router/intermediate node updates its local available $R(t)$ value.

To determine the available rate, RCP relies on the router information. If that information is correct and if there are no delays between the link and the source, the available rate is simply:

$$R\left(t\right) = \frac{C}{N\left(t\right)} \tag{2.21}$$

where $R\left(t\right)$ is the given rate out of the flows, $N\left(t\right)$ is the number of ongoing flows and C is the link capacity. However, as there is feedback delay and it is difficult to know $\frac{C}{N(t)}$, an adaptive algorithm is used that updates the rate assigned to the flows, allowing to simulate processor sharing in the presence of feedback delay and not knowing the number of flows. For this processor sharing approximation, RCP determines the available rate by:

$$R\left(t\right) = R\left(t - d_0\right) + \frac{\left[\alpha \cdot \left(C - y\left(t\right)\right) - \beta \cdot \frac{q(t)}{d_0}\right]}{N\left(t\right)} \tag{2.22}$$

where d_0 is a moving average of the RTT measured across all flows, $R\left(t - d_0\right)$ is the last updated rate, C is the link capacity, $y\left(t\right)$ is the measured input traffic rate during the last update interval (d_0 in this case), $q\left(t\right)$ is the instantaneous queue size, $N\left(t\right)$ is the router's

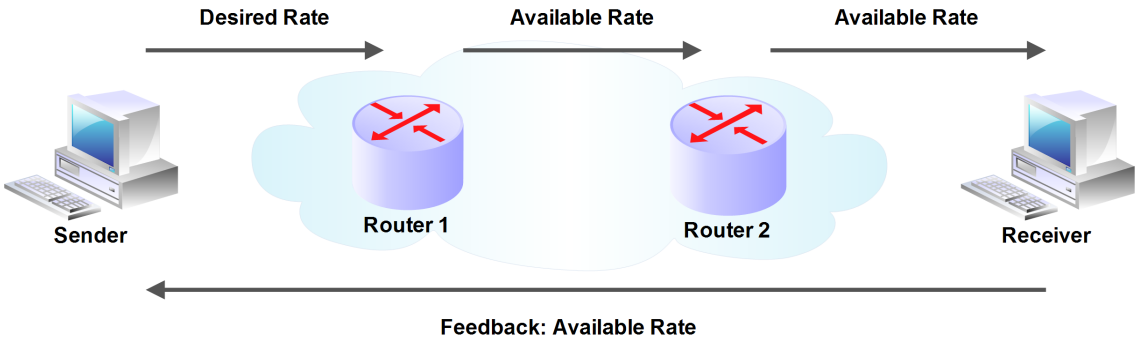


Figure 2.5: RCP operation.

estimate of the number of ongoing flows (i.e., number of flows actively sending traffic) at time t, and α, β are parameters chosen for stability and performance.

Figure 2.5 shows a basic RCP operation. In the beginning of the operation, the sender sets the desired rate with an infinite value. The next router calculates the available rate and overrides the rate value in the congestion header. This value is then compared by the final router. If the value is smaller then the available rate, the router does not change the value in the congestion header. The rate value is only updated if the available rate has a smaller value. Finally, the receiver feedbacks the rate value in the acknowledgement packet. If the flow lasts longer than one RTT, the subsequent rates are piggybacked on the data and acknowledgment packets.

[16] and [68] show that RCP can be stable under broad operating conditions, and its performance is independent of the distribution of flow-size and RTT. Experimental simulations show that RCP allows a sender to pick a fast starting rate and adapt quickly to network conditions, and thus, RCP achieves completion times one order of magnitude faster than TCP. The inter-RTT fairness of RCP is high as the fast convergence of flows is ensured due to the fair-share rate at every router. In addition, RCP allocates bandwidth quickly to new arriving flows. Also, the work in [69] studies the impact of buffer sizes on RCP. This subject is of importance since packet bursts of explicit congestion control protocols (e.g., due to sudden increase in sending rates) require large buffer sizes.

Both studies show that RCP is particularly well suited for traffic with bursts: since bandwidth allocation is instantaneous, small transfer such as web pages take less time to be transmitted with RCP than using TCP or XCP. Such dynamic behavior, however, comes at the cost of increased jitter, as queues oscillate to compensate the variation of flows over the network.

In high dynamic wireless networks, the increased jitter can cause performance degradation. However, the use of RCP explicit congestion notification can keep a wireless link fully utilized, thus improving network performance. Another important aspect of RCP in wireless networks is the correct evaluation of the available rate. This is a major issue, as RCP heuristics do not rely in information from other layers, considering only transport

layer information in the rate calculation, which is not very accurate.

Taking into account RCP main properties, it has been proposed the Wireless RCP (WRCP) [70]. WRCP uses explicit feedback based on capacity information to achieve a max-min fair rate allocation over a collection tree. In WRCP, a receiver capacity model is applied. This model associates capacities with nodes instead of links. The receiver model is also used to develop and implement the explicit and distributed rate based congestion control protocol for wireless sensor networks. WRCP needs that each node has the following information: a list of its neighbors, the total number of senders, the total number of neighborhood descendants, a relationship between the receiver available capacity and the number of broadcasts, the number of its own descendants and the total number of the node's flows. Each WRCP node determines the per flow rate that it can support at the end of a periodic interval. The per flow rate can be obtained by first finding the total bandwidth consumed in that periodic interval. To calculate the amount of consumed bandwidth, it is necessary to know the rate at which packets are transmitted (obtained locally), and the transmission rates of all the neighbors in the interval (explicitly learned from neighbors). Then, the node estimates its available capacity by subtracting the bandwidth consumed from its receiver capacity and, finally, the per flow capacity/rate of the node is obtained dividing the available capacity by the total number of consumers. The results obtained are then exchanged between neighborhood nodes.

WRCP has only been tested, through simulation, in wireless sensor networks (WSN). The most important drawback of WRCP is the introduction of overhead in the network; moreover, it does not consider link capacities, but nodes capacity estimations which are usually over-estimated.

2.3.4 Hybrid Congestion Control Schemes

As an attempt to benefit from the best of the two worlds, new approaches based on both AIMD and rate control schemes, normally denoted as hybrid schemes, have been proposed. TCP with Adaptive Pacing (TCP-AP) [20], The Cooperative Neighborhood Airtime-limiting (CNA) [71], HOP [72], WCPCap [21] and EZ-Flow [73] are some examples of hybrid mechanisms designed for wireless networks.

TCP-AP [20] is a hybrid scheme between a pure rate based transmission control, and TCP's use of the congestion window to trigger new data packets to be sent into the network. A TCP sender adaptively sets its transmission rate using an estimate of the current 4-hop propagation delay, and the coefficient of variation of recently measured round-trip times. The 4-hop propagation delay describes the time elapsed between transmitting a TCP packet by the TCP source node and receiving the packet at the node, which lies 4 hops apart from the source node along the path to the destination. In TCP-AP, congestion detection is done proactively by using RTT fluctuations to detect path link contention, while congestion control is done by pacing the transmission based on the contention measurement and the delay until the sender can send a new packet. The delay is obtained using RTT, link capacity and the number of hops informed by the routing layer. TCP-AP uses two parameters for the 4-hop propagation delay estimation: α that represents the averaging

weight in the exponentially weighted moving average, and the number of recent RTT samples to be taken into consideration.

TCP-AP was developed having into consideration that it should just be a sender modification, to be inter-operable with any valid TCP implementation and/or modification. This allows an incremental deployment: it should use the knowledge of how traffic is forwarded in multi-hop wireless environments to avoid congestion, and should be able to reduce losses and retransmissions.

Based on the results of [74], the authors of TCP-AP used the 4-hop propagation delay (FHD). TCP-AP sender calculates the FHD using the RTT estimation and the number of hops of the connection. The RTT is composed of the sum of the delay experienced by the data packet on the way from the sender to the receiver, and the delay experienced by the ACK packet sent from the receiver to the sender. Each of these delays comprise the time to forward the packet over h hops, where each forwarding requires a queuing delay t_q and transmission delays t_{data} and t_{ACK}, respectively. The measured RTT is then:

$$RTT = h(t_q + t_{data}) + h(t_q + t_{ACK}) \tag{2.23}$$

Being,

$$t_{data} = \frac{S_{data}}{b} \tag{2.24}$$

and

$$t_{ACK} = \frac{S_{ACK}}{b} \tag{2.25}$$

where b is the wireless interface bandwidth, S_{data} is the size of the TCP data packet, and S_{ACK} is the size of the TCP ACK packet t_q can then be obtained by:

$$t_q = \frac{1}{2}(\frac{RTT}{h} - \frac{S_{data} - S_{ACK}}{b}) \tag{2.26}$$

FHD is, therefore, estimated by:

$$FHD = 4(t_q + \frac{S_{data}}{b}) = 2(\frac{RTT}{h} + \frac{S_{data} - S_{ACK}}{b}) \tag{2.27}$$

TCP-AP uses also the estimated link-layer contention. The authors of [20] proposed the coefficient of variation of recently measured RTT (COV_{RTT}) as a measure of contention. With this mechanism, the transmission rate depends on the FHD index and on the link-layer contention. The coefficient of variation of recently measured proposed is obtained by:

$$COV_{RTT} = \frac{\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(RTT_i - R\bar{T}T)^2}}{R\bar{T}T} \tag{2.28}$$

where N is the number of RTT samples, $R\bar{T}T$ is the mean value of the RTT samples, and RTT_i is the value of the *i-th* RTT sample.

The evaluation presented in [20] shows that this new proposal obtains good results in static wireless networks. However, an important drawback is that it is not clear how the pacing mechanism works in a hybrid mobile network with wired links on the path, or where there is only a single wireless hop. Another major issue of TCP-AP is the lack of accuracy when obtaining the link capacity; moreover, it does not take into account node available bandwidth contention along the wireless path.

CNA [71] explicitly allocates the channel resources, but provides only imprecise feedback to the source. In CNA each link is assigned an airtime limit. In any time interval T, each link can only use up to $\frac{1}{6}$ of T for its transmission. In a CNA system, if there are different links with different rates, they will be allowed to transmit different amounts of traffic, thus, guaranteeing a fair goodput. Each retransmission is then considered for airtime usage. TCP is used by CNA to obtain congestion control. When a TCP sender communicates in a airtime limited link, the TCP congestion control mechanism adapts to the most airtime constrained link.

CNA achieves efficient airtime allocation by distributing available airtime within a wireless neighborhood, then monitoring the air utilization and dynamically redistributing unused airtime to improve overall airtime usage. The authors of CNA claim that it achieves transparency, low overhead and responsiveness. However, CNA considers airtime to be the fraction of the time that a wireless link can occupy the shared channel; it does not consider the time a node is waiting to transmit.

HOP [72] is a clean-slate design of hop-by-hop congestion control. HOP tries to use reliable per-hop block transfer as a building block. HOP protocol consists of six components: reliable per-hop block transfer, virtual retransmissions for end-to-end reliability, back-pressure congestion control, handling routing partitions, acknowledgement *withholding* to avoid the hidden terminal, and a per-node packet scheduler. A block is the reliable transmission unit in HOP. Virtual transmissions are used, together with networking cache, to reduce overhead of retransmitting large blocks of data. Back-pressure congestion control of HOP is done by having each node monitoring the difference between the number of blocks received and the number of reliably transmitted blocks to its next-hop. HOP limits this difference to a small fixed value, implementing it with no additional overhead. As HOP transfers a block in a hop-by-hop way, it can continue to transmit even when the network is partitioned. HOP uses a novel ACK *withholding* technique to avoid the hidden terminals problem. A receiver only acknowledges one packet at a time, and withholds acknowledgement to other concurrent packets until the outstanding block has completed. This allows the receiver to ensure that it is only receiving one block from a sender at any given time. Once the block has been completely received, the receiver transmits an ACK backward to one of the other waiting senders, which starts transmitting its block.

HOP is referred by its authors as: fast, because it eliminates many sources of overhead as well as noisy end-to-end rate control; robust to partitions and route changes, because of hop-by-hop control as well as in-network caching; and simple, because it obviates complex end-to-end rate control as well as complex interactions between the transport and link layers.

EZ-Flow [73] is a hop-by-hop congestion-control mechanism that operates at the MAC

layer. EZ-Flow was developed to be fully backward-compatible with the existing IEEE 802.11 deployments, and it works without any form of message passing. EZ-Flow takes advantage of the broadcast nature of the wireless medium in order to passively derive the queue size at the next-hop node. This information is then used by each node to adapt accordingly its channel access probability, through the contention window parameter of IEEE 802.11. Each node in EZ-Flow maintains two independent queues: one for its own traffic and the other for the forwarded traffic. Furthermore, a node that has multiple successors should maintain one queue per successor (two if it acts as source and relay). This allows EZ-Flow to perform better as it can adapt the channel access probability per successor. To achieve congestion control, EZ-Flow uses two different modules: a buffer occupancy estimator (BOE) module that derives the queue status of the successor node along a flow; and a channel access adaptation (CAA) module that uses the information from the BOE to adapt the channel access probability through the congestion window.

The main issue of EZ-Flow is its scalability: if a node has a significant number of successors, the node has to maintain a large number of queues, introducing overhead and reducing network performance.

WCPCap [21] is a distributed rate controller that estimates the available capacity within each neighborhood, and divides this capacity to contending flows. WCPCap, as XCP and RCP, also uses explicit feedback. WCPCap uses the achievable rate estimation to estimate the achievable bandwidth, and then gives the rate feedback to the sources. WCPCap assumes that all control message broadcasts are exchanged instantaneously and without loss, and that each node has complete information about the entire network instead of just its neighborhood. Each node is aware of the data rate at each link in the network and the global network topology. As stated in [21], the last assumption is needed because residual capacity at a link depends on the global topology and not merely on the local neighborhood topology. WCPCap is then able to obtain an approximate value of the residual capacity, while ideal WCPCap will obtain an exact value of the residual capacity.

It is evident that considering wireless congestion collectively over a neighborhood of a link is essential to any future design of wireless congestion control. WCPCap uses a sophisticated stochastic model to estimate the achievable rate region, given packet loss rates, topology, and flow information. It then allocates the achievable capacity fairly across flows, sending feedback to the sources.

2.3.5 Summary

Congestion control and transmission control in wireless networks pose many challenges for the design and implementation of different congestion control protocols. TCP congestion control mechanisms are proved to not react well to the dynamic behavior of wireless nodes. TCP assumes that the probability of a lost packet is higher than the one of a corrupted packet, which is not true in wireless networks.

Several congestion control mechanisms were proposed to enhance TCP's behavior in a wireless environment. The base approaches concentrate on improving TCP's throughput by freezing TCP's congestion control algorithm during link-failure induced losses, especially

when route changes occur. These TCP developments differ in the manner in which losses are identified and notified to the sender, and in their details of freezing TCP's congestion control algorithm. Even though these schemes do not recognize the need of congestion detection and signaling over a neighborhood, their congestion metric implicitly takes some degree of neighborhood congestion into account.

Other proposals, specially developed for wireless environments, use a rate based equation to control the transmission, thus, achieving congestion control. However, they do not correctly evaluate the available rate, making sometimes the assumption that all links have equal rate. Other schemes use analytical models to estimate the available bandwidth and link capacity within each communication neighborhood, and distribute the capacity to contending flows, using a distributed rate controller. They lack of sender and receiver node cooperation.

New congestion control techniques identify the importance of having intermediate nodes to explicitly and precisely send feedback to the sources. They use explicit feedback based on capacity information to achieve a max-min fair rate allocation over a collection tree. However, the capacity and available bandwidth estimation are based in analytical models, and they do not use real-time efficient and accurate information for congestion control.

It is, thus, important to define new congestion control mechanisms in wireless networks that are able to correctly infer and use link capacity and available bandwidth to improve congestion control. These mechanisms should differentiate between poor channel conditions and collisions as the source of transmission failures, and consequently invoke proper congestion control, reducing throughput degradation and improving overall network performance.

Based on the previous considerations and analysis, it is recommended that a next generation congestion control algorithm shall have the following requirements: (1) consider effective, accurate and real-time available bandwidth and link capacity for rate control; (2) react and consider the wireless behavior effect in congestion control; (3) consider node and network interaction for a more reliable congestion control; (4) consider TCP friendliness and, whenever possible, TCP compatibility; (5) use effectively estimation information to infer how close the system is to saturation, and adapt the transmission rate according to the state the system is currently operating at.

Having in mind the previous considerations, we then propose the extension of some existing congestion control mechanisms, namely XCP, RCP and TCP-AP as described in Chapters 5 and 6.

2.4 Collision Probability

A wireless network is performance dependent on its medium access control scheme [75]. In CSMA-CA, a node is allowed to transmit only if it determines the medium to be idle. CSMA-CA, however, cannot prevent packet collisions caused by nodes that are located within the transmission range of the receiver, but not in the transmission range of both receivers (hidden nodes problem [76]). To prevent DATA packet collisions due to

hidden nodes, IEEE 802.11 supports the RTS/CTS mechanism [5]. However, it must be noticed that in ad-hoc networks, this assumption does not hold in general. Neighboring nodes are often unable to receive the control packets because they are masked by on-going transmissions from other nodes near them. [77] states that, if nodes are mobile, then a node that did not hear an RTS or CTS may migrate into the footprint of a receiver and destroy a DATA packet with its own transmission. The probability of such a scenario increases with the mobility of the nodes. [78] also shows that in an ad-hoc network, a successful exchange of RTS and CTS is not sufficient to prevent DATA packet collisions.

Since the development of the IEEE 802.11 standard, significant research has been conducted trying to define new models that effectively deal with the collisions problem of the shared medium. Some contributions linearize the non-linear expression for collision probability and use a simplified model to determine the global collision probability.

Studies like [79], [80], [81] also address the effect of collisions in wireless networks, considering that the MAC IEEE 802.11 protocol cannot prevent hidden node collisions from happening.

Previous works - [82], [78], [83] - tried to infer collision probabilities through extensive mathematical formulas. These models assume that the channel is ideal, that there is a finite number of nodes, and that collision probability is uniform for all transmissions. The model refers the principles of the exponential backoff protocol and both RTS/CTS and basic access mechanisms. The model uses two discrete Markov processes to track the backoff and transmission attempts to infer collision probability. One of the processes is the backoff counter ($b(t)$). For each start of a backoff, $b(t)$ is decremented; when $b(t)$ is equal to zero, the station can transmit. When the transmission is finished, or even when the transmission was not successful, a new backoff value is assigned to $b(t)$.

As the size of the contention window from which the $b(t)$ values are obtained after collision depends on the transmission history of the station, another Markov process ($s(t)$) is responsible for keeping the transmission history. The value of $s(t)$ represent the number of times the current segment has been transmitted. This is referred to as backoff stage. After each successful retransmission, $s(t)$ is reset to the initial backoff stage. After any failed transmission, $s(t)$ is incremented, and when it reaches the maximum contention window, it will persist at that value until the transmission is successful.

The combination of the two processes defines a set of non-null one step transition probabilities, as described next:

$$\begin{cases} P\{i,k|i,k+1\} = 1 \; k\,\epsilon\,[0,W_i-2] \; i\epsilon[0,m] \\ P\{0,k|i,0\} = \frac{1-p}{W_0} \; k\,\epsilon\,[0,W_0-1] \; i\epsilon[0,m] \\ P\{i,k|i-1,0\} = \frac{p}{W_i} \; k\,\epsilon\,[0,W_i-1] \; i\epsilon[1,m] \\ P\{m,k|m,0\} = \frac{p}{W_m} \; k\,\epsilon\,[0,W_m-1] \end{cases} \tag{2.29}$$

where p is the probability of a transmission experiencing a collision, W_i is the contention window size in the backoff stage i, W_0 is the initial contention window, and m is the maximum contention window size.

Considering that the probability of a station transmitting (τ) in any backoff time slot

is uniform and defined by:

$$\tau = \sum_{i=0}^{m} b_{i,0} \tag{2.30}$$

Then, the collision probability p for a transmission in a network of n stations can be expressed as:

$$p = 1 - (1 - \tau)^{n-1} \tag{2.31}$$

The validation of this model against simulation results shows that it is quite accurate. However, it is also shown that this model is effectively accurate when the number of nodes is considerably large. This is due to the fact that the treatment of the backoff mechanism is not the most correct. In IEEE 802.11, backoff counters are decremented at the end of each idle backoff time; and upon backoff reaches zero, the station initiates transmission immediately. This procedure goes against the approximation that collision probability is uniform for each transmission attempt.

Other studies, like [84], use the size of TCP congestion window (CW) to control the collision probability. The increase of the CW decreases congestion probability. For every successful transmission, the IEEE 802.11 standard defines that the CW must be reseted to a minimum value (CW_{min}), which may result in collision bursts [85]. As a result of collisions, CW is increased, and if it is reset after a successful packet delivery, it increases the probability that the next packet to be transmitted will have a collision. This technique will waste link capacity as a result of the collision bursts. In [85] it is also presented an adaptive way of dealing with the CW reset, proportionally to the collision rate. These techniques are not very accurate and lack of performance.

In [86] it is introduced the concept of average conditional collision probability. Average conditional probability is calculated by modeling the evolution of contention zones during a contention period defined by a Markov chain. Using this approach, timescales can be decoupled and the simplicity of the model is preserved [83]. With such consideration, the model is then used to calculate collision probability.

Research in [87] and [88] has focused on the physical layer approach based on Signal to noise ratio (SNR) and Received signal strength indication (RSSI) measurements. It was shown that these approaches have a weak correlation with the MAC layer, being very dependent on the type and power consumption of the equipments.

In [89] it is proposed a new scheme to infer collision probability and to improve accuracy on available bandwidth estimation. It designed a per-link available bandwidth estimation method that uses an estimation of the collision probability on a link. It considers a scheme with the following postulate defined by [90]: *"collisions in the IEEE 802.11 networks occur before congestion, since packets are lost when the queue size in some nodes exceeds a limited value. This threshold is far away from the congestion limit, making the system unstable".*

2.4.1 Summary

The previously mentioned considerations show that, due to the large random topology and mobility in wireless networks, collision probability affects the network performance. Collision probability influences the determination of underlying parameters that are major factors of network performance, such as link capacity and available bandwidth. More often wireless networks need to accommodate multimedia traffic which is bursty and has diverse quality of service requirements; thus, collision probability is important to improve bandwidth allocation and to efficiently manage such traffic.

Therefore, it is evident that collision probability must be taken into consideration when improving congestion control and link capacity estimation techniques. It is important to know the extra time introduced when a node is waiting to transmit as a result of collisions. When a node wants to transmit and enters contention mode, due to collision detection, this will affect link evaluation, leading to over-estimated capacity, available bandwidth values and congestion control, resulting in low network performance. Without considering in-line packet collision estimations, throughput and delay will be affected: throughput will decrease and the delay will increase, which will increase the performance degradation of IEEE 802.11 networks.

2.5 Cross Layer Design

Network design has mainly been based on the TCP/IP reference model, that is a development of the ISO/OSI reference model. Those models have been extremely important in allowing protocol standardization as they facilitate an easy and clean protocol design, thus, enabling interoperability among different networks. They were inspired in the following rules:

- A new layer has to be created where a different abstraction is needed;
- each layer should perform a well defined function;
- information flow across layers should be minimized;
- the number of layers should be large enough such that different functions are in different layers;
- the function of each layer should be easily mapped on a protocol.

These rules recall that each layer has to appear as a closed container to all the others layers.

As each protocol resides inside a single layer needing only to realize the functions for which that layer is in charge, it allows an easy protocol design. Each protocol can be implemented independently of all others, as interaction with higher and lower layers is limited to the knowledge of the general input/output specifications provided by the interface used to connect with neighboring layers.

However, the recent need to reach network performance limits has showed that this closed layer approach has some implicit limitations. It is clear that, due to all the recent and complex networking functions, layering itself imposes barriers to network performance optimization and behavior. These assumptions make it clear that enhanced network performance can be reached with neighboring and non-neighboring layer interaction.

Moreover, recently, it became evident that a traditional layering design is not efficient for wireless links [91], due to the interaction of links through interference, that implies that a change in one link can induce changes in the capacities of all links in the surrounding area, and changes in the performance of flows that do not pass over the modified link. A more relevant and clear interaction among neighboring and non-neighboring layers seems, thus, to be one of the most important steps for network performance enhancement. It may even be required that a completely new layering model needs to be used to drive more efficient solutions.

The cross layer design concept in networking is not recent [92], but it still lacks a clear definition. The main reason for this is due to network structure complexity.

However, as stated by the numerous developments in wireless congestion control, there is an increase use of cross layer information exchange that is used to improve the transport layer functionality. The whole performance of the network can be increased if the transport layer operations are optimized. If the transport layer is aware of some key parameters of the underlying layers, this will allow to make smarter congestion control decisions. For instance, instead of measuring the available bandwidth and the link capacity as perceived at the transport layer (like many approaches do), the congestion control mechanisms that exist on that layer can be enhanced with the information on the raw rate - namely at the MAC layer - at which a transmission towards a specific node will be performed. To fully optimize wireless networks, it is important that information from the MAC layer, and if possible physical layer, can be taken into consideration. It is clear that knowledge has to be shared between layers to obtain the highest possible network performance and stability.

This has led to the development of some frameworks that, using cross layer information, try to quantify the problem of transport layer rate control: FAST TCP [93] is one of such examples. Other proposals like XCP and RCP were developed having into consideration the cross layer principles. While the adoption of these proposals in wired networks has faced some problems, specially due to the TCP nature of these networks, their use in wireless networks is more forthcoming.

A cross layer design can be achieved through different techniques. Different proposals have tried to enhance the well known ISO/OSI and TCP/IP layered models, aiming to improve network performance. One of the approaches is shown in Figure 2.6 [94], the Cross Layer Signaling Shortcuts (CLASS). It focus on specific interactions between layers, and it introduces the respective message functions. This proposal clearly identifies the interaction between the application and the MAC layer, which can be used to set the user's need according to the MAC bandwidth and vice-versa.

A more general concept refers to the definition of general methods for exchanging control messages between different layers, enhancing the existing layered architecture. Figure 2.7 presents the proposal in [95], called coordination plane, where the standard protocol stack

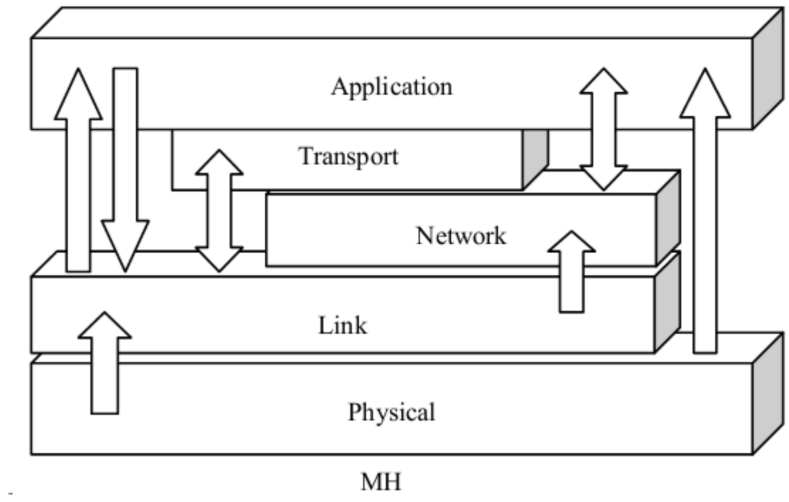


Figure 2.6: Cross Layer Signaling Shortcuts (CLASS) Architecture.

is modified by using transversal control planes. Each control plane is in charge of a different function and can interact with all layers in order to achieve its optimization goal. Thus, each plane acts both as a communication and a control plane.

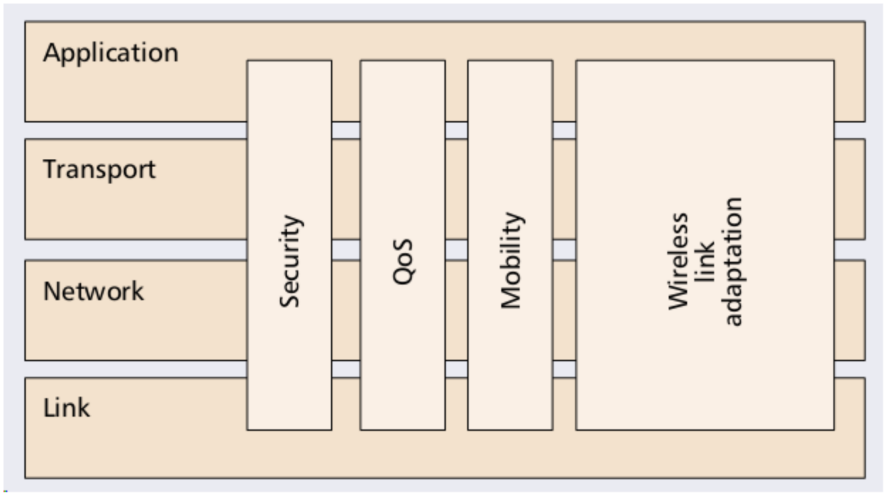


Figure 2.7: Coordination Plane Architecture.

Cross layer design can be achieved through layer triggers. Layer triggers are signals used to notify special events between protocols. One example is the so called Layer 2 (L2) triggers [96]. These are internal stack signals added between the Logical Link (LL) and IP layer, allowing to trigger changes in the wireless link to be efficiently detected and shared

among these two layers, thus the name L2.

Another technique is the EventHelix protocol design [97]. EventHelix is an open source streaming protocol that uses a standardized interface between layers. EventHelix allows a dynamic insertion of protocol layers from a stack. Another approach is the MobileMan [2] architecture. MobileMan main goal was to implement a system-wide cross layer design in a mobile network protocol stack using 802.11. Figure 2.8 shows the protocol design of MobileMan. MobileMan uses, for its layer communication, a shared database architecture. This architecture provides a set of get/insert methods into a database that is common to all stack layers. In the MobileMan proposal, all layers communicate with a single control plane, which is devoted to controlling all layers functions in a unified way, according to a high optimization criteria. The control plane, which becomes the core of the network node, can actually be used to create a new abstraction of the network functionalities. MobileMan main comprise its cross layer optimization for all network functions, and an improved local and global adaption and reduced overhead.

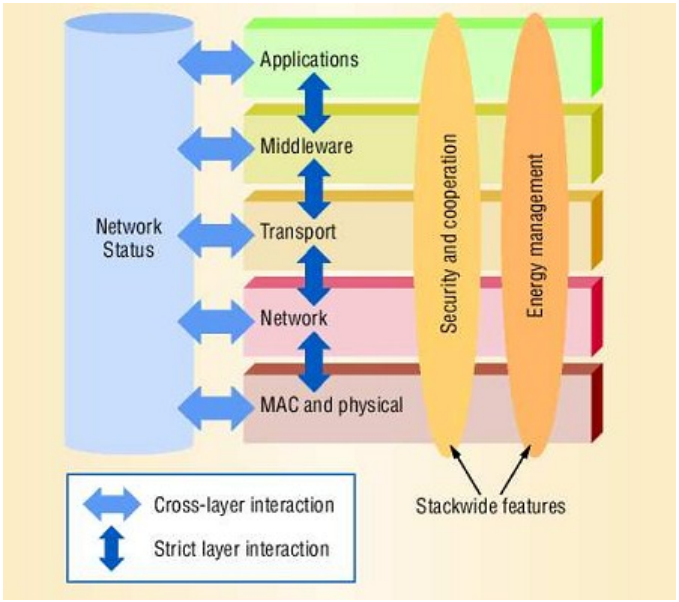


Figure 2.8: MobileMan Architecture. [2]

Several works propose the optimization of the wireless networks performance with cross layer design. Works like [98], [99] and [100] consider time-varying channels and provide stochastic optimization. The stochastic optimizations are based on the neighborhood queue backlog occupancy to make in-line decisions. The connection between stochastic optimization and the static optimization approaches are still being discussed, but those works clearly state that these stochastic measures are useful in wireless contexts. The main issue with these approaches is that they use very complex heuristics and indirect parameters.

An example of the need to break the layered structure is also represented by the interaction between transport layer and physical/MAC layer in a wireless system. In a wireless

link, the assumption of congestion control being detected by losses does not hold anymore, due to the high probability of a packet loss being caused by a channel impairment.

Research, like [101], [102], [103] and [104], focus their work on improving the transport layer behavior with a joint transport and PHY/MAC layer optimization. They consider the use of the underlying bandwidth to improve and optimize the way the transport layer performs congestion control decisions. They use in-line information and try to keep the process very simple and accurate.

It is clear that performance enhancement and optimization in wireless networks represents, still, an open research field where many issues remain unsolved. Being capable of knowing and acting simultaneously on many parameters is extremely important; thus, the concept of cross layer design is specially suited for this type of networks.

Using information from different layers can significantly improve network performance, allowing to use more efficiently and more fairly the medium and the network capabilities. However, the use of cross layer design can impose some difficulties and, if not clearly defined, it can become ineffective. It is, then, important that the use of a cross layer technique does not introduce excessive overhead in the network, and can guarantee a local and global network improvement.

Thus, in the following sections we will use the *MobileMan* cross layer architecture in the context of multi-hop wireless networks, to improve their congestion control performance, as one of the contributions of this Thesis.

2.6 Summary

This chapter introduced the concepts and provided the background information for the work developed and described in this Thesis. We described techniques and tools used to estimate available bandwidth and link capacity in both wired and wireless networks. We also described the most important proposals of congestion control mechanisms in the area of wireless and wired networks. The proposals were grouped according to their congestion control scheme - AIMD, rate control or hybrid schemes. We also introduced the problem of collision probability in the context of wireless networks, and the research that has been done in this area. Finally, we introduced the concept of cross layer design and described proposed solutions for the cross layer implementation.

All related work presented in this chapter shows that network performance in wireless networks is still an open research area where many issues are still unsolved and performance behavior is still not well understood. The need for performance improvements has surely to be done with layer interaction, as it is necessary to know different parameters from different network layers. Thus, a cross layer design model is specially suited for these types of networks. This model should allow some kind of high optimization criteria and should allow to define new network functionalities in a transparent and abstract manner.

It also became clear that parameters, like link capacity and available bandwidth from the MAC layer, can improve greatly network congestion control behavior and, thus, network performance. It is then advisable to use or define estimation mechanisms that use

information already present in the network, without introducing overhead or increasing network traffic. IdleGap is a recent estimation tool that gathers information without interfering in the network. However, this mechanism is using the *Data Rate* value obtained directly from the IEEE 802.11 header. This is not a very accurate value, making IdleGap to overestimate the available bandwidth estimation. It is then important to improve IdleGap estimation process, making the estimation more accurate and with node interaction, and obtaining effectively link capacity.

Congestion control is also an important area of research in wireless networks. New proposals have been defined explicitly for wireless environments. These proposals, however, use information obtained directly from the transport layer, which is not accurate and effective. Being link capacity and available bandwidth major factors that influence network congestion control, it is important to use this information for improving congestion control. Such information, inferred at the MAC layer, can be used directly by rate based congestion control mechanisms, or to improve congestion control mechanisms specially proposed for wireless environments.

As TCP is still the most used congestion control protocol, it is important to understand how it behaves against new congestion control mechanisms. New rate based congestion control mechanisms defined for wired networks clearly perform better than TCP. They use analytical models to obtain link capacity and, then, to adapt the transmission rate, improving congestion control. However, in wireless networks there is still lack of knowledge of their behavior against standard TCP and against wireless defined congestion control mechanisms. It is thus important to first evaluate these mechanisms and protocols in wireless environments, to obtain the most possible knowledge about these protocols. This knowledge will allow first to address practical issues and, then, will allow a better definition of new optimization and improved solutions in congestion control.

Chapter 3

Congestion Control Evaluation in Multi-Hop Wireless Networks

3.1 Introduction

Wireless mesh networks (WMNs) and wireless ad-hoc networks (usually called mobile ad-hoc networks - MANETs) are in a fast undergoing progress, since they are flexible, easy to deploy and very dynamic [6]. WMNs are comprised of two types of nodes: mesh routers and mesh clients. Other than the routing capability for gateway/bridge functions as in a conventional wireless router, a mesh router contains additional routing functions to support mesh networking. A MANET is established when a collection of wireless nodes organize themselves to constitute a wireless network without the use of any infrastructure, such as access points or base stations. In a MANET it is assumed that all nodes cooperate with each other to form a flexible and dynamic communication network. This type of cooperative model allows communication among mobile nodes also when no communication infrastructure exists in remote locations. Ad-hoc networks do not rely on dedicated routers for forwarding data packets, as each node uses routing functions to discover routes and forward packets.

When compared to conventional wireless networks, MANETs and WMNs are built based on the same principles and similar hardware and software platforms; however, both MANETs and WMNs increase and add many technical issues in congestion control, due specially, to the fact that correspondent clients are constantly moving and changing information with each other and mesh routers, in the case of WMNs, or between correspondent nodes in MANETs. As any other wireless network, MANETs and WMNs use air as the access media, which, being a shared medium, is more sensitive to interferences and to congestion.

The developed congestion protocols do not take into account the problems and particularities of wireless networks. The Transmission Control Protocol (TCP) [9], the most widely used congestion protocol in the Internet, was developed for wired networks. TCP uses the *Van Jacobson* [105] congestion control algorithms, which have been highly suc-

cessful over many orders of magnitude of Internet bandwidth and delay. However, with the increase of computer networks, the high Internet demand and the proliferation of wireless networks, TCP became unsuitable for highly dynamic environments. Some of these performance problems led to the development of new congestion control protocols. The eXplicit Control Protocol (XCP) [15] and the Rate Control Protocol (RCP) [16] are two of the most recent ones. They rely in network interaction for congestion control improvement, since they need intermediate nodes, such as routers, to work and interact to support the congestion control. In wired networks they increase efficiency in the congestion control. However, these protocols were developed taking only in consideration the characteristics of wired networks. Their efficiency relies in some of the wired network features that are different, or are not present, in wireless networks.

Due to the fast progress in the use of wireless networks, new congestion control schemes specially developed for these kinds of networks have been developed. Schemes, such as the Wireless Control Protocol (WCP) [21] and the TCP with Adaptive Pacing (TCP-AP) [20], try to reduce TCP problems in wireless environments and enhance TCP behavior. WCP and TCP-AP use on their main operating principles the Additive Increase Multiplicative Decrease (AIMD) process of TCP, updated with analytical models to infer wireless capacity and, then, the transmission rate is changed accordingly. Wireless congestion control schemes based in rate control protocols were also defined. One such example is XCP-blind (XCP) [22]. XCP-b also uses an analytical model and complex heuristics as its main operating model.

Many performance evaluations of TCP, XCP and RCP have been conducted, specially in wired networks, but, to the best of our knowledge, there is no known study that evaluates these congestion control protocols against each other in a wireless environment. Since these protocols have a large acceptation on the research field, and simultaneously, wireless networks are in undergoing rapid progress, it is important to evaluate how XCP and RCP behave in both mesh and ad-hoc wireless networks, as compared to TCP.

Some other evaluation efforts have only measured performance of the new wireless proposals in a static topology and against TCP. However, these efforts have largely focused on special topological configurations, and the evaluation of proposed solutions has mostly been performed in limited scenarios and with limited comparison protocols.

Motivated by the above observations, this chapter makes the following contributions. It presents a performance evaluation study on the effects of mobility on several congestion control protocols against TCP. Rate based protocols such as XCP, RCP and XCP-b, and AIMD based protocols such as WCP and TCP-AP are considered. Although defined as an hybrid mechanism, TCP-AP evaluation will be considered against AIMD based protocols as it also relies on AIMD operations. The main objective of the evaluation is to explore AIMD based and rate based protocols under certain network conditions to understand the advantages and disadvantages of them comparatively. Notably, the results show that TCP outperforms, in some situations, the behavior of the new rate based congestion control schemes, and its behavior is very close to the one obtained by some wireless specific congestion control techniques.

The rest of the chapter is organized as follows. In section 3.2 we describe the tools

used for the simulation, the parameters and the main scenarios. Section 3.3 describes the main ns-2 implementations used on the simulations, referring some important configuration parameters. Then, in section 3.4 we discuss the performance results of the considered protocols. Section 3.5 concludes the chapter and gives research directions.

3.2 Performance Evaluation Methodology

The congestion protocols evaluation is performed through the ns-2 [23] simulator with a wide set of topologies. The ns-2 simulator is well suited for the study and research of wireless networks and it also contains several functionalities required by the congestion control approaches. In the Medium Access Control (MAC) layer, we simulated IEEE 802.11 with RTS/CTS (Request-to-Send/ Clear-to-Send) enabled. All simulations last 300 seconds. The configured default transmission range is 250 meters, the default interference range is 500 meters, and the channel data rate is 11 Mbps. For the data transmissions, an FTP (File Transfer Protocol, is a standard network protocol used to copy a file from one host to another over a TCP-base network) application has been used on top of the wireless network, with packets of 1440 bytes. The buffer size is set to 100 packets per queue, and the MAC retry limit is set to its default value. All our simulations are conducted considering packet losses due to collisions. In the mobility scenarios, the ns-2 *setdest* tool is used. This tool generates a random node movement pattern. We configured *setdest* with a minimum speed of 10 m/s, a maximum speed of 30 m/s and a topology boundary of 1000x1000 meters. All results were obtained from ns-2 trace files, with the help of *trace2stats* scripts [106] adapted to our own needs. The simulations were repeated 30 times with different ns-2 seed values. The mean and 95% confidence intervals are presented in the results. The routing protocol used is the Destination-Sequence Distance-Vector (DSDV) [107].

Our evaluation uses the following metrics: average network throughput/goodput, the average end-to-end delay and the average number of received packets. The average throughput ($\bar{TH}$) of all flows in a network is computed as:

$$\bar{TH} = \frac{S_{received} \times 8}{T_{simulation}}(kbps) \tag{3.1}$$

where $S_{received}$ is the size of the received packets and $T_{simulation}$ represents the simulation time. The average throughput can be defined as the average data rate of a source sending packets and received by the receiver. The average end-to-end delay ($\bar{D}$), which refers to the time taken for a packet to be transmitted across a network from source to destination, is computed as:

$$\bar{D} = N \times [D_{trans} + D_{prop} + D_{proc}](s) \tag{3.2}$$

where D_{trans} is the transmission delay, D_{prop} is the propagation delay, D_{proc} is the processing delay and N is the number of flows. The combination of these metrics also allow us to infer the fairness of a protocol. Fairness is defined as the bandwidth sharing between flows of the same type.

The test scenarios simulated comprise various $N \times N$ grid mesh topologies and various ad-hoc topologies with different mobile nodes and flows. In the mesh grid scenarios, the horizontal and vertical distance of successive nodes is set to 200 meters, and the number of mesh nodes, N, varied from 5 to 16. In all mesh topologies, it is used a combination of 3, 4, 5, 6 and 7 mobile nodes. Figure 3.1 represents a mesh topology of 5 mesh nodes and 5 mobile nodes. The mobile nodes are simultaneously sources and sinks and are randomly distributed throughout the simulation area.

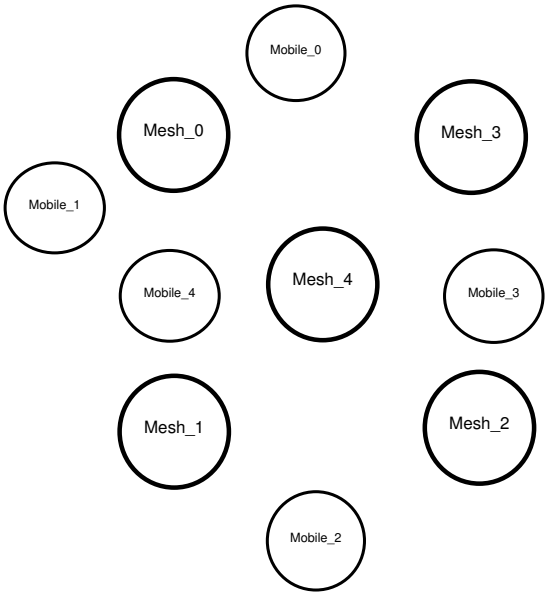


Figure 3.1: NxN Grid Mesh Topology (N=2).

The parameters of the mesh simulations are presented in Table 3.1.

The ad-hoc scenarios consist of mobile nodes moving around the workspace area. The scenarios are composed of 2^N mobile nodes, where N is defined as 3, 4, 5, 6, 7, 8 resulting in 8, 16, 32, 64, 128 and 256 mobile nodes. For each scenario, it was configured 2^{N-1} flows, i.e for each scenario there are 4, 8, 16, 32, 64 and 128 simultaneous flows. The flows are randomly generated and the sources and sinks are also randomly defined. Mobility was also dynamically generated through different seed values and nodes positions are also randomly generated. Figure 3.2 represents an ad-hoc scenario with 8 mobile nodes.

Table 3.2 shows the parameters used on the ad-hoc networks simulations.

3.3 NS-2 Implementations

This section presents some considerations regarding ns-2 implementations of the proposed evaluated protocols and mechanisms. Most of the protocols and mechanisms evaluated have a public available implementation.

Simulation Parameters	
Topology Area	1000m x 1000m
Simulation Time	300 sec.
Simulation repetition	30 times
Mesh Scenarios Number of Mobile Nodes	3, 4, 5, 6, 7
Mesh Scenarios Number of Mesh Fixed Nodes	5, 9, 12, 16
Mesh Scenarios Number of Flows	6, 8, 10, 12, 14
Mesh Nodes Position	Random
Path Loss Model	Two Ray
Mobility Model	Random Way Point
Maximum Movement Speed	30 m/s
Minimum Movement Speed	10 m/s
Mac layer	IEEE 802.11
Propagation Model	Two Ray Ground
Routing Protocol	DSDV

Table 3.1: Mesh Simulation Environment Parameters.

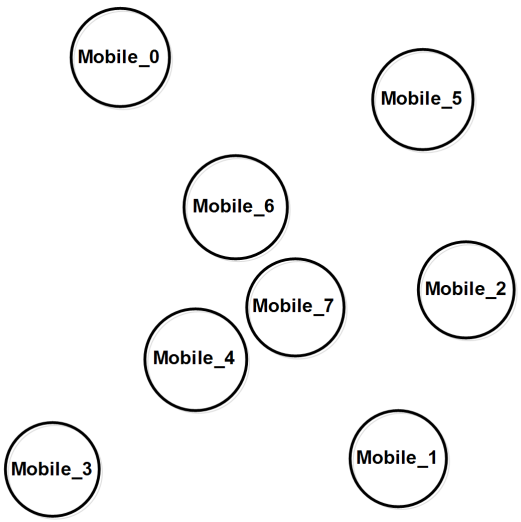


Figure 3.2: 8 Mobile Nodes Ad-Hoc Topology.

3.3.1 TCP

TCP is widely deployed on the ns-2 simulator. The most widely used variant of TCP is the NewReno variant. TCP NewReno has proven both its effectiveness and adaptability in most environments. Therefore, TCP NewReno will be used as the standard baseline to compare the other approaches. TCP NewReno is available by default in ns-2. The default parameters used in the simulation are shown below (see [23] for details).

Simulation Parameters	
Topology Area	1000m x 1000m
Simulation Time	300 sec.
Simulation repetition	30 times
Ad-Hoc Scenarios Number of Mobile Nodes	8, 16, 32, 64, 128, 256
Ad-Hoc Scenarios Number of Flows	4, 8, 16, 32, 64, 128
Mesh Nodes Position	Random
Path Loss Model	Two Ray
Mobility Model	Random Way Point
Maximum Movement Speed	30 m/s
Minimum Movement Speed	10 m/s
Mac layer	IEEE 802.11
Propagation Model	Two Ray Ground
Routing Protocol	DSDV

Table 3.2: Ad-Hoc Simulation Environment Parameters.

Agent/TCP/Newreno set newreno_changes_ 0 - used for fixing unnecessary fast retransmits.

Agent/TCP/Newreno set newreno_changes1_ 1 - forces the impatient variant of TCP NewReno from RFC 3782 [108], with the retransmit timer reset after each partial new ACK.

Agent/TCP/Newreno set partial_window_deflation_ 1 - allows TCP Newreno to ensure that, when the fast recovery process eventually ends, approximately *ssthresh* amount of data will be outstanding in the network.

Agent/TCP/Newreno set exit_recovery_fix_ 0 - to use the fast recovery algorithm modification presented in RFC 3782.

3.3.2 XCP

The ns-2 also includes an implementation of XCP. The parameters used are the same as the original XCP proposal [15]. Thus it was used an α value of 0.4 and a β of 0.226. Thus, in the simulation script file we included:

Queue/XCP set alpha_ 0.4 - the XCP α factor as described in section 2.3.3.1.

Queue/XCP set beta_ 0.226 - the XCP β factor as described in section 2.3.3.1.

3.3.3 RCP

In [16] it is publicly available a ns-2 RCP implementation. We used all the default values recommended. We included RCP ns-2 files in base ns-2 to support RCP.

3.3.4 XCP-b

The XCP-b code for ns-2 was provided by XCP-b's authors. We configured the needed parameters for our specific simulations.

3.3.5 TCP-AP

TCP-AP ns-2 implementation is based in TCP NewReno. The TCP-AP code was provided by its authors and is publicly available in [109]. In the simulations we used the optimal settings, as suggested by TCP-AP authors, which contain the default parameters shown below (see [20] for more details):

Agent/TCP/Newreno/AP set n_factor_ 4 - the spatial reuse constraint factor which mainly depends on ratio between transmission range and interference range (default is 4 for 250 meters transmission range and 550 meters interference/cs ranges).

Agent/TCP/Newreno/AP set rate_interval_ 0.05 - the time between successive packet transmissions.

Agent/TCP/Newreno/AP set n_hop_delay_ 0 - how much to delay the transmission to avoid hidden terminal induced collisions.

Agent/TCP/Newreno/AP set avg_n_hop_delay_ 0 - allows to estimate the propagation delays for only the four hops.

Agent/TCP/Newreno/AP set coeff_var_ 0 - the coefficient of variation of *n_hop_delay_* samples.

Agent/TCP/Newreno/AP set adev_ 0 - mean absolute deviation for the *n_hop_delay_* samples.

Agent/TCP/Newreno/AP set history_ 50 - the history size for the computation of the coefficient of variation.

Agent/TCP/Newreno/AP set delaybound_ 0.5 - an upper bound for the delay samples.

Agent/TCP/Newreno/AP set alpha_ 0.7 - the smoothing factor for *avg_n_hop_delay_*.

Agent/TCP/Newreno/AP set ll_bandwidth_ 2e6 - sets the link layer bandwidth (in bits/s).

3.3.6 WCP

WCP does not have a native ns-2 implementation. WCP code is publicly available in [110] for the QualNet simulator [111] and as a Click [112] implementation. We then ported the Click WCP implementation to ns-2 using Nsclick [113]. After integrating Nsclick into ns-2 it is possible to use the WCP code under ns-2. In [113] there is a complete description of how to integrate Nsclick in ns-2 and how to configure basic parameters. In the simulations we used the recommended WCP parameters as shown below:

Agent/RAW/WCCP set alpha_ 0.1 - the recommended additive increase factor.

Agent/RAW/WCCP set util_ 0.7 - the recommended channel utilization factor.

3.4 Performance Evaluation Results

In this section we present the performance evaluation results, using the performance metrics previously referred. The results are presented considering the comparison of TCP against AIMD and rate based protocols. The AIMD based protocols used are WCP and TCP-AP. For the rate based protocols we evaluate XCP, RCP and XCP-b.

3.4.1 AIMD Based Protocols Evaluation

Figure 3.3, Figure 3.4 and Figure 3.5 show the performance results of TCP, WCP and TCP-AP for the mesh scenarios. The presented results are for the 4x4 grid mesh nodes and variable number of mobile nodes. From the results it is possible to observe that WCP has the best performance results. WCP improved AIMD mechanism makes it perform more efficiently. TCP-AP also obtains better overall results than TCP, but with more mobile nodes in the network its results become similar to standard TCP results. This is due to the fact that the link capacity evaluation of TCP-AP becomes more inaccurate when the number of nodes increases. WCP has also lower performance when the number of nodes increases. As WCP signals all flows in a neighborhood of congestion and sets the control interval to the maximum round trip time (RTT), this strategy, based on the AIMD standard scheme, makes WCP to be very conservative not using efficiently the medium. With respect to the number of received packets, it is possible to observe that both WCP and TCP have better results than TCP-AP. This is due to the fact that the four hop propagation delay of TCP-AP is very conservative, making it receive less packets for the same throughput. With the previous results, it is also possible to conclude that TCP-AP is not fair, since throughput values are not obtained with good received packets values. WCP fairness decreases with the increase in mobility.

To understand how the variation of the number of mesh nodes affects network performance, Figure 3.6, Figure 3.7 and Figure 3.8 show the results for mesh scenarios with a fixed number of 7 mobile nodes and a variable number of mesh nodes (5, 9, 12 and 15 mesh nodes).

The obtained results show that WCP has the best results. WCP presents, however, a very irregular behavior as opposed to TCP when the number of mesh nodes increases. This is due to the fact that, with the increased number of mesh routing messages, WCP is not capable to correctly infer and monitor the round-trip time of every flow. With more mesh nodes and more flows in the network, WCP becomes inefficient and less accurate. With the increase of nodes and flows, the routing messages in the network also increase, and WCP does not evaluate correctly the interference range of the congested link, becoming more aggressive and obtaining poor performance.

TCP-AP, while presenting better overall throughput results than TCP, obtains lower average results for delay and received packets. TCP-AP is based on the estimation of the current four hop propagation delay and in the coefficient of variation of recently measured RTTs. The increase on the number of mesh routing messages in the network causes TCP-AP rate control mechanism to be excessively proactive making it send few packets, resulting

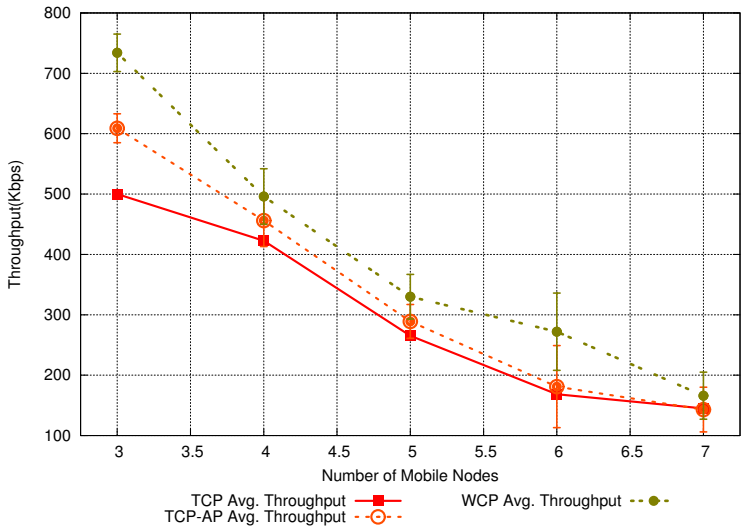


Figure 3.3: AIMD Based Protocols Average Throughput - 16 Mesh Nodes, Variable Number of Mobile Nodes.

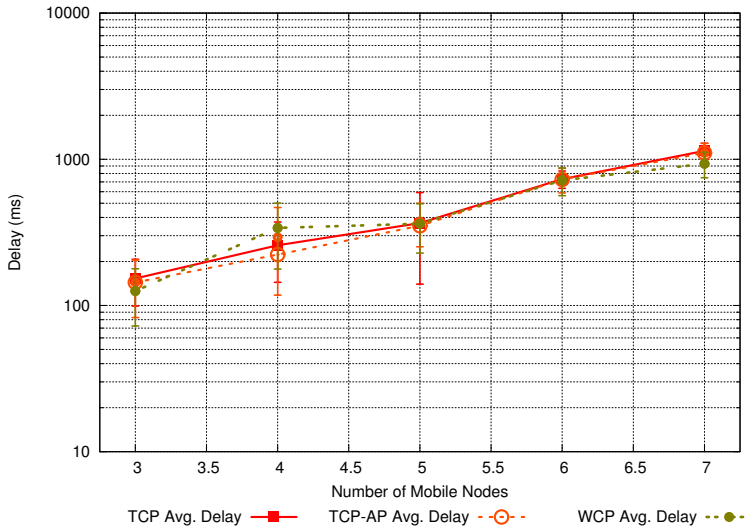


Figure 3.4: AIMD Based Protocols Average Delay - 16 Mesh Nodes, Variable Number of Mobile Nodes.

in less received packets and irregular throughput values. Another important conclusion is that TCP-AP with multiple flows and higher number of mesh routing messages is not able to correctly infer available bandwidth assuming that the bandwidth at each node is identical.

TCP is using more fairly and more efficiently the medium when the number of mesh nodes, and consequently the number of mesh routing messages, increases. Its standard congestion control mechanisms, while very conservative in the slow start phase at the

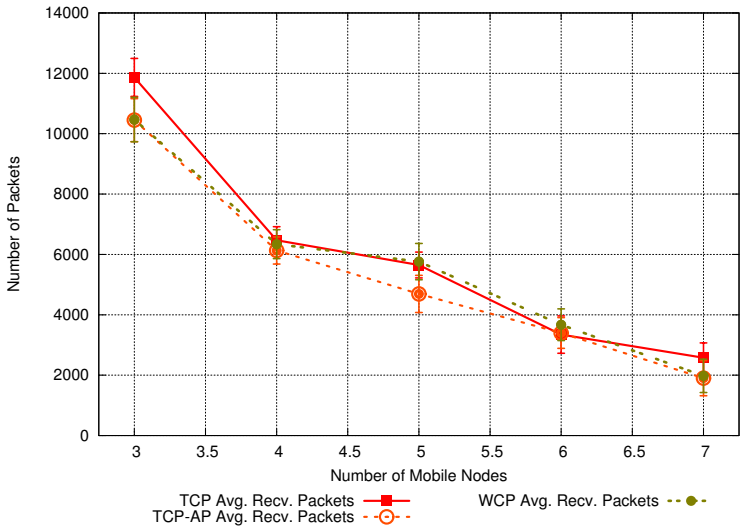


Figure 3.5: AIMD Based Protocols Average Received Packets - 16 Mesh Nodes, Variable Number of Mobile Nodes.

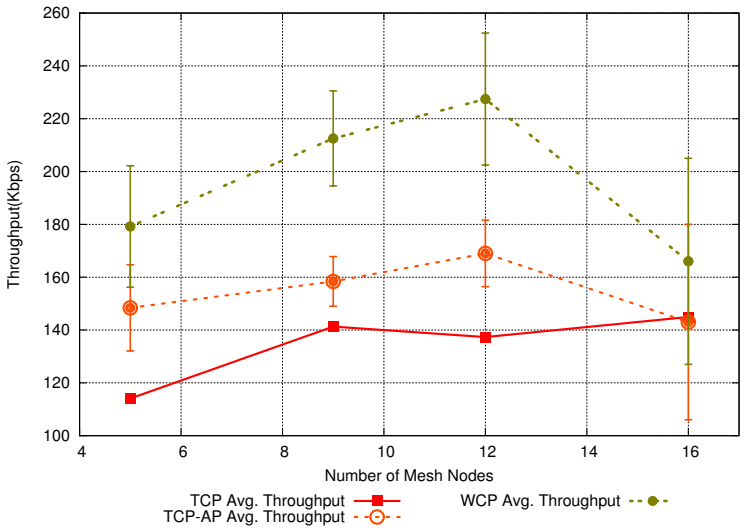


Figure 3.6: AIMD Based Protocols Average Throughput - Variable Number of Mesh Nodes, 7 Mobile Nodes.

beginning, it improves then the network performance with the retransmit/fast recovery phase. Thus, TCP shows a very regular and stable behavior even when the number of mesh nodes increases.

The results of ad-hoc scenarios are shown in Figure 3.9, Figure 3.10 and Figure 3.11. WCP obtains the best average throughput values, while TCP and TCP-AP show very similar results. However, in terms of delay results, TCP gets the best results. Fairness can be outperformed with throughput and bandwidth allocation. Thus, combining Figure

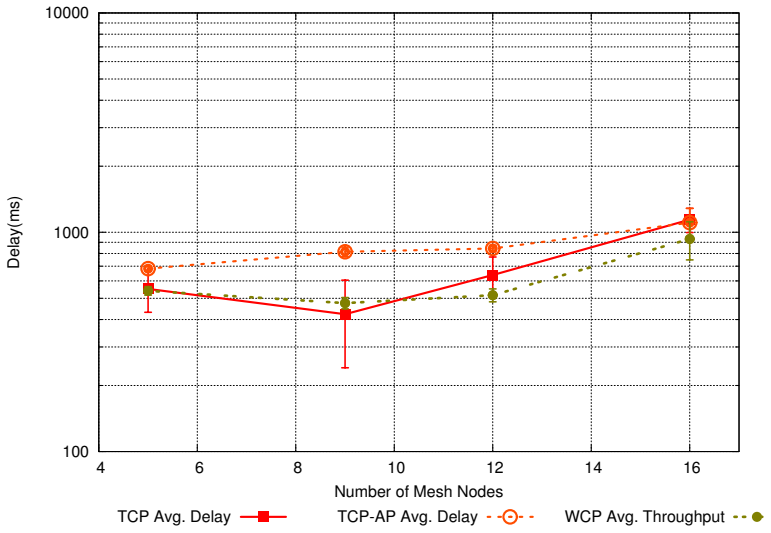


Figure 3.7: AIMD Based Protocols Average Delay - Variable Number of Mesh Nodes, 7 Mobile Nodes.

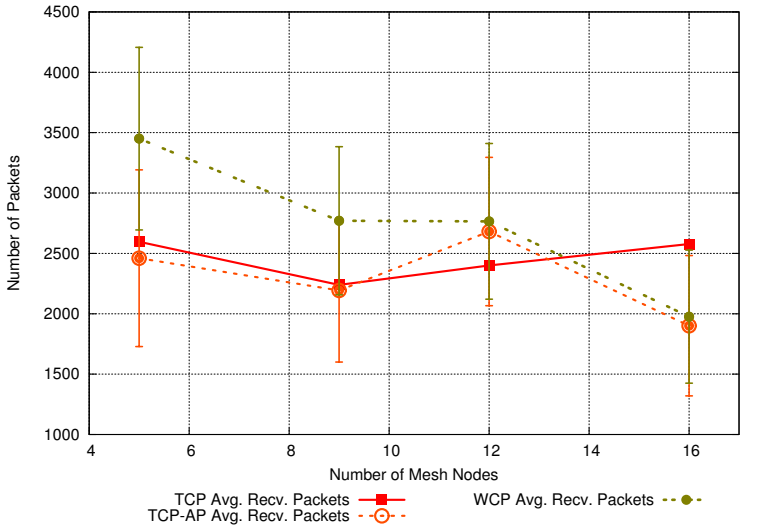


Figure 3.8: AIMD Based Protocols Average Received Packets - Variable Number of Mesh Nodes, 7 Mobile Nodes.

3.9 and Figure 3.10, it is possible to conclude that TCP has a more fair behavior than both WCP and TCP-AP, allowing to use more efficiently the medium when the network is heavily utilized. In terms of number of received packets, WCP gets the best values and TCP-AP the worse values. In terms of number of received packets, TCP results are not very far way from the ones obtained by WCP. WCP modified AIMD mechanism tries to explicitly recognize and account for congestion within a neighborhood; in high mobility and congested networks WCP is not very accurate introducing higher packet losses, thus,

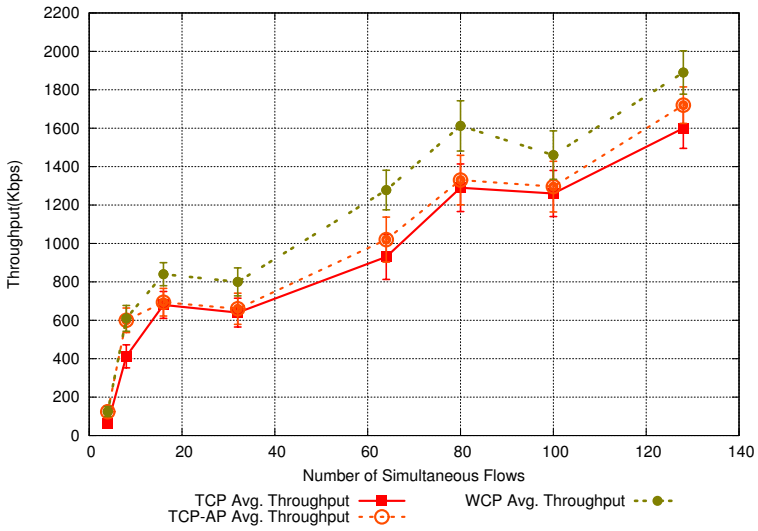


Figure 3.9: AIMD Based Protocols Average Throughput - Ad-Hoc Scenario.

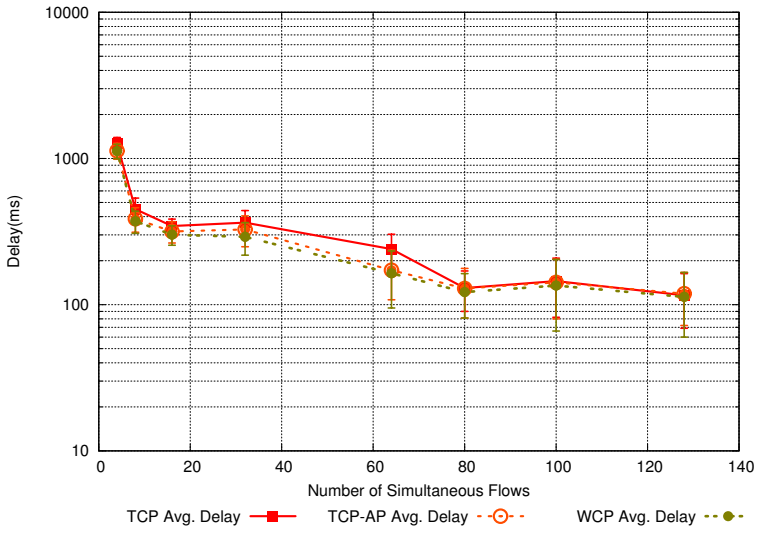


Figure 3.10: AIMD Based Protocols Average Delay - Ad-Hoc Scenario.

obtaining very conservative received packets results. Regarding TCP-AP, its results show that it uses very conservative mechanisms, and that being an hybrid mechanism, when considering high density and high mobility networks, its estimation mechanism together with the AIMD strategy makes it behave very inefficiently and inaccurately, thus, obtaining moderate evaluation results, sometimes even worse than TCP. Surprisingly, TCP shows a very stable and fair behavior.

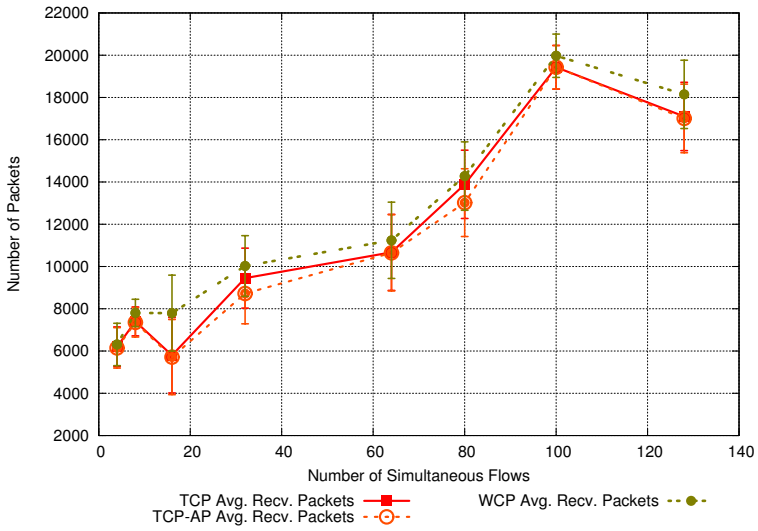


Figure 3.11: AIMD Based Protocols Average Received Packets - Ad-Hoc Scenario.

3.4.2 Rate Based Protocols Evaluation

This section presents the throughput, end-to-end delay and number of received packets results of rate based congestion techniques compared to TCP. The rate based congestion control mechanisms used are XCP, XCP-b and RCP. These mechanisms use network interaction for rate adaptation and congestion control.

Figure 3.12, Figure 3.13 and Figure 3.14 show the obtained results for the 4x4 grid mesh nodes scenario. Figure 3.12 shows the larger throughput of XCP-b. It must be noticed, however, that as XCP-b uses an analytical model that relies on the maximum buffer size of nodes and in complex heuristics, with the increase in the number of mobile nodes, it becomes less accurate and efficient, making XCP-b to obtain results that are very similar to TCP. From the obtained results, it is also possible to see that TCP has a very regular and fair behavior, while RCP and XCP are less efficient, less fair and sometimes show a very erratic and irregular behavior. Although while regarding throughput, the conclusion of more efficiency is not very clear from the delay and number of received packets, TCP has a very clear and improved performance when compared with both XCP and RCP. As previously mentioned, it is possible to outperform fairness with throughput and bandwidth allocation. Thus, from the combination of Figure 3.12 and Figure 3.13, it is possible to conclude that TCP is more fair than XCP and RCP. Comparing to XCP-b, and as more nodes exist in the network, TCP is using more efficiently the medium. It is evident that the use of an analytical model that relies in buffer size is making XCP-b to use the medium inefficiently.

From the obtained results, it is evident that both XCP and RCP do not perform well in a wireless mesh network. XCP and RCP need, to operate, that all nodes in the network exchange information, and they also need to correctly infer the link capacity and available bandwidth. Since they are not inferring correctly those parameters, their performance is

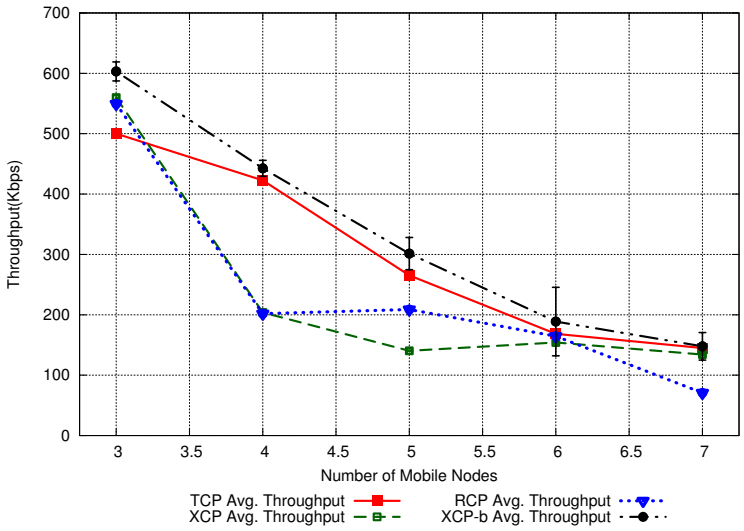


Figure 3.12: Rate Based Protocols Average Throughput - 16 Mesh Nodes, Variable Number of Mobile Nodes.

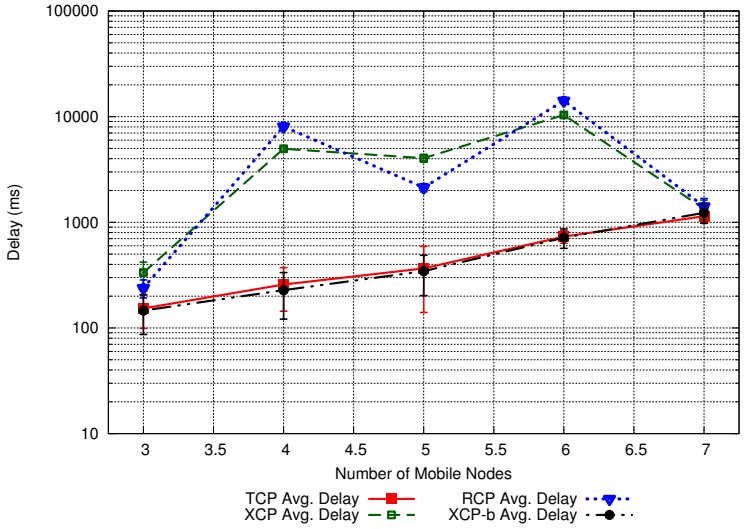


Figure 3.13: Rate Based Protocols Average Average Delay - 16 Mesh Nodes, Variable Number of Mobile Nodes.

significantly reduced, increasing the number of collisions and delay, and obtaining lower throughput values. Figure 3.14 shows that TCP has very good results in terms of received packets, thus, having fewer packet losses. Since XCP-b uses node buffer size, when the network is fully utilized, it is not accurately obtaining available bandwidth values resulting in higher packet losses. The results show that XCP and RCP are less efficient and fair than TCP. While XCP-b has the best results, when mobility and number of nodes increases, its results are comparable to the ones obtained by TCP.

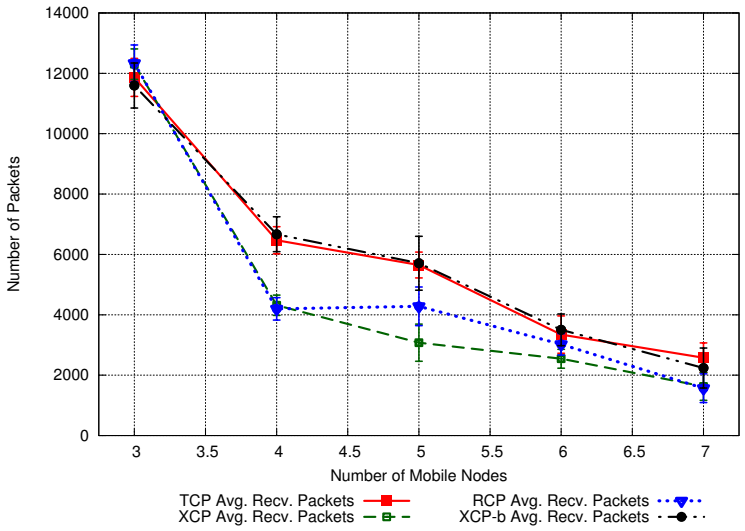


Figure 3.14: Rate Based Protocols Average Average Received Packets - 16 Mesh Nodes, Variable Number of Mobile Nodes.

Figure 3.15, Figure 3.16 and Figure 3.17 show the evaluation parameters results for mesh scenarios with a fixed number of 7 mobile nodes and a variable number of mesh nodes (5, 9, 12 and 15 mesh node). The results show that XCP-b obtains the best throughput results. However, as the number of mesh nodes increases XCP-b results are very similar to TCP results. Moreover, XCP-b obtains worse results than TCP for the number of received packets and for delay when the number of mesh nodes is 16. With more routing messages crossing the network, XCP-b is not able to correctly distinguish between transmission traffic from routing traffic, and considering also both routing and communication retransmissions, it becomes less efficient and with higher number of losses.

TCP, once more, presents a very regular and stable behavior. RCP and XCP behaviors are sometimes very erratic and irregular with different values in terms of throughput, delay and received packets. As XCP and RCP need, to operate, that all nodes in the network exchange information, the number of collisions increases, leading to higher losses, and consequently lower number of packets received and lower throughput and high delay. Thus, XCP and RCP are not allowing nodes to correctly manage their queues, nor obtaining correctly available bandwidth and link capacity values. XCP and RCP rate based control mechanisms clearly suffer from incorrect control parameters estimation in wireless environments, due specially to the underlying shared medium.

The ad-hoc scenario results are presented in Figure 3.18, Figure 3.19 and Figure 3.20.

In Figure 3.18, it is shown that in this particular simulation, the throughput values are very similar for all the evaluated proposals, obtaining XCP-b the best results. It must be noticed that, regarding delay and the number of received packets, it is not so evident that XCP-b outperforms the other evaluated protocols. The best results of XCP-b are, once more, when the network is not heavily utilized, where the XCP-b model is able to

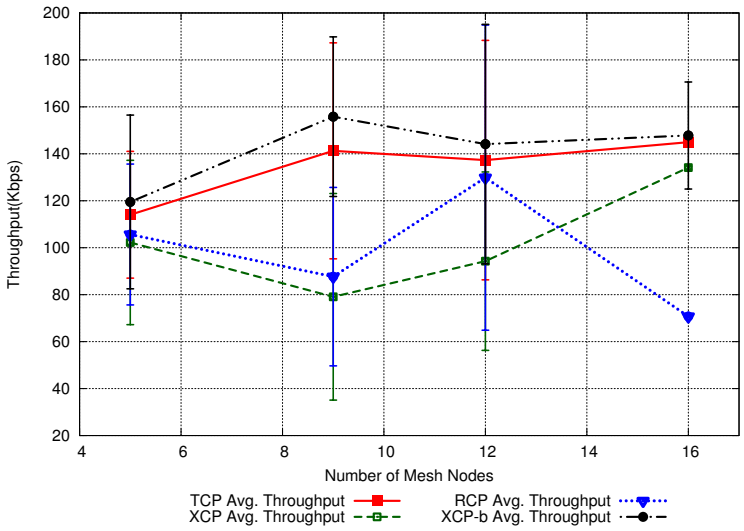


Figure 3.15: Rate Based Protocols Average Throughput - Variable Number of Mesh Nodes, 7 Mobile Nodes.

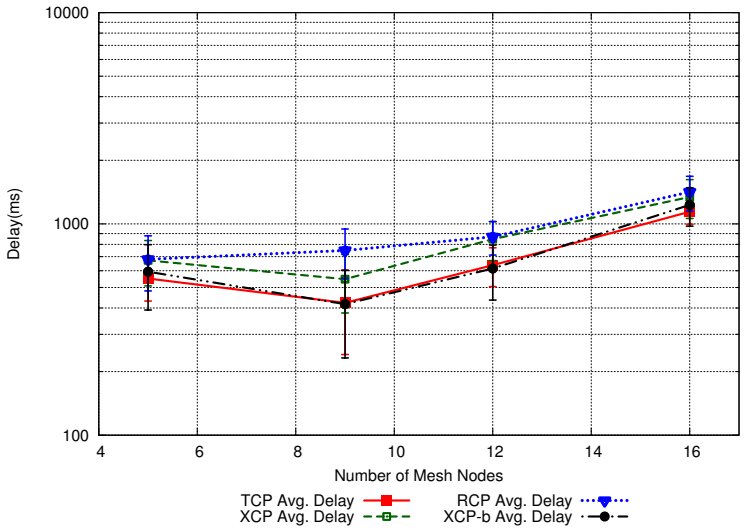


Figure 3.16: Rate Based Protocols Average Average Delay - Variable Number of Mesh Nodes, 7 Mobile Nodes.

correctly obtain the available bandwidth and consequently the rate, allowing for a more effective congestion control. However, as the network load increases, XCP-b becomes more inefficient not using properly the medium and reducing its performance. With more nodes and flows in the network, XCP-b behavior is very unstable not coping correctly with the higher number of losses (Figure 3.20).

The Figures also show that TCP clearly outperforms the results obtained by XCP and RCP. The pure AIMD process of TCP allows it, when the network conditions are severe,

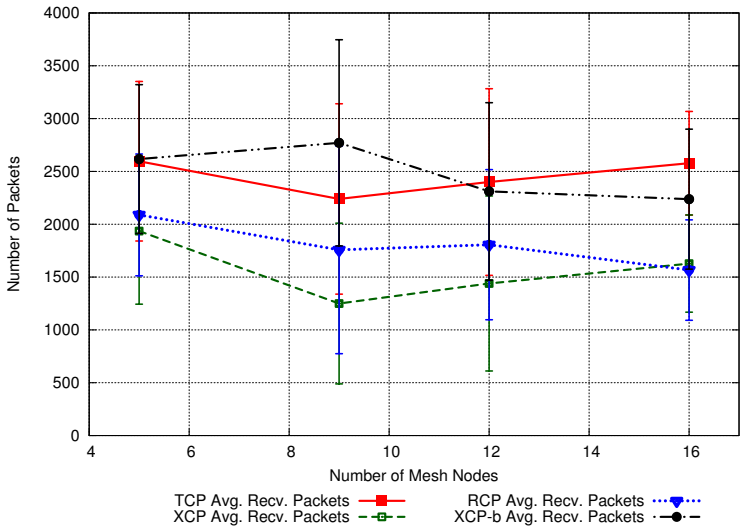


Figure 3.17: Rate Based Protocols Average Average Received Packets - Variable Number of Mesh Nodes, 7 Mobile Nodes.

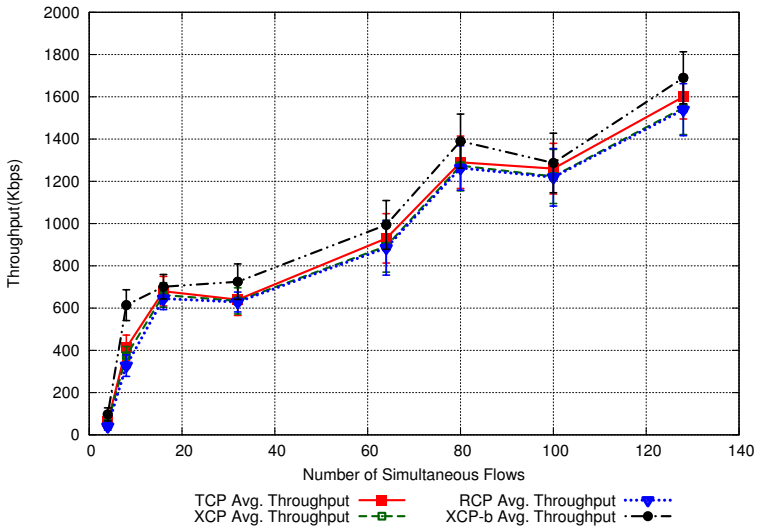


Figure 3.18: Rate Based Protocols Average Average Throughput - Ad-Hoc Scenario.

to maintain its stable behavior as it does not rely in information that must be updated in real-time, such as link capacity and available bandwidth. In a high utilized network with a high load of messages in transit, XCP and RCP are very unstable as they are not able to correctly infer the rate. This makes the rate based congestion control mechanism of XCP and RCP very inefficient and less fair than TCP. This behavior is also shown by Figure 3.18, Figure 3.19 and Figure 3.20, where it is possible to observe similar throughput values as TCP, with less received packets and worse delay vales.

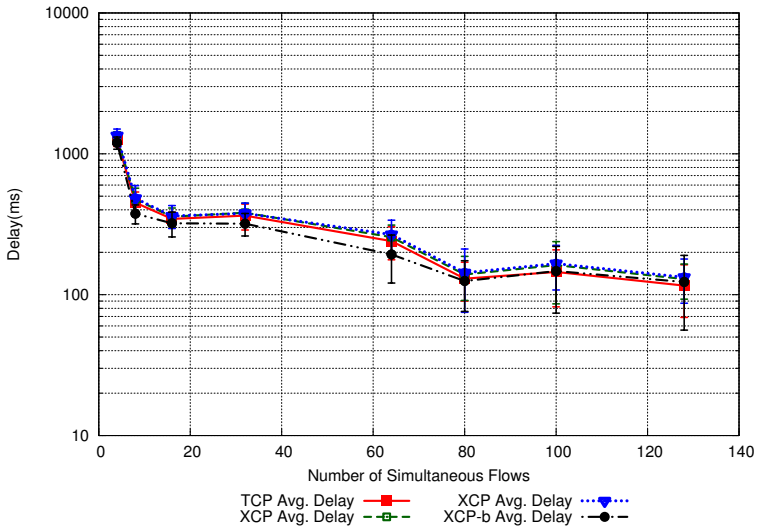


Figure 3.19: Rate Based Protocols Average Average Delay - Ad-Hoc Scenario.

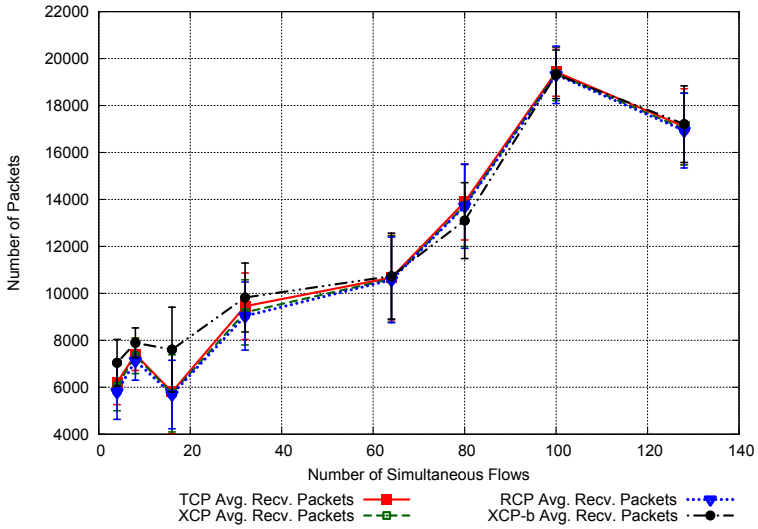


Figure 3.20: Rate Based Protocols Average Average Received Packets - Ad-Hoc Scenario.

3.5 Conclusions

In this chapter we performed an evaluation of the most used congestion control protocol, TCP, against a set of new AIMD and rate based congestion control mechanisms. The AIMD approaches used for the evaluation were WCP and TCP-AP. The rate based were XCP, XCP-b and RCP. These mechanisms use network interaction for rate adaptation. The simulations were conducted with the ns-2 simulator. The performance parameters analyzed were throughput, received packets and delay. For the evaluation, it was used various mesh network topologies and several ad-hoc network topologies.

From the obtained results, it is possible to conclude that WCP overall outperforms the other AIMD based protocols. WCP uses a congestion control mechanism that explicitly reacts to congestion and, it also uses a cooperative communication process between neighbor nodes. These processes allow WCP to react more efficiently to the network conditions, making it use more efficiently the medium and the network resources.

TCP-AP results show that this mechanism, as a consequence of using AIMD and rate based congestion control processes, can obtain improved throughput results, when compared to TCP, but with less received packets. A TCP-AP sender is using, in whatever type of scenario, an estimate of the current four hop propagation delay, being very conservative and becoming more inefficient as the number of hops increases. Another important characteristic of TCP-AP is that it does not use information directly from the MAC layer as the baseline for rate control equation, relying in information from the transport layer, making it react with overall poor performance.

Considering now the rate based approaches, the performed evaluation results show that TCP obtains good results specially when the network is fully utilized and with high mobility scenarios. XCP-b in those scenarios is not taking into consideration packet loss, considering packet loss as a buffer overflow, resulting in an inefficient behavior and introducing slowdown on the network, that are reflected by throughput and delay results.

The results also show surprisingly that TCP has a better behavior than XCP and RCP, being more efficient than XCP and RCP in wireless mesh scenarios. TCP is more fair and stable than XCP and RCP. This is due to the AIMD strategy of TCP. XCP is the less efficient protocol, as it increases delay in the communications. To obtain the available network capacity, both XCP and RCP need that all nodes in the network cooperate, which increases network overhead, specially when dealing with wireless mesh networks. Moreover, TCP, RCP and XCP are not taking into consideration losses due to interference or weak signal strength; this is more relevant in XCP and RCP as they need to evaluate the available capacity. Also, nodes in XCP and RCP are not evaluating precisely network capacity, thus leading to a poor network performance.

However, we know that TCP is not a good congestion control protocol for these networks, since it does not behave correctly when there are losses due to weak signal strength or interference. In wireless environments, TCP is unable to distinguish between losses due to network congestion or bit errors and handoffs, making TCP-based applications suffering of poor performance. Another problem of TCP is its well known unfairness.

One of XCP and RCP problems is the wrong inference of available bandwidth in wireless networks. For this purpose, cross layer communication may help: MAC layer can be used and be a source of good available rate planning and decision to improve the effective calculation of the available bandwidth of the channel. One possible information is the one obtained by the Network Allocation Vector (NAV). As referred by [47], the NAV is a timer that indicates the amount of time the medium will be reserved. This important information combined, for example, with the times provided by RTS/CTS packets and/or probing packets, can be very important to improve the congestion protocol performance. This cross layer communication mechanism would, then, allow the congestion protocol to decide if it would increase or decrease rate communication, improving throughput and

fairness as bandwidth allocation would also be improved.

Another problem observed in the evaluation was the lack of feedback information in these transport protocols. The interaction of mesh routers to keep track of feedback information, would surely improve XCP and RCP based protocols performance in WMNs. Another point to have in consideration regarding this aspect is that the correct determination of the available bandwidth implies the correct definition of the network achievable capacity. Therefore, queues in all nodes need to be small, leading to more precise feedback information with better channel utilization.

Based on these problems, it is important to define a new wireless inference mechanism that can obtain, without affecting the network dynamics, link capacity and available bandwidth estimation. The link capacity and available bandwidth results could then be used, through cross layer interaction in real time, by rate based congestion control techniques, like XCP and RCP, to improve congestion control performance and, thus, the overall network performance. Available bandwidth and link capacity should be inferred using MAC layer information.

Chapter 4

Real Time Wireless Inference Mechanism - *rt-Winf*

4.1 Introduction

Knowledge of link capacity and available bandwidth is important for wireless network design, management and, of course, utilization. [6] states that "The deployment of wireless networks reveals that despite the advances in physical-layer transmission technologies, limited capacity, and consequently available bandwidth, continues to be a major factor that limits the performance of wireless networks and severe congestion collapses are pervasive". It is then important to have a capacity and available bandwidth estimation mechanism that can actively and effectively, using network coordination and interaction, measure the referred link parameters. Such technique can then be used for more efficient congestion control

As referred in Chapter 2, link capacity and available bandwidth have been widely studied. A recent mechanism, named *IdleGap* [1], uses information from the Medium Access Control (MAC) layer focusing on its mechanisms in the CSMA-CA scheme of wireless networks. *IdleGap* takes the Network Allocation Vector (NAV) into consideration, where NAV represents the time other nodes will occupy the medium. *IdleGap* introduces a new layer between the MAC and the Network Layer, called *Idle Module*. The introduction of the *Idle Module* has an important disadvantage, that is the modification of the Open Systems Interconnection (OSI) Model [48] by the introduction of a new sublayer. The *Idle Module* is triggered by the NAV vector, registering the occupation time of the wireless link allowing to infer the available bandwidth and the rate value, which is obtained from the IEEE 802.11 header. IdleGap uses the pre-defined IEEE 802.11 header *Data Rate* value, which is not real, resulting in inaccurate estimation values.

Taking into consideration all the previous considerations we propose a new wireless capacity and available bandwidth estimation technique. The proposed algorithm does not influence network dynamics, uses network cooperation and is a real time estimation technique, called *rt-Winf*. *rt-Winf* is based in *IdleGap*, and aims to eliminate its main

problems: it aims to accurately estimate the capacity and available bandwidth.

This chapter is organized as follows. Section 4.2 describes important background information. Then, section 4.3 describes the main *rt-Winf* algorithm, presenting the two possible functioning variants, with the RTS/CTS handshake (section 4.3.1) and with probe packets (section 4.3.2). Section 4.4 presents an evaluation of *rt-Winf* through the obtained results with an emulator (section 4.4.1) and the ns-2 simulator (section 4.4.2). This chapter is closed, on section 4.5, with a summary of its main conclusions.

4.2 Preliminaries

In a network path we have a sequence of H store-and-forward links that transfer packets from a sender to a receiver. Each link i can transmit data at a rate C_i, referred as link capacity. Then, the wireless link end-to-end capacity can be defined as:

$$C \equiv \min_{i=1...H} C_i \tag{4.1}$$

The available bandwidth can, thus, be defined as the fraction of the links capacity that has not been utilized during a period of time. If we extend this concept to the entire path, the end-to-end available bandwidth is the minimum available bandwidth among all links in the path. Thus, the available bandwidth can be defined as:

$$AB \equiv \min_{i=1...H} AB_i \tag{4.2}$$

The MAC layer is very important for management performance in a shared wireless medium. Shared wireless medium has inherent problems caused by its particular characteristics, namely the interference, collisions and the hidden nodes problem. In order to minimize these problems, there is a need to manage each communication so that two near nodes do not communicate simultaneously.

Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) [41] is the MAC layer protocol that solves this communication control. CSMA-CA is a carrier sense transmission protocol with 2 schemes. In a simple scheme, each node checks if the channel is not being used before the transmission. If it is not used, the packet is sent. Otherwise, the node waits for a random period of time before it checks again if the channel is being used.

This scheme, however, does not solve the well known hidden terminal problem, which is very common in wireless networks. It occurs in a multi-hop wireless network where two different nodes, which are distant enough and not detect each other, want to communicate with another node. They will both sense the medium as clear and start the communication, with the other node, causing undesired collisions. In order to solve this problem, a more complex CSMA-CA scheme was designed. It requires a handshake before the actual communication, which is achieved by the exchange of Request-To-Send (RTS) and Clear-To-Send (CTS) control packets (Figure 4.1).

The usage of this scheme is optional, but it is nowadays widely supported by all wireless equipments. Its usage, in a traditional wireless network, represents a negligible cost in terms

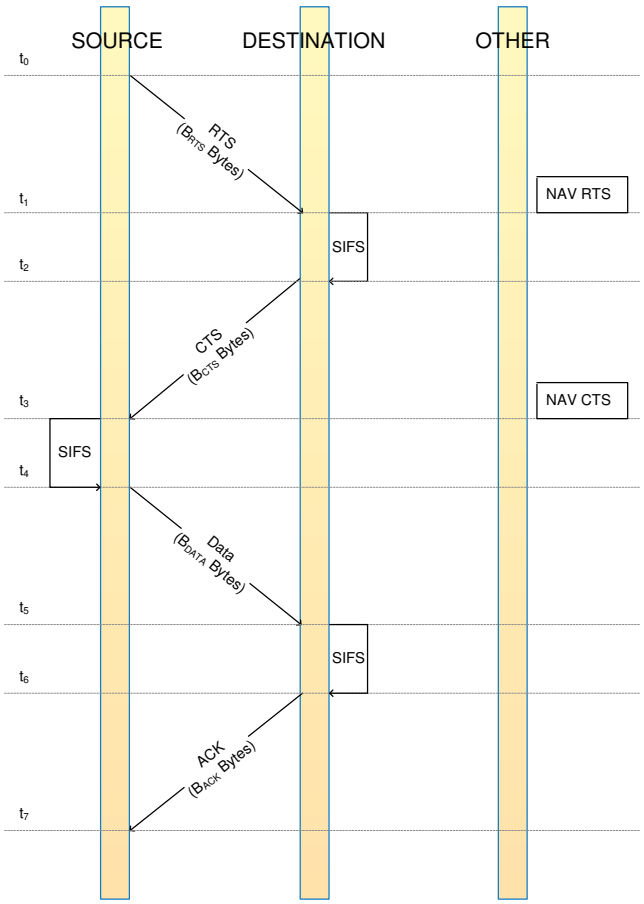


Figure 4.1: RTS/CTS CSMA-CA Scheme.

of overhead [114]. In this context, it is possible to use those control packets to determine the path capacity and the available bandwidth of a wireless path/link. This may be seen as a very important advantage of using RTS/CTS packets.

4.3 *rt-Winf* Mechanism

The real time wireless inference (*rt-Winf*) mechanism was developed with the purpose to mitigate the problems previously mentioned of *IdleGap* (section 2.2.2), being compatible with all systems and determining both the link capacity and available bandwidth without over-loading and influencing the network. *rt-Winf* does not introduce any change to the OSI Model, as opposed to *IdleGap*, being able to obtain all the necessary times to calculate the path capacity and available bandwidth. One of the main issues of *IdleGap* is that it uses the *Data Rate* value of the IEEE 802.11 header [5], while *rt-Winf* effectively calculates the capacity. The operational principles of *rt-Winf* allows it to rely in the RTS/ CTS handshake or in small probe packets.

4.3.1 RTS/CTS Packets

A RTS/CTS enabled *rt-Winf* mechanism relies on the RTS/CTS handshake to correctly retrieve the NAV values, and uses the same node states as defined by *IdleGap*, i.e. the *Sender*, *Onlooker* and *Receiver* states. In the *Sender* state the node is transmitting data; in the *Receiver* state the node is receiving data; and, in the *Onlooker* state the node is not participating in the transmission.

The RTS/CTS packets have accurate duration values, which can be used to trigger *rt-Winf* calculations. To understand how RTS/CTS packets can be used by each node state, a set of handshake captures was performed. The obtained captures showed information on how each node state manages the received packets. CTS, DATA and ACK packets are captured in the case of the *Sender* state. In the *Receiver* state, a node is able to capture the RTS and the DATA packets, while a node in the *Onlooker* state is able to capture the complete set of packets: RTS, CTS, DATA and ACK. This different knowledge implied the conception of different algorithms for each state. Then, we propose that each node state uses a different method to determine the *Idle Rate*, as can be seen in Table 4.1.

To obtain the link capacity and the available bandwidth estimations, a node in the Sender state relies on the NAV information of the CTS packets. Thus, a *Sender* obtains the link capacity (C_{Sender}) using:

$$C_{Sender} = \frac{S}{T_{ACK} - T_{CTS} - 2T_{SIFS}} \tag{4.3}$$

where S is the DATA packet size, T_{ACK} is the actual clock time when the ACK packet is received, T_{CTS} is the clock time of the CTS packet reception, and T_{SIFS} is the duration of the occurred Short Inter-Frame Spacing (SIFS). The time that the channel is busy can then be represented by:

$$T_{Busy} = T_{ACK} - T_{CTS} - 2T_{SIFS} \tag{4.4}$$

When the *Sender* obtains the capacity, it can determine the available bandwidth (AB) by:

$$AB = C_{Sender} \times (1 - \frac{\sum NAV_{CTS}}{T_{Total}}) \tag{4.5}$$

where NAV_{CTS} is the NAV information on a CTS packet, and T_{Total} represents the total elapsed transmission time obtained by the difference between the last captured ACK time and the initial transmission time.

In the *Receiver* state a node uses the NAV information on the RTS packets to obtain the capacity ($C_{Receiver}$), or *Idle Rate*, by:

$$C_{Receiver} = \frac{S}{T_{DATA} - T_{RTS} - 3T_{SIFS}} \tag{4.6}$$

where T_{RTS} is the time when the RTS packet was received, and T_{DATA} is the clock information of the reception of the first DATA packet. The *Receiver* available bandwidth estimation is then calculated through:

$$AB = C_{Receiver} \times (1 - \frac{\sum NAV_{RTS}}{T_{Total}}) \tag{4.7}$$

where NAV_{RTS} represents the NAV value on the RTS packet, and T_{Total} is defined as the difference between the last send ACK packet time and the initial RTS reception time.

The *Onlooker* state uses the NAV value according to the existence, or not, of the RTS packet to obtain both the available bandwidth and capacity. If a node in the *Onlooker* state captures a CTS packet of a communication without capturing the RTS packet, this implies that the communication is suffering from the *hidden nodes* problem. Thus, the algorithm will only use the NAV from the CTS packet to retrieve the correct values. An *Onlooker* node obtains the link capacity (C) by:

$$C_{Onlooker} = \frac{S}{T_{ACK} - T_{CTS} - 2T_{SIFS}} \tag{4.8}$$

If the node only captures a CTS packet the AB is obtained by:

$$AB = C_{Onlooker} \times (1 - \frac{\sum NAV_{CTS}}{T_{Total}}) \tag{4.9}$$

Where T_{Total} is equal to $NAV_{CTS} + T_{CTS} + 2SIFS$. If the *Onlooker* receives a RTS packet then

$$AB = C_{Onlooker} \times (1 - \frac{\sum NAV_{RTS}}{T_{Total}}) \tag{4.10}$$

And T_{Total} is defined as $NAV_{RTS} + T_{RTS}$.

Table 4.2 shows, for a better understanding of the available bandwidth and capacity evaluation, the variable symbols and its description used on *rt-Winf* calculations.

A general *rt-Winf* system, with its main functioning principles and states transitions, can be observed in Figure 4.2. The estimation process in the *Receiver* starts when a RTS packet is received. If the node transitions from the *Onlooker* state to the *Receiver* state, it stores the *Onlooker* state capacity ($C_{Onlooker}$) estimated value. First, the node stores the time at which it received the RTS packet (T_{RTS}). Then, it sends a CTS packet to the *Sender* initiating the communication process. When the *Receiver* receives the actual data, it stores its reception time (T_{DATA}) and then, using the corresponding capacity estimation equation, it obtains its link capacity. If the *Receiver* has stored $C_{Onlooker}$, meaning that there was a transition from the *Onlooker* state to the *Receiver* state, the node compares its capacity estimation value ($C_{Receiver}$) with $C_{Onlooker}$, storing the smallest value of the

State	**Available Bandwidth**	**Capacity**
On-looking	Captured RTS packet? YES: $AB = C_{Onlooker} \times (1 - \frac{\sum NAV_{RTS}}{T_{Total}})$ NO: $AB = C_{Onlooker} \times (1 - \frac{\sum NAV_{CTS}}{T_{Total}})$	$C_{Onlooker} = \frac{S}{T_{ACK} - T_{CTS} - 2T_{SIFS}}$
Sender	$AB = C_{Sender} \times (1 - \frac{\sum NAV_{CTS}}{T_{Total}})$	$C_{Sender} = \frac{S}{T_{ACK} - T_{CTS} - 2T_{SIFS}}$
Receiver	$AB = C_{Receiver} \times (1 - \frac{\sum NAV_{RTS}}{T_{Total}})$	$C_{Receiver} = \frac{S}{T_{DATA} - T_{RTS} - 3T_{SIFS}}$

Table 4.1: *rt-Winf* Algorithm.

Variable	Description
C	Capacity
C_{Sender}	Sender State Capacity
$C_{Receiver}$	Receiver State Capacity
$C_{Onlooker}$	On-looking State Capacity
AB	Available Bandwidth
S	Packet Size
$T_{Transfer}$	Transfer Time
T_{Total}	Total Elapsed Time
T_{ACK}	Time ACK Packet
T_{DATA}	Time DATA Packet
T_{BO}	Backoff Time
$MSDU$	MAC Service Data Unit
NAV_{CTS}	NAV Value in CTS Packet
NAV_{RTS}	NAV Value in RTS Packet
T_{CTS}	CTS Packet Time
T_{RTS}	RTS Packet Time
T_{MSDU}	Delay per MSDU
T_{80211b}	Maximum 802.11b theoretical throughput

Table 4.2: *rt-Winf* Variables.

two capacities:

$$\begin{aligned} & C_{Receiver} - C_{Onlooker} \\ \textbf{if } (>0) &=> C_{Receiver} = C_{Onlooker} \\ \textbf{else if } (<0) &=> C_{Receiver} = C_{Receiver} \end{aligned} \tag{4.11}$$

Then the $C_{Receiver}$ value is sent to the *Sender* on an ACK packet.

In the *Sender* state, and when a CTS packet is captured, the node starts to evaluate the available bandwidth and link capacity. First, the node stores the time at which the CTS packet is received (T_{CTS}), and then starts to send the actual data. When it receives an ACK packet acknowledging data reception, it stores the time at which the ACK packet was received (T_{ACK}) and obtains from the received ACK packet header the $C_{Receiver}$. Then, it uses the corresponding equation of Table 4.1 to estimate its link capacity (C_{Sender}). If the node made a transition from the *Onlooker* state to the *Sender* state, it first compares the C_{Sender} with $C_{Onlooker}$, and updates C_{Sender} to the minimum value of the capacities values:

$$\begin{aligned} & C_{Sender} - C_{Onlooker} \\ \textbf{if } (>0) &=> C_{Sender} = C_{Onlooker} \\ \textbf{else if } (<0) &=> C_{Sender} = C_{Sender} \end{aligned} \tag{4.12}$$

Then, it compares C_{Sender} with the one received on the ACK packet ($C_{Receiver}$) using always the inferior capacity value, avoiding queue fill-ups and bottlenecks. If the node was not previously on the *Onlooker* state, it compares immediately its C_{Sender} estimated value with $C_{receiver}$ on the ACK packet.

$$\begin{aligned} & C_{Sender} - C_{Receiver} \\ \textbf{if } (>0) &=> C_{Sender} = C_{Receiver} \\ \textbf{else if } (<0) &=> C_{Sender} = C_{Sender} \end{aligned} \tag{4.13}$$

With the found capacity value, it determines the corresponding available bandwidth. This cooperation process between the *Sender* and the *Receiver* is a great improvement when compared to *IdleGap*. The onlooking capacity is obtained as described in Table 4.1.

4.3.2 Probe Packets

If RTS/CTS packets are not present, *rt-Winf* can use probe packets in order to retrieve the transfer time values. Probe packets can be sent between nodes. Probe packets are just sent in the beginning of the communication and, for each new communication, new probe packets are sent. To achieve good results, we use packets with 1500 bytes and frequency of 4 samples before the actual transmission starts (as stated in [37]). it must be noticed,

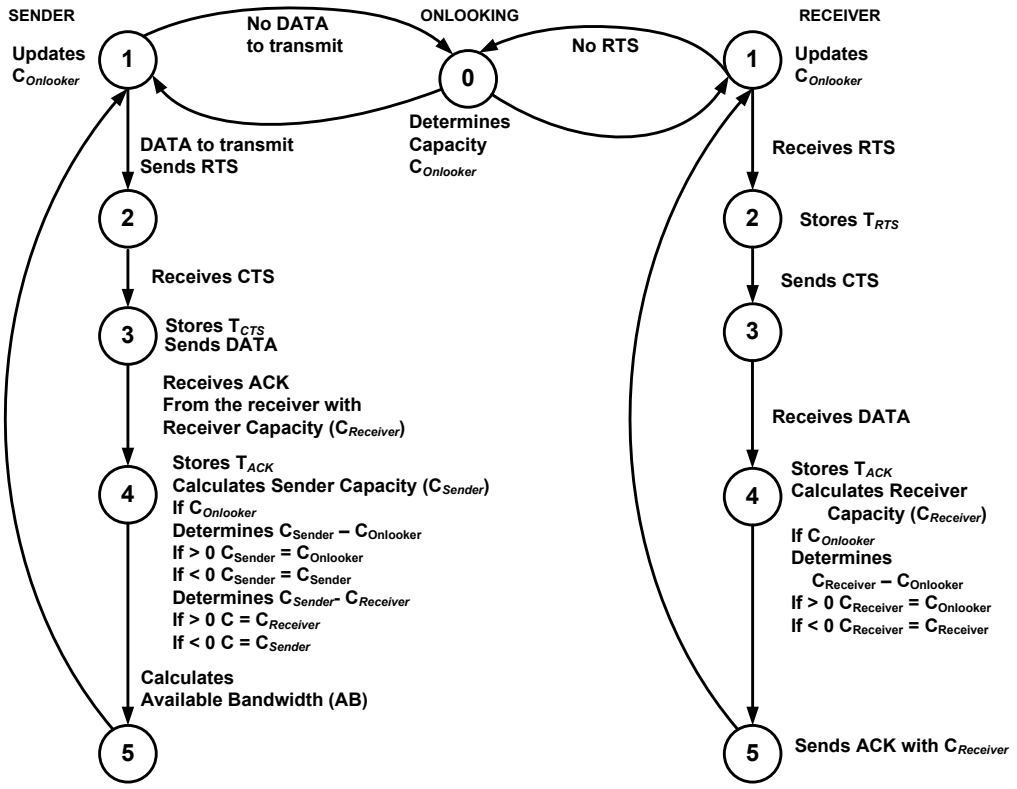


Figure 4.2: *rt-Winf* Sender, Receiver and Onlooking State Diagrams.

however, that probe packets with reduced sizes can be used. The probe packets must be UDP generated packets with altered Frame Control IEEE 802.11 header: Type Data and Subtype Reserved. We use packets with Frame Control Type set to 10 (data) and Subtype to 1001 (Reserved). This way the *Sender* and the *Receiver* can successfully differentiate these packets from the ordinary data packets. The IEEE 802.11 standard defines that, for each successfully received packet, it must be sent a MAC ACK packet [5]. The whole process is very similar to the one with the RTS/CTS handshake.

The generated packets are used to retrieve the capacity (C) and available bandwidth (AB) values, according to Equation 4.14 and Equation 4.15.

$$C = \frac{S}{T_{Transfer}} \tag{4.14}$$

where S is the packet size and $T_{Transfer}$, the transfer time, is equal to $T_{ACK} - T_{DATA}$.

$$AB = 1 - \left(\frac{\sum T_{Transfer}}{T_{Total}} \right) \times C \tag{4.15}$$

where T_{Total} is the total elapsed time since the beginning of the process.

The generated packets are only sent before a node starts a transmission and in the absence of traffic. This allows the system to initially determine the available bandwidth

and capacity. Then, the existing traffic and the MAC layer ACK will be used to trigger the calculations. As NAV values are not correctly defined in DATA packets, *rt-Winf* uses clock time information to determine the busy time. Each node state has to manage independently its clock information. Therefore, NAV values are not considered in this specific implementation with probe packets. To be fully operational, both *Sender* and *Receiver* must be running the *rt-Winf* mechanism.

It must be stated that in a normal VoIP call using G.711 codec [115], the overhead introduced by *rt-Winf* probe packets mechanism is, if using one packet of 1500 bytes, $\sim 1.66\%$. For a flow with more than 1Mbps, the overhead introduced if using packets of 1500 bytes with a frequency of four samples per flow is less than $\sim$ 0.6%.

4.4 *rt-Winf* Performance Evaluation

rt-Winf was implemented in the Carnegie Mellon University (CMU) Wireless Emulator [24] and in the ns-2 simulator. All three states - *Sender*, *Receiver* and *Onlooking* - defined by the *rt-Winf* mechanism and the cooperation between them and between the nodes was developed in *C* language. Although the emulator provides more realistic results than a traditional network simulator, we also present ns-2 simulations for comparison purposes. In base *rt-Winf*, the system is configured with enabled RTS/CTS/ACK handshake packets. In *rt-Winf* probe, RTS/CTS/ACK handshake is not enabled, and probe packets are implemented. The maximum achievable data rate is set to 11 Mbps. Nodes are placed in such a distance that the path loss effect is considered negligible.

Several scenarios were used, varying the number of nodes and the traffic load. Most of the scenarios used an Access Point (AP), for the management of the wireless communication. An ad-hoc and a mesh scenario were also considered to evaluate *rt-Winf*. Path capacity and available bandwidth are evaluated in different mesh and ad-hoc wireless scenarios.

4.4.1 CMU Wireless Emulator Results

To evaluate *rt-Winf* in the CMU wireless emulator, we had to implement all *rt-Winf* functions and states in *C* language and then update the CMU emulator nodes with our enhancements. This emulator is able to provide a real environment, which is very important to assess the truth efficiency of the proposed estimation mechanism.

In the path capacity evaluation, *rt-Winf* results are compared against *AdHoc Probe* results and maximum throughput (that represents the maximal theoretical throughput) in a simple 2 ad-hoc nodes testbed. Figure 4.3 represents the scenario used, where node 1 is the sender and node 2 is the receiver. In this scenario both nodes use *rt-Winf*, but the presented results are retrieved in the sender node. The nodes use random mobility.

AdHoc Probe measures efficiently the path capacity in a wireless communication when compared to other mechanisms [37]. An UDP flow with Constant Bit Rate (CBR) of 64

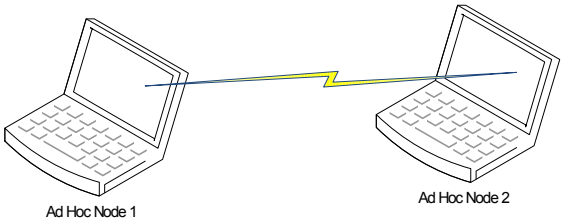


Figure 4.3: 2 Ad-Hoc Nodes Testbed.

Kbps is transferred between the two nodes. Accordingly to [116], the maximal theoretical throughput is obtained through

$$TH_{80211b} = \frac{MSDU}{T_{MSDU}}(Kbps) \tag{4.16}$$

where MSDU is the MAC Service Data Unit and

$$T_{MSDU} = T_{DIFS} + T_{SIFS} + T_{BO} + T_{CTS} + T_{ACK} + T_{RTS} + T_{DATA}(s) \tag{4.17}$$

where T_{DIFS} is the DIFS time, T_{SIFS} is the SIFS time, T_{BO} is the backoff time, T_{CTS} is the time needed to transmit a CTS packet, T_{RTS} is the time needed to send a RTS packet, T_{ACK} is the ACK packet time and T_{DATA} is the time to transmit the actual data.

The maximum throughput represents, in ideal conditions, the maximum achievable capacity. Figure 4.4 shows the path capacity results. With a CBR flow in the network, the expected capacity shall be lower than the maximum throughput. The simulations validate this assumption, showing that *rt-Winf* results are close to the maximum throughput values, but lower than this value. It is then possible to observe that *rt-Winf* uses efficiently the information present in the channel in order to obtain the resulting capacity. This is because *rt-Winf* takes into consideration all traffic flows and then measures more accurately the channel occupation time. Comparing with the *AdHoc Probe* mechanism, and with a similar probing time, *rt-Winf* gathers more information to perform the desired calculations, thus being able to be statistically more precise and less sensitive to flow variations. *AdHoc Probe* is a very intrusive approach and it only takes into consideration the information given by its probing packets, which can suffer dispersion and collisions (introducing a negative impact in the capacity evaluation). The effect of bad estimation, retrieved by the probe packets can be seen between 100 and 150 seconds which present lower values for *AdHoc Probe*. *AdHoc Probe* probe packets are overloading the network, which increases collisions and packet losses. Therefore, *AdHoc Probe* is estimating incorrectly the network usage.

We also performed available bandwidth evaluation. The results are compared against *IdleGap* [1], *IPerf* [30], maximum throughput and maximum achievable data rate (11Mps)

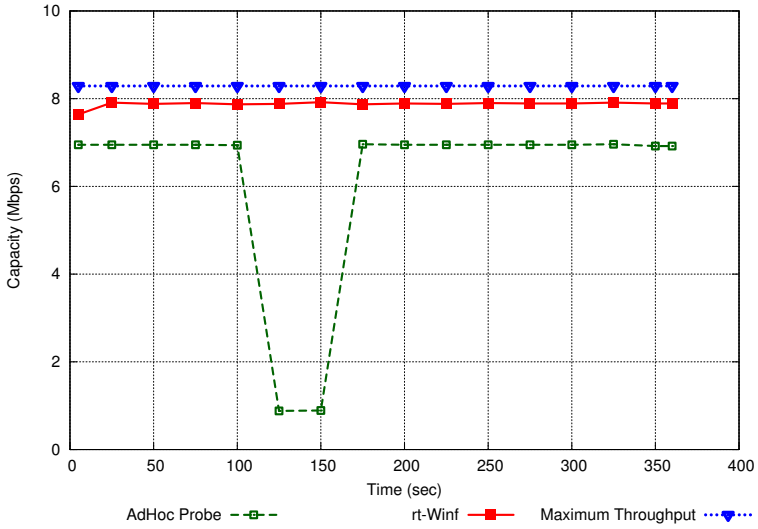
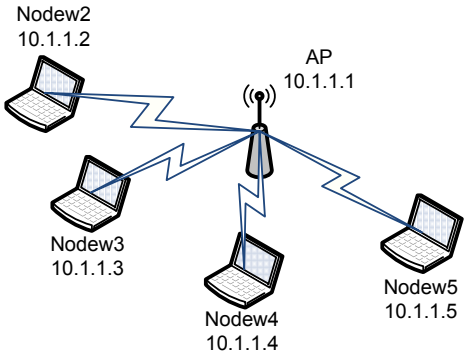


Figure 4.4: AdHoc Probe and rt-Winf Path Capacity Estimation.

values. According to [39] the ground truth of the achievable available bandwidth of a Dynamic Wireless Local Area Network (WLAN) is aproximated by the downstream throughput of a single saturated CBR UDP flow with a packet size of the Maximum Transmission Unit (MTU). Then, an UDP flow with a very high constant bit rate (10Mbps) was considered and injected through the channel. The result of the downstream throughput is the curve IPerf UDP. For the IPerf TCP results, we used its default parameters, only changing the sampling time. The "*Maximum Throughput*" is obtained from equation 4.16, which corresponds to the maximal theoretical throughput for each scenario.

Figure 4.5 shows the used testbed which contains four mobile nodes and an AP. In this testbed we analyze different scenarios that vary according to the number of active nodes and traffic flows. This is shown in the table of Figure 4.5. Each scenario is labeled and differs on the concurrent traffic flow and number of used nodes. A value of zero represents nonexistence of the node in the considered scenario. We conducted 50 emulations for each scenario, resulting in a confidence interval of 95%.

In Figure 4.6, we show the variation of available bandwidth as a time function obtained from scenario E as described in the table of Figure 4.5. It is possible to observe the results of each mechanism compared with both the maximum achievable data rate and the maximum theoretical throughput. The maximum theoretical throughput does not take in consideration the cross traffic influence. Then, it can be considered a valuable information as an upper bound of the expected results. On the other hand, *IPerf UDP* values are obtained through the overloading of the wireless channel, thus being considered a lower bound of the expected results. As *rt-Winf* obtains results within those bounds, we can conclude that the determined values are nearer the real ones, and then, *rt-Winf* is more precise. This is due to the fact that this mechanism actually measures the channel occupation time of each packet, thus obtaining more precise values for the capacity. Available



Flows	F1(Mbps) N4→N3	F2(Mbps) N5→N3	F3(Mbps) N5→N4
A	0	0	0
B	1	0	0
C	1	1	0
D	1	1	1
E	2	1	1

Figure 4.5: Available Bandwidth Estimation Scenario and Specification.

bandwidth, as described in section 4.3, depends directly on the channel capacity. The *IPerf UDP* values are close to the ones obtained with *rt-Winf*, but they are obtained through a very intrusive way. As expected, the *IPerf TCP* results are the worse, as a result of its congestion control mechanisms, namely, congestion avoidance and slow start [105]. This is represented in Figure 4.6 by *IPerf TCP* variations.

IdleGap values are larger than the ones obtained with other mechanisms and are very close to the maximum *DataRate*. To observe this fact, maximum *DataRate* values are also shown in Figure 4.6. *IdleGap*, as described in section 2.2.2, does not determine the real link capacity, and it considers the value registered in each *Data Rate* header field of each packet. This value is defined by the standard [5] and varies only accordingly with signal propagation and network traffic. The standard allows a predefined set of data rate values to determine, in real time, the channel capacity. As the available bandwidth is obtained using link capacity, *IdleGap* available bandwidth estimations are overestimated.

Each bar on Figure 4.7 represents the average emulation results for each scenario presented in Figure 4.5. We compare the results provided by *IPerf UDP*, *IdleGap* and *rt-Winf*. The line in the figure represents the variation of the concurrent traffic on the considered scenario. The concurrent traffic varies from 1 Mbps to 4 Mbps.

From the Figure 4.7, it is possible to observe that *IdleGap* results are higher than *rt-Winf* and *IPerf*. Again, this is related with the fact described before. *IdleGap* values vary in real time and *IdleGap* evaluates the occupied time in a similar way as *rt-Winf*. However, when this occupation rate is obtained and is applied with the IEEE802.11 header *Data Rate* value, the resulting available bandwidth is being erroneously magnified. As noted on Figure 4.7, with the increase of concurrent traffic, the difference of values between *IdleGap* and the other mechanisms is significantly reduced. This is due to the fact that, for more concurrent traffic, the device drivers automatically adjust the data rate value to a lower one. In scenarios A, B and C, it is set to 11Mbps, and in scenarios D and E, it is adjusted to 5.5 Mbps.

Path capacity and available bandwidth evaluations are also conducted on a wireless mesh scenario (Figure 4.8). The two mobile nodes, *Mobile Node 1* and *Mobile Node 2*,

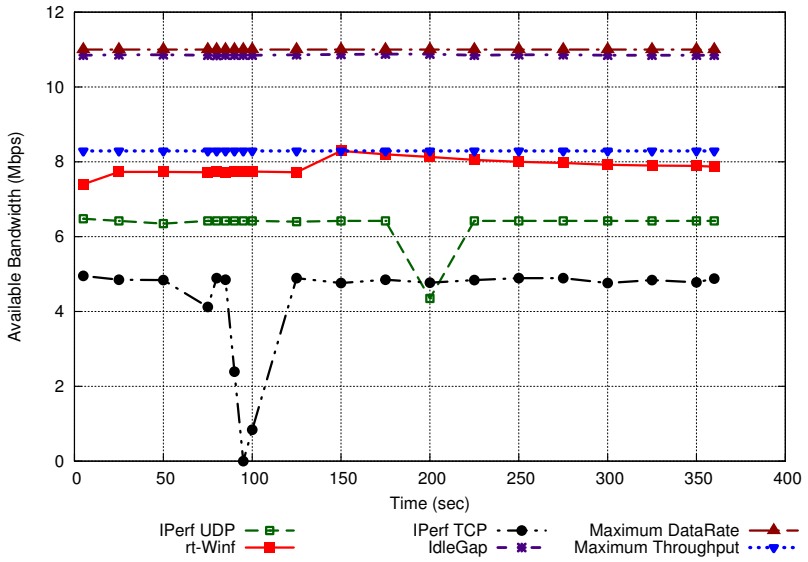


Figure 4.6: rt-Winf, IdleGap and Iperf Available Bandwidth.

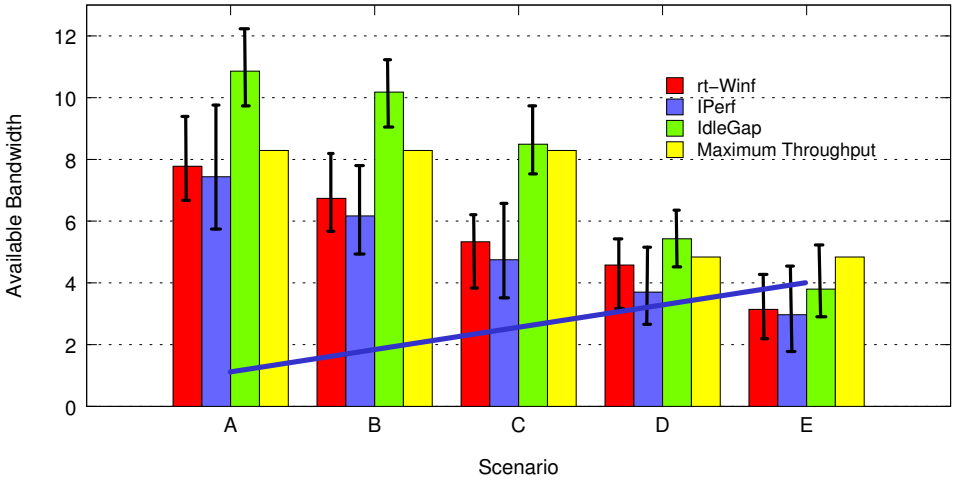


Figure 4.7: Available Bandwidth by Scenario

communicate with each other through two mesh nodes that are responsible for the routing and link management. The mobile nodes are in such a distance that the traffic is always routed by the mesh nodes. It was considered as the link traffic a CBR UDP flow of 1 Mbps and packets of 1500 bytes.

Path capacity results of *AdHoc Probe* and *rt-Winf* are shown in Figure 4.9. Available bandwidth results are shown in Figure 4.10. Figure 4.10 contains the results of *rt-Winf*, *IPerf UDP* [30] and *IdleGap*. Maximum throughput values are also presented, being considered as an upper bound of the results, as described before. *IPerf UDP* results are considered the lower bound.

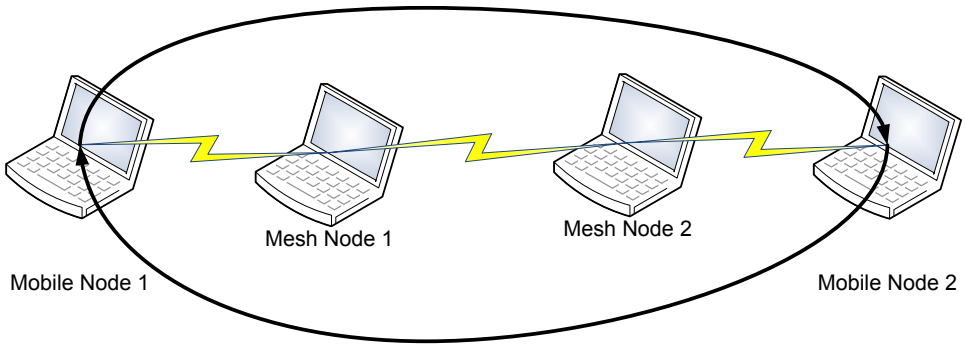


Figure 4.8: Wireless Mesh Scenario.

As observed in Figure 4.9, *rt-Winf* is less sensitive to variations when compared to *AdHoc Probe*. This is because *rt-Winf* is taking into consideration all packets in the network and is measuring the channel occupation time of each packet, while *AdHoc Probe* is only considering the packets that it generates, thus, being more sensitive to flow variations.

The results presented in Figure 4.10 show that *IdleGap* available bandwidth values have a small variation, without reflecting the network dynamics. This is due to the fact that *IdleGap* considers the *Data Rate* header value as the link capacity to estimate the available bandwidth, which is not an accurate value. In *rt-Winf*, the obtained results are within an upper bound, the maximum theoretical throughput, and a lower bound, *IPerf UDP*.

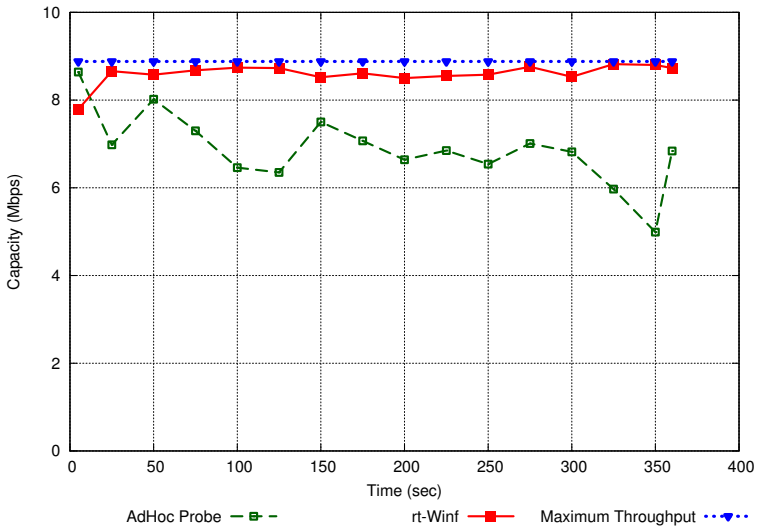


Figure 4.9: Path Capacity in the Wireless Mesh Scenario.

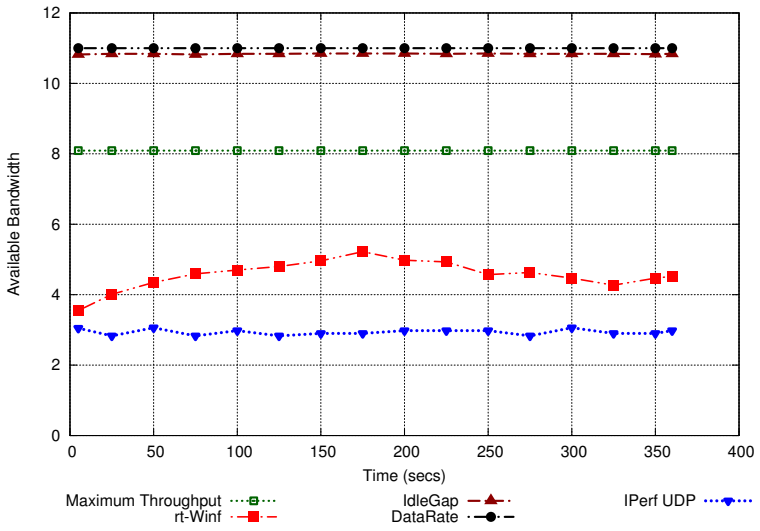


Figure 4.10: Available Bandwidth in the Wireless Mesh Scenario.

4.4.2 Ns-2 Simulator Results

In order to compare the values of the emulator with the ones of the simulator, and also to investigate the behavior of *rt-Winf* with probe packets in a wireless mesh scenario, we also conducted simulations in the ns-2 simulator. The standard ns-2 simulator does not support *rt-Winf*. We then modified ns-2 IEEE 802.11 MAC layer to support all functions and node states as defined by *rt-Winf*, which allowed us to conduct the simulations. Ns-2 was also modified to support *rt-Winf* with probe packets. The ns-2 *IdleGap* code was provided by its authors; however, it was required to further improve the development by adding support to more types of traffic flows.

As *rt-Winf* is based in *IdleGap*, the simulations also allow a baseline comparison of those mechanisms. The scenario used is the mesh wireless scenario shown in Figure 4.8, with two mobile nodes and two mesh nodes. In the simulations, it is used a FTP transfer from a source to a sink with packets of 1500 bytes. Different simultaneous flows were used. The wireless link capacity was configured to be 10 Mbps. The maximum throughput is calculated using ns-2 default values and using Equation 4.16. In the simulations with *rt-Winf probes*, we used probe packets with sizes of 1000 and 300 bytes.

Figure 4.11 summarizes the obtained results. Each value is an average of 20 runs lasting 300 seconds of simulated time, and nodes are stationary placed on a straight line with 200 meters in between. As expected, the estimated capacity value decreases as traffic flows increases. This is a consequence of more cross traffic in the network. We can also observe that *IdleGap* results are almost equal to the maximal theoretical throughput. *IdleGap* is using the IEEE 802.11 header *Data Rate* in the calculations. This will affect available bandwidth estimations as *IdleGap* is overestimating the capacity values. Another important observation is that *rt-Winf probes* estimate value increases as the packet size increases. This is consistent with Equation 4.14. Therefore, when using *rt-Winf* with

probe packets, it is important to use packets with the network MSS. The results obtained by *rt-Winf* are very close to the ones obtained by *rt-Winf probes*, allowing to conclude that in highly dynamic environments *rt-Winf* will behave efficiently using the NAV values defined in the RTS/CTS packets. Another important consideration is that, with more flows in the network, *rt-Winf* estimates capacity more effectively than the probe packets versions. This is a consequence of using extra packets in the network that can result in more collisions and network overload, reducing the efficiency of capacity estimations.

These results validate the ones obtained with the CMU Emulator, since the results for 1 flow in Figure 4.11 are similar to the ones of Figure 4.9.

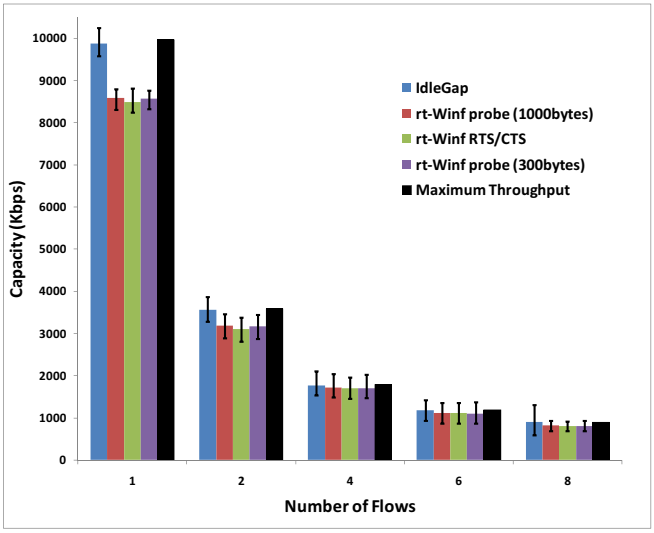


Figure 4.11: Ns-2 Capacity Results.

4.5 Conclusions

In this chapter we presented a new wireless inference mechanism called *rt-Winf*, that estimates both link capacity and available bandwidth. This mechanism performs real time evaluations and introduces the concept of network and node cooperation. We gave an overview of the *rt-Winf* inference technique and architecture. *rt-Winf* uses information already available on the network and is based on *IdleGap*. It can rely on the CTS/RTS/ACK messages handshake or on small probes. These packets provide time information, allowing to know each node's channel allocation.

IdleGap adds a new module between the MAC and network layer to the nodes system, altering its protocol stack. Moreover, its results are not accurate, as *IdleGap* considers the IEEE 802.11 header *Data Rate* value as the link capacity.

Besides being able to accurately determine the capacity and available bandwidth, *rt-Winf* does not introduce any changes to the equipment protocol stack and uses the real

Data Rate value. This means that *rt-Winf* can be supported by any existing wireless equipment.

The simulation results obtained through the CMU Wireless Emulator and the ns-2 simulator, conducted in wireless infrastructure, ad-hoc and wireless mesh networks, indicate that *rt-Winf* efficiently performs the desired calculations, providing accurate results without the need to negatively influence the network. *rt-Winf* can be used in a passive way, measuring the existing traffic of the wireless links, without the need to introduce more traffic in the network.

The *rt-Winf* mechanisms is capable of providing good estimates of link capacity and available bandwidth in wireless environments. We have shown in Chapter 3 that it is important to use accurate link capacity and available bandwidth estimations by rate congestion control mechanisms. Therefore, it is important that we investigate the suitability of the *rt-Winf* estimations as part of rate based congestion control mechanisms, and develop new congestion control mechanisms with accurate and dynamic network state estimation.

Chapter 5

Improved XCP and RCP Wireless Congestion Control Techniques

5.1 Introduction

Congestion control has largely been based in the Additive Increase Multiplicative Decrease (AIMD) scheme of the Transport Control Protocol (TCP). However, AIMD based protocols suffer from long queue backlogs as the requested rates exceed the actual available bandwidth; these schemes also suffer from long convergence times to reach the achievable data rate. TCP, as stated in Chapter 2, assumes that a packet loss is always due to congestion in the network and, but not as often, of packet reordering. TCP does not respond well to packet losses due to bit errors and handoffs, making TCP-based applications suffering from poor performance in wireless environments.

The main objective of congestion control is to manage the traffic generated and injected at the sender into the network to prevent overloading of paths, while maximizing the network utilization for good throughput performance. As discussed in section 2.3.1.1, TCP's AIMD congestion control mechanism is not suitable for wireless networks as it overloads the wireless channel.

Recent studies ([117], [118]), specially conducted in wireless ad-hoc networks, have shown that rate control mechanisms based on rate feedback obtained from the network are a better alternative for multi-hop wireless environments than the standard TCP window based congestion control mechanism. For efficient transport of data over wireless networks, it is important for the source nodes to dynamically regulate traffic inserted into the network to avoid network congestion. Particularly in wireless networks, congestion results in packet losses due to buffer overflowing, but it also increases the contention and collision probability for wireless channel access.

The rate feedback obtained from the nodes along the intended transmission path is a more accurate indicator of bandwidth available for transmission than packet losses, since a node in the network is in a better position to monitor the transmission resources available in its local neighborhood. The support of a more accurate and timely feedback allows faster

and more precise rate adaptation, thus maximizing channel utilization while preventing network congestion. In addition, it offers an easy way of decoupling the congestion control from the loss recovery mechanisms, potentially leading to a better design of loss detection and recovery mechanism to deal with losses caused by congestion, channel bit error and node mobility. Moreover, in rate based flow control packets are sent at regular time intervals instead of bursty transmissions as in TCP's window-based flow control.

It must be stated that in both wired and wireless networks, robust and timely estimation of link capacity and available link bandwidth is a necessary component not only for effective congestion control, but also for quality of service support mechanisms such as admission control, *QoS* routing and resource reservation.

In wireless networks link capacity and available bandwidth are two major important parameters that limit overall network performance, and specially congestion control performance and behavior. The congestion control capacity of a wireless congestion control protocol is also a key factor to make the whole network more reliable, fair and with higher performance.

Rate estimation is clearly a key component of a rate based control mechanism. The accuracy of the rate estimation, thus link capacity and available link capacity estimation, affects the stability of the control loop. In Chapter 4, we have proposed the *rt-Winf* link capacity and available bandwidth estimation technique for multi-hop wireless networks, and we have evaluated it in comparison with other recent wireless estimation techniques. It has been found that *rt-Winf* has a very effective and accurate behavior, while other techniques present over-estimated values.

As discussed before, a rate based protocol is a good candidate for improving congestion control behavior as it uses direct path capacity to adapt its congestion window according to the instantaneous rate measured. A rate based congestion control mechanism based on explicit rate feedback from the network effectively decouples congestion control mechanism from the reliability mechanism; that is, it does not depend on packet losses to detect congestion, and the rate feedback provides explicit input for congestion control.

The eXplicit Control Protocol (XCP) [15] and the Rate Control Protocol (RCP) [16] are two of the most recent congestion control protocols for wired networks. They rely on network interaction for congestion control improvement, since they need intermediate nodes, such as routers, to work and interact to support the congestion control. In wired networks they increase efficiency in the congestion control. However, as studied in Chapter 3, their performance in wireless networks (wireless mesh networks and wireless ad-hoc networks), is decreased (their performance is even worse than TCP), since they are not able to accurately measure the available bandwidth of the wireless links.

In order to overcome the limitations in wireless networks and to increase the congestion control efficiency of XCP and RCP, making them a first option in wireless congestion control, we propose, in this chapter, two new congestion control mechanisms, *XCP-Winf* and *RCP-Winf*, that use link capacity and available bandwidth estimation of *rt-Winf* in their congestion control techniques to accurately evaluate the congestion status of the wireless links, and then take proper actions to avoid and reduce congestion and improve network performance.

This chapter is organized as follows. Section 5.2 presents a brief overview of a generic rate based congestion control mechanism. Then, in section 5.3 we present the conceptual approach of both *XCP-Winf* and *RCP-Winf*, through a mathematical model. Section 5.4 proposes the extension of the congestion control with packet collisions effect. In section 5.5 we perform a complete evaluation of the congestion control approaches compared against base TCP, XCP, RCP and a state of the art approach such as XCP-blind (XCP-b) [22]. We extend the evaluation with the analysis of the effect of collision probability and the improvement of the results when introducing this effect on the congestion control approaches (section 5.5.3). We complete the evaluation with the assessment of both *XCP-Winf* and *RCP-Winf* in terms of TCP friendliness (section 5.5.4), and we also extend the proposals with the improvement of the TCP friendliness behavior (section 5.6). This chapter ends with a reference to its main conclusions on section 5.7.

5.2 Rate Based Congestion Control

The main goals of congestion control are to avoid, or react accordingly, to network congestion and to maximize network utilization as new applications arise. The most important factor in a rate based feedback technique is to correctly estimate the available bandwidth and use it to provide rate feedback. Thus, the underlying link capacity and available bandwidth estimation technique is required to provide accurate estimation, allowing the source node to control its sending rate so that the bottleneck node is able to maximize the throughput and to maintain its average queue delay within a certain limit or threshold.

Figure 5.1 shows an example of a simple rate based feedback congestion control system. Along with each data sent, a special control option in the transport header is used to allow information exchange between the source node and the nodes along the path. The source node uses the control option to convey its required rate to the intermediate nodes which perform flow rate allocation according to a flow allocation policy, and the intermediate nodes use it to convey the flow's allowed rate, or the advertised rate, to the destination node. On receiving a data packet, the destination node keeps an exponential average of the allowed flow rate (i.e. bottleneck rate), and periodically sends a rate-feedback packet (ACK packet) to convey the allowed flow rate to the source, which adjusts its sending rate accordingly. This completes the feedback flow control loop.

A rate based congestion control is implemented as part of transport protocol to regulate the sending rate. Figure 5.2 shows the transport layer packet format. The DATA packet is used for carrying payload data from the source to the destination, and the ACK packet is used by the destination to send rate feedback information in *feedback rate* to the source periodically. Each DATA packet transmitted is assigned a unique packet identifier (i.e. *pkt_id*), and the *pkt_id* in an ACK packet is the packet identifier of the latest DATA packet received; in here, the *pkt_id* does not have a significant role, except it is used to avoid counting duplicated packets in the number of packets received.

Nodes in a rate based congestion control system perform a set of operations. The

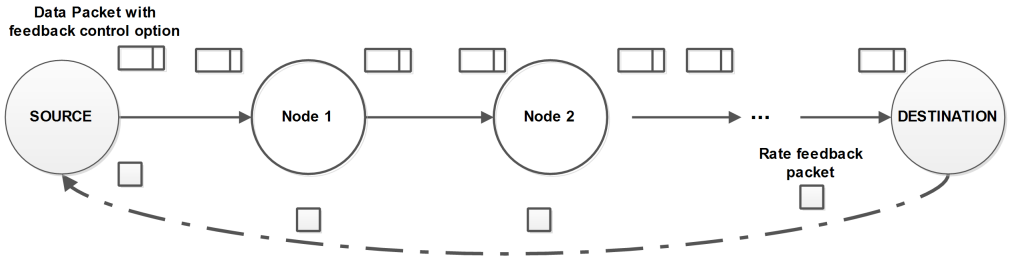


Figure 5.1: Rate Based Feedback Congestion Control System.

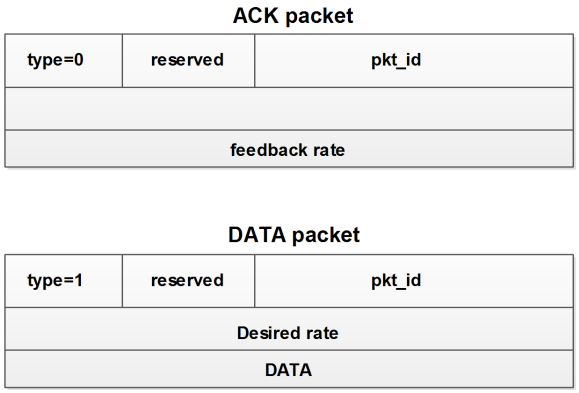


Figure 5.2: Rate Based Feedback Congestion Control Packets.

aggregation of this set of operations is what defines the rate based congestion control mechanism. The operations are defined at the source, intermediate and destination nodes when a packet is received/sent. In a general rate based system, when the source receives an ACK packet, it uses the *feedback rate* to adapt its sending rate. When a sender sends a packet, it defines the required rate. This is known as the advertised rate. At each intermediate node along the path, when a DATA packet is received, the required rate is read from the packet header, and checked if the rate is possible. On forwarding a DATA packet, the bottleneck rate is updated, as written in the advertised rate field. In this way, the DATA packet header will carry the minimum advertised rate of the node along the path when it reaches the destination. When the destination node receives the DATA packet, the values in the *desired rate* field are passed to the transport layer, and the receiver sends that value in an ACK packet to the source. The ACK packets are not modified by the intermediate nodes. One main advantage of this process is that the return path of the ACK is not required to be the same as the DATA packets path.

If we assume that there is a bandwidth estimator mechanism at each node to provide the actual link capacity and available bandwidth estimation, it is required a flow rate

allocation scheme. While a simple equal rate allocation where the available bandwidth is equally divided among the flows has the benefit of not requiring maintenance of flow state information at each node, it may not fully utilize the bandwidth allocated.

Both XCP and RCP congestion control mechanisms are supported by rate allocation algorithms. The rate allocation algorithms first allocate rate to the smaller flows (i.e. with smaller rate requirement) before allocating to the larger flows, thereby ensuring that the smaller flows are not marginalized by the presence of the large flows. This ensures a certain degree of fairness among flows. Secondly, it can allocate rate according to the rate requirements of each flow, thereby allowing the rate available at a node to be fully utilized by the flows, and maximize the network performance.

In this section we described a general rate based congestion control system. To meet the flow control design goals of maximizing the channel/network utilization while avoiding congestion of the wireless channel, queue delay should be kept to a certain limit and throughput should be nearly to its maximized value. To accomplish this, we employ the *rt-Winf* estimation technique to provide precise rate feedback to the source node for adaptive flow control, allowing the network to maximize its performance, as found in Section 5.5. In the next Sections we describe how *rt-Winf* is integrated in both XCP and RCP.

5.3 *XCP-Winf* and *RCP-Winf* Congestion Control Mechanisms

To improve the performance of congestion control techniques in dynamic wireless networks, we define a new congestion control approach, based in XCP and RCP. The solution proposed adopts the explicit congestion control scheme enhanced with the interaction of a link and available bandwidth estimation mechanism. Both base XCP and RCP take the link capacity at the interface to compute the rate feedback. This introduces capacity overestimation which will generate inflated feedback, and the senders will send more traffic than the link can transfer. In the new approach, *rt-Winf* estimates the available bandwidth that will be used by the congestion control mechanisms to update their transmission rate. The estimation mechanism is integrated both at *senders*, *receivers* and *onlookers* nodes, as defined in IdleGap [1].

rt-Winf estimation values are obtained in the Medium Access Control (MAC) layer, and therefore, this information has to be accessed by *XCP-Winf* and *RCP-Winf*. The *rt-Winf* information is sent to the network layer through a simple, but effective, cross layer communication process. For this communication system, it is used a shared database architecture, with a set of methods to get/insert information in a database accessible by all protocol layers. One example of such architecture is the *MobileMan* cross layered network stack [2].

A generic *XCP-Winf/RCP-Winf* mechanism relies on the main functioning principles of XCP and RCP and is represented in Figure 5.3. *rt-Winf* inserts the available bandwidth and the link capacity information in the shared database and, then, *XCP-Winf* and

RCP-Winf access that information and use it to update their functions. In the following sub-sections we present the algorithms performed on both *XCP-Winf* and *RCP-Winf* mechanisms.

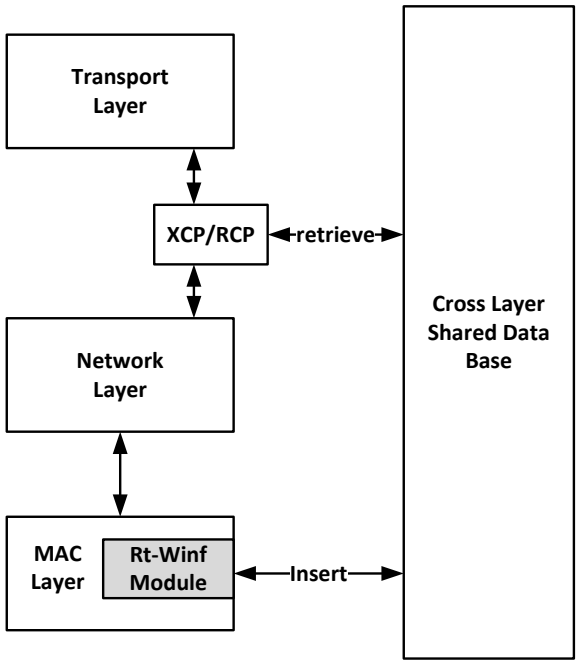


Figure 5.3: XCP-Winf/RCP-Winf System.

For a better understanding of the implemented functions described in the following sections, Table 5.1 shows the variable symbols used and its description.

5.3.1 *XCP-Winf* Functions

In this section we describe the proposed *XCP-Winf* functions. The main changes are performed on the XCP *Sender* and XCP *Router* functions. The XCP *Sender* uses the *Sender* state of the *rt-Winf* algorithm and the XCP *Router* uses the *Onlooker* state. The *XCP-Winf Receiver* is responsible for copying the throughput value required by the sender (in the packet), represented by $C_{desired}$, to the reverse feedback field of outgoing packets. A *XCP-Winf Receiver* operates in a similar way as a XCP *Receiver*. When acknowledging a packet, the *XCP-Winf* Receiver copies the congestion header from the data packet to the corresponding acknowledgment packet, and acknowledges the data packet in the same way as a TCP receiver.

When operating as a *XCP-Winf Sender*, several calculations need to be performed for each packet. In a *XCP-Winf* system, the necessary change in the throughput (TH_{change})

Variable	Description
TH_{change}	Calculated throughput change
$TH_{inverse}$	Inverse throughput
ϕ	Total input traffic
H_{RCP}	RCP header size
H_{IP}	IP header size
T_{pacing}	Pacing interval
Γ	RTT by throughput
S	Packet size
M	Maximum Segment Size
q	Queue size
$\breve{q}$	Instant queue
T_{RTT}	Sender estimate of the round-trip time (RTT)
$T_{\bar{RTT}}$	Average T_{RTT}
T_{ci}	Maximum allowable control interval
T_e	Queue estimation timer
T_m	Time between the transmission of two consecutive packets
T_{RTT}	Sender estimate of the round-trip time (RTT)
T_{DIFS}	IEEE 802.11 DCF Interframe Space Time
T_{SIFS}	IEEE 802.11 Short Interframe Space Time
$T_{backoff}$	IEEE 802.11 backoff time
C_{winf}	*rt-Winf* obtained Capacity
$C_{desired}$	Sender desired capacity change
C_{φ}	Amount of capacity shuffled
C_{change}	Allocated feedback change
C_w	Weighted link capacity
C_{extra}	Extra bandwidth consumed due to backoff
AB_{winf}	*rt-Winf* obtained Available Bandwidth
F_{Winf}	Aggregated feedback
R	RCP rate
p_f	Positive feedback factor
n_f	Negative feedback factor
f_p	positive feedback
f_n	negative feedback
Pc	Collision probability
CW_{Sender}	Sender congestion window

Table 5.1: XCP-Winf and RCP-Winf Variables.

is obtained by

$$TH_{change} = \frac{C_{desired} - C_{winf} \times 1000}{C_{winf} \times \frac{T_{RTT}}{M}} \tag{5.1}$$

where T_{RTT} is the current Round Trip Time (RTT) and M is the Maximum Segment Size (MSS) used on the network. C_{winf} is the link capacity obtained from *rt-Winf* and $C_{desired}$ is a desired change in the capacity value, that might be supplied by an application, or it might be the speed of the local interface. If no additional capacity is needed or desired, $C_{desired}$ will be equal to zero, and the packet will be immediately sent. If the value of TH_{change} exceeds the available bandwidth value AB_{winf}, obtained by *rt-Winf*, it is reduced to the current value of AB_{winf}.

The *XCP-Winf Router/Node* system operations in the *Onlooker* state are divided in four moments: when a packet arrives, when a packet departs, when the control interval timeout packet arrives, and when it is required to assess the persistent queue. Once more, *rt-Winf* available bandwidth and capacity are used in the calculations. On a packet arrival the total input traffic (ϕ) seen at the XCP queue is incremented by the size of the packet received. The sum of the inverse throughput is used for capacity allocation. The inverse throughput ($TH_{Inverse}$) uses the capacity value obtained by *rt-Winf* and the packet size (S), allowing to have more precise values when compared to standard XCP:

$$\sum TH_{inverse} += \frac{S}{C_{winf}} \tag{5.2}$$

Another important parameter is the sum of T_{RTT} by throughput; this parameter is used to obtain the control interval; on its calculation, it uses *rt-Winf* capacity values:

$$\sum \Gamma += \frac{T_{RTT} \times S}{C_{winf}} \tag{5.3}$$

This algorithm also checks if the round trip time (T_{RTT}) of each flow is exceeding the maximum allowable control interval (T_{ci}), with a default value of 0.5 seconds. If the round trip time exceeds the threshold, the maximum allowable control interval to avoid delays when new flows are started is updated to:

$$\sum \Gamma += \frac{T_{ci} \times S}{C_{Winf}} \tag{5.4}$$

When operating on the *Onlooker* state and when the control timer expires, *rt-Winf* values are used to determine the aggregated feedback (F). The aggregated feedback value depends on the link available bandwidth. The aggregate feedback represents the desired variation, on number of bytes, that the traffic is able to allow in a time interval, normally the average RTT. A *XCP-Winf Router* obtains the aggregate feedback [15] based in *rt-Winf* information:

$$F_{Winf} = \alpha \times (C_{winf} - AB_{winf}) - \beta \times \frac{q}{\bar{T}_{RTT}} \tag{5.5}$$

where α and β are constant parameters, q represents the queue value, $\bar{T}_{RTT}$ represents the average RTT, and $\frac{q}{\bar{T}_{RTT}}$ represents the persistent queue. AB_{Winf} is the available bandwidth value of the *rt-Winf* mechanism.

Then, as stated in [15], it is obtained the amount of capacity that will be shuffled (C_φ) in the next control interval. This allows new flows to acquire capacity in a full loaded system. This parameter is obtained by $max(0, 0.1 \times AB_{winf} - |F_{Winf}|)$. This will allow to obtain the increase - positive feedback scale factor (p_f) - or decrease - negative feedback scale factor (n_f) - on the traffic:

$$p_f = \frac{C_\varphi + max(F_{Winf}, 0)}{\sum \Psi} \tag{5.6}$$

$$n_f = \frac{C_\varphi + max(-F_{Winf}, 0)}{\phi} \tag{5.7}$$

When a packet departs, the node has to calculate a per-packet capacity change, that will be compared to the T_{change} value in the packet header. As stated in [15] "using the AIMD rule, positive feedback is applied equally per flow, while negative feedback is made proportional to each flow's capacity". The allocated feedback (C_{change}) for the packet is the positive per-packet feedback (f_p) minus the negative per-packet feedback (f_n). The positive feedback is obtained using p_f and the flows inter packet time, then,

$$f_p = p_f \times \frac{S}{C_{winf}} \tag{5.8}$$

The negative feedback is obtained using the packet size and the n_f:

$$f_n = n_f \times S \tag{5.9}$$

Thus, the capacity change requested is:

$$C_{change} = f_p - f_n \tag{5.10}$$

This value may be positive or negative. The node verifies whether the packet is requesting more capacity (via the packet's $C_{desired}$ field) than the node has allocated. If so, this means that the sender's desired throughput needs to be reduced and verified against *rt-Winf* available bandwidth (AB_{winf}). If the node has allocated more capacity than the available bandwidth, the desired throughput is updated to the *rt-Winf* available bandwidth. If the allocated capacity is less than the available bandwidth, the $C_{desired}$ field in the packet header is updated with the feedback allocation.

XCP-Winf, as XCP, needs to calculate a queue that does not drain in a propagation delay, that is the persistent queue. This queue is intended to be the minimum standing queue over the estimation interval. Each time a packet departs, the queue length ($\breve{q}$) is checked and the minimum queue size is calculated. When the queue estimation timer T_e expires, the persistent queue length is equal to the minimum queue value over the last T_e interval. For obtaining the duration of the T_e interval, it is used the capacity value of *rt-Winf*:

$$T_e = max(q, (\bar{T_{RTT}} - \frac{\breve{q}}{\frac{C_{winf}}{2}})) \tag{5.11}$$

Comparing *XCP-Winf* with XCP, it is possible to conclude that both use the same principles but differ in the way link capacity and available bandwidth are obtained and used.

5.3.2 *RCP-Winf* Functions

RCP-Winf updates RCP operations using *rt-Winf* link capacity values. A *RCP-Winf Receiver* operates in the same way as a standard RCP *Receiver*. The *RCP-Winf Receiver* just updates the RCP congestion header with the bottleneck rate, i.e. the rate of the most congested link, in the ACK packet, and then sends it to the sender.

RCP-Winf relies only on link capacity evaluation and there is no need for determining the aggregate feedback (F_{Winf}), thus there is also no need for explicitly using the available bandwidth. A *RCP-Winf* implementation also keeps unchanged the standard operations performed by RCP routers when a packet arrives and departs.

When operating as a sender, *RCP-Winf* needs to perform operations that allow it to modulate the congestion window. The *RCP-Winf Sender* will evaluate the desired change in throughput $C_{desired}$ through the value obtained in the ACK packet congestion header field and the link capacity obtained by the *rt-Winf*, gathered through the cross layer communication process:

$$C_{winf} - C_{desired} \leq 0 \tag{5.12}$$

According to this evaluation, *RCP-Winf* will update or not the desired change in throughput. If the evaluation returns a negative value, the $C_{desired}$ value will be updated to the C_{winf} value. Then, it modulates the sender congestion window (CW_{Sender}):

$$CW_{Sender} = \frac{C_{desired} \times T_{RTT}}{M + H_{RCP} + H_{IP}} \tag{5.13}$$

where H_{RCP} is the RCP header size (12 bytes) and H_{IP} is the IP header size. Then, it calculates the pacing interval (T_{pacing}), that will be used to send packets from the queue:

$$T_{pacing} = \frac{M}{C_{desired}} \tag{5.14}$$

When the rate timer of a *RCP-Winf* Router expires, the node first gets the *rt-Winf* capacity values. Then, it assumes that the aggregate incoming traffic rate is defined by the *rt-Winf* capacity value (C_{winf}). Next, it obtains the average round-trip time of the traffic that has arrived in the rate estimation interval (T_e). After that, the node updates the RTT estimate ($\bar{T_{RTT}}$), and then, it updates the rate value that will be offered to the flows (R) using the *rt-Winf* capacity:

$$R = R \times (1 + [\frac{(\frac{T_e}{\bar{T_{RTT}}}) \times (\alpha \times (C_w \times C_{winf} - C_{winf}) - \beta \times \frac{q}{\bar{T_{RTT}}})}{(C_w \times C_{winf})}] \tag{5.15}$$

The node tests the rate value and updates it. The rate value cannot be under a minimum rate value (R_{min}), or above a weighted (C_w) link capacity value:

$$\begin{aligned} \textbf{if } (R < R_{min}) &=> R = R_{min} \\ \textbf{else if } (R > C_w \times C_{winf}) &=> R = C_w \times C_{winf} \end{aligned} \tag{5.16}$$

Then, the node decides the length of the next rate estimation interval. Before finishing, the node resets the variables and restarts the timer. C_w controls the target link-utilization that can be any value in the range $0.95 < C_w < 1$. It is important to choose a value less than 1 as it allows some comfort to drain excess traffic before building up a queue. In extreme congestion scenarios, the minimum rate value allowed (R_{min}) is considered to be:

$$R_{min} = \frac{(00.1 \times MTU)}{\bar{T_{RTT}}} \tag{5.17}$$

where MTU is the maximum transmission unit. The result of the operation will be the value of the minimum rate.

A *RCP-Winf Router* also has to perform per packet operations, namely when a packet arrives and when a packet departs. Whenever a packet arrives carrying a valid round trip time, its value is added to the stored sum of RTTs and the number of packets carrying a valid RTT is incremented. This allows for a more precise calculation of the average RTTs. This is also the normal operation of a RCP standard system. *RCP-Winf Router* uses the obtained values of *rt-Winf* as their underlying value. When a packet requests an unspecified value or a value that exceeds the link capacity, the system is updated with the *rt-Winf* obtained capacity value, i.e.:

$$C_{desired} = C_{winf} \tag{5.18}$$

5.4 Collision Probability

The access to the shared medium in IEEE 802.11 [5] is performed using the Distribution Coordination Function (DCF) access method. The DCF access method is based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) principle in which a host wishing to transmit senses the channel, waits for a period of time and then transmits if the medium is still free. If the packet is correctly received, the receiving host sends an ACK frame after another fixed period of time. If the ACK frame is not received by the sending host, a collision is assumed to have occurred. The sending host attempts to send the packet again when the channel is free for the period augmented with a random interval of time. If there are multiple hosts attempting to transmit, the channel may be sensed busy, and in this case hosts enter the collision avoidance phase. It is possible to conclude that each access to the medium is independent from the previous one and, when a collision occurs, a transmitting host waits a random number of slots distributed geometrically. The

transmitting node is responsible for all collision dynamics; thus, all information regarding collision probability can be driven from the transmitting node.

To improve efficiency and reliability of *XCP-Winf* and *RCP-Winf* available bandwidth and link capacity inference mechanism, it is of extreme importance to account for the collision probability. As *rt-Winf* makes a real time analysis of the network status, it is important to know the extra time introduced when a node is waiting to transmit as a result of collisions. When a node wants to transmit and enters contention mode due to collision detection, this will affect link evaluation, leading to over-estimated capacity and available bandwidth values. To obtain more real values for the available bandwidth and link capacity, it is important to take in consideration all possible collisions in a period of time [119]. When a sender cannot transmit due to collision, the backoff mechanism is activated. This mechanism is also consuming bandwidth. C_{extra} is the extra bandwidth consumed due to backoff, as defined by equation 5.19:

$$C_{extra} = \frac{T_{DIFS} + T_{backoff}}{T_m} \tag{5.19}$$

where T_{DIFS} represents the IEEE 802.11 DCF Interframe Space obtained by $T_{SIFS} + (2 \times Slottime)$ [5], $T_{backoff}$ is the medium backoff time, and T_m is the time between the transmission of two consecutive packets. Whenever a station wants to transmit and the medium is busy, it has to wait some time before sensing again the medium. This time, defined in the IEEE 802.11 standard, is the $backoff$ time and is always known by the sender. When the Request-to-Send (RTS)/ Clear-to-Send (CTS) handshake is enabled, T_m can be obtained through:

$$T_m = 2 \times T_{DIFS} + 3 \times T_{SIFS} + T_{backoff} + T_{CTS} + T_{RTS} + T_{DATA} \tag{5.20}$$

where T_{CTS} represents time to transmit a CTS packet, T_{RTS} is the time of the RTS packet and T_{DATA} represents the data transmission time. The collision probability (Pc) can then be defined as $1 - C_{extra}$. Applying this result to the available bandwidth (AB) inference mechanism we have:

$$AB = Pc \times AB_{Winf} \Rightarrow AB = (1 - C_{extra}) \times AB_{Winf}$$
$$\Rightarrow AB = (1 - \frac{T_{DIFS} + T_{backoff}}{T_m}) \times AB_{Winf} \tag{5.21}$$

Equation 5.21 is only used when a node is in the *Sender* state of *XCP-Winf* and *RCP-Winf*. It is only possible to control and obtain the T_m and $T_{backoff}$ times in the *Sender* state. The use of information available in the *Sender* improves queue management, leading to less packet losses, and also to queue management improvement in the intermediate nodes. A node in the *Onlooker* state only updates its Network Allocation Vector (NAV), not being capable of determining the overall collision probability on the transmission side, as it is not aware of some important time control information. In the *Receiver* state, a node can only account with updated NAVs, that are important for determining the *Idle Rate*, which

do not consider the collision of the sender side. As collision can only occur when a node is sending packets, the information obtained by the sender is the one that can improve collision control. As *rt-Winf* is a cooperative inference mechanism, the sender is able to infer collision probability and, then, it can use that information to compare to the *rt-Winf* values of the receiver and use the ones that are most suitable, i.e the inferior capacity and available bandwidth values; this allows *rt-Winf* to use more precise information in its decision process, resulting in a more fair and accurate rate control.

5.5 Simulation Results

In this section we introduce the simulation setup to evaluate the performance of the proposed congestion control mechanisms and the simulation results. The results are obtained using the ns-2 [23] simulator. The standard ns-2 simulator supports neither *XCP-Winf* and *RCP-Winf*. Therefore, we have further improved ns-2 by adding *XCP-Winf* and *RCP-Winf* features. We have also modified ns-2 to support cross layer communication between *rt-Winf* and both *XCP-Winf* and *RCP-Winf*, using ns-2 class commands calls and procedures, thus simulating the *MobileMan* cross layer communication process. To evaluate the performance three metrics are used: throughput, delay and number of received packets. The proposed control mechanisms, *XCP-Winf* and *RCP-Winf*, are evaluated against the base protocols, TCP, XCP and RCP, and against the XCP-b, which is specially designed for wireless networks.

Different wireless mesh and ad-hoc scenarios were used. The parameters of the simulations are presented in Table 5.2. The configured default transmission range is 250 meters, the default interference range is 500 meters, and the channel data rate is 11 Mbps. For the data transmissions, it is used a File Transfer Protocol (FTP) application with packets of 1500 bytes or a Constant Bit Rate (CBR) application. The mobility is emulated through the ns-2 *setdest* tool to provide a random node movement and position pattern. We configure *setdest* with a minimum speed of 10 m/s, a maximum speed of 30 m/s, an average pause time between movements of 2 seconds and a topology boundary of 1000x1000 meters. All results were obtained from ns-2 trace files, with the help of *trace2stats* scripts [106] adapted to our own needs. The routing protocol used was the Destination-Sequence Distance-Vector (DSDV) [107]. The presented results show the mean values obtained through different simulation runs with different seeds, which include the 95% confidence interval.

5.5.1 Wireless Mesh Scenarios

In this section we analyze and compare the results of the congestion control approaches in the mesh topology scenarios. The mesh topologies defined comprise a grid of 5, 9, 12 and 16 fixed mesh nodes. In all mesh topologies, it is used a combination of 3, 4, 5, 6 and 7 mobile nodes. Figure 3.1 represents the same type of mesh topology that will be evaluated here. The mobile nodes are simultaneously sources and sinks.

Simulation Parameters	
Topology Area	1000m x 1000m
Simulation Time	300 sec.
Simulation repetition	30 times
Ad-Hoc Scenario Number of Mobile Nodes	8, 16, 32, 64, 128, 256
Mesh Scenario Number of Mobile Nodes	3, 4, 5, 6, 7
Mesh Scenario Number of Mesh Fixed Nodes	5, 9, 12, 16
Mesh Nodes Position	Random
Path Loss Model	Two Ray
Mobility Model	Random Way Point
Maximum Movement Speed	30 m/s
Minimum Movement Speed	10 m/s
Average pause time	2 s
Mac layer	IEEE 802.11
Propagation Model	Two Ray Ground
Routing Protocol	DSDV

Table 5.2: Simulation Environment.

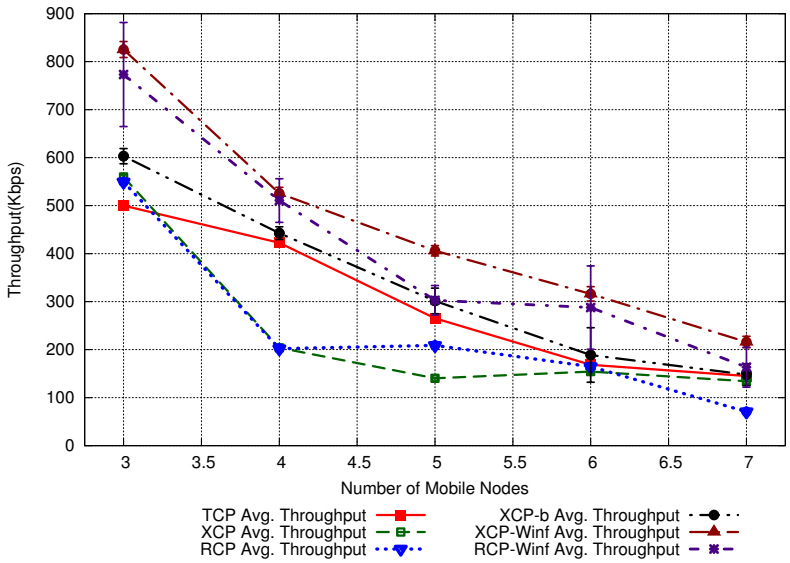


Figure 5.4: Average Throughput - 16 Mesh Nodes, Variable Number of Mobile Nodes.

Figure 5.4, Figure 5.5 and Figure 5.6 show the previously referred performance metrics for five different scenarios. In each scenario, it was used a fixed number of 16 mesh nodes (4x4 grid) and a variable number, from 3 to 7, of mobile nodes. Each mobile node, as previously stated, is simultaneously sending and receiving data.

Figure 5.4 clearly shows how throughput is improved in XCP and RCP with *rt-Winf*. The throughput values of *XCP-Winf* are $\sim 47\%$ to $\sim 60\%$ better than the ones with TCP,

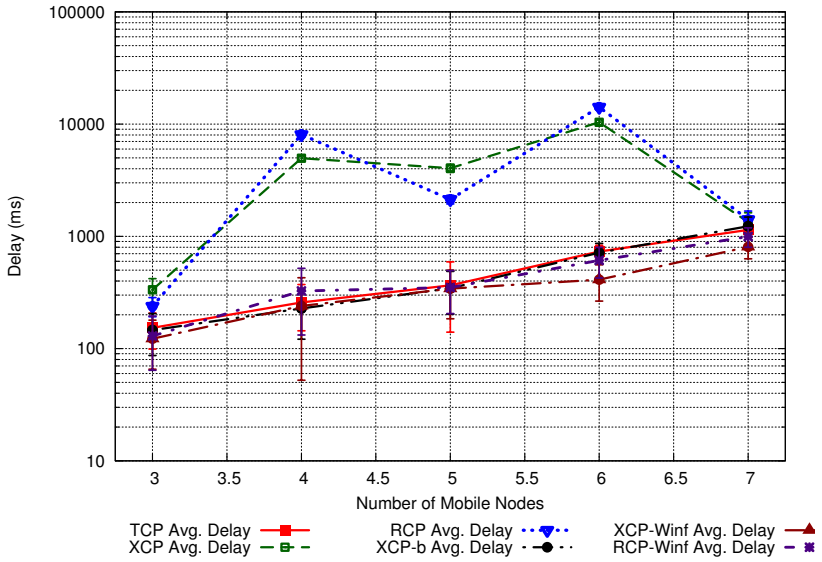


Figure 5.5: Average Delay - 16 Mesh Nodes, Variable Number of Mobile Nodes.

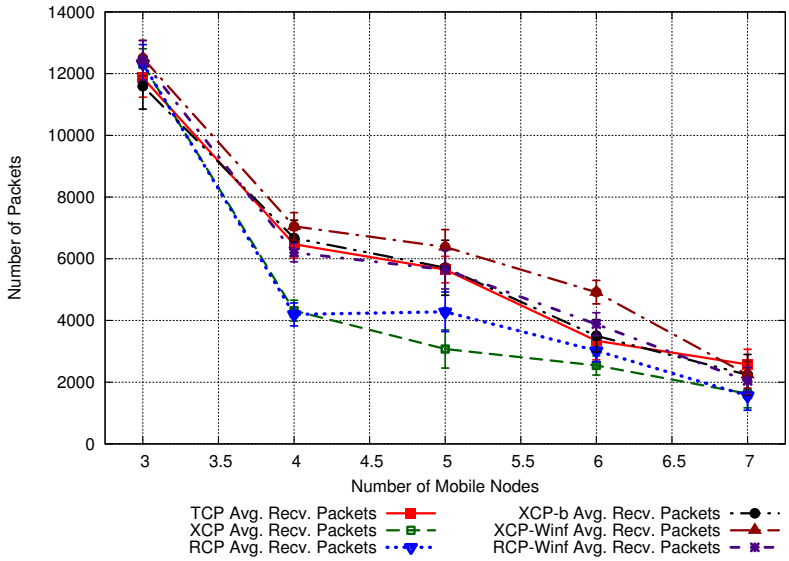


Figure 5.6: Average Received Packets - 16 Mesh Nodes, Variable Number of Mobile Nodes.

while with the base XCP, throughput values were worse than TCP. For *RCP-Winf*, the percentages, when compared to TCP, are between $\sim 17\%$ and $\sim 56\%$. In terms of received packets, as observed in Figure 5.6, it is also possible to see that with *rt-Winf* integrated, both XCP and RCP can receive more packets, which reflects a lower rate of lost packets. This is due to the fact that *XCP-Winf* and *RCP-Winf*, with accurate link capacity and available bandwidth, are using more efficiently the medium and improving each node queue management, thus, reducing congestion. Then, the nodes, and of course the network,

can transmit with a higher rate and less losses. As more packets are transmitted, more throughput is obtained and the medium is better used; it is possible to infer that both *XCP-Winf* and *RCP-Winf* are more stable and fair, since in the same conditions, it is possible to send more information with a higher rate. The delay values, Figure 5.5, are significantly decreased by one order of magnitude in the *XCP-Winf*/*RCP-Winf* versions, from around 1 sec to 100 msec.

The obtained results show that the integration of *rt-Winf* in XCP and RCP improves significantly their behavior, which makes XCP and RCP behave more efficiently and with better channel utilization, which also leads to less channel losses (more received packets) and to less congestion. The use of *rt-Winf* in the mesh nodes (onlooking state) makes the feedback mechanism more accurate, as all nodes in the network can determine available bandwidth and capacity, and send that information to the other nodes that are participating in the communication. XCP-b presents better results than TCP and standard XCP and RCP. However, they are outperformed by both *XCP-Winf* and *RCP-Winf*. XCP-b relies on the maximum buffer size of nodes and in indirect measurements to obtain the available bandwidth, and therefore, with the current scenario conditions, XCP-b is less efficient and less accurate than both *XCP-Winf* and *RCP-Winf*. While obtaining better throughput values than TCP, XCP-b does not provide good received packets results. This means that XCP-b is not capable of effectively using the medium, being less fair than *XCP-Winf* and *RCP-Winf*.

Figure 5.7, Figure 5.8 and Figure 5.9 show the same results for mesh scenarios with a fixed number of 7 mobile nodes and a variable number of mesh nodes (5, 9, 12, 16 mesh nodes).

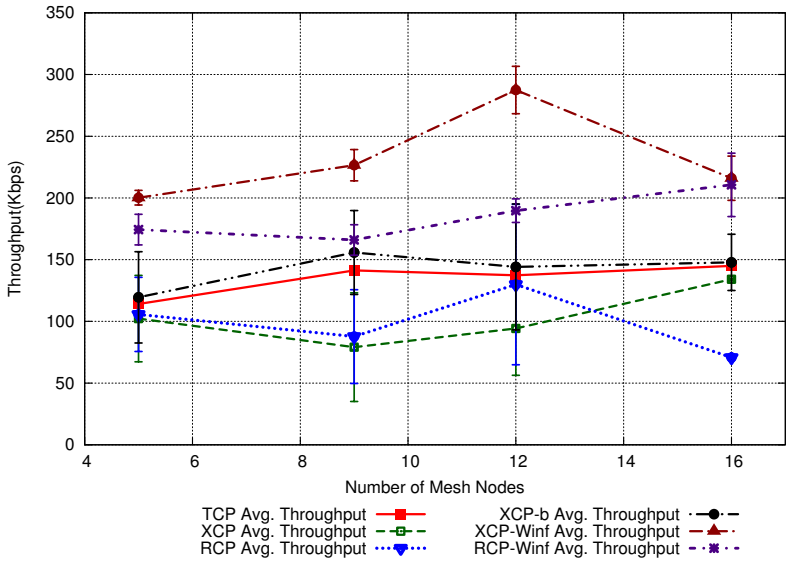


Figure 5.7: Average Throughput - Variable Number of Mesh Nodes, 7 Mobile Nodes.

The results show that *rt-Winf* significantly improves XCP and RCP behavior. *rt-Winf*

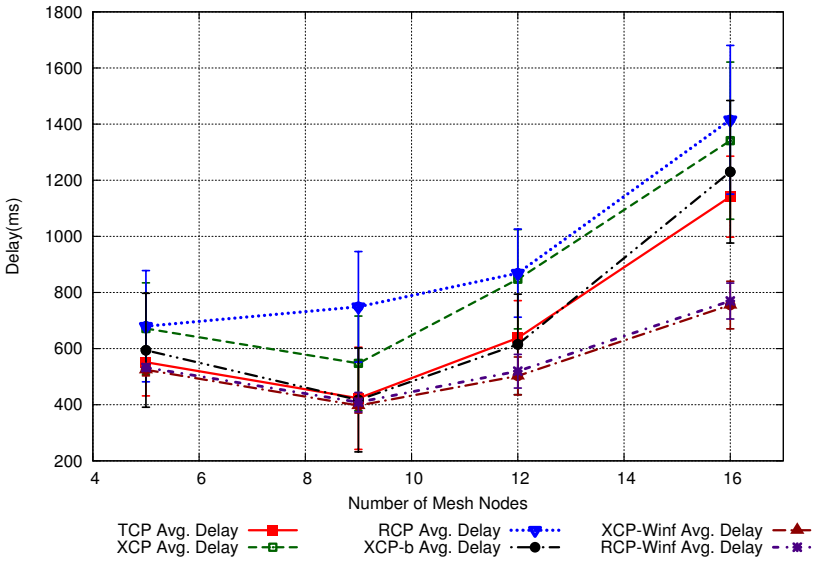


Figure 5.8: Average Delay - Variable Number of Mesh Nodes, 7 Mobile Nodes.

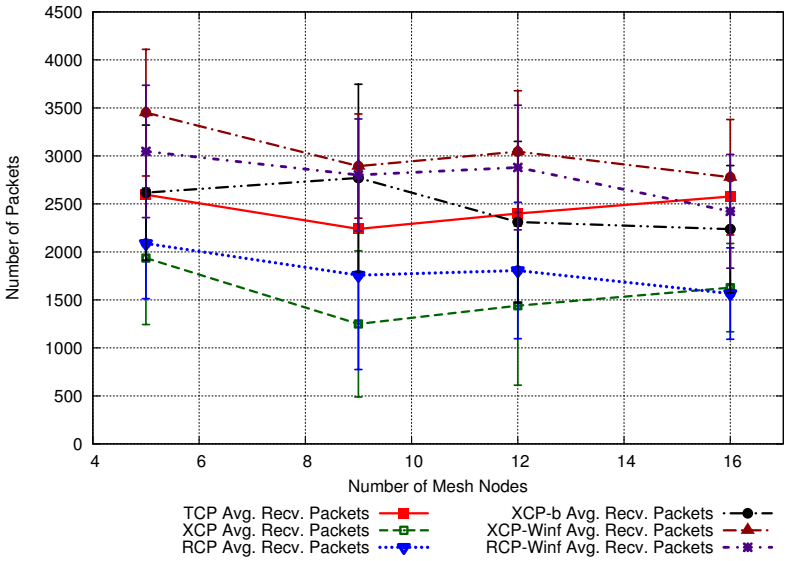


Figure 5.9: Average Received Packets - Variable Number of Mesh Nodes, 7 Mobile Nodes.

estimations, which are accurate and use node cooperation, allow XCP and RCP to use more efficiently and effectively the medium, clearly outperforming XCP-b and the standard TCP, XCP, RCP in terms of throughput, received packets and delay. *XCP-Winf* presents the best results, which is due to the fact that in its operations it is using both link capacity and available bandwidth estimation. This allows *XCP-Winf* to have good throughput results, from $\sim 50\%$ to $\sim 80\%$ better than the ones with TCP, while with the base XCP throughput, the values were worse than TCP. It is possible to observe that as the number of mesh nodes

increases, *RCP-Winf* results are better. This is due to the underlying RCP development for bursty traffic. With more mesh nodes, more control messages are exchanged, more burtsy traffic is present and *RCP-Winf* behaves more efficiently. XCP-b results become worse with the increase of mesh nodes and the consequent control messages. As XCP-b uses complex heuristics to obtain link capacity which are based on the maximum buffer size of nodes, the nodes buffers are heavily utilized when the number of nodes increases, which makes XCP-b less efficient and accurate. Figure 5.8 and Figure 5.9 show that, with 16 mesh nodes, XCP-b has worse performance than TCP.

The use of multimedia applications, such as audio and video streaming, videoconferencing, Voice-over-IP (VoIP) and multi-player games, increases every day. Therefore, it is important to evaluate *XCP-Winf* and *RCP-Winf* against such type of traffic. Thus, we simulated VoIP streams with G.729 codec characteristics using 64 kbps CBR User Datagram Protocol (UDP) traffic to evaluate *XCP-Winf* and *RCP-Winf* in the 16 mesh nodes scenario and variable number of mobile nodes. Figure 5.10, 5.11 and 5.12 show the obtained results. Without *rt-Winf* enabled, XCP obtains better results than RCP for a lower number of mobile nodes. This is due to the fact that RCP was developed having in mind Internet burst traffic. With less mobile nodes exchanging information, the number of collisions is lower, and less retransmissions and burst traffic are present in the network. It is also possible to conclude that both XCP and RCP are not evaluating correctly the link capacity and do not have the necessary mechanisms to overcome this situation.

The results show that *XCP-Winf* has the best overall performance, improving throughput in more than 60% when compared to XCP. We may conclude that the rate estimation mechanism of XCP is not accurate, and XCP is not using effectively the medium, transmitting less packets and with higher delay values. *rt-Winf* allows *XCP-Winf* to use more efficiently the medium, and more fairly, as more packets are transmitted with better delay values. The rate estimation technique of *XCP-Winf*, provided by *rt-Winf*, allows it to provide a better and more precise congestion control, even when the traffic is UDP based. The results also show that, as more nodes are present in the network, the better is *RCP-Winf* performance. This reflects *RCP-Winf* base development for bursty traffic. The CBR application is sending data at a constant rate; with more mobile nodes sending data, more collisions will occur and more bursts of traffic will be present in the network. This situation will allow RCP to react more precisely and reach higher throughput. The results also show that XCP-b is better suited for the scenarios where the number of mobile nodes is small. XCP-b estimation mechanism is not suited for situations of high mobility and high network utilization.

5.5.1.1 Building Scenario Results

For a more real evaluation, we defined a new mesh network scenario to simulate a public building, with public services and a public garden (Figure 5.13). When compared to Table 5.2, the different parameters of this scenario are the following: the number of mobile nodes is 10, 20 and 30; the number of fixed mesh nodes is 6 and two different types of flows are used. In this scenario mobile nodes start to transmit in a random way

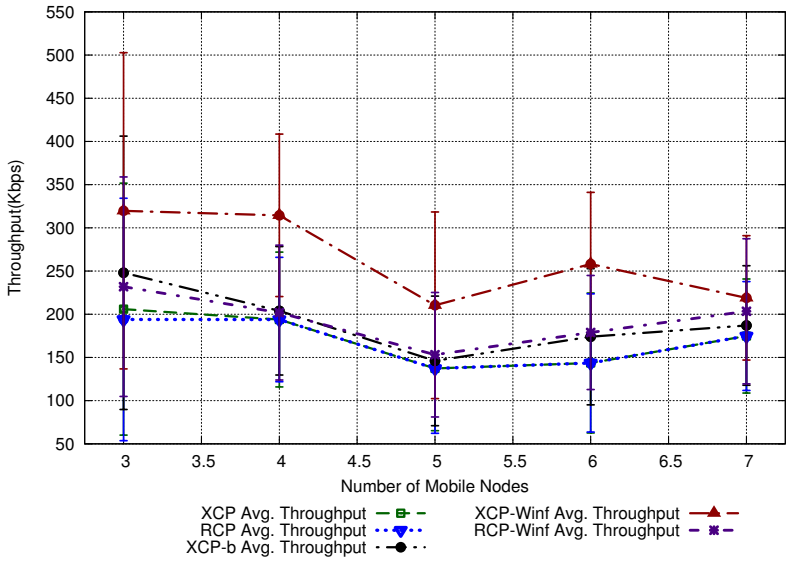


Figure 5.10: Average CBR Throughput - 16 Mesh Nodes, Variable Number of Mobile Nodes.

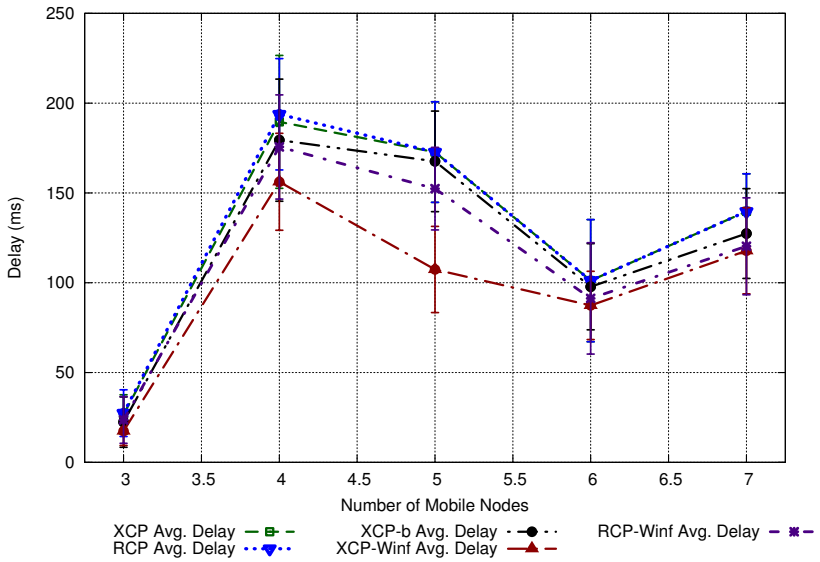


Figure 5.11: Average CBR Delay - 16 Mesh Nodes, Variable Number of Mobile Nodes.

and their transmission lasts 240 seconds. The mesh nodes position is randomly defined in the inside area. A randomly chosen mesh node is shutdown for 100 seconds during the simulation period. This includes larger dynamics in the network. We simulated FTP and CBR traffic applications. For the FTP traffic application we configured flows with packets of 1500 bytes. For the CBR traffic, two types of flows are used, to represent a light traffic flow (4 CBR 64 Kbps flows, sent each 400 ms) and a heavy traffic flow (4 CBR 128 Kbps

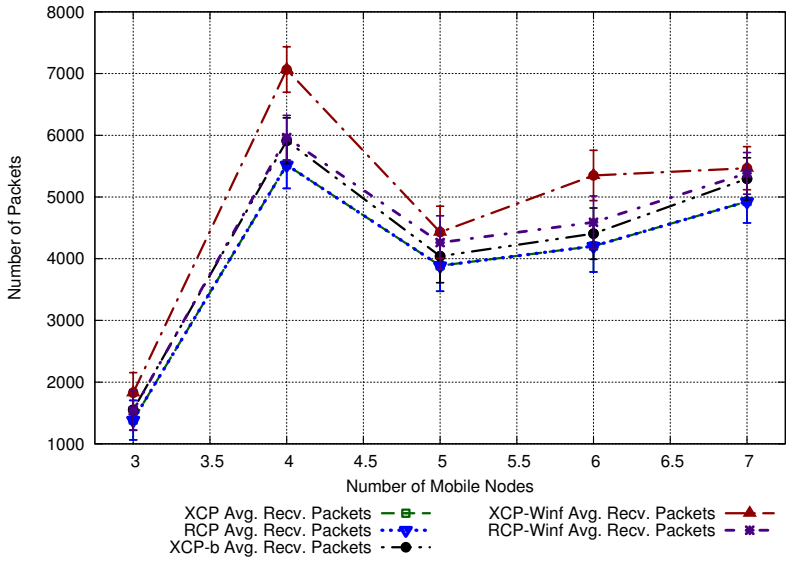


Figure 5.12: Average CBR Received Packets - 16 Mesh Nodes, Variable Number of Mobile Nodes.

flows, sent each 100 ms).

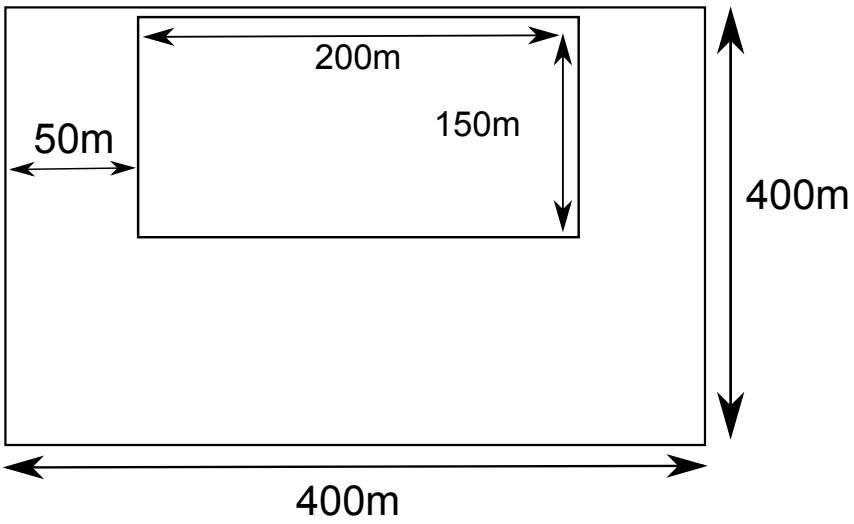


Figure 5.13: Building Layout Simulation.

The results for the FTP application are shown in Figure 5.14, Figure 5.15 and Figure 5.16. It is possible to observe that *XCP-Winf* and *RCP-Winf* have better results than TCP, XCP, RCP and XCP-b. The use of *rt-Winf* allows both *XCP-Winf* and *RCP-Winf* to adapt quickly to the network conditions, and, thus, to increase network performance. The *rt-Winf Onlooker* state allows nodes that do not participate in the communication to constantly evaluate the network performance, when network conditions change; when

these nodes start to participate in the communication, they have a previous knowledge of network conditions that make them to react faster and with higher efficiency. This also results in a more fair network usage. XCP-b, while having good results, is not adapting its behavior efficiently to the network conditions, as it is using indirect parameters for its link estimations. *XCP-Winf* obtains the best results, as it uses both link capacity and available bandwidth estimations for congestion control. *XCP-Winf* can then transmit more traffic, with node queues better utilized, resulting in a more effective congestion control with better throughput, less delay and more received packets. *XCP-Winf* performance results are $\sim$ 25% to $\sim$ 48% better than XCP-b, and $\sim$ 37% to $\sim$ 65% better than TCP. *RCP-Winf* obtains $\sim$ 12% to $\sim$ 38% better results than XCP-b, and $\sim$ 26% to $\sim$ 53% better results than TCP.

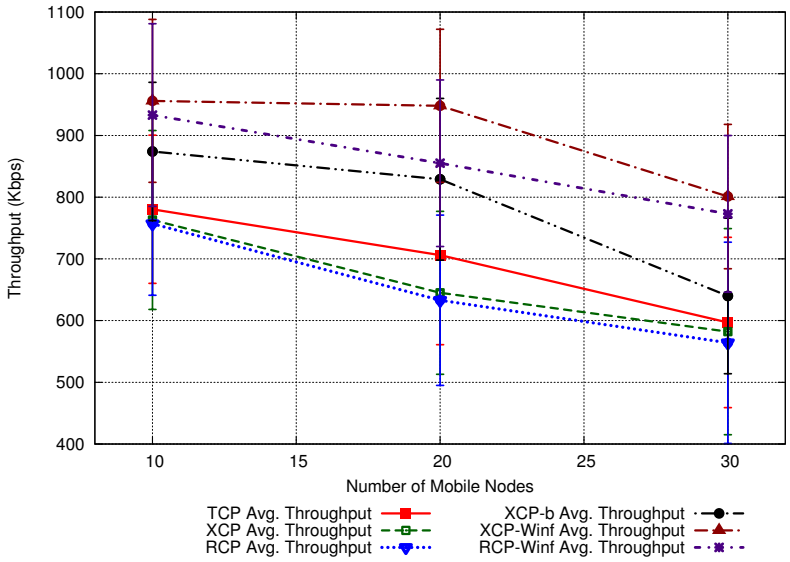


Figure 5.14: Public Building FTP Application Throughput.

Comparing FTP results of both the wireless mesh scenario and the public building scenario, we can observe that *XCP-Winf* and *RCP-Winf* perform better than the other proposals in the wireless mesh scenario. In the building scenario we have more mobile nodes, thus less route breakages as more nodes are available. This allows TCP, XCP and RCP to have better performance results. In the wireless mesh scenario the small number of nodes allows *rt-Winf* cooperation process to provide a better network utilization, as less feedback information is needed.

The obtained results of the light CBR traffic flows are shown in Figure 5.17, Figure 5.18 and Figure 5.19; the results of the heavy CBR traffic flows are presented in Figure 5.20, Figure 5.21 and Figure 5.22. As can be observed from the results, the integration of *rt-Winf* in both XCP and RCP improve their standard behavior. Since the nodes that are not participating in the communication enter the *Onlooker* state and evaluate the network performance, it is possible to have a state by state and rate by rate overall performance

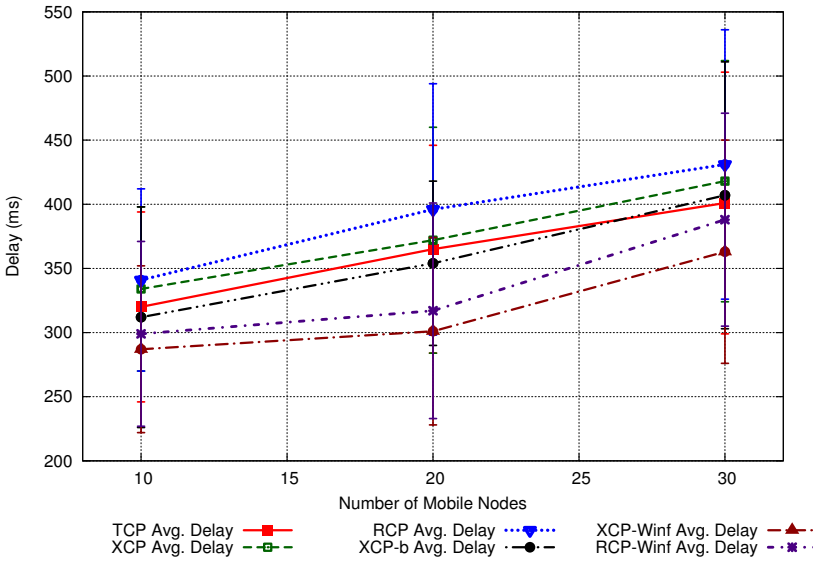


Figure 5.15: Public Building FTP Application Delay.

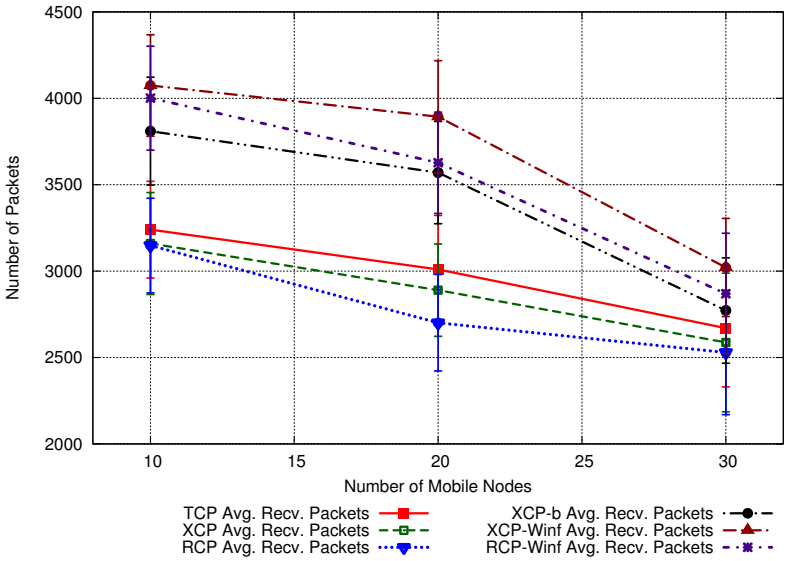


Figure 5.16: Public Building FTP Application Received Packets.

evaluation. As *rt-Winf* uses three different states and network cooperation, it is possible to have a more effective and efficient performance evaluation. This results in a more efficient evaluation and use of the channel capacity. As *XCP-Winf* and *RCP-Winf* also have more capability to adapt to the changing conditions of the network, this is expressed in better transmission rates and better channel usage. XCP-b has again poor performance for high load scenarios: XCP-b is not taking into consideration packet loss, considering packet loss as a buffer overflow, thus having a more inefficient behavior and introducing unnecessary

capacity slowdowns on the network.

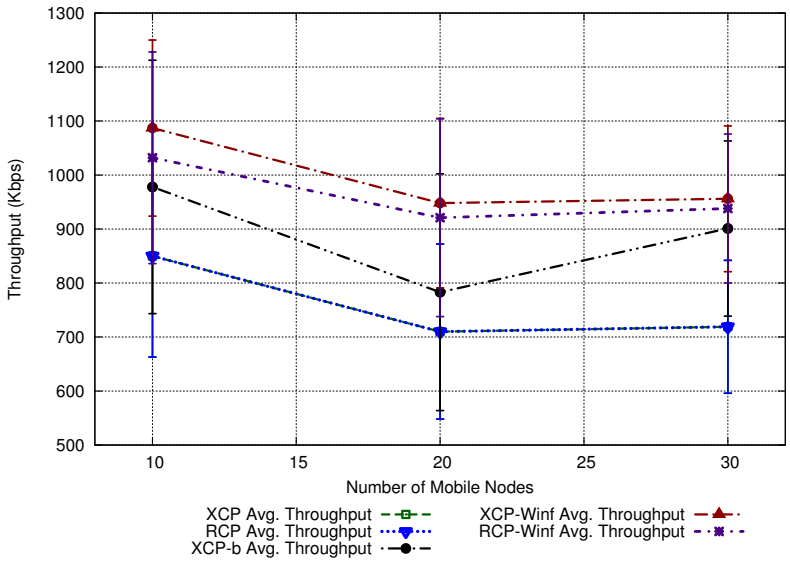


Figure 5.17: Public Building CBR Light Flow Throughput.

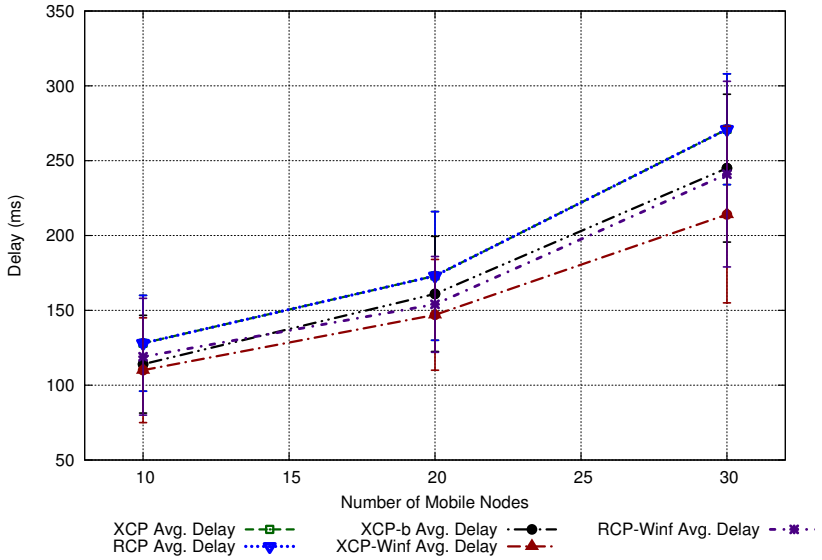


Figure 5.18: Public Building CBR Light Flow Delay.

When observing both CBR heavy and light flows results, it is clear that both *XCP-Winf* and *RCP-Winf* results are improved in the CBR light flow scenario. This is a consequence of the number of packets "in flight", in the light flow scenario, that allows the feedback mechanism to be more efficient, making both *XCP-Winf* and *RCP-Winf* to have a better performance. Another important aspect to be taken into consideration is

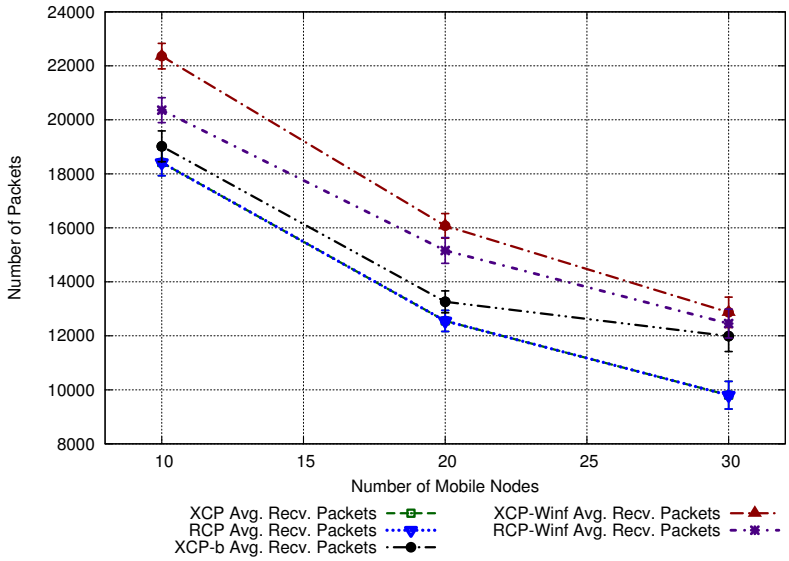


Figure 5.19: Public Building CBR Light Flow Received Packets.

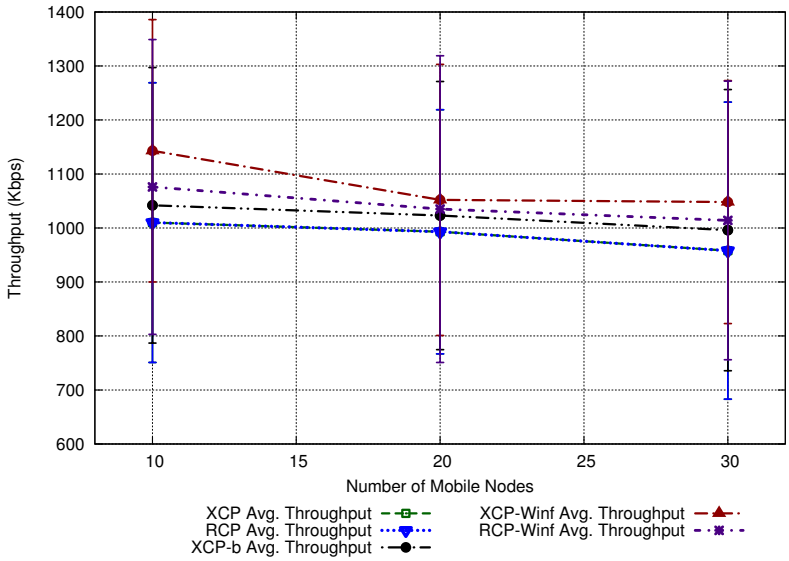


Figure 5.20: Public Building CBR Heavy Flow Throughput.

that both XCP and RCP standard algorithms suffer of slow allocation of bandwidth to new flows in a situation with high link utilization. As *XCP-Winf* and *RCP-Winf* rely on those algorithms, that is more noticed in high flow scenarios.

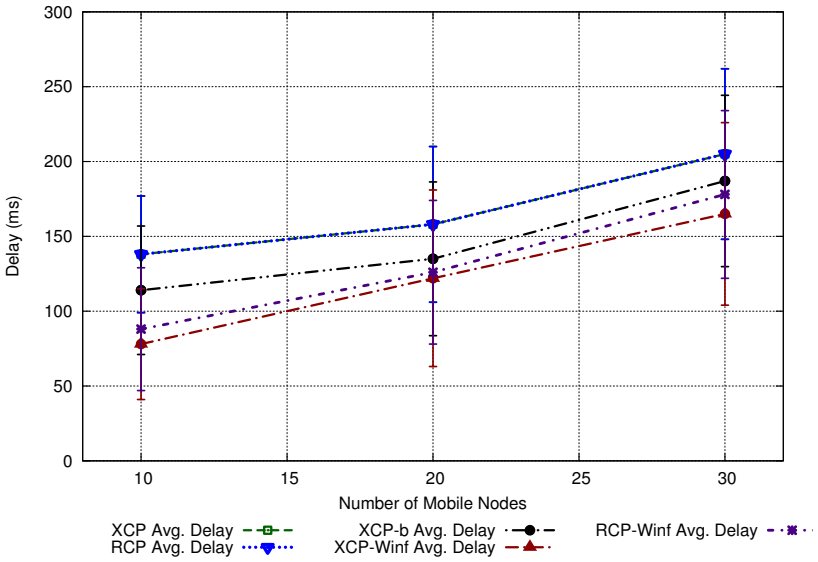


Figure 5.21: Public Building CBR Heavy Flow Delay.

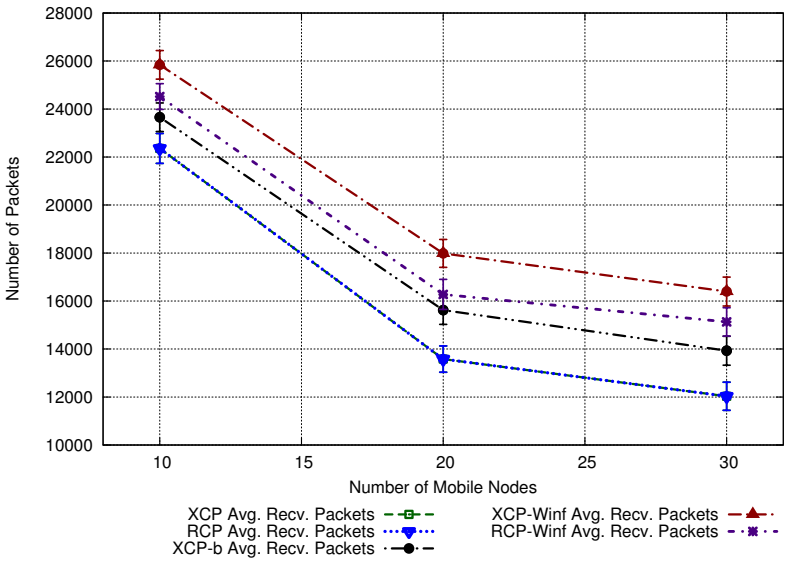


Figure 5.22: Public Building CBR Heavy Flow Received Packets.

5.5.2 Wireless Ad-hoc Scenarios Results

The congestion control approaches are also evaluated in ad-hoc scenarios using FTP and VoIP traffic. The scenarios are composed by 8, 16, 32, 64, 128 and 256 nodes; for each scenario there are 4, 8, 16, 32, 64 and 128 simultaneous flows: that is, the scenario with 8 nodes generates 4 flows, the 16 nodes scenario has 8 flows and so on, up to 256 nodes and 128 flows. The flows are randomly generated through the ns-2 *gencbr.tcl* tool. The

mobility was also dynamically generated through different seed values on the *setdest* tool of ns-2. The *setdest* tool was configured with a minimum speed of 10 m/s, a maximum speed of 30 m/s and an average pause time between movements of 2 seconds.

The FTP application results are shown in Figure 5.23, Figure 5.24 and Figure 5.25.

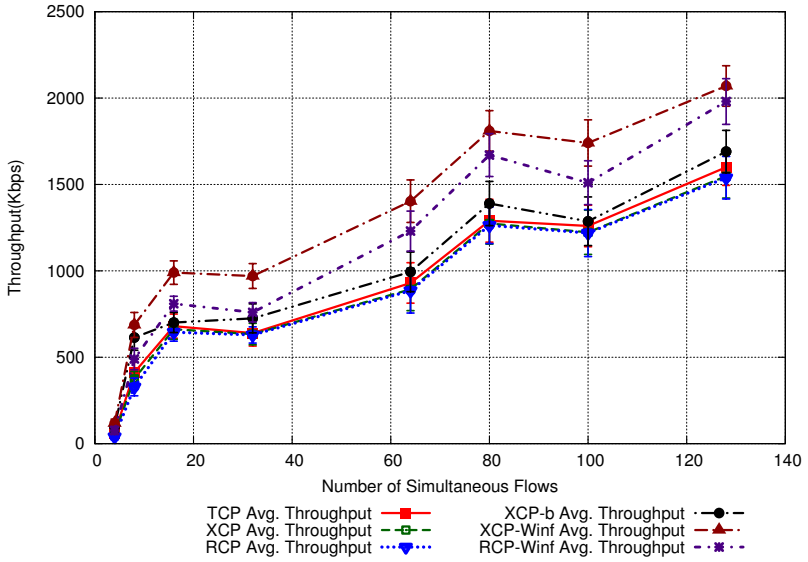


Figure 5.23: FTP Ad-Hoc Scenario: Variable Number of Flows Throughput.

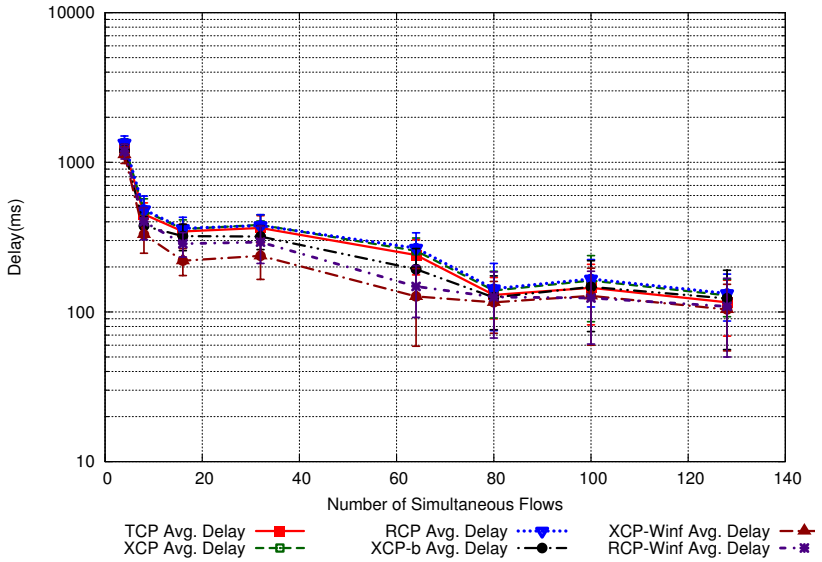


Figure 5.24: FTP Ad-Hoc Scenario: Variable Number of Flows Delay.

In these particular scenarios, *XCP-Winf* and *RCP-Winf* obtain good delay and received packet results. These results indicate that those two mechanisms provide good congestion

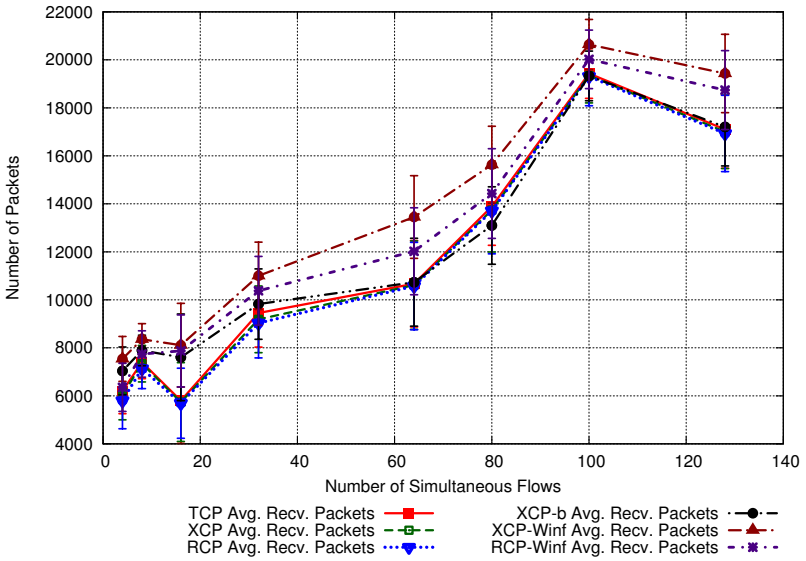


Figure 5.25: FTP Ad-Hoc Scenario: Variable Number of Flows Received Packets.

control, not saturating the network. This allows both mechanisms to be more fair and efficient than TCP, XCP, RCP and XCP-b. Using *rt-Winf* as the underlying estimation tool, it makes XCP and RCP to have a good network perspective, thus, they can control flows more accurately resulting in better network performance. TCP, XCP, RCP and XCP-b under utilize the network, as their mechanisms are not controlling effectively the network. As congestion occurs more often, packet losses increase and their performance results become worse. Due to its underlying control mechanisms XCP-b obtains good results when the network is not heavily utilized. This allows XCP-b to perform its heuristics more efficiently and precisely - as they depend on less indirect parameters.

Comparing the ad-hoc scenario FTP results with the FTP results of the wireless mesh and public building scenarios, we can observe that *XCP-Winf* clear outperforms the other proposals, while *RCP-Winf* has better results, when compared to the other proposals in the ad-hoc scenarios. In terms of throughput improvement, *XCP-Winf* results are better in the wireless mesh and public building scenarios. In those scenarios, we have fixed mesh nodes that are in the *Onlooker* state of *rt-Winf*, their information allows *XCP-Winf* to improve network performance. *RCP-Winf* as RCP is more efficient with traffic bursts. In the ad-hoc scenarios as we have more mobile nodes, more route changes and more control messages, thus traffic bursts occur more often making *RCP-Winf* to perform better.

The obtained results for the VoIP traffic are presented in Figure 5.26, Figure 5.27 and Figure 5.28.

From the obtained results, it is possible to conclude that standard XCP and RCP have the same behavior when the traffic is UDP; however, the integration of *rt-Winf* makes them react differently as they both use the information from the MAC sublayer in a different way. It is also possible to see that with *rt-Winf* integrated, both XCP and RCP can receive

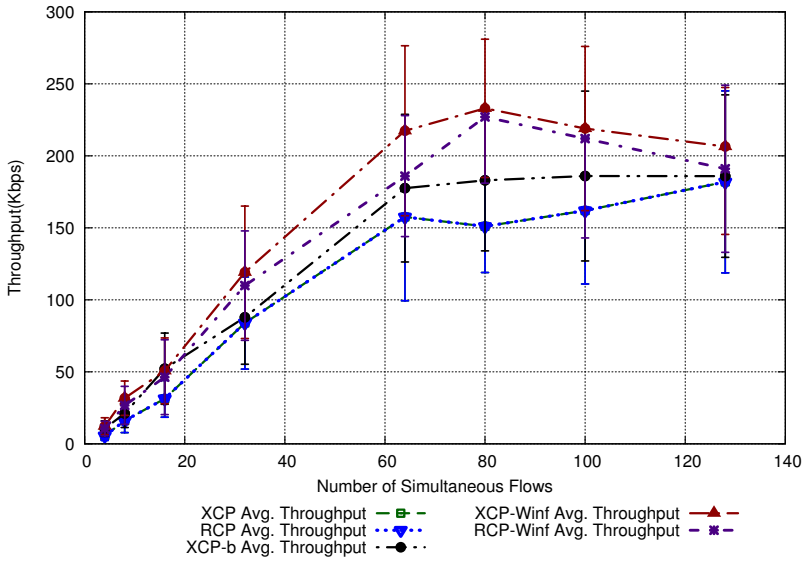


Figure 5.26: VoIP Ad-Hoc Scenario: Variable Number of Flows Throughput.

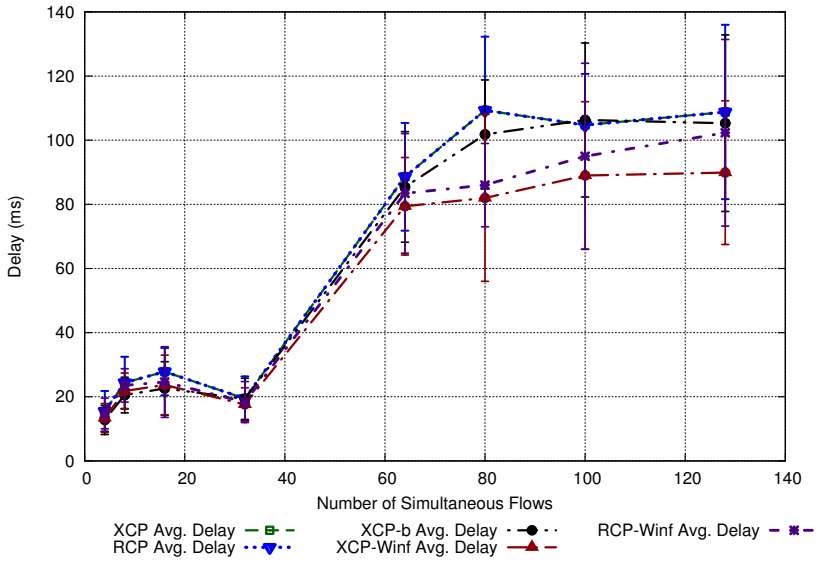


Figure 5.27: VoIP Ad-Hoc Scenario: Variable Number of Flows Delay.

more packets, which reflects a lower rate of lost packets. It must be noticed that the results also reflect the mobility randomness, where more nodes are in each other influence area. Another factor that is influencing the results is the routing information and the exchanged routing messages: as flows increase, the collisions and delay also increase, which is also reflected in throughput values. Once more XCP-b obtains good results when the network is not heavily utilized, where XCP-b increases its available bandwidth, and consequently, its rate. However, with more nodes and flows in the network, XCP-b behavior is less efficient

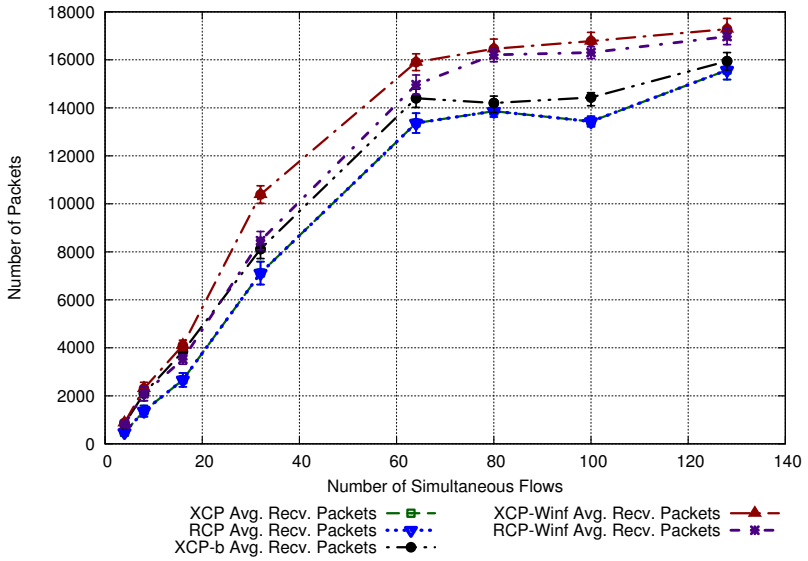


Figure 5.28: VoIP Ad-Hoc Scenario: Variable Number of Flows Received Packets

due to the higher number of losses, reducing the flow rate.

The comparison of the ad-hoc VoIP results with the mesh scenario results shows that *XCP-Winf* outperforms the other proposals, including the *RCP-Winf* results. As base RCP, *RCP-Winf* is better suited for traffic bursts or short flows; when the flows are larger, *XCP-Winf* is more efficient and effective allowing to have better network performance. Another important consideration is that, in the mesh scenarios some nodes are only in the *rt-Winf Onlooker* state: this allows to have a more effective network knowledge and to not under-utilize the available resources. This contributes to a more efficient available rate allocation and to improve network usage and performance.

5.5.3 Collision Probability Results

The influence of the collision probability is also analyzed in the performance evaluation. We carried out a set of new simulations, in the building simulation layout, changing the speed of the nodes. The new simulations were defined in three different scenarios: with no mobility, *Normal Mobility* which represents a random velocity with a maximum speed of 30 m/s, and mobility with a maximum speed of 100 m/s. Figure 5.29 shows the obtained results for *XCP-Winf* and Figure 5.30 shows the results with *RCP-Winf*. We observe that, with increased speed, the inclusion of collision probability improves the performance: with higher speed and more traffic, more collisions will occur, making the collision rate an important factor in the performance evaluation. Using the collision probability parameter allows, thus, to achieve better medium usage that is then reflected in more efficient available bandwidth and capacity evaluation. Collision probability improves throughput performance results from ~ 6.5 % in *RCP-Winf* to ~ 8.5 % in *XCP-Winf* when nodes are moving very fast and the traffic is saturating the network. In such conditions, the collision

probability rate is an important factor.

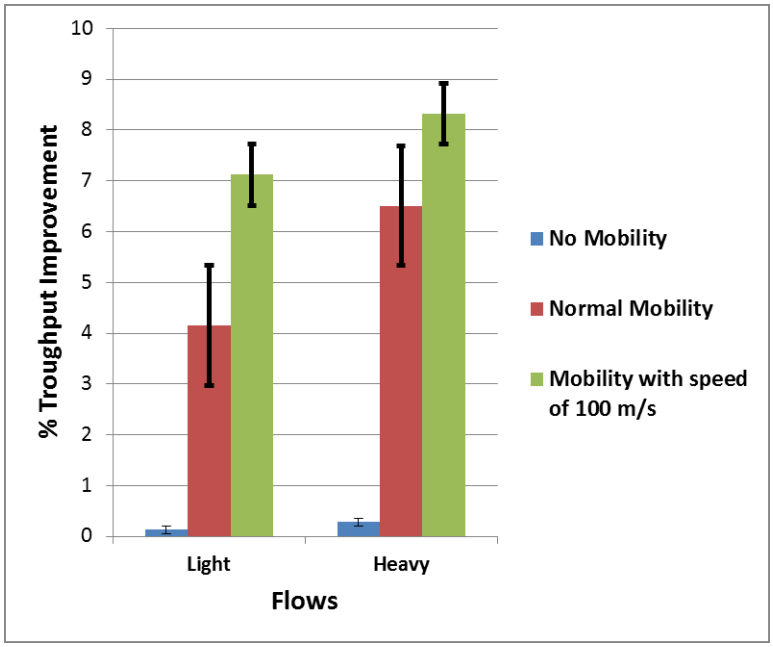


Figure 5.29: *XCP-Winf* collision probability.

5.5.4 Utility Results

As TCP is the most used and deployed congestion control protocol on the Internet, it is important (as described on [120]) to analyze how *XCP-Winf* and *RCP-Winf* flows interact and compete with TCP. For this purpose, we use the average data rate over time for each flow, thus allowing to observe how bandwidth is being managed between TCP and the *Winf* proposals. This is called *utility* of a networking protocol.

Two scenarios were defined: one using *XCP-Winf* and TCP, and another using *RCP-Winf* and TCP. The two scenarios consist of a 1000m x 1000m area, divided on three distinct parts: an area of 250m x 250m where there are two mobile node sources, one with TCP and the other with *XCP-Winf* or *RCP-Winf*; a middle area of 500m x 500m with two mobile nodes with the *rt-Winf* mechanism activated (the average data rate is measure on these two nodes, as they will have TCP and *Winf* like flows competing); finally, another area of 250m x 250m for the mobile nodes sinks. Figure 5.31 represents the scenarios used. Each source generates two FTP flows with packets of 1500 bytes. The simulation lasts for 120 seconds.

Figure 5.32 and Figure 5.33 show the obtained results. We to observe that, on both situations, the TCP flow grows faster and gains more bandwidth on the beginning. In the beginning, TCP is trying to fill the nodes queues, expecting that packet loss due to

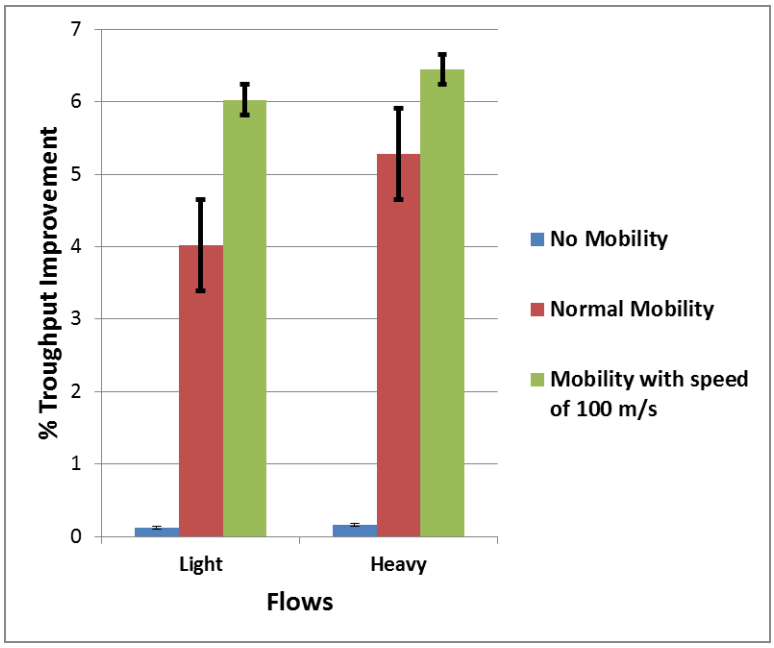


Figure 5.30: *RCP-Winf* collision probability.

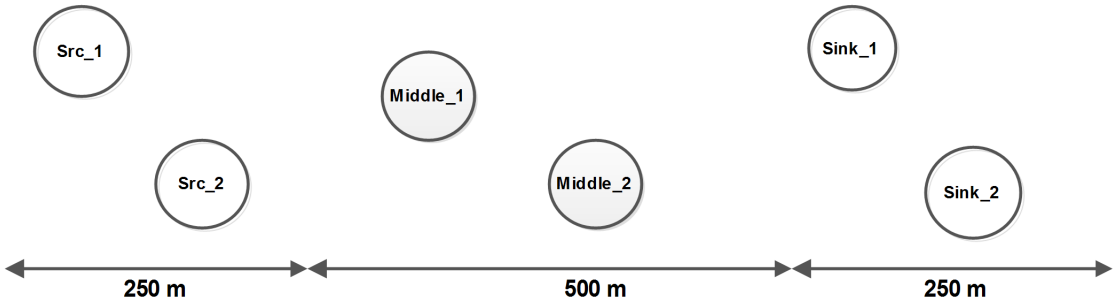


Figure 5.31: Utility scenario.

queue overflow will signal congestion. However, *RCP-Winf* and *XCP-Winf* are evaluating, and measuring consistently, how the network is behaving and adjusting the bandwidth requirements to that information. As *RCP-Winf* is based on RCP with its bursty traffic development, it has a more unstable behavior during the first phase of the simulation, but as more traffic enters the network, *RCP-Winf* becomes more stable.

The results also show that the *winf* mechanisms are TCP friendly on the long term and are adjusting to the unfairness nature of TCP. *XCP-Winf* reacts earlier to these constraints, and starts to compete for the same bandwidth as TCP earlier than *RCP-Winf*. However, it must be noticed that the results show that both *RCP-Winf* and *XCP-Winf* take some

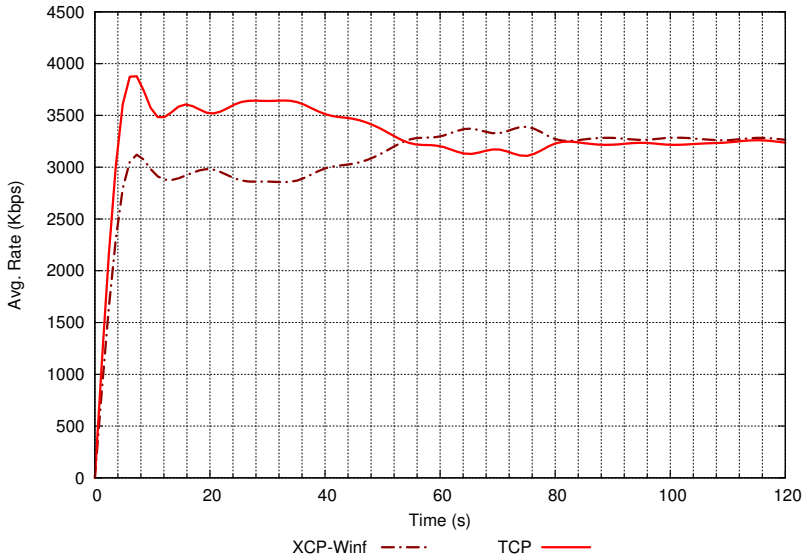


Figure 5.32: *XCP-Winf* utility results.

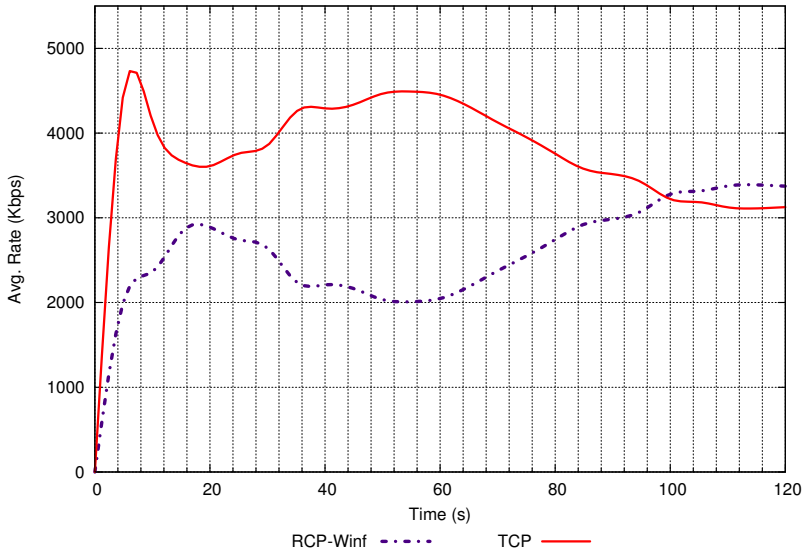


Figure 5.33: *RCP-Winf* utility results.

time to allow a fair share of bandwidth between their native flows and TCP. It is advisable that this fair share is obtained as quickly as possible, thus allowing a more efficient share and coexistence of network resources.

5.6 XCP-Winf and RCP-Winf TCP Aware Improvement

As TCP friendliness is an important issue when designing a new congestion control protocol, and accordingly to the results of the previous section, it is important to make *XCP-Winf* and *RCP-Winf* TCP aware and, thus, make them more TCP friendly. The modified versions of *XCP-Winf* and *RCP-Winf* try to reflect the existence of TCP flows. Therefore, to obtain *Winf* like protocols that are TCP friendly, *rt-Winf* link capacity has to reflect the presence of TCP flows. *rt-Winf* link capacity is, then, updated as follows:

$$C_{util} = \left(\frac{NF_{Winf}}{NF_{Winf} + NF_{TCP}}\right) \times C_{Winf} \tag{5.22}$$

where C_{util} is the update link capacity with TCP awareness, NF_{Winf} represents the number of *Winf* like flows, and NF_{TCP} represents the number of TCP flows that go through a node. It is clear that if the number of TCP flows is equal to zero, than $C_{util} = C_{Winf}$.

New simulations were conducted to evaluate the TCP awareness improvement. Figure 5.34 and Figure 5.35 show the results for the two FTP flows with packets of 1500 bytes.

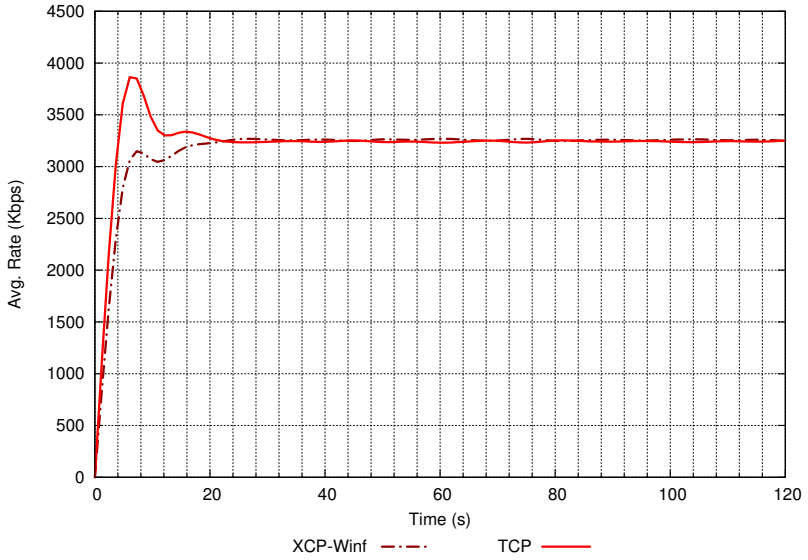


Figure 5.34: *XCP-Winf* utility results, TCP aware.

It is possible to observe that the introduction of these modifications allows a more fair data rate share between flows. This allows TCP to work more naturally and to behave more efficiently. Another important observation is that *XCP-Winf* converges more quickly than *RCP-Winf*, as it is not dependent of bursty traffic for its efficient operation.

For a better evaluation of the new *XCP-Winf* and *RCP-Winf* improvement, new simulations were conducted using the same network topology, but now the sender nodes generate concurrently 8 *Winf* like flows and 4 TCP flows. Figure 5.36 and Figure 5.37 show the

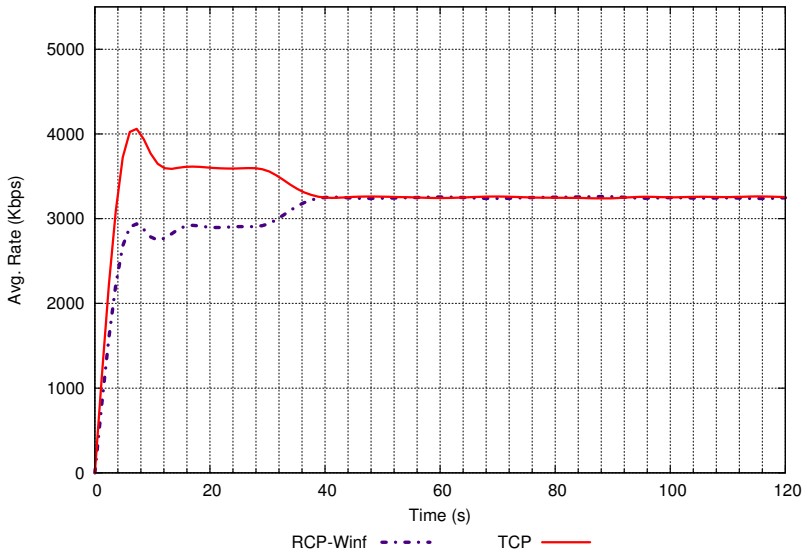


Figure 5.35: *RCP-Winf* utility results, TCP aware.

results without the TCP awareness update. The obtained results confirm that both *RCP-Winf* and *XCP-Winf* take some time to allow a fair share of bandwidth between their native flows and TCP.

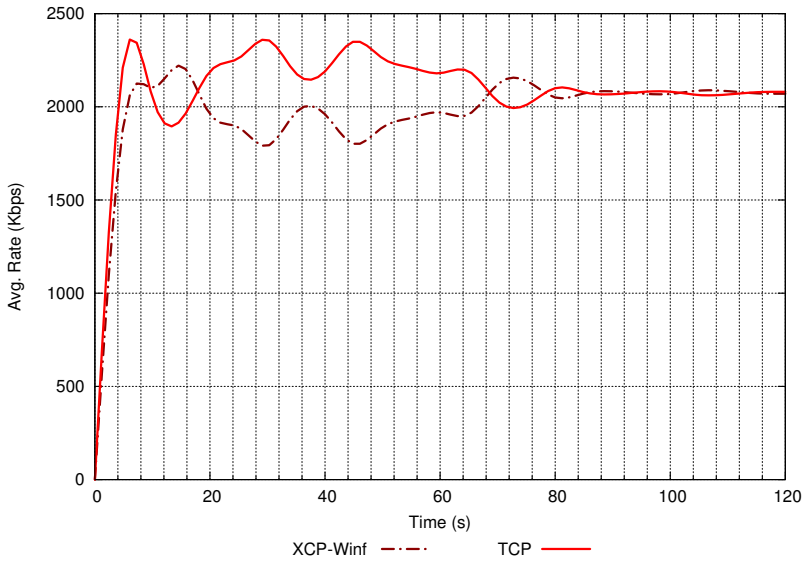


Figure 5.36: *XCP-Winf* utility results (8 Winf flows - 4 TCP flows).

The obtained results with the TCP aware update, Figure 5.38 and Figure 5.39, clearly show that the TCP aware modification achieves better utility fairness among the different flows. Both *XCP-Winf* and *RCP-Winf* gradually converge to a fair-share point during the

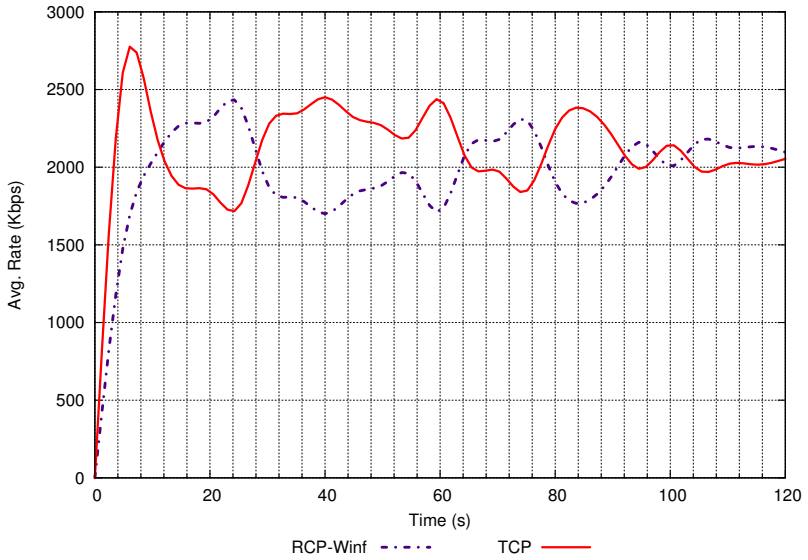


Figure 5.37: *RCP-Winf* utility results (8 Winf flows - 4 TCP flows).

connection time. It is possible to observe that, in a long term, *XCP-Winf* and *RCP-Winf* compete for the bandwidth in the same way as TCP, resulting in a TCP friendly behavior.

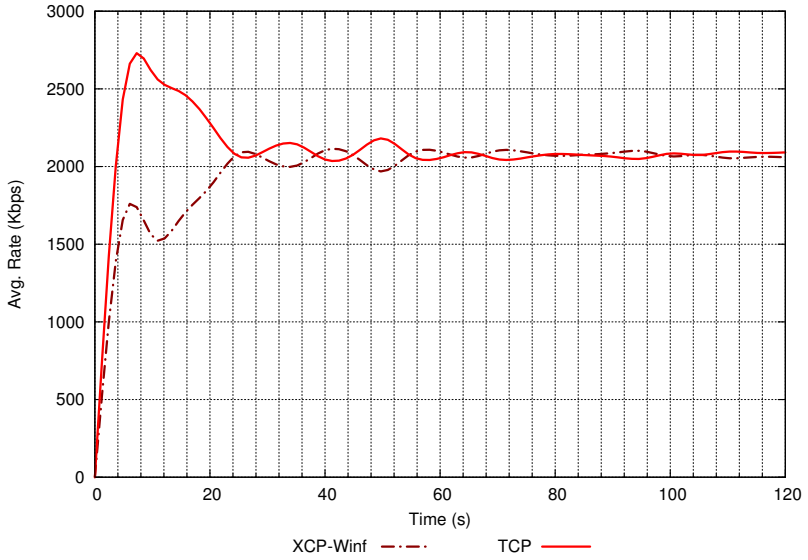


Figure 5.38: *XCP-Winf* utility results (8 Winf flows - 4 TCP flows), TCP aware.

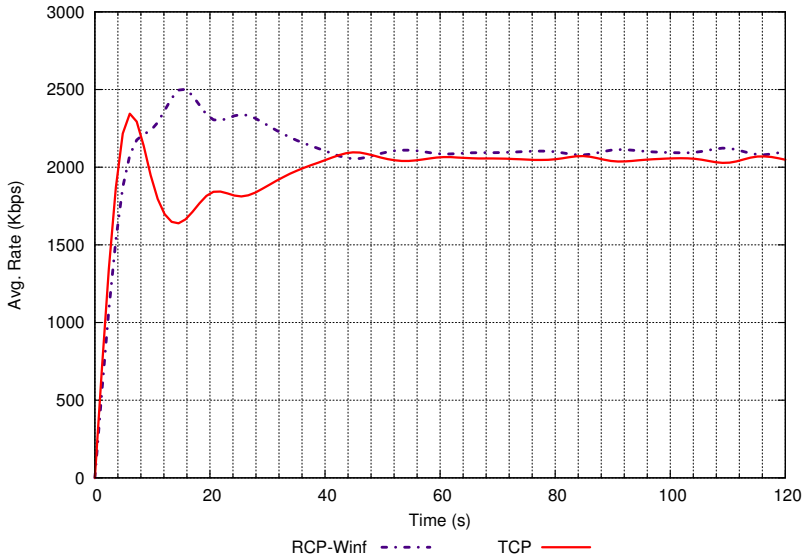


Figure 5.39: *RCP-Winf* utility results (8 Winf flows - 4 TCP flows, TCP aware).

5.7 Conclusions

In this chapter we presented two new wireless congestion control approaches. Those approaches are based on two recent rate based congestion control mechanisms that use explicit congestion notifications, XCP and RCP, integrated with *rt-Winf* information. The proposed mechanisms are called *XCP-Winf* and *RCP-Winf*. *rt-Winf* MAC layer information is used by XCP and RCP through a cross layer communication process to accurately determine the network status and act accordingly.

To improve both efficiently and reliability *XCP-Winf* and RCP-Winf, collision probability is also accounted. This allows to reflect the extra time introduced when a node is waiting to transmit as a result of collisions, thus, consuming more bandwidth. Collision probability can only be used by a sender node to improve communications. Using *rt-Winf* as the underlying estimation tool, a sender can infer collision probability and use more precise information, resulting in a more fair and accurate rate control.

The evaluation results of both *XCP-Winf* and *RCP-Winf*, obtained through ns-2 simulations, against TCP, XCP, RCP and XCP-b, show that the *rt-Winf* algorithm significantly improves XCP and RCP behavior, making them more efficient and stable. To obtain the available network capacity, both XCP and RCP need that all nodes in the network cooperate, which increases network overhead, specially when dealing with special wireless environments, such as wireless mesh networks and ad-hoc networks. Using *rt-Winf*, that works in the MAC layer, it is possible to perform link capacity and available bandwidth calculations without interfering in the network dynamics, allowing to significantly improve XCP and RCP performance. It was also possible to conclude that *XCP-Winf* and *RCP-Winf* behave more efficiently and use the network available information more effectively than XCP-b, a XCP based congestion control mechanism specifically designed for wire-

less environments. It is then possible to conclude that both *XCP-Winf* and *RCP-Winf* outperform TCP, XCP, RCP and XCP-b in terms of medium usage, thus allowing a more stable behavior and a faster convergence to the active network conditions.

TCP friendliness is an important issue when designing a new congestion control protocol. The first evaluation results of TCP friendliness show that both *XCP-Winf* and *RCP-Winf* take some time to allow a fair share of bandwidth between their native flows and TCP. To overcome this problem a new improvement to the capacity estimation is proposed. The proposed improvement takes into consideration the type of flows that coexist in a network. The new simulation results clearly show that this improvement makes both *XCP-Winf* and *RCP-Winf* to behave more TCP friendly.

Chapter 6

Enhanced TCP-AP Congestion Control Protocol

6.1 Introduction

The results in the previous chapter show that, using an accurate and reliable estimation technique together with a rate based congestion control mechanism, it is possible to significantly increase network performance and behavior. It is thus important that, for efficient data transport over wireless networks, the source nodes need to regulate traffic inserted into the network. It is also clear that, for effective congestion control, a robust and effective estimation of link capacity and available bandwidth is a main component.

Due to several decades of reliable utilization in wired communications, the Transmission Control Protocol (TCP) is, together with the Internet Protocol (IP), the de-facto suite that governs network communications and developments. TCP is well known to be a reliable end-to-end transport protocol used for data services in wired networks, and has become (as new research has been developed) very efficient and robust in such networks. However, developed research and experiments show that TCP does not perform well in wireless networks, specially in wireless mesh and ad-hoc networks [53]. TCP in wireless networks presents degraded throughputs, being very unfair among flows.

Since TCP is the dominant protocol for most of today's applications and communications, it is a constant object of research in several computer networks technologies, namely in wireless computer communications. Some attempts to improve TCP performance in wireless mesh and ad-hoc networks have been reported. Such networks have become, during the last years, increasingly important as they allow ubiquitous connectivity ahead traditional wired networks. They are comprised of highly mobile nodes, and introduce new congestion control challenges and considerations raised by the unique characteristics of the wireless medium and their dynamic nature.

TCP with Adaptive Pacing (TCP-AP) [20] is a congestion control mechanism based on TCP specifically designed for ad-hoc multi-hop wireless networks. TCP-AP uses a hybrid scheme between a pure rate based transmission control and TCP's use of the standard

congestion control algorithms. A TCP sender adaptively sets its transmission rate using an estimate of the current four hop propagation delay and the coefficient of variation of recently measured round-trip times (RTT). TCP-AP relies only on the four hop propagation delay to evaluate link available bandwidth and capacity, thus, not taking into consideration all the factors that influence link evaluation.

One important aspect of TCP-AP is that it is TCP compatible and compliant, and only the sender needs some minor changes. This allows TCP-AP to cohabit with current TCP implementations and applications. TCP-AP rate based transmission control key components are link capacity and available bandwidth; if these components are not efficiently obtained, TCP-AP performance is clearly compromised. In Chapter 4 we proposed the *rt-Winf* estimation technique for wireless multi-hop networks that uses Medium Access Control (MAC) layer information for its estimations. It was found that it accurately and efficiently obtained link capacity and available bandwidth. In this chapter we aim to investigate how TCP-AP can be improved using the integration of *rt-Winf* for capacity and available bandwidth estimation.

This chapter presents in section 6.2 an evaluation study of TCP-AP against several approaches for wireless congestion control, using the ns-2 simulator. The evaluation shows that TCP-AP is not evaluating accurately the capacity and available link bandwidth in wireless networks. In section 6.3 we introduce the main principles of improving TCP-AP performance in wireless environments. Section 6.3.1 presents the principles of using *rt-Winf* through cross layer communication in TCP-AP. Then, section 6.3.2 describes how TCP-AP with *rt-Winf* behavior can be enhanced, introducing the node path contention count factor. We call this new proposal *Wireless Enhanced TCP-AP WE TCP-AP*. A performance evaluation based on simulation results is presented in section 6.4. We analyze throughput, delay and received packets parameters as well as TCP friendliness. The results show that the enhanced proposal of TCP-AP clear outperforms base TCP-AP results. The chapter ends in section 6.5 with the summary of the new congestion control proposal and the main conclusions of its evaluation against other proposed approaches.

6.2 TCP-AP Evaluation in Wireless Ad-Hoc Networks

As TCP-AP tries to retain the end-to-end semantics of TCP, without any modifications on the link layer, routing layer or the need of cross layer information, it is important to understand how it reacts under high density and high dynamic environments.

TCP-AP is a hybrid scheme that introduces the concept of four hop propagation delay, which is the estimated elapsed time between transmission of a packet by the source and its reception by a node that is four hops away. This estimation is based on the round trip time (RTT) of packets. TCP-AP hybrid scheme implements rate based packet transmissions within the TCP congestion window, and considers TCP standard Additive Increase Multiplicative Decrease (AIMD) for the network congestion beyond the four hops. Thus,

TCP-AP uses rate based as well as congestion window based congestion control, being considered as a hybrid approach.

For better understanding the TCP-AP behavior and performance, we evaluate TCP-AP using ns-2 [23] simulations, against XCP-b [22], *XCP-Winf* and the Wireless Control Protocol (WCP) [21]. As WCP is also an AIMD and rate based approach, it is a good baseline for comparison purposes. The network scenario used is an ad-hoc network similar to the one in Chapter 3, with nodes varying from 8 to 256 nodes (8, 16, 32, 64, 128, 256). Nodes are randomly distributed throughout the simulation area as presented in Figure 3.2.

Flows also vary according to the number of nodes, with 8 nodes we have 4 flows, with 16 nodes we have 8 flows, and so on. The routing protocol used was the Destination-Sequence Distance-Vector (DSDV) [107]. The configured default transmission range was 250 meters, the default interference range is 500 meters, and the channel data rate is 11 Mbps. The performance metrics used are: throughput, delay and number of received packets. Each flow represents a File Transfer Protocol (FTP) application, simulating a large file download. The mobility is emulated through the ns-2 *setdest* tool to provide a random node movement pattern. We configure *setdest* with a minimum speed of 10 m/s, a maximum speed of 30 m/s and a topology boundary of 1000x1000 meters. All results were obtained from ns-2 trace files, with the help of *trace2stats* [106] scripts adapted to our own needs. All simulations last 300 seconds. The simulations are repeated 30 times with different ns-2 seed values. The mean and 95% confidence intervals are presented in the results. Table 6.1 resumes the main used parameters.

Simulation Parameters	
Topology Area	1000m x 1000m
Simulation Time	300 sec.
Simulation repetition	30 times
Ad-Hoc Scenarios Number of Mobile Nodes	8, 16, 32, 64, 128, 256
Ad-Hoc Scenarios Number of Flows	4, 8, 16, 32, 64, 128
Flow Type	FTP
Mesh Nodes Position	Random
Path Loss Model	Two Ray
Mobility Model	Random Way Point
Maximum Movement Speed	30 m/s
Minimum Movement Speed	10 m/s
Mac layer	IEEE 802.11
Propagation Model	Two Ray Ground
Routing Protocol	DSDV

Table 6.1: Ad-Hoc Simulation Environment Parameters.

TCP-AP ns-2 implementation is based in the TCP NewReno implementations and it is available at [109]. In the simulations we used the optimal settings, suggested by its authors with the same parameters as in Chapter 3.

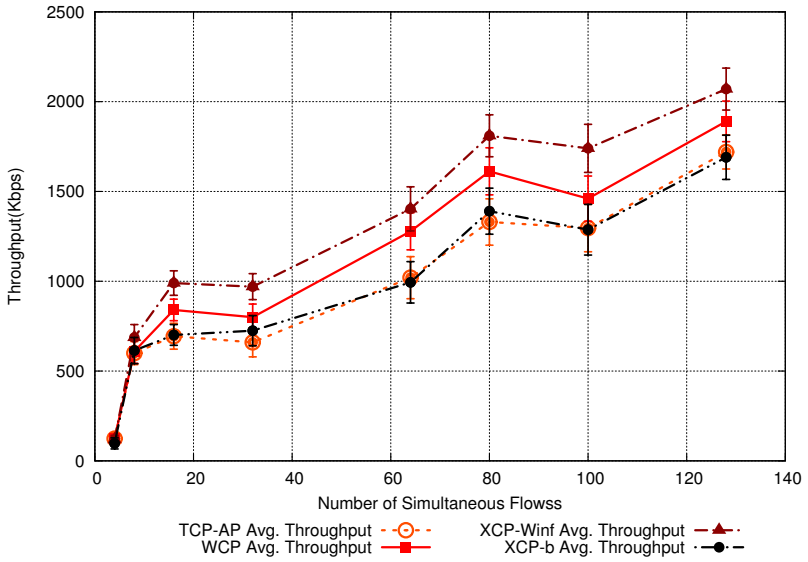


Figure 6.1: TCP-AP Ad-Hoc Evaluation Scenario, Throughput.

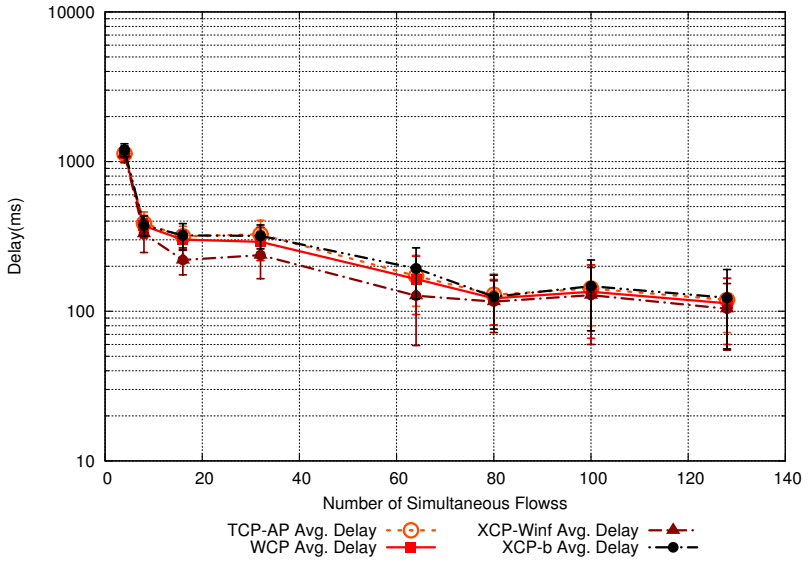


Figure 6.2: TCP-AP Ad-Hoc Evaluation Scenario, Delay.

From Figure 6.1, Figure 6.2 and Figure 6.3 we can observe that TCP-AP is the protocol with the worst results, with poor performance when compared to the other approaches. From the Figures it is possible to observe that TCP-AP presents a lower performance in terms of delay, throughput and received packets. This is due to the fact that TCP-AP is not obtaining correctly the network's maximum capacity, thus not avoiding congestion and not using efficiently the medium. TCP-AP is over-estimating the available rate producing congestion, that it is represented, in the results, by higher delays and less received packets.

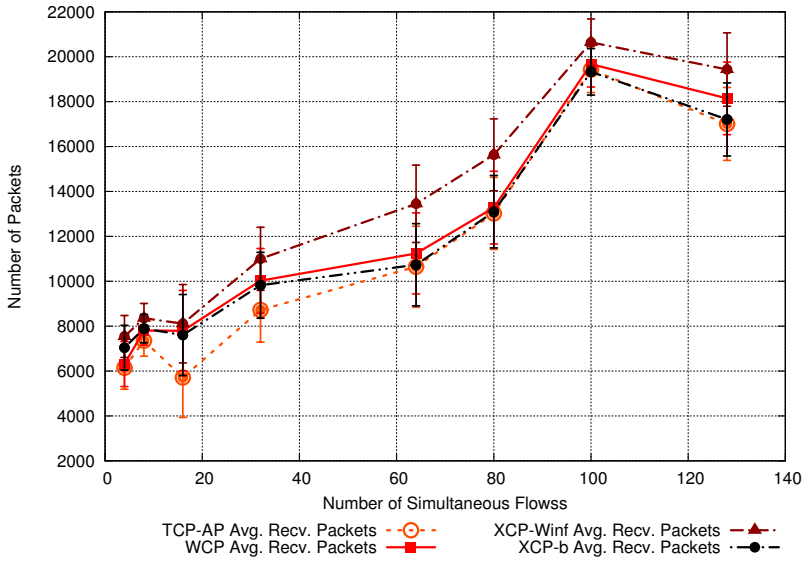


Figure 6.3: TCP-AP Ad-Hoc Evaluation Scenario, Received Packets.

As TCP-AP rate estimation technique is not correctly evaluating the medium, the sender is generating more traffic than the medium supports, resulting in more packets queued and less packets in transit; this results in decreased throughput, increased delay and a considerably decreased number of received packets. Another characteristic of TCP-AP is that it uses the standard AIMD process in most of its behavior. This process is not suitable for wireless networks as it overloads the wireless channel. This behavior in conjunction with the estimation technique of TCP-AP, that only uses measures of the four hop neighborhood, results in evident inaccurate available bandwidth estimations and higher delays.

We can then conclude that TCP-AP obtains poor throughput and behaves in a very conservative way. TCP-AP is also not considering a fair share of the bandwidth to all flows, not using correctly the medium and having a significant degradation of performance.

XCP-Winf is the approach that achieves overall best results. *XCP-Winf*, being a rate based protocol where bandwidth and capacity estimation is based on MAC layer information, and providing node cooperation, it can effectively and quickly adapt to the links conditions, thus, improving network performance and making the network behave more fairly.

From the presented results, it is also possible to observe that WCP has better overall results than TCP-AP. WCP has a rate control mechanism that reacts explicitly to congestion, and a cooperative communication process between neighbor nodes that make WCP to react more efficiently to the network conditions, allowing to have a better medium usage. The results of received packets in WCP become very close to the ones obtained by TCP-AP, as the number of flows increases. This is a clear indication that WCP with more nodes and mobility is not using efficiently the medium, being clearly affected by the network conditions. XCP-b results are better than the ones obtained by TCP-AP. However,

its results are worse than the ones obtained by *XCP-Winf* and WCP.

Although TCP-AP scheme is a hybrid scheme of sender rate control and congestion control, TCP-AP is based on two assumptions: that the rate control mechanism is efficient and the contention and spatial reuse is accurate, whether they are effective in some network topologies remains unknown. These assumptions are clearly not effective in high mobility wireless scenarios. The conservativeness of TCP-AP is observed through throughput (Figure 6.1) results and received packets (Figure 6.3). While having good throughput results, they are obtained with less received packets. This is a consequence of using the hybrid scheme for congestion control, considering only the four hop propagation delay neighborhood.

TCP-AP is not using information from the MAC layer, and it relies on the transmission of packets at the transport layer. This principle is failing effectively to transmit packets at the MAC layer, making it react with poor performance in terms of received packets. As TCP-AP is not relying in an effective available bandwidth and link capacity estimation mechanism at the MAC layer, the sender assumes that the bandwidth of all links in the path is the same and the medium usage is clearly not efficient. Due to its four hop propagation delay assumption, TCP-AP available bandwidth and link capacity estimation are not considering all the nodes along the path: the nodes that contend for available bandwidth along the path. This is specially relevant when we are dealing with a high density and high mobility network, introducing inaccuracy and lack of fairness on TCP-AP performance.

6.3 Improving TCP-AP performance

The previous section has shown that, due to an inaccurate and ineffective available bandwidth estimation technique, TCP-AP performance is significantly reduced when compared to other protocols and approaches. The hybrid scheme of TCP-AP combined with an inaccurate estimation technique reduces medium efficiency and usage. TCP-AP also considers a maximum neighborhood of four hops, not considering all nodes along the path, introducing unfairness and conservativeness. Ways for improving end-to-end TCP-AP congestion control must, then, be defined and examined. Incorporating more accurate and efficient estimation techniques into the rate based algorithm of TCP-AP and allowing TCP-AP to be aware of the influence of all nodes that contend for available bandwidth on the path can be a way of improving TCP-AP performance, fairness and efficiency.

This section presents the main principles for improving TCP-AP behavior in wireless environments. First, it is presented the use of *rt-Winf* as the MAC layer information estimation technique to improve TCP-AP. Then, as TCP-AP uses the four hop propagation delay for its main operating principles, and due to the shared nature of the wireless medium, nodes along the path contend for access the medium. This must be accounted in the actual available bandwidth estimation, and thus, we introduce the effect of the nodes contending along the path. Simulation results show that this effect contributes for a fair share of the bandwidth.

6.3.1 TCP-AP with *rt-Winf* - *TCP-AP-Winf*

The base TCP-AP considers network and transport layer information (RTT values) for its capacity and available bandwidth estimations. This technique is therefore not very accurate introducing inefficiency in the congestion control process. This was already shown in wired networks, in works [27], [121], which introduced packet dispersion to analyze the capacity and available bandwidth estimations. This problem is even increased when dealing with wireless networks, since their variation and instability increase.

It is important to have a cross layer approach for bandwidth and link capacity estimation: use information provided by several layers, including the MAC layer, to increase reliability and efficiency of the congestion control mechanism. Therefore, we propose TCP-AP with *rt-Winf*, which relies on the main functioning principles of TCP-AP, but uses information provided by *rt-Winf* (Chapter 4) to determine in real-time the link capacity and available bandwidth.

As *rt-Winf* obtains the link capacity and available bandwidth in the MAC layer, this information has to be accessed by TCP-AP through a cross layer communication process, through the *MobileMan* [2] cross layered network stack.

Our approach, when compared to the base TCP-AP, changes the way each node calculates the four hop delay (FHD) and the average packet queuing delay per node (t_q), with the *rt-Winf* link capacity and available bandwidth values. The standard TCP-AP rate control algorithm is, thus, updated to:

$$t_q = \frac{1}{2}\left(\frac{T_{RTT}}{h} - \frac{S_{data} - S_{ack}}{C_{Winf}}\right) \tag{6.1}$$

where T_{RTT} represents the RTT value, h represents the number of hops between sender and receiver, S_{data} is the size of the data packet and S_{ack} the size of the acknowledgement (ACK) packet. Finally, C_{Winf} corresponds to the *rt-Winf* link capacity. Considering the previous equation, we also update the 4-hop delay (FHD) by:

$$FHD = 4 \times \left(t_q + \frac{S_{data}}{AB_{Winf}}\right) \tag{6.2}$$

where AB_{Winf} is the *rt-Winf* available bandwidth. As considered in Chapter 4, high density and high mobility networks, as wireless mesh and ad-hoc networks, suffer from a large number of collisions. *rt-Winf* updated mechanism with the effect of collision probability is also considered when updating TCP-AP behavior.

6.3.2 Wireless Enhanced TCP-AP - *WE TCP-AP*

In a wireless network, nodes along a multi-hop path (NP) contend among themselves for access to the medium, i.e, they contend for available bandwidth. Considering that TCP-AP only implements adaptive pacing at the sender side, available bandwidth and capacity estimation must take into consideration nodes along the path between the source

Algorithm 1: *WE TCP-AP* Source Node Operations.

```
foreach ACK packet do
    Node estimates node path (NP) from MAC ACK
    Node computes NP-1
    if NP − 1 <= 4 then
        HD = FHD;
    else
        R = (NP+1)/NP;
        HD = R × FHD
```

and the sink, that is, the bandwidth contending successors and predecessors on the route path. However, this is not true in TCP-AP, since it is considering only the four hop neighborhood for these contending estimations. Therefore, to eliminate this inaccuracy, we changed TCP-AP with *rt-Winf* to use a coefficient (R is the unused bandwidth) that represents the proportion of bandwidth contention among other nodes on the path, thus, maximizing the throughput while guaranteeing fairness.

If we consider NP as all nodes along the path and if $NP-1$ is equal or less than 4, then TCP-AP with *rt-Winf* is kept unchanged; if $NP-1$ is higher than 4 then the FHD equation, now called the hop delay (HD) is updated to:

$$HD = FHD \times R \tag{6.3}$$

where

$$R = 1 + \frac{1}{NP} \tag{6.4}$$

then,

$$HD = 4 \times (\frac{NP+1}{NP}) \times (t_q + \frac{S_{data}}{AB_{Winf}}) \tag{6.5}$$

For better understanding on how a source TCP-AP node algorithm is updated, Algorithm 1 shows the pseudo-code of a *WE TCP-AP* source node.

As R represents the unused bandwidth due to node contention and queue management along the path, it introduces the fairness factor allowing an improved fair share of the available bandwidth among all contending nodes, not only the ones within the four hop propagation delay, improving *WE TCP-AP* behavior and making it behave more accurately and efficiently.

6.4 WE TCP-AP Performance Evaluation

In this section, we evaluate the performance of TCP-AP with *rt-Winf* and *WE TCP-AP*. We first describe the simulation methodology, and then present the simulation results

obtained on different scenarios and topologies.

6.4.1 Methodology

We have implemented both TCP-AP modifications in the ns-2 [23] simulator. We updated the base TCP-AP implementation with the cross layer communication process and the *rt-Winf* information, accordingly to Equation 6.3.1 and Equation 6.3.1. The cross layer communication in ns-2 is done using class commands. To perform the *WE TCP-AP* implementation, we updated the TCP-AP with *rt-Winf* code, as described in Algorithm 1.

All our simulations were conducted using an unmodified 802.11 MAC implementation. The underlying *rt-Winf* mechanism is configured with enabled Request-to-Send (RTS)/ Clear-to-Send (CTS)/ACK handshake packets. The proposed mechanisms are evaluated against the base TCP-AP protocol, WCP and *XCP-Winf*. Two different scenarios were used: the same ad-hoc scenario presented in Section 6.2, and a wireless mesh topology scenario that is presented to understand how the new proposals behave under different conditions, similar to the one on section 3.2. This scenario is defined with a grid of 16 (4x4) mesh nodes and a variable number of mobile nodes. The number of mobile nodes changes from 3 to 7. Figure 6.4 shows the mesh scenario with 16 mesh nodes and 3 mobile nodes. For the data transmissions, it is used a FTP application with packets of 1500 bytes. It was used the same mobility tools and trace scripts as in the TCP-AP evaluation in section 6.2. First, we present and analyze the results of the TCP-AP with *rt-Winf*, and then we present and analyze the results with the enhanced contention approach, the complete *WE TCP-AP*. For studying the performance of *WE TCP-AP* in the presence of Constant Bit Rate (CBR) traffic, we simulated 64 Kbps VoIP flows as the background traffic. All mobile nodes generated and received VoIP traffic. Finally, we also analyze the influence of the factor R in *WE TCP-AP* behavior and the *WE TCP-AP* TCP friendliness.

6.4.2 TCP-AP with *rt-winf* Simulation Results

Figure 6.5, Figure 6.6 and Figure 6.7 show the performance metrics for the mesh topology scenario. From the observation of the results, it is possible to observe that TCP-AP with *rt-Winf* integrated clearly improves TCP-AP performance behavior, but it is still below the performance of *XCP-Winf*. TCP-AP with *rt-Winf* is only taking into consideration *rt-Winf* information for the last four hop nodes; TCP-AP, as opposed to *XCP-Winf*, uses the standard behavior of TCP for the other hops of the network, considering that all links have the same bandwidth.

Another important drawback of TCP-AP with *rt-Winf* is the fact that it does not have a fairness module, resulting in a more conservative and less fair operation. The fairness module is a native mechanism used by *XCP-Winf* and also present in XCP [15]. As TCP-AP with *rt-Winf* uses, in most of its functioning, the standard AIMD process of TCP and is not entirely using the available information, between the source and the sink, its results are not similar to the ones of *XCP-Winf*. *XCP-Winf* also relies on total node path

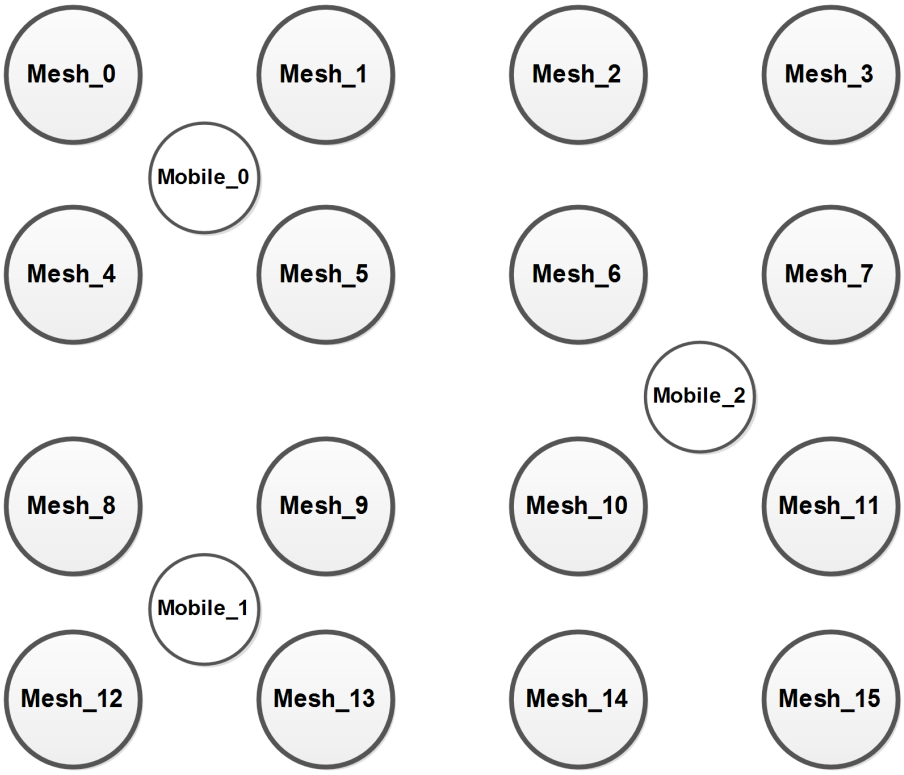


Figure 6.4: 16 Mesh Nodes - 3 Mobile Nodes Scenario.

interaction, using a cooperative approach to obtain the best available bandwidth and link capacity usage. In TCP-AP with *rt-Winf*, as the number of nodes or flows increases, it uses conservative mechanisms, reducing its performance especially concerning received packets.

WCP obtains better results than TCP-AP with *rt-Winf*. WCP uses explicit congestion information between nodes that trigger rate changes, making it behave with good efficiency and fair. As *XCP-Winf* uses the *rt-Winf* mechanism as its base estimation tool, it has a precise feedback communication mechanism between all the nodes along the path using total network cooperation, and it is able to better use the channel with less losses, resulting in a more efficient and accurate behavior.

Figure 6.8, Figure 6.9 and Figure 6.10 show the results for the ad-hoc topology scenario. In this scenario, we can see that *rt-Winf* clearly improves TCP-AP performance, compared to TCP-AP and WCP. However, TCP-AP with *rt-Winf* still reflects some of TCP-AP flaws. With the increase of the number of flows, TCP-AP with *rt-winf* becomes less efficient, as it is only relying on the four hop propagation delay and the AIMD process, not considering the entire network topology for its rate changes. This is shown by being able to obtain good throughput results, compared to *XCP-Winf*, when the network is not heavily loaded. When increasing the number of nodes, number of flows and the mobility density, TCP-AP with *rt-Winf* becomes more inefficient, reducing significantly its throughput and number

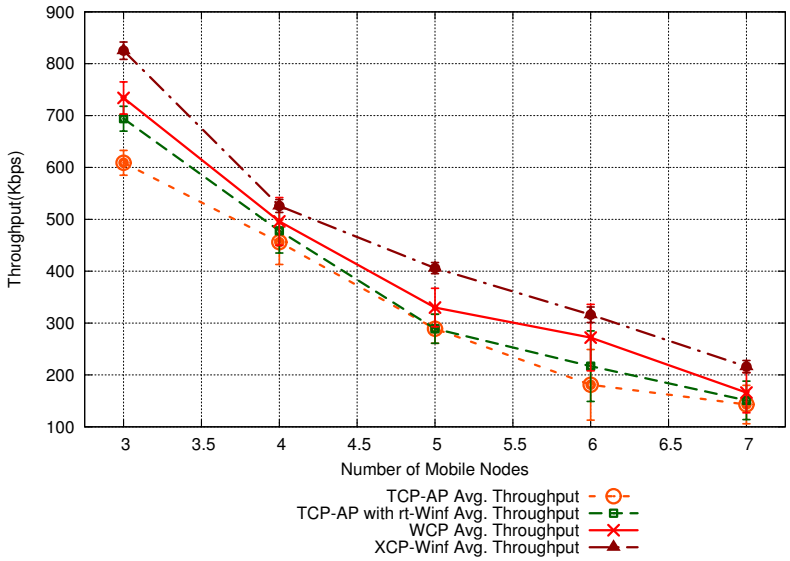


Figure 6.5: 16 Mesh Nodes - Variable Number of Mobile Nodes, TCP-AP with *rt-Winf* Throughput.

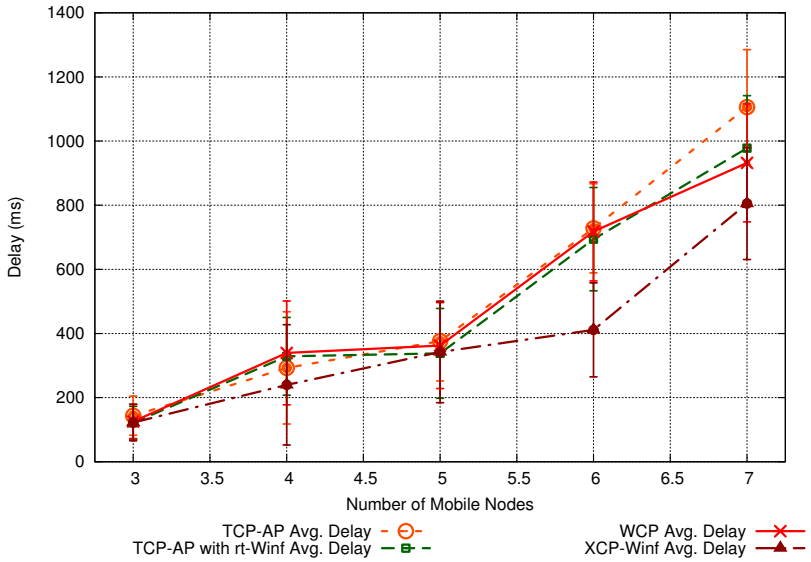


Figure 6.6: 16 Mesh Nodes - Variable Number of Mobile Nodes, TCP-AP with *rt-Winf* Delay.

of received packets when compared to the other approaches.

TCP-AP with *rt-Winf* is also more fair than TCP-AP to mobility changes, but still shows an unstable behavior. WCP has overall good results: although being an hybrid approach, it uses a more effective congestion and control interval, as all nodes within the congestion neighborhood mark packets with congestion indicators, triggering rate reduc-

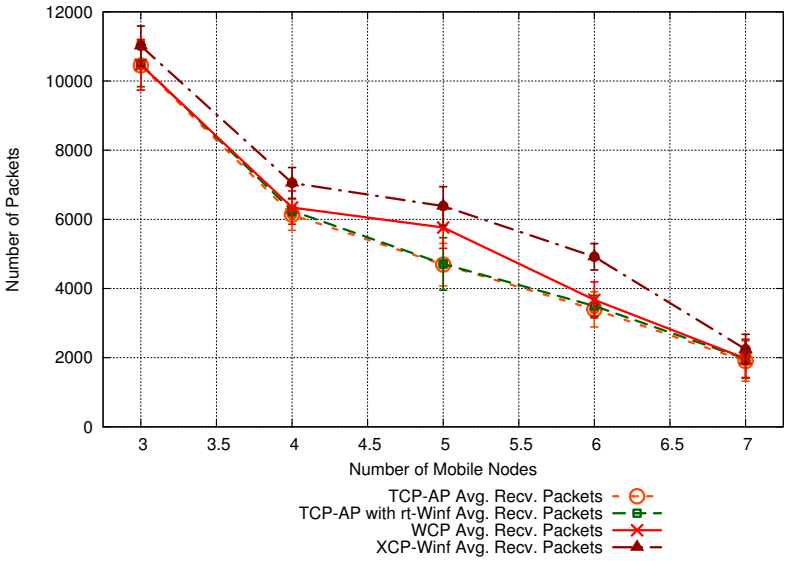


Figure 6.7: 16 Mesh Nodes - Variable Number of Mobile Nodes, TCP-AP with *rt-Winf* Received Packets.

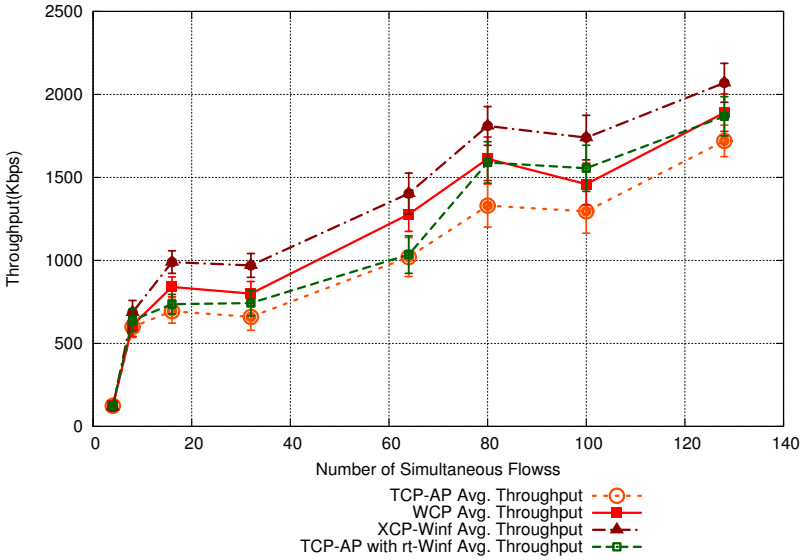


Figure 6.8: Variable Number of Flows Ad-Hoc Scenario, TCP-AP with *rt-Winf* Throughput.

tions more efficiently at the source.

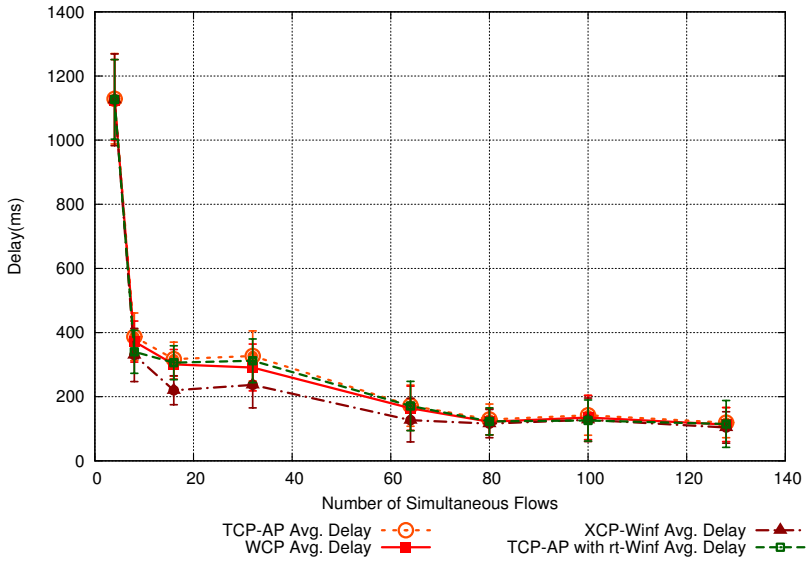


Figure 6.9: Variable Number of Flows Ad-Hoc Scenario, TCP-AP with *rt-Winf* Delay.

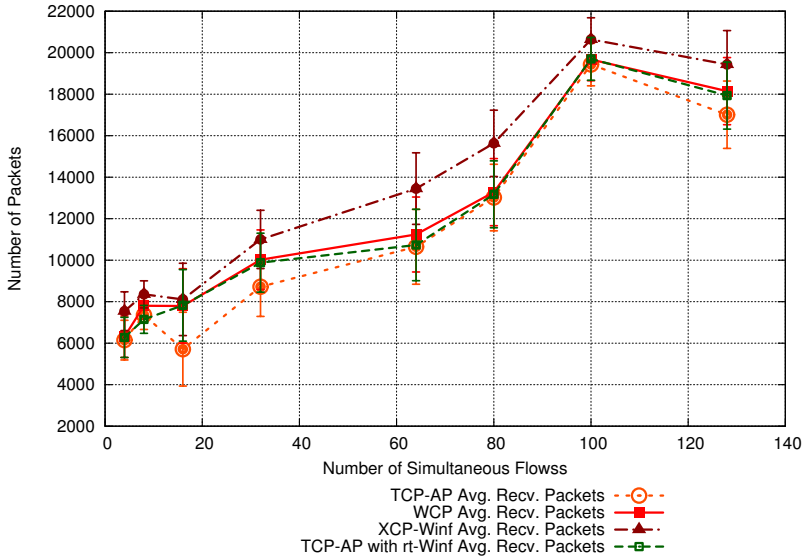


Figure 6.10: Variable Number of Flows Ad-Hoc Scenario, TCP-AP with *rt-Winf* Received Packets.

6.4.3 *WE TCP-AP* Simulation Results

This section presents the simulation results of *WE TCP-AP* in both mesh scenario and ad-hoc scenarios.

Figure 6.11, Figure 6.12 and Figure 6.13 show the performance metrics. Figure 6.11 shows how throughput is improved in *WE TCP-AP*. The *WE TCP-AP* throughput values are $\sim$ 20% to $\sim$ 40% better than the ones with the standard TCP-AP, and $\sim$ 14% to

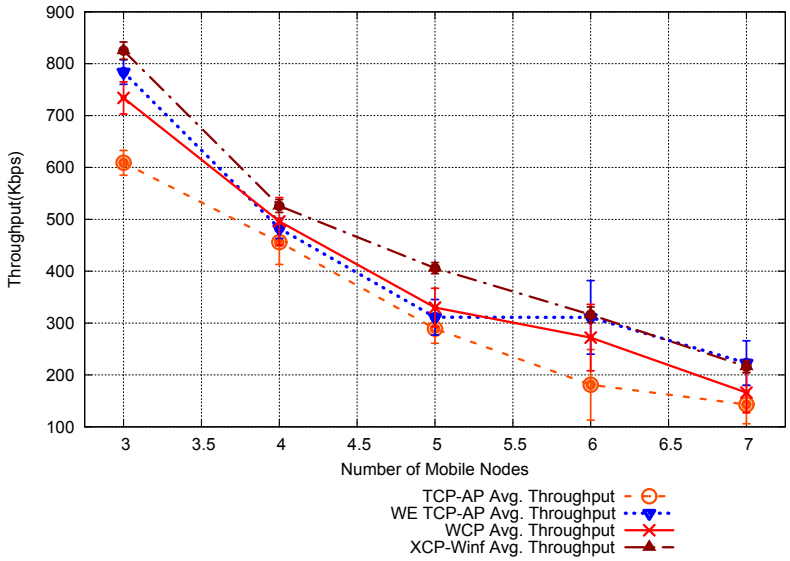


Figure 6.11: 16 Mesh Nodes - Variable Number of Mobile Nodes Scenario, *WE TCP-AP* Throughput.

$\sim$ 25% better than TCP with *rt-Winf*. In terms of received packets, as observed in Figure 6.13, it is possible to see that *WE TCP-AP* is able to use more efficiently the medium, as it can transmit more packets increasing overall throughput results. More received packets means that more transmissions are allowed, thus *WE TCP-AP* is behaving more fairly. With these improvements, the network can transmit with a higher rate and incurring less losses. As more packets are transmitted, more throughput is obtained and the medium is better and more efficiently used. This allows to have a more stable and fair behavior. It is, however, important to say that TCP-AP has a very conservative behavior, as it allows a good throughput with less received packets. This behavior is clearly improved with *WE TCP-AP*. The delay values, in Figure 6.12, are also reduced reinforcing the fact that this new proposal is much more efficient and fair, with better medium usage, than the base protocol. The better results are still obtained by *XCP-Winf*, but it is closely followed by *WE TCP-AP*. It is clear that the use of MAC layer information and the node path contention count is making *WE TCP-AP* to react more efficiently to the network dynamics. Another important conclusion is that a pure rate based congestion control mechanism with explicit rate feedback, like *XCP-Winf*, improves network performance allowing to satisfy end users quality requirements.

Figure 6.14, Figure 6.15 and Figure 6.16 show *WE TCP-AP* results in the ad-hoc networks scenarios, as defined before. From the observation of the results, it is possible to observe that TCP-AP with *rt-Winf* integrated and node path contention clearly improves base TCP-AP performance behavior. It is possible to conclude that, with more nodes and flows in the network, *WE TCP-AP* is more efficient than the standard TCP-AP proposal. *XCP-Winf* is again able to operate more efficiently than *WE TCP-AP*, specially concerning

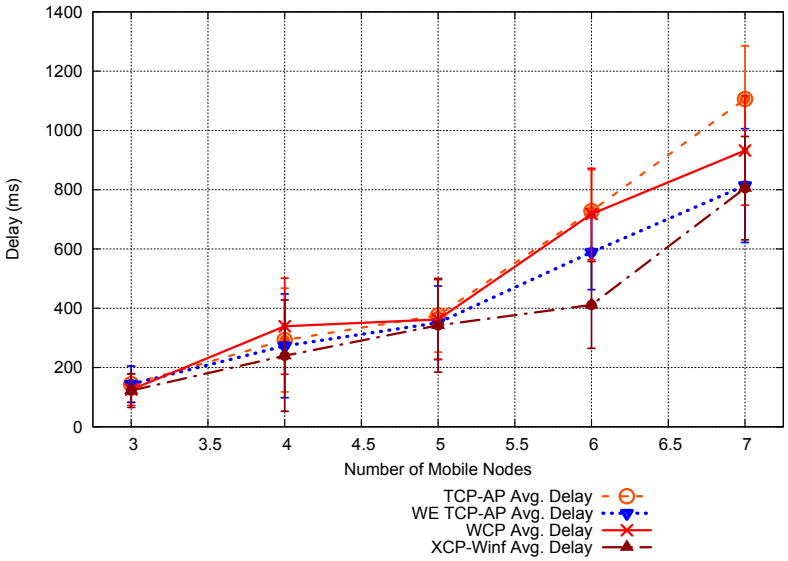


Figure 6.12: 16 Mesh Nodes - Variable Number of Mobile Nodes Scenario, *WE TCP-AP* Delay.

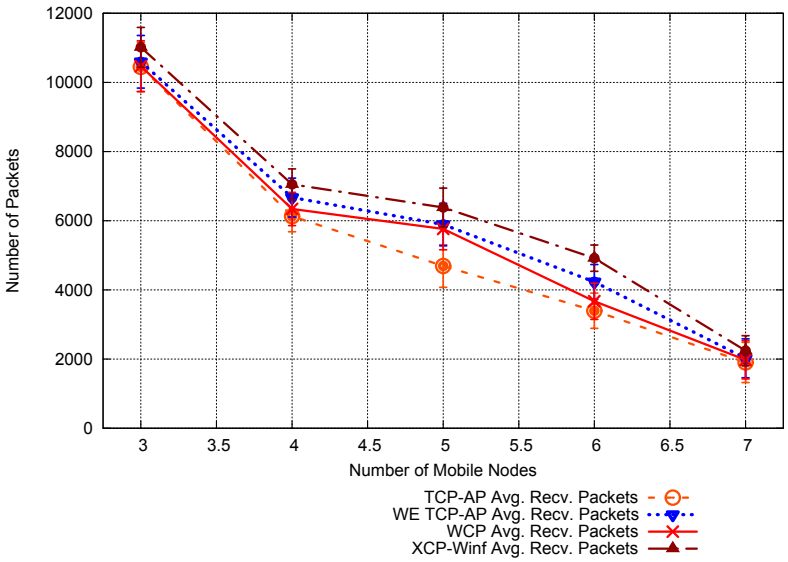


Figure 6.13: 16 Mesh Nodes - Variable Number of Mobile Nodes Scenario, *WE TCP-AP* Received Packets.

the number of received packets. *WE TCP-AP* is not a pure rate based congestion control mechanism with explicit feedback, thus it is not reacting quickly to network changes. The AIMD process of *WE TCP-AP* still introduces some instability and behavior problems.

WE TCP-AP, as opposed to TCP-AP, is considering a fair share of the unused bandwidth, that results from the use of the node path contention count, allowing it to increase

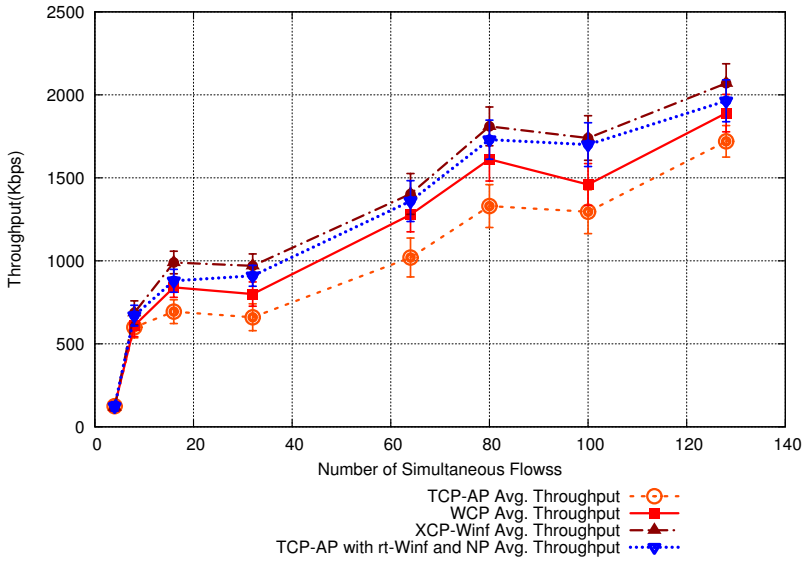


Figure 6.14: Variable Number of Flows Ad-Hoc Scenario, *WE TCP-AP* Throughput.

the flow rate, and consequently increase the number of received packets and reducing the overall delay. We can conclude that the available bandwidth and capacity evaluation of *rt-Winf*, estimated at the MAC layer, the collision probability and the node contention count factors are relevant and make *WE TCP-AP* achieving better channel utilization, which also leads to less channel losses. Comparing both ad-hoc and mesh results, it is evident that *WE TCP-AP* results have better performance on the ad-hoc environments; this is due to the fact that base TCP-AP was developed having in mind ad-hoc networks; moreover, its underlying hybrid scheme is better suited for ad-hoc networks with high density and mobility.

Because of the undesirable properties of TCP in wireless networks, many new applications often choose the User Datagram Protocol (UDP), with or without congestion control mechanisms (on the application layer). These long lasting UDP flows present a potential threat of network collapse if they do not include a congestion control mechanism. Thus, to analyze *WE TCP-AP* behavior in the presence of CBR background traffic, we simulated the same FTP application as before, but we also used UDP flows simulating a VoIP call of 64 Kbps as the background traffic on the ad-hoc scenarios. Each node generated and received background flows.

Figure 6.17, Figure 6.18 and Figure 6.19 show the obtained results. The results show that *WE TCP-AP* behavior is clearly improved when compared against standard TCP-AP and WCP. *WE TCP-AP*, as opposed to TCP-AP, allows to increase the flow rate, and consequently increase the number of received packets and reduce the overall delay. With a relative small number of flows in the network, and consequently with a relative small number of nodes, *WE TCP-AP* results are very close to the ones obtained by *XCP-Winf*. With more nodes and flows in the network, *WE TCP-AP* results become worse than

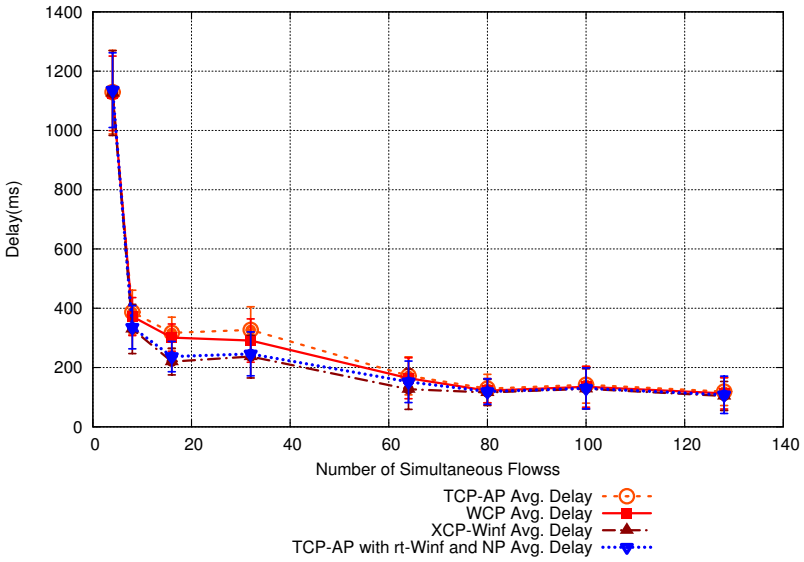


Figure 6.15: Variable Number of Flows Ad-Hoc Scenario, *WE TCP-AP* Delay.

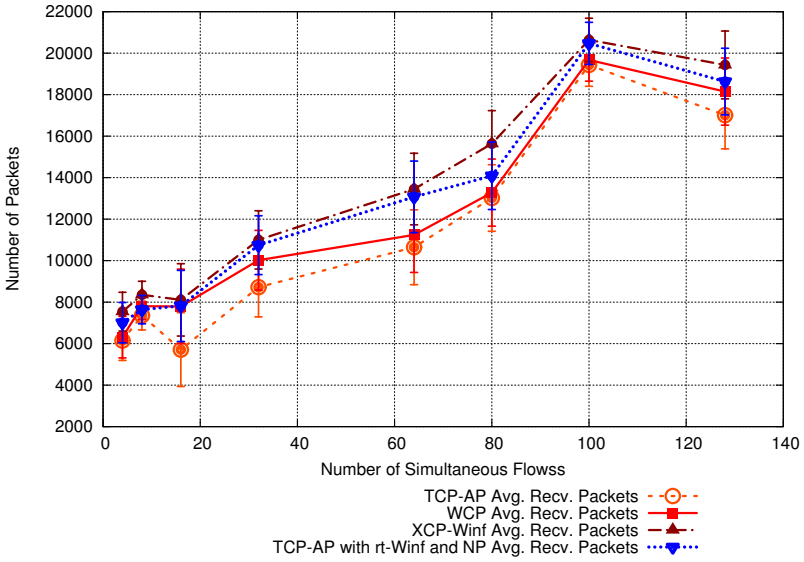


Figure 6.16: Variable Number of Flows Ad-Hoc Scenario, *WE TCP-AP* Received Packets.

XCP-Winf results.

WCP results show that, with a fully utilized network, its algorithm is not correctly estimating the maximum allowable rate, thus obtaining bad performance results. WCP results are very similar to the ones obtained by the standard TCP-AP. It is clear that these two proposals are relying in inaccurate bandwidth estimation results and are not considering the network as a cooperative environment.

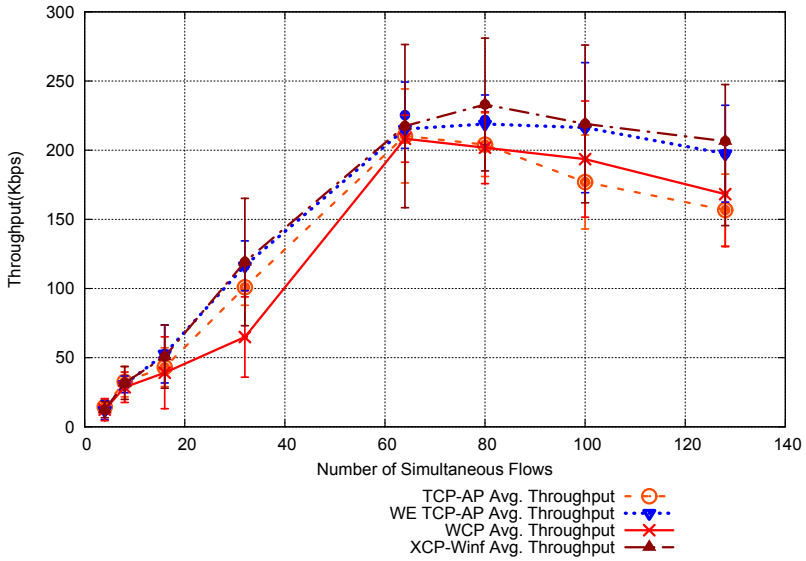


Figure 6.17: VoIP Background Traffic Variable Number of Flows Ad-Hoc Scenario, *WE TCP-AP* Throughput.

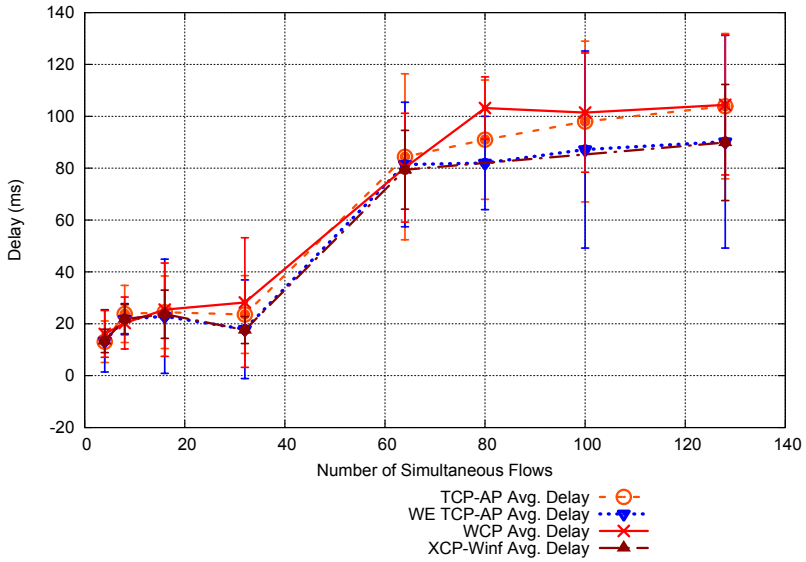


Figure 6.18: VoIP Background Traffic Variable Number of Flows Ad-Hoc Scenario, *WE TCP-AP* Delay.

6.4.4 Factor R Results

For a better understanding of how the factor R is influencing *WE TCP-AP* behavior, a central network chain scenario was defined. It must be noted that the standard TCP-AP four hop propagation delay assumes that "*every fourth node can transmit in a multi-hop*

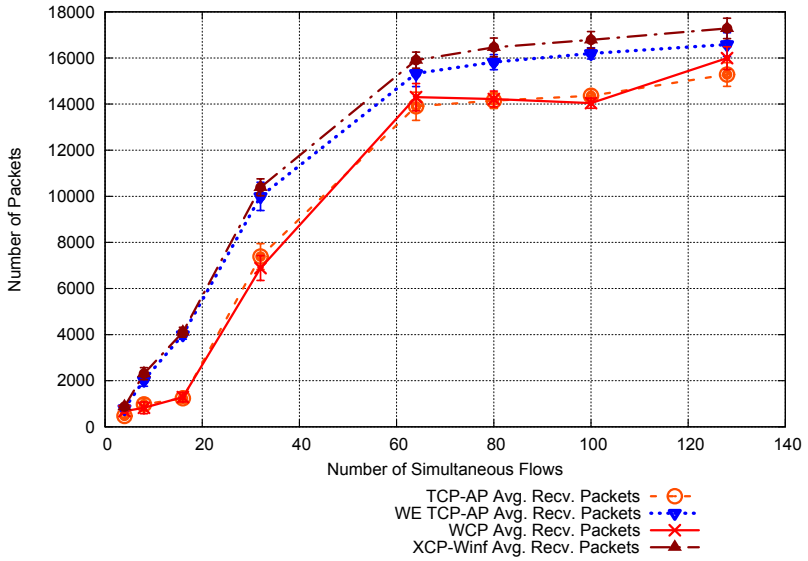


Figure 6.19: VoIP Background Traffic Variable Number of Flows Ad-Hoc Scenario, *WE TCP-AP* Received Packets.

chain topology". In this scenario, it was used the proposed version of *WE TCP-AP* and the TCP-AP with *rt-Winf* version. The chain scenario consists of a network divided in three parts. Figure 6.20 depicts the network topology with four chain nodes. The application used simulates a FTP transfer.

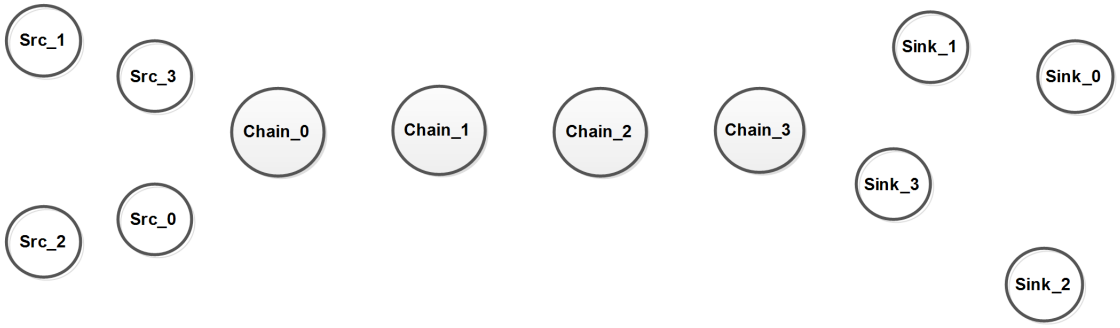


Figure 6.20: Chain Scenario.

The results are shown in Figure 6.21, Figure 6.22 and Figure 6.23. The presented results show that, with the increase of the chain nodes, TCP-AP with *rt-Winf* has worse results: it becomes less efficient and less accurate, as it is not considering the unused share of bandwidth. *WE TCP-AP*, on the other hand, is more accurate, since the available bandwidth and capacity estimation are considering the nodes along the path between the sources and the sinks nodes, that is, the contending successors and predecessors on

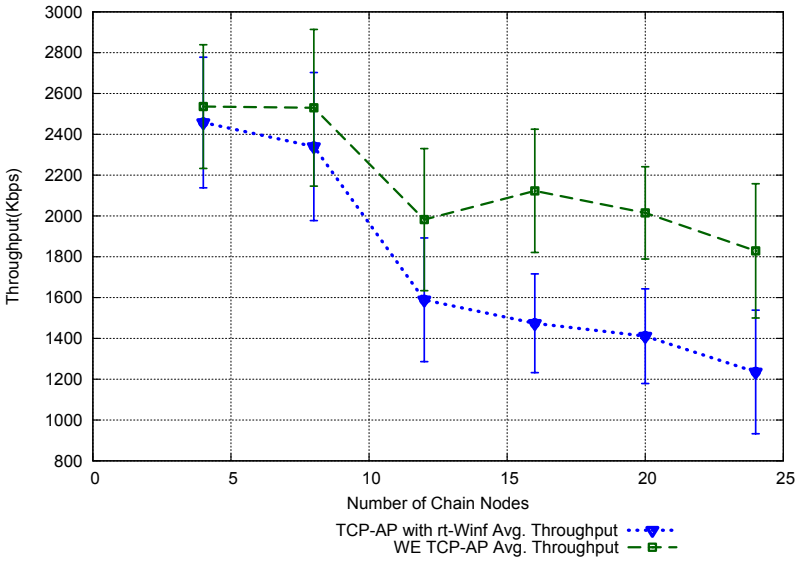


Figure 6.21: Chain Scenario, Throughput.

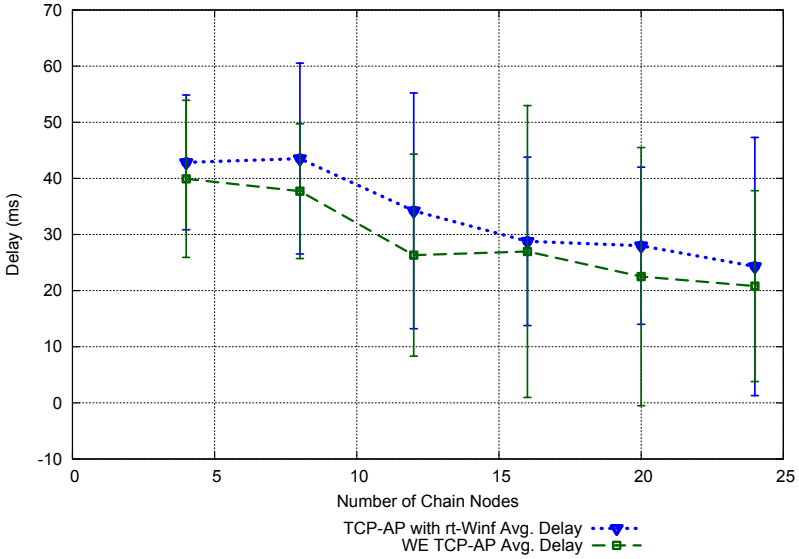


Figure 6.22: Chain Scenario, Delay.

the route path. It is then shown that the factor R, which represents the proportion of bandwidth contention among other nodes on the path, is maximizing the throughput while guaranteeing fairness.

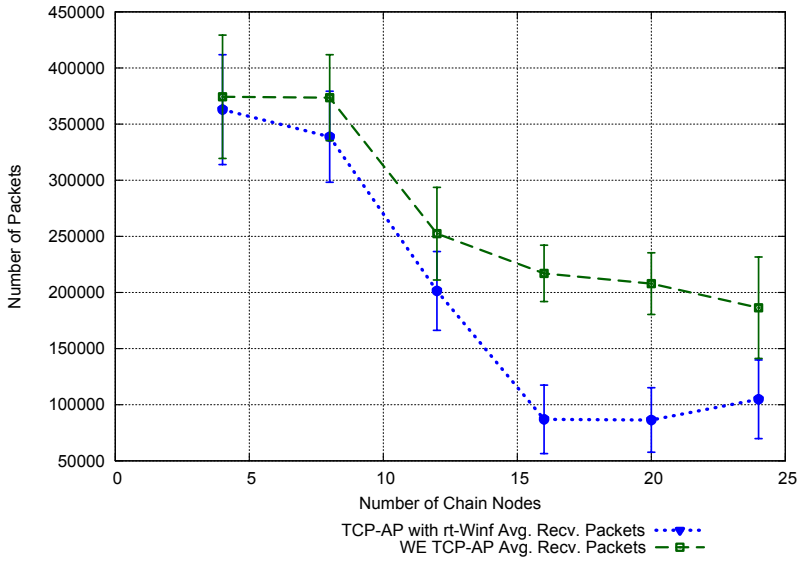


Figure 6.23: Chain Scenario, Received Packets.

6.4.5 Utility Results

In this section we analyze how *WE TCP-AP* flows interact and compete with TCP, i.e. the TCP friendliness of a congestion control mechanism. To analyze how friendly *WE TCP-AP* is, we use the average data rate over time for each flow, thus allowing to observe how bandwidth is being managed between TCP and the *WE TCP-AP* proposal. This is called the utility of a congestion control mechanism against TCP.

The evaluation scenario, is the same as the one presented in section 5.5.4 and consists of a 1000mx1000m area, divided on three distinct parts. In the left side area, with 250mx250m, we have two mobile source nodes: one source node is configured to use only the standard TCP, and the other source node uses the *WE TCP-AP* congestion control mechanism. The right side of the area has the same characteristics of the left area but, instead of source we have sink nodes. Finally, the middle area has two mobile nodes configured with the *WE TCP-AP* mechanism as their main congestion control mechanism. The average data rate is measured in these two nodes, as they will have TCP and *WE TCP-AP*-like flows competing.

We have defined two evaluation scenarios. In one scenario we have each source generating eight FTP flows, with packets of 1500 bytes. In the other scenario we have each source generating sixteen FTP flows. The simulations last 120 seconds. The obtained results are shown in Figure 6.24 and Figure 6.25. From the utility results, it is possible to observe that, on both situations, the TCP flow grows faster and gains more bandwidth on the beginning. However, as *WE TCP-AP* is a hybrid approach, keeping unchanged the AIMD process of TCP and being updated with an evaluation and measurement process, it quickly adjusts to TCP behavior, thus, allowing a fair share of network resources and being TCP friendly.

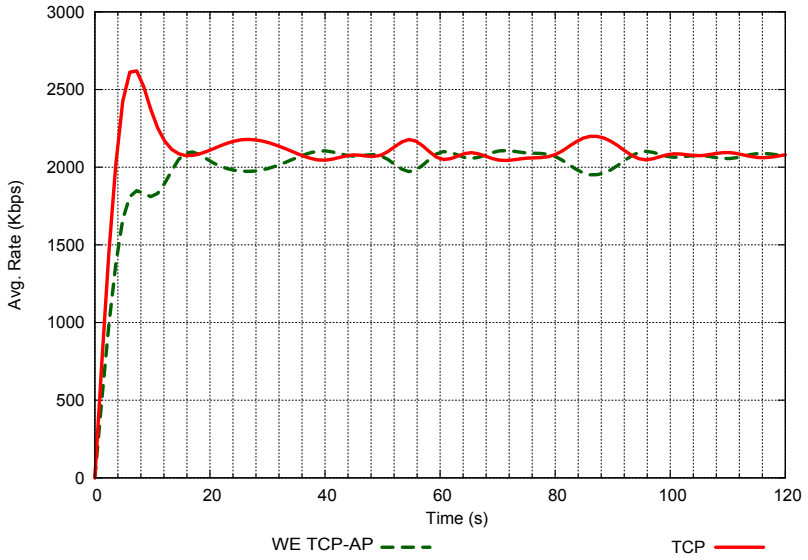


Figure 6.24: Utility Results, 2 x 8 Flows.

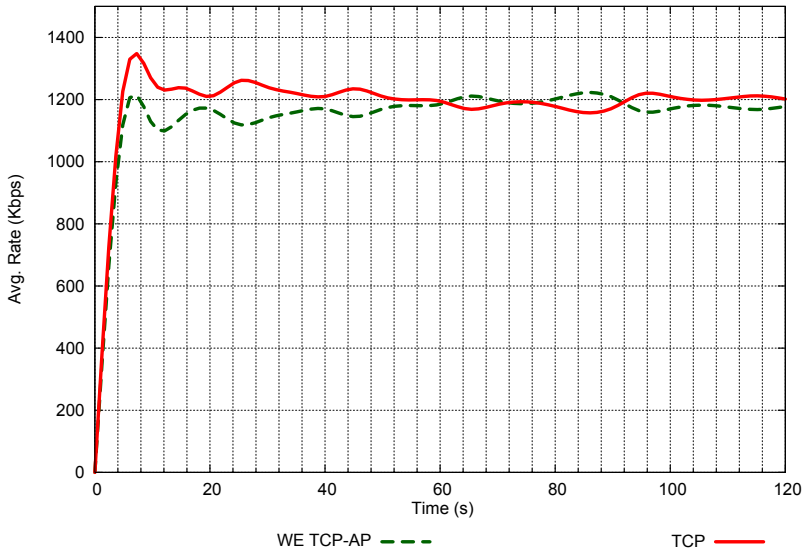


Figure 6.25: Utility Results, 2 x 16 Flows.

6.5 Conclusions

In this chapter we presented first a performance evaluation of the TCP-AP protocol. TCP-AP is a new congestion control protocol, for wireless ad-hoc networks, that uses TCP as its underlying base. TCP-AP is an hybrid congestion control protocol, using rate based control on the four hop propagation delay neighborhood, and the standard AIMD algorithm over the four hops. The performance evaluation conducted against several

wireless congestion control mechanisms, namely XCP-b, XC-Winf and WCP, shows that TCP-AP is not very effective and efficient using the medium, obtaining poor performance results.

To improve TCP-AP performance, it was first proposed the integration of the *rt-Winf* estimation technique. This allows to have a more effective and accurate measurement of the wireless capacity and the available bandwidth of wireless links. *rt-Winf* information is then sent to the TCP-AP rate control algorithm through a cross layer communication process. This proposal is called TCP-AP with *rt-Winf*. A performance evaluation of TCP-AP with *rt-Winf*, conducted in mesh and ad-hoc networks, shows that *rt-Winf* clear improves base TCP-AP behavior and performance, allowing base TCP-AP to use more efficiently the medium.

However, TCP-AP with *rt-Winf* keeps unchanged the four hop propagation delay estimation neighborhood technique, not using all the information available on the network. The use of the values measured only within the four hop propagation delay is not accurate. Also, due to the shared nature of the wireless medium, nodes along the path contend for access the medium. This must be accounted for in the available bandwidth estimation. It was then defined a new variant of TCP-AP with *rt-Winf* that introduces the effect of nodes contending for available bandwidth along the path. This introduces a fairness factor allowing an improved fair share of the available bandwidth among all contending nodes. This new enhanced proposal is designated by *Wireless Enhanced TCP-AP* (*WE TCP-AP*).

A complete performance evaluation of *WE TCP-AP*, including utility results, was performed. The results show that *WE TCP-AP* has a better behavior in terms of fairness, and allows to use more efficiently the medium with good network performance results. The utility results show that *WE TCP-AP* quickly adjusts its behavior to TCP behavior, allowing a fair share of network resources.

Chapter 7

Conclusions and Future Research

New research has been propelled on wireless networks environments, by the so called "wireless revolution". It is well known the great expansion in the deployment of wireless networks. The most recently researched wireless networks are mesh wireless networks (WMN) and ad-hoc wireless networks (normally called Mobile Ad-Hoc Networks - MANETs). Nevertheless, these kinds of networks face several problems. Limited capacity and available bandwidth are two of the major factors that limit wireless networks performance that result in pervasive congestion collapses.

Wireless networks use air as the access media, which is a shared medium that is more sensitive to interferences and congestion. The Transmission Control Protocol (TCP) [9], the most widely used congestion protocol in the Internet, was developed for wired networks. Its congestion control algorithms offer reliability in wired networks. However, with the increased demand in wireless networks, TCP became unsuitable on such highly dynamic environments.

Despite the advance in physical-layer transmission technologies, network performance still suffers from unstable and inefficient behavior over wireless networks. Congestion control mechanisms that ensure the efficient use of available bandwidth and link capacity are crucial to the efficiency and development of wireless multi-hop networks. It is of major importance to obtain accurately link capacity and available bandwidth and, then, use these parameters actively in wireless networking congestion control.

Congestion control is an essential building block of modern network design and architecture. Network performance issues lead to the development of new congestion control protocols. The eXplicit Control Protocol (XCP) [15] and the Rate Control Protocol (RCP) [16] are two of the most recent ones. They use network interaction and use rate as the principal congestion parameter. In wireless networks their behavior can be significantly improved if they interact and use available bandwidth and link capacity information obtained in the Medium Access Control (MAC) layer.

This thesis represents our contribution in link capacity and available bandwidth estimation techniques and congestion control mechanisms in wireless environments, in particular tackling three important aspects:

- The active, accurate, real-time, estimation of link capacity and available bandwidth

of wireless links, using the MAC layer information.

- Improve congestion control mechanisms with the support of an accurate capacity and available bandwidth computation algorithm.

- Improve the congestion control approaches by introducing the effect of packet collisions on the measurement of available bandwidth.

In our work, we successfully solved these issues by defining a new available bandwidth estimation algorithm (*rt-Winf*), which was used, through a cross layer communication process, to improve XCP and RCP performance in wireless networks, these new congestion control approaches are called *XCP-Winf* and *RCP-Winf*. We have also extended our work, using the same cross layer communication principles, and proposed to improve TCP-AP with *rt-Winf* information and node path contention count. TCP-AP is a congestion control protocol based on TCP specially designed for ad-hoc wireless networks. We call this new proposal *Wireless Enhanced TCP-AP* (*WE TCP*). In the next section, section 7.1, we describe what we accomplished and the results we obtained, while in section 7.2 we outline further research directions.

7.1 Conclusion

As discussed previously, wireless networks are very easy to deploy and are rapidly spreading to everywhere and used for all kinds of situations. This increase use and deployment has lead to the need of obtaining underlying information, such as link capacity and available bandwidth, efficiently and accurately. One of the most challenging aspect of wireless congestion control is to effectively know the network status, thus, the underlying link capacity and available bandwidth. Therefore, we proposed a method, *rt-Winf* - *Real Time Wireless Inference Mechanism*, to estimate both link capacity and the available bandwidth of a wireless link. *rt-Winf* is a lightweight active estimation tool that is easily implemented with MAC layer information of a wireless node. *rt-Winf* measures the available bandwidth and link capacity through the information included in the Request-to-Send (RTS)/ Clear-to-Send (CTS) packets, measuring the effective transmission time of the existent traffic, through the estimation of each node's channel allocation. In case RTS/CTS are not enabled, *rt-Winf* is able to perform the same operation with a small amount of probing packets. The evaluation results of *rt-Winf*, obtained with th ns-2 simulator and the CMU wireless emulator, show that capacity and available bandwidth are correctly estimated and that the interference and overhead included in the network are very small (smaller than 1%). *rt-Winf* was shown to: (a) be applicable to active congestion control mechanisms, (b) be simple and effective in estimating the network status and (c) introduce low network overhead. *rt-Winf* is based on IdleGap [1], but is a more accurate mechanism, as it determines the real capacity instead of using the IEEE 802.11 header *Data Rate* value. Moreover, it does not introduce any changes in the OSI model.

We have investigated and evaluated, through network simulation, the most used congestion control protocol TCP, against recent rate based and Additive Increase Multiplicative Increase (AIMD) based congestion control mechanisms. The rate based congestion control mechanisms used in the evaluation were XCP, RCP and XCP-blind (XCP-b). They all use explicit feedback information. XCP-b is a congestion control mechanism specially defined for wireless networks that uses indirect heuristic for rate control. The AIMD based congestion control mechanisms evaluated against TCP were the WCP and TCP-AP. The evaluation was conducted in wireless mesh and ad-hoc networks. We discussed the obtained results. The ideas learned from the performance evaluation gave us the new directions to propose improvements in wireless congestion control mechanisms.

Having in mind the results of the previously referred performance evaluation study, we then presented the design of *XCP-Winf* and *RCP-Winf*, two new congestion control mechanisms that use explicit feedback based on precise link capacity and available bandwidth. *XCP-Winf* and *RCP-Winf* are based respectively on XCP and RCP, that are extended with the support of *rt-Winf* accurate capacity and available bandwidth computation algorithm. The link capacity and available bandwidth values obtained through *rt-Winf* are transmitted, through cross layer techniques, to XCP and RCP that use that information in their native congestion control techniques. Due to the special nature of wireless networks and also due to the fact that collisions in the IEEE 802.11 networks occur before congestion, we further improved the congestion control approaches by introducing the effect of packet collisions on the measurement of the available bandwidth. The proposed congestion control approaches were evaluated in different scenarios, mesh and ad-hoc, and were evaluated against state of the art congestion control mechanisms. The obtained results show that the integration of *rt-Winf* allows to clearly improve base XCP and RCP network performance against standard congestion control protocols and, also, against specific congestion protocols for wireless environments. We have also shown the advantages of using explicit capacity information in designing rate control algorithms when compared to AIMD approaches.

XCP-Winf and *RCP-Winf* show considerable gains in terms of throughput and received packets as compared to TCP, XCP, XCP-b and RCP, and also in terms of end-to-end packets delays. The main gains of using *XCP-Winf* and *RCP-Winf* are the better medium usage and more efficient congestion control leading to faster flow completion times and, thus, being more fair. Due to the fact that TCP is still the most used congestion control protocol, it was also evaluated the TCP friendliness of *XCP-Winf* and *RCP-Winf*. The first results showed that both *XCP-Winf* and *RCP-Winf* were at long time TCP friendly. However, to improve their TCP friendliness it was, then, proposed the implementation of a TCP awareness algorithm on both congestion control approaches. New TCP friendly results showed that the TCP awareness algorithm improved TCP friendliness on both *XCP-Winf* and *RCP-Winf*.

Finally, it was also improved the behavior of a specific congestion control protocol for wireless ad-hoc networks, TCP-AP. TCP-AP is a TCP based congestion protocol that uses a hybrid approach with rate based and congestion window control. Within the four hop propagation delay, a TCP source uses rate based mechanisms to do congestion control,

while in the rest of the network it relies in the standard AIMD process. It is showed, through simulation results, that TCP-AP lacks of efficiency and is not using correctly the medium, thus, not evaluating correctly the parameters that are real constraints in wireless environments. To improve TCP-AP behavior, we proposed the integration of *rt-Winf* in TCP-AP, through the same cross layer principles as defined on *XCP-Winf* and *RCP-Winf*. Simulation results showed that the *rt-Winf* integration improved TCP-AP performance. However, TCP-AP with *rt-Winf* still reflects some of TCP-AP constraints, especially concerning fairness and the fact that it does not use the entire network information, as it does not rely in an explicit feedback mechanism.

It was showed, then, that to improve TCP-AP with *rt-Winf* behavior, it was important to introduce the knowledge of all nodes along the path contending for available bandwidth and capacity, introducing an important fairness factor. We called this new approach *Wireless Enhanced TCP-AP* (*WE TCP-AP*). The new simulations conducted showed that the consideration of the node path contention effect and the integration of *rt-Winf* clearly improves base TCP-AP performance. The simulation results were conducted on both ad-hoc and mesh wireless networks and represent, in terms of wireless networks congestion control and behavior, a significant step towards their knowledge. It was also evaluated how TCP-AP with the new improvements was TCP friendly. The obtained results showed that, as TCP-AP relies for the majority of its operations in TCP AIMD process, being just updated with an evaluation and measurement process on the sender side, it quickly adjusts to TCP behavior which allows a fair share of network resources between the two congestion control protocols.

The integration of *rt-Winf* as the underlying estimation tool used by such different congestion control mechanisms highlights its versatility, making us believe that *rt-Winf* can be integrated in other group of congestion control mechanisms and, also, in new routing strategies or policies based on MAC information.

7.2 Future Research

The ongoing research on wireless networking is still looking for better architectures and better congestion control solutions. This Thesis has several contributions both on deploying a mechanism to gather network status information from th MAC layer and proposing some feasible congestion control solutions, that overcome some major issues of the traditional TCP congestion control behavior in wireless networks. The proposed solutions open new paths, on which further research can be done and other issues explored.

This work presents mainly results obtained through simulation evaluation. Future work may involve exploring the implementation of the proposed solutions against other routing protocols, and also implement them directly in the Linux Kernel, allowing to create a small testbed for testing and evaluating the solutions in real environments, and in different conditions.

Another important issue that needs to be addressed is to optimize all system parameters, as the values α and β in XCP, as well as defining mechanisms to dynamically set

those values as, for example, the instant queue size of a node. A dynamic setting of those parameters, adapting them to the changing characteristics of the networks would increase network performance in all possible network scenarios. Thus, research is needed to adapt the definition of those parameters to the network status and behavior.

The proposed solutions and implementations in this Thesis are defined mainly as end-to-end rate based congestion control (XCP and RCP), and four hop propagation delay schemes (TCP-AP). *XCP-Winf* and *RCP-Winf* perform rate based congestion control at end hosts, based on the rate feedback of the intermediate nodes. In very large networks this can introduce some delay in responding to link capacity and available bandwidth change. Future research could define hop-by-hop congestion control, or clustering congestion control systems. In the hop-by-hop congestion control, each node performs rate congestion control with information from the rate feedback of the next hop, sending only with the allowed rate of the next hop. In cluster rate based congestion control, areas of congestion control would be defined, and a node would be responsible for managing rate feedback from other cluster node, and to inform the other nodes of its area. This would possibly reduce the rate changes and potentially increase network performance with better congestion control.

Multi-path transport protocols, like R-MTP (Reliable Multiplexing Transport Protocol) [122], are new transport protocols that aim to provide a way for the use of simultaneous paths at the transport layer and load balancing traffic on these paths. Multi-path transport protocols use network information to react to congestion by moving traffic away from congested paths. Nowadays mobile equipments have often more than one single network interface. For instance, laptops have usually at least both a wired and a wireless network interface adapters. In the same way, smartphones and tablet PCs can use wireless or a cellular network to access Internet. It is clear that, in this context many paths can exist between any two endpoints. Thus, research can be performed in investigating the overall network enhancements of incorporating *rt-Winf* estimations in a multi-path transport framework for wireless environments.

Future work may also involve exploring possibilities of using cross layer information and *rt-Winf* estimations for QoS based routing selection, introducing new estimation parameters, such as power consumption information from the physical layer, and use them together with MAC layer information to optimize network behavior. New extensions of existing routing protocols integrating QoS capabilities based in link capacity and available bandwidth parameters, obtained from the MAC layer, would allow a faster and precise routing decision, improving overall network performance. Available link bandwidth and link capacity estimation would serve as a basis for a routing strategy based on admission control for data flows. This would greatly increment the packet delivery ratio and, thus, network performance. It would also be advisable to use other techniques to determine new underlying control parameters.

The above list is not exhaustive, showing only some possibilities that could be addressed for the success of wireless networks. While wired networking has reached a steady state after the development of the legacy protocol stack, wireless networking is relatively young. The IEEE 802.11 standard has triggered an unpredicted success of the wireless networking. The amount of research done in wireless networking so far has produced its results, but

more is still to come. Time and research will deepen our understandings, allowing us to mature the wireless networking technology.

Bibliography

[1] H. Lee, V. Hall, K. Yum, K. Kim, and E. Kim, "Bandwidth estimation in wireless LANs for multimedia streaming services," *Advances in Multimedia*, vol. 2007, no. 1, pp. 9–9, 2007.

[2] M. Conti, G. Maselli, G. Turi, and S. Giordano, "Cross-layering in mobile ad hoc network design," *Computer Journal*, vol. 37, pp. 48–51, August 2004.

[3] J. Geier, *Designing and Deploying 802.11n Wireless Networks*. Cisco Press, 2010.

[4] M. Rahnema, "Overview of the GSM system and protocol architecture," *Communications Magazine, IEEE*, vol. 31, no. 4, pp. 92–100, April 1993.

[5] *IEEE Std 802.11 - 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 Computer Society Standard, 2007. [Online]. Available: http://standards.ieee.org/getieee802/download/802.11-2007.pdf

[6] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, no. 4, pp. 445 – 487, March 2005.

[7] J. G. Andrews, A. Ghosh, and R. Muhamed, *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*, 1st ed. Prentice Hall, March 2007.

[8] E. C. Efstathiou, P. A. Frangoudis, and G. C. Polyzos, "Stimulating participation in wireless community networks," in *INFOCOM*. Barcelona, Catalunya, Spain: IEEE, April 2006.

[9] J. Postel, "Transmission control protocol," RFC 793, Defense Advanced Research Projects Agency, September 1981. [Online]. Available: http://www.rfc-editor.org/rfc/rfc793.txt

[10] S. Vangala, , and M. A. Labrador, "Performance of TCP over wireless networks with the snoop protocol," in *In Proceedings of the 27th Annual IEEE Conference on Local Computer Networks (LCN 2002)*, November 2002.

[11] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and Y. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," in *IEEE/ACM Transactions on Networking*, vol. 5, December 1997, pp. 756–769.

[12] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Jounal of Computer Networks and ISDN Systems*, vol. 17, pp. 1–14, June 1989.

[13] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, ser. MobiCom '99. New York, NY, USA: ACM, September 1999, pp. 219–230.

[14] P. S. J. Monks and V. Bharghavan, "Limitations of TCP-ELFN for ad hoc networks," in *7th International Work on Mobile Multimedia Communications (MoMuC 2000)*, Tokyo, Japan, October 2000.

[15] A. Falk and D. Katabi, *Specification for the Explicit Control Protocol (XCP), draft-falk-xcp-spec-03.txt*, Information Sciences Institute Internet-Draft. [Online]. Available: http://www.isi.edu/isixcp/docs/draft-falk-xcp-spec-00.html

[16] N. Dukkipati and N. McKeown, "Why flow-completion time is the right metric for congestion control," *ACM SIGCOMM Computer Communications Review*, vol. 36, pp. 59–62, January 2006.

[17] H. Zhai, Y. Kwon, and Y. Fang, "Performance analysis of IEEE 802.11 MAC protocols in wireless lans," *Wireless Communications and Mobile Computing*, vol. 4, no. 8, pp. 917–931, December 2004.

[18] N. Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques," *Selected Areas in Communications, IEEE Journal on*, vol. 21, no. 6, pp. 879–894, August 2003.

[19] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, "Multimedia streaming via TCP: An analytic performance study," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 4, pp. 16:1–16:22, May 2008.

[20] S. M. ElRakabawy, A. Klemm, and C. Lindemann, "TCP with adaptive pacing for multihop wireless networks," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, ser. MobiHoc '05. New York, NY, USA: ACM, May 2005, pp. 288–299.

[21] S. Rangwala, A. Jindal, K.-Y. Jang, K. Psounis, and R. Govindan, "Understanding congestion control in multi-hop wireless mesh networks," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, ser. MobiCom '08. ACM, September 2008, pp. 291–302.

[22] F. Abrantes and M. Ricardo, "A simulation study of XCP-b performance in wireless multi-hop networks," in *Proceedings of the 3rd ACM workshop on QoS and security for wireless and mobile networks*, ser. Q2SWinet '07. ACM, 2007.

[23] "The network simulator - ns-2," 2001. [Online]. Available: http://www.isi.edu/nsnam/ns/

[24] "CMU wireless emulator." [Online]. Available: http://www.cs.cmu.edu/emulator/

[25] S. M. Buhari, M. H. Habaebi, and B. M. Ali, "A new congestion control algorithm for active networks," *Pertanika Journal of Science and Technology*, pp. 187–211, April 2005.

[26] C. Dovrolis, R. Prasad, M. Murray, and k. claffy, "Bandwidth estimation: metrics, measurement techniques, and tools," *IEEE Network*, no. 6, pp. 27–35, Apr 2003.

[27] C. Dovrolis, P. Ramanathan, and D. Moore, "Packet dispersion techniques and a capacity estimation methodology," *IEEE/ACM Transactions on Networking*, vol. 12, pp. 963–977, December 2004.

[28] R. Carter and M. Crovella, "Dynamic server selection using bandwidth probing in wide-area networks," Boston, MA, USA, Tech. Rep., 1996.

[29] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics and relation with TCP throughput," *ACM SIGCOMM*, pp. 537–549, August 2002.

[30] "IPerf." [Online]. Available: http://dast.nlanr.net/Projects/Iperf/

[31] D. Antoniades, M. Athanatos, A. Papadogiannakis, E. P. Markatos, and C. Dovrolis, "Available bandwidth measurement as simple as running wget," *Passive and Active Measurement (PAM) Workshop*, March 2006.

[32] V. Jacobson, "Pathchar: A tool to infer characteristics of internet paths," *MSRI talk*, April 1997.

[33] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cot, "Pathchirp: Efficient available bandwidth estimation for network paths," *Passive and Active Measurement (PAM) Workshop*, April 2003.

[34] R. Kapoor, L. J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi, "Capprobe: A simple and accurate capacity estimation technique." *ACM SIGCOMM*, vol. 34, no. 4, August 2004.

[35] K. Lakshminarayanan, V. N. Padmanabhan, and J. Padhye, "Bandwidth estimation in broadband access networks," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. ACM, October 2004, pp. 314–321.

[36] S. ju Lee, P. Sharma, S. Banerjee, S. Basu, and R. Fonseca, "Measuring bandwidth between planetlab nodes," in *Passive and Active Network Measurement, 6th International Workshop*, ser. Lecture Notes in Computer Science, March 2005, pp. 292–305.

[37] L.-J. Chen, T. Sun, G. Yang, M. Sanadidi, and M. Gerla, "Adhoc probe: Path capacity probing in wireless ad hoc networks," *Wireless Networks*, vol. 15, no. 1, pp. 111–126, January 2009.

[38] T. Sun, L. jyh Chen, G. Yang, M. Y. Sanadidi, and M. Gerla, "SenProbe: Path capacity estimation in wireless sensor networks," in *Wireless Traffic Measurements and Modeling Workshop (SenMetrics2005)*, January 2005.

[39] M. Li, M. Claypool, and R. Kinicki, "WBest: a bandwidth estimation tool for IEEE 802.11 wireless networks," *IEEE LCN*, October 2008.

[40] L.-J. Chen, C.-W. Sung, H.-H. Hung, T. Sun, and C.-F. Chou, "Tsprobe: A link capacity estimation tool for time-slotted wireless networks," in *Wireless and Optical Communications Networks, 2007. WOCN '07. IFIP International Conference on*, July 2007, pp. 1 –5.

[41] P. Brenner, "A technical tutorial on the IEEE 802.11 protocol," Department of Engineering, BreezeCom Wireless Communications, July 1997. [Online]. Available: http://sss-mag.com/pdf/802_11tut.pdf

[42] T. Oetiker, "Mrtg: The multi router traffic grapher." in *LISA*. USENIX, 1998, pp. 141–148. [Online]. Available: http://dblp.uni-trier.de/db/conf/lisa/lisa1998.html#Oetiker98

[43] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot, "Packet-level traffic measurements from the sprint ip backbone," *IEEE Network*, vol. 17, no. 6, pp. 6–16, November 2003. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1248656

[44] T. En-Najjary and G. Urvoy-Keller, "PPrate: A passive capacity estimation tool." in *Fourth IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services, E2EMON 06, 3rd April, 2006, Vancouver, BC, Canada*. IEEE, 2006, pp. 82–89. [Online]. Available: http://dblp.uni-trier.de/db/conf/e2emon/e2emon2006.html#En-NajjaryU06

[45] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "Simple Network Management Protocol (SNMP)," RFC 1157, Internet Engineering Task Force, United States, May 1990. [Online]. Available: http://tools.ietf.org/html/rfc1157

[46] K. Lakshminarayanan, V. N. Padmanabhan, and J. Padhye, "Bandwidth estimation in broadband access networks," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, October 2004, pp. 314–321.

[47] M. S. Gast, *802.11 Wireless Networks - The Definitive Guide*, 4th ed. O'Reilly, 2002.

[48] H. Zimmermann, "OSI reference model- the ISO model of architecture for open systems interconnection," *IEEE Transactions on Communications*, vol. 28, no. 4, April 1980.

[49] W. R. Stevens, *TCP/IP illustrated (vol. 1): the protocols*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1994.

[50] B. A. Fourouzan, *TCP/IP Protocol Suite*, 2nd ed. McGraw-Hill Higher Education, 2002.

[51] A. DeSimone, M. C. Chuah, and O.-C. Yue, "Throughput performance of transport-layer protocols over wireless LANs," in *Proc. IEEE Global Telecommunications Conference, including a Communications Theory Mini-Conference. Technical Program Conference Record, IEEE in Houston. GLOBECOM '93*, December 1993, pp. 542–549.

[52] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A feedback based scheme for improving TCP performance in ad-hoc wireless networks," pp. 472–, April 2001.

[53] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," in *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. ACM, 1999, pp. 219–230.

[54] D. Kim, C.-K. Toh, and Y. Choi, "TCP-BuS: improving TCP performance in wireless ad hoc networks," vol. 3, June 2000, pp. 1707 –1713 vol.3.

[55] C. L. T. Man, G. Hasegawa, and M. Murata, "ImTCP: TCP with an inline measurement mechanism for available bandwidth," *Elsevier Computer Communications*, vol. 29, pp. 1614–1626, June 2006.

[56] J. Liu and S. Singh, "ATCP: TCP for mobile ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 19, no. 7, pp. 1300 –1315, July 2001.

[57] G. Anastasi, E. Ancillotti, M. Conti, and A. Passarella, "TPA: a transport protocol for ad hoc networks," in *Computers and Communications, 2005. ISCC 2005. Proceedings. 10th IEEE Symposium on*, June 2005, pp. 51 – 56.

[58] C. Cordeiro, S. Das, and D. Agrawal, "Copas: dynamic contention-balancing to enhance the performance of TCP over multi-hop wireless networks," in *Computer Communications and Networks, 2002. Proceedings. Eleventh International Conference on*, October 2002, pp. 382 – 387.

[59] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP performance," *Mobile Computing, IEEE Transactions on*, vol. 4, no. 2, pp. 209 – 221, April 2005.

[60] K. Xu, M. Gerla, L. Qi, and Y. Shu, "Enhancing TCP fairness in ad hoc wireless networks using neighborhood red," in *Proceedings ACM MobiCom*. ACM Press, September 2003, pp. 16–28.

[61] K. Tan, F. Jiang, Q. Zhang, and X. Shen, "Congestion control in multihop wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 2, pp. 863–873, March 2007.

[62] C. keong Toh, "Associativity-based routing for ad-hoc mobile networks," *Jounal of Wireless Personal Communications*, vol. 4, pp. 103–139, March 1997.

[63] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, August 1993.

[64] S. H. Low, F. Paganini, J. Wang, S. Adlakha, and J. Doyle, "Dynamics of TCP/RED and a scalable control," in *INFOCOM*, November 2002.

[65] D. Clark, "Window and Acknowledgement Strategy in TCP," RFC 813, Internet Engineering Task Force, July 1982. [Online]. Available: http://www.ietf.org/rfc/rfc813.txt

[66] Y. Su and T. Gross, "WXCP: Explicit congestion control for wireless multi-hop networks," in *In IWQoS 05: Proceedings of the 12th International Workshop on Quality of Service*, November 2005.

[67] "100x100 clean state project." [Online]. Available: http://100x100network.org/

[68] C.-H. Tai, J. Zhu, and N. Dukkipati, "Making large scale deployment of rcp practical for real networks," in *INFOCOM 2008. 27th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 13-18 April 2008, Phoenix, AZ, USA*. IEEE, 2008, pp. 2180–2188.

[69] A. Lakshmikantha, R. Srikant, N. Dukkipati, N. McKeown, and C. Beck, "Buffer sizing results for rcp congestion control under connection arrivals and departures," *SIGCOMM Comput. Commun. Rev.*, vol. 39, pp. 5–15, December 2008.

[70] A. Sridharan and B. Krishnamachari, "Explicit and precise rate control for wireless sensor networks," in *SenSys '09: Proceedings of the 7th ACM conference on Embedded network sensor systems*. New York, NY, USA: ACM, November 2009.

[71] K. young Jang, K. Psounis, and R. Govindan, "Simple yet efficient, transparent airtime allocation for TCP in wireless mesh networks," in *Proceedings of the 6th international conference on Emerging networking experiments and*

technologies, ser. CoNEXT '10, December 2010, pp. 28:1–28:12. [Online]. Available: http://www-bcf.usc.edu/~kpsounis/Papers/cna.pdf

[72] M. Li, D. Agrawal, D. Ganesan, and A. Venkataramani, "Block-switched networks: A new paradigm for wireless transport," in *Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, ser. NSDI'09, April 2009, pp. 423–436.

[73] A. Aziz, D. Starobinski, P. Thiran, and A. E. Fawal, "EZ-flow: Removing turbulence in IEEE 802.11 wireless mesh networks without message passing," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, ser. CoNEXT '09, December 2009, pp. 73–84.

[74] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla', "The impact of multihop wireless channel on TCP throughput and loss," *IEEE INFOCOM*, March 2003.

[75] R. Rom and M. Sidi, *Multiple Access Protocols: Performance and Analysis*. Springer. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0387972536

[76] F. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part ii–the hidden terminal problem in carrier sense multiple-access and the busy-tone solution," *Communications, IEEE Transactions on*, vol. 23, no. 12, pp. 1417 – 1433, December 1975.

[77] Z. Haas, "On the performance of a medium access control scheme for the reconfigurable wireless networks," in *Military Communications Conference (MILCOM) 97 Proceedings*, vol. 3, November 1997, pp. 1558 –1564 vol.3.

[78] S. Ray, J. B. Carruthers, and D. Starobinski, "Evaluation of the masked node problem in ad-hoc wireless LANs," *IEEE Transactions on Mobile Computing*, vol. 4, pp. 430–442, September 2005.

[79] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP throughput and loss," 2003, pp. 1744–1753.

[80] P. Chatzimisios, A. Boucouvalas, and V. Vitsas, "Effectiveness of RTS/CTS handshake in IEEE 802.11a wireless lans," *Electronics Letters*, vol. 40, no. 14, pp. 915 – 916, July 2004.

[81] K. Xu, M. Gerla, and S. Bae, "How effective is the IEEE 802.11 rts/cts handshake in ad hoc networks," in *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, vol. 1, November 2002, pp. 72 – 76 vol.1.

[82] H. Khalife and N. Malouch, "Interaction between hidden node collisions and congestions in multihop wireless ad-hoc networks," in *IEEE International Conference*

on Communications, 2006. ICC '06., vol. 9, June 2006, pp. 3947 –3952. [Online]. Available: www-rp.lip6.fr/~khalife/extended.pdf

[83] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535 –547, March 2000.

[84] C. H. Foh, J. W. Tantra, and S. Member, "Comments on IEEE 802.11 saturation throughput analysis with freezing of backoff counters," *IEEE Communication Letters*, vol. 9, pp. 130–132, February 2005.

[85] L. Romdhani, Q. Ni, and T. Turletti, "Adaptive EDCF: Enhanced service differentiation for IEEE 802.11 wireless ad-hoc networks," in *IEEE 802.11 Wireless Ad Hoc Networks, IEEE WCNC03 (Wireless Communications and Networking Conference*, vol. 2, March 2003, pp. 16–20.

[86] J. W. Robinson and T. S. Randhawa, "Saturation throughput analysis of IEEE 802.11e enhanced distributed coordination function," *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 5, pp. 917–928, June 2004.

[87] D. Qiao and S. Choi, "Goodput enhancement of IEEE 802.11a wireless lan via link adaptation," in *in Proceedings of IEEE ICC2001*, vol. 7, June 2001, pp. 1995–2000.

[88] Q. Pang, S. C. Liew, and V. C. M. Leung, "Design of an effective loss-distinguishable MAC protocol for 802.11 WLAN," *IEEE Communications Letters*, vol. 9, no. 9, September 2005.

[89] C. Sarr, C. Chaudet, G. Chelius, and I. G. Lassous, "Improving accuracy in available bandwidth estimation for IEEE 802.11 based ad-hoc networks," in *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, October 2006, pp. 517 –520.

[90] M. Franceschetti, M. Migliore, and P. Minero, "The capacity of wireless networks: Information-theoretic and physical limits," *Information Theory, IEEE Transactions on*, vol. 55, no. 8, pp. 3413 –3424, August 2009.

[91] A. Goldsmith and S. Wicker, "Design challenges for energy-constrained ad hoc wireless networks," *IEEE Wireless Communications*, vol. 9, no. 4, pp. 8 – 27, August 2002.

[92] D. D. Clark and D. L. Tennenhouse, "Architectural considerations for a new generation of protocols," *SIGCOMM Comput. Commun. Rev.*, vol. 20, pp. 200–208, August 1990.

[93] C. Jin, D. Wei, S. H. Low, J. Bunn, H. D. Choe, J. C. Doyle, H. Newman, S. Ravot, and S. Singh, "Fast TCP: From theory to experiments," *IEEE Network*, vol. 19, pp. 4–11, January 2005.

[94] Q. Wang and M. A. Abu-Rgheff, "Cross-layer signalling for next-generation wireless systems," in *IEEE Wireless Communications and Networking, 2003. IEEE WCNC 2003*, vol. 2, March 2003, pp. 1084–1089 vol.2.

[95] G. Carneiro, J. Ruela, and M. Ricardo, "Cross-layer design in 4g wireless terminals," *IEEE Wireless Communications*, vol. 11, no. 2, pp. 7 – 13, apr 2004.

[96] S. Kunniyur and R. Srikant, "End-to-end congestion control schemes: utility functions, random losses and ecn marks," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 689–702, October 2003.

[97] "Eventhelix cross layer." [Online]. Available: http://www.eventhelix.com

[98] M. J. Neely, "Energy optimal control for time varying wireless networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, March 2005, Miami, FL, USA*. IEEE, 2005, pp. 572–583.

[99] L. Georgiadis, M. J., and R. Tassiulas, "Resource allocation and cross-layer control in wireless networks," in *Foundations and Trends in Networking*, vol. 1. Hanover, MA, USA: Now Publishers Inc., April 2006, pp. 1–149.

[100] Y. Xi and E. M. Yeh, "Throughput optimal distributed power control of stochastic wireless networks," *IEEE/ACM Transactions on Networking*, vol. 18, pp. 1054–1066, August 2010.

[101] B. Johansson, S. Member, P. Soldati, and M. Johansson, "Mathematical decomposition techniques for distributed cross-layer optimization of data networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 1535–1547, August 2006.

[102] J. Liu, B. Li, Y. Hou, and I. Chlamtac, "On optimal layering and bandwidth allocation for multisession video broadcasting," *IEEE Transactions on Wireless Communications*, vol. 3, no. 2, pp. 656 – 667, March 2004.

[103] X. Wang and K. Kar, "Cross-layer rate optimization for proportional fairness in multihop wireless networks with random access," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 8, pp. 1548–1559, August 2006.

[104] J. Yuan, Z. Li, W. Yu, and B. Li, "A cross-layer optimization framework for multihop multicast in wireless mesh networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 11, pp. 2092–2103, November 2006.

[105] V. Jacobson and M. J. Karels, "Congestion avoidance and control," *ACM SIGCOMM Computer Communication Review*, August 1988.

[106] M. Fiore, "trace2stats." [Online]. Available: http://www.tlc-networks.polito.it/fiore/index.html

[107] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *ACM SIGCOMM Computer Communications*, October 1994.

[108] S. Floyd, T. Henderson, and A. Gurtov, "The NewReno Modification to TCP's Fast Recovery Algorithm," Internet Engineering Task Force, April 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3782.txt

[109] "Google code: TCP-AP ns-2 files," 2012. [Online]. Available: http://code.google.com/p/uu-cope/source/detail?r=170

[110] "WCP source code," 2012. [Online]. Available: http://enl.usc.edu/enl/trunk/wcp/click-code/

[111] "Qualnet," 2012. [Online]. Available: http://www.scalable-networks.com/products/

[112] "The click modular router project," 2012. [Online]. Available: http://read.cs.ucla.edu/click/click

[113] "Nsclick," 2012. [Online]. Available: http://read.cs.ucla.edu/click/nsclick

[114] N. BenAmmar and J. S. Baras, "Incentive compatible medium access control in wireless networks," in *GameNets '06: Proceeding from the 2006 workshop on Game theory for communications and networks*. New York, NY, USA: ACM, October 2006, p. 5.

[115] "Recommendation ITU-T G.711.0," ITU, 2005. [Online]. Available: http://www.itu.int/rec/T-REC-G.711/

[116] eljko Ili, A. Baant, I. olak, and D. Jaki, "Optimal MAC packet size in wireless LAN," *International conference on information and communication technology, electronics and microelectronics (MIPRO)*, June 2005.

[117] K. Sundaresan, V. Anantharaman, H.-Y. Hsieh, and A. R. Sivakumar, "ATP: a reliable transport protocol for ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 6, pp. 588–603, November 2005.

[118] K. Chen, K. Nahrstedt, and N. Vaidya, "The utility of explicit rate-based flow control in mobile ad hoc networks," *Wireless Communications and Networking Conference, 2004. WCNC 2004 IEEE*, vol. 3, pp. 1921–1926, March 2004.

[119] H. Zhao, E. Garcia-Palacios, J. Wei, and Y. Xi, "Accurate available bandwidth estimation in IEEE 802.11 based ad-hoc networks," *Elsevier Computer Communications*, vol. 32, no. 6, pp. 1050 – 1057, April 2009.

[120] J. F. L. Xu, "TCP-friendly CBR-like Rate Control," *IEEE Internation Conference on Network Protocols 2008 (ICNP)*, pp. 177 – 186, October 2008.

[121] C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?" *IEEE INFOCOM*, January 2001.

[122] L. Magalhaes and R. Kravets, "Transport level mechanisms for bandwidth aggregation on mobile hosts," in *Network Protocols, 2001. Ninth International Conference on.* IEEE Computer Society, November 2001, pp. 165 – 171.