# PYLON-LITE:
# AN ARCHITECTURAL MODEL
# FOR CROSS-DOMAIN QOS

By:

Yasser L. Morgan

A thesis submitted to the
Faculty of Graduate Studies and Research
in final fulfillment of the requirements for
the degree of Doctor of Philosophy

Ottawa Carleton Institute of Computer Science
Carleton University
Ottawa, Ontario, Canada, K1S 5B6

May 2005

# Abstract

In recent years topics related to mobile wireless networks have evolved into a major research interest fueled by the latest advances in radio technologies, and the special appeal of wireless devices to the end users. Quality of service research in mobile ad-hoc networks is a specifically difficult topic due to the network dynamics, variations in radio link quality, limited capabilities of mobile nodes, and lack of central authority.

This research proposes the PYLON-Lite[1] QoS model that deals with the cross-domain QoS connectivity issues. A PYLON-Lite gateway operates between the ad-hoc network, with its unique characteristics on one side, and the fixed topology access network on the other side. The fundamental differences between these two networks represent the heterogeneous environment of the PYLON-Lite model. The PYLON-Lite QoS model provides a, seemingly, homogeneous cross-domain QoS solution. PYLON-Lite has a supple model design that facilitates a lightweight implementation to benefit from the already existing QoS models.

The development of PYLON-Lite is motivated by the need of mobile nodes in the ad-hoc network to access the Internet. PYLON-Lite relies on the QoS models implemented on each side of the gateway to provide detailed services in the relevant network, while it remains focused on the QoS concatenation issues. The PYLON-Lite gateway design follows the principles of cascaded network services. No comparable models are found in the literature to cover the defined problem to this date; therefore, PYLON-Lite is unique.

This research identifies the challenges in designing the cross-domain QoS model. PYLON-Lite presents specific mechanisms to deal with those challenges while maintaining the lightweight approach. It defines methods to interact with other QoS models on both sides of the gateway, methods to police traffic, and conditions to guarantee model scalability. This research provides intensive analysis and evaluation of the model performance and behavioral characteristics. PYLON-Lite is shown to consistently improve the QoS provided for real-time traffic with limited impact on best-effort traffic.

[1] In ancient Egyptian and Greek languages, PYLON is the gateway, the major entrance to temples and the sky.

# Acknowledgement

Appreciate all the angels who were there when I needed most.

GOD from the first step till now; always felt your presence.

… …

I am grateful.

# Table of Contents

# List of Figures

# List of Tables

CHAPTER 1

# Introduction and Overview

Mobile ad-hoc networks have captured the attention of many data network researchers in the current decade. The Internet Engineering Taskforce (IETF) has crystallized the interest in mobile ad-hoc networks by forming a working group called MANET (Mobile Ad-hoc Networking). Since the MANET working group issued its charter [25] on January 1999, researchers have benefited from focusing on the challenges it defines. The MANET charter views the ad-hoc domain as a network that can be formed by a group of mobile nodes which are able to operate as hosts and routers at the same time. Ad-hoc mobile nodes may submit, consume, or route the network traffic. The charter [25] also outlines the limited processing, and storage capabilities of mobile nodes, in addition it illustrates the difficulties introduced by various mobility scenarios and network dynamics. The MANET networking group has been focusing on solving the main challenges facing the deployment and commercialization of ad-hoc networks by specifically defining a suitable set of routing algorithms. MANET has decided to materialize the workgroup efforts of the last five years by promoting four routing proposals to the *Internet Engineering Standard Group* IESG [122] by the end of 2004.

Non-routing ad-hoc issues such as Quality of Service (QoS), energy consumption, ad-hoc gateway design and discovery, ad-hoc IP auto-configuration and addressing schemes, and many other issues are not considered a core part of the MANET by definition. Thus research on areas like QoS has evolved inconsistently and without a high level view of what an ad-hoc QoS should really mean and provide. The current QoS research efforts are a collection of independent research, and some military research projects.

## 1.1 QoS Guarantees

RFC 2386 [26] characterizes QoS as a set of service requirements to be met by the network while transporting a packet stream from source to destination. Intrinsic to the notion of QoS is an agreement or a guarantee by the network to provide a set of measurable pre-specified service attributes to the user in terms of trans-network delay, delay variance (jitter), provided bandwidth, and probability of packet loss as described in RFC 2386 [26]. The network may provide the services with different levels of guarantees as follows:

- **Deterministic Guarantee:** Deterministic guarantees assure the delivery of services equal to or better than the requested services.

- **Statistical Guarantees:** Statistical guarantees assure the delivery of the requested services to a certain percentage of the traffic over a predefined period of time. An example could be the delivery of 95% of packets with the requested service level over a long time period [10].

- **Soft Guarantees:** Soft guarantees means the network tries to achieve the targeted level of service, but can not guarantee the targeted level of services. Soft guarantees fit into the mobile ad-hoc environment. Soft guarantees are equivalent to statistical guarantees in a zero mobility environment [20].

- **Best Effort (No Guarantees):** Best-effort services provide no service prediction, and no preferable treatment to different traffic packets.

## 1.2 The Need for QoS in Ad-hoc Domains

Ad-hoc domains are known for having limited resources in terms of bandwidth, storage, or processing power of mobile nodes. Lack of resources fuels the need for sound QoS models that are lightweight, distributed, and require minimum signaling. However, arguments against IP QoS surface in research literature from time to time like in [18] and [27]. Classical quarrels against QoS in ad-hoc networks stand on the following arguments:

- **Availability of infinite bandwidth:** The argument is that bandwidth will be available, and at a low cost. However, this low cost availability is not expected to happen soon. In addition, wireless networks are expected to continue to provide less bandwidth than wire-line networks, and therefore, wireless bandwidth remains a scarce resource.

- **Simple priority is sufficient:** The argument here could be true in many situations. However, assume that all real-time flows get higher priority, congestion becomes more possible, and this leads to users loosing their connectivity during the lifetime of a session. A better approach can be to serve the already admitted flows, and deny new requests.

- **Applications can adapt:** The development of adaptive real-time applications draws the attention to adding intelligence to applications. However, while applications can help hiding some network delays, for example, the human need for interaction and intelligibility limit the possibility of relying on application capabilities.

The *"wireless lag tenet"* is the tenet that wireless networks will always be more than a decade behind wire-line networks in terms of bandwidth or resources like storage and processing power. Therefore, QoS is expected to remain a fundamental component of the wireless ad-hoc networks. In conclusion, there is an inescapable need for mobile nodes to be able to treat traffic differently and in a distributed, decentralized, and reliable manner.

QoS models are embedded within the network service interface, and are invoked by the application, which defines desired QoS parameters. The actual resources used to satisfy the desired QoS exist in the network infrastructure. Both the underlying network and the user application will evolve over time; the service model will have to evolve as well to respond to changing requirements. However, the service model has to maintain the same service interface for compatibility reasons. This means that the service model must be flexible enough to accept adding new services, but keeping the same interface.

## 1.3 QoS Challenges in Ad-hoc Domains

A domain is, typically, defined to be a single network that is administrated by a single authority. However, in ad-hoc networks the concept of administrative authority is absent. This research uses the term ad-hoc domain to refer to a single ad-hoc network. Ad-hoc mobile nodes facilitate interconnections between remote nodes by relying on peer-to-peer wireless communications where mobile nodes operate as routers on behalf of source-destination pairs. The dynamics of ad-hoc networks in terms of node mobility, limited battery power, and variable radio quality, make it difficult to support real-time applications with appropriate QoS. The network dynamics also make it difficult to assign a central controller to maintain connection state and reservations. Major QoS challenges facing ad-hoc domains can be summarized as follows:

1- QoS challenges due to mobility of nodes. These challenges make it difficult to maintain resources on specific routes. The network dynamics impose inherent limitations to QoS promises in terms of connectivity, and robustness.

2- QoS challenges due to unpredictable link properties such as interference with other wireless devices, signal fading, or hidden node issues. This problem results in variant resources even on a fixed route and even assuming no mobility, for instance, due to interference with, potentially, wireless devices outside the ad-hoc domain. The unpredictability of a wireless link causes potential variations in link capacity, and therefore, inherent limitations on the expected QoS guaranties.

3- QoS challenges due to limited capabilities of mobile nodes in terms of processing power, storage capacity, or energy. The limited capabilities challenge, influence, and shape the QoS design for instance by forcing a distributed approach, avoiding lookup tables, accommodating dormant devices, or adopting simpler lightweight algorithms.

4- QoS challenges due to the lack of a central authority that can maintain central information on flows, routes, or connections. The challenge here is to design a decentralized QoS schemes.

All these challenges lead to serious concerns specially when designing scalable, robust, and distributed QoS architectural models.

## 1.4 QoS Approaches from a Layered Perspective

Current ad-hoc QoS research can be categorized into major design approaches from a layered perspective.

### 1.4.1 Media Access Control (MAC) QoS

The MAC approach provides QoS support at the media access control (MAC) layer. Radio channels are shared media, and can be shared differently to provide service differentiation for instance by assigning larger time slots for higher priority packets. However, implementing such a simple principle turns out to be a fairly complicated process. A common mechanism uses a distributed control scheme as in ([29], [30], [64], [78], [99], and [104]).

Best-effort distributed MAC controllers are widely used in existing wireless ad-hoc networks. The IEEE 802.11 *Distributed Coordination Function* (DCF) is a good example of a best-effort distributed MAC. The *Enhanced Distributed Coordination Function* (EDCF) is a growing IEEE 802.11 alternative that facilitates prioritized packet transmission [28]. Recently, there have been a number of proposals to support service differentiation at the MAC layer using distributed control schemes like [99] and [104]. Kanodia [54] adopts a priority-scheduling algorithm for sharing the radio media efficiently based on QoS requirements. HAVANA [41] proposes an algorithm to predict the fading of a wireless channel, and suggests compensating flows experiencing bad link quality.

Other MAC QoS solutions are designed specifically for cellular networks or Wireless Local Area Networks. Those solutions are reviewed in Appendix A. WLAN QoS solutions, in particular; always fall into the MAC QoS category. MAC QoS approaches provide valuable mechanisms for achieving per-link service differentiation [65] and [118], complementary to other approaches, but are incomprehensive since they cannot select an end-to-end QoS path.

### 1.4.2 QoS-aware Routing

QoS-aware routing considers the QoS dimension when performing route selection and packet scheduling. Embedding QoS in routing mechanisms can solve many of the problems faced during the QoS implementation on fixed topology networks running classical routing algorithms such as OSPF (*Open Shortest Path First*) [76]. For instance, it is possible to avoid building MPLS (*Multi-Protocol Label Switching*) tunnels [96] and [109] as illustrated by [20] and [115].

The QoS-aware routing approach is still in a very early research phase in the ad-hoc environment, and limited numbers of proposals have evolved so far. For example, the *Ad-hoc On-demand Distance Vector* (AODV) protocol proposed [86], the *Dynamic Source Routing* (DSR) protocol proposed [63], and the *Optimized Link State Routing* (OLSR) protocol proposed [77]. Those proposals are yet in a draft form, and limited evaluation of their efficacy has taken place, see for example [5]. QoS routing is valuable in finding optimal QoS routes, complementary to other approaches, but incomprehensive since it cannot perform service recovery, route maintenance, or process QoS reports.

### 1.4.3 Inter-layer QoS Models

The third research approach has been focused on providing inter-layer QoS solutions that can operate over different routing mechanisms and various media access layer. The inter-layer QoS model approach, in a sense, follows the flavor of QoS solutions for fixed topology networks namely by viewing routing mechanisms as one distinct component that can interact with the QoS model.

This approach has lead to less than a handful of QoS solutions that operate independent of the underlying routing algorithm or MAC layer implementation. This approach has started by importing QoS solutions from fixed topology networks as in the *Flexible QoS Model for Mobile* ad-hoc networks (FQMM) [110] or the *Dynamic QoS* for mobile ad-hoc networks (dQoS) [70] and [71]. Then this approach evolved to realize the unique characteristics of ad-hoc networks as in *In-band Signaling* (INSIGNIA) [62] and *Stateless Wireless Ad-hoc Networks* (SWAN) [2]. The inter-layer QoS approaches provide comprehensive solutions, but are less efficient since they cannot perform optimization at both MAC and routing layers.

### 1.4.4 QoS-aware Applications

QoS-aware applications form a set of applications that can adapt to limited variations in the service provided by the network and hide such variations from the application user. For instance, [39] provides a thorough analysis of QoS-aware applications, and tries to enhance application adaptability to variable levels of services. QoS-aware applications have significantly evolved over the last decade. Used techniques vary between compression algorithms, layered encoding, rate shaping, and adaptive error control. It is conceivable that some modifications and improvements are required for the ad-hoc environment. QoS-aware applications cannot solve all MANET QoS challenges like fading of wireless channels which can be solved by rerouting. QoS-aware applications can help providing the user with seemingly reasonable network performance.

### 1.4.5 Other QoS Solutions

Other ad-hoc QoS researchers follow hybrid approaches or focus on finding solutions to specific issue within the QoS problem domain. For instance, [108] provides a tool to maintain DiffServ QoS when mobile nodes perform handoff. Corson proposes a *Five-Phase Reservation Protocol* (FPRP) that can run over *Time Division Multiple Access* TDMA networks in order to perform resource reservation. Another example is *RSVP-Mobile IP* (RSVP-MP), which is a variant version of the *Resource Reservation Protocol* (RSVP), designed to operate in mobile environments [84]. These, and similar efforts are important contributions that solve different pieces of the QoS problem domain.

## 1.5 Research Motivation

This research is, generally, motivated by the increasing interest in wireless networks. The increasing demand, affordability, and the special appeal of location independent wireless devices fuel the need for specially designed gateways that can integrate the wireless and wire-line services into a, seemingly, homogeneous global network. The characteristics of the wireless evolution are taking shape through several parallel wireless development approaches as follows:

- The deployment of Wireless Local Area Networks (WLAN) is increasing rapidly even for home-based networks. Industrial, educational, and corporate use of WLAN bandwidth is growing fast and is pushing the need for more resources.
- The growth in cellular networks, specifically the Universal Mobile Telecommunications Service (UMTS) and the General Packet Radio Service (GPRS). Cellular network providers are currently adding data services to their voice services.
- The increasing use of satellite networks such as the UMTS Satellite Radio Access Network (USRAN).
- The development of Community Area Networks (CAN) that utilize the unlicensed radio frequency. A CAN is capable of reaching sparsely populated areas to provide services at very little cost.
- The growing interest in the emerging wireless MESH network concepts which combines different wireless solutions. MESH networks have gained noticeable success in special applications and are growing to more commercial and main stream applications.

- The evolution of many other peer-to-peer wireless networks like the vehicular ad-hoc networks.

In all the illustrated wireless deployments, the wireless networks maintain connectivity to the Internet via wireless access gateways. Wireless access gateways are a vital part in providing robust connection to the information and services available in the wire-line and the global network. The wireless access gateways deal with different QoS mechanisms on the wireless side and the wire-line side.

The differences in QoS mechanisms implemented on either sides of the wireless access gateway impose various challenges to its design. While the wireless access gateways operate in a heterogeneous environment, the endpoint applications expect homogeneous services regardless of their location. Therefore, the design of a QoS wireless access gateway is a central issue in the wireless network evolution. This research is motivated by our desire to solve the wireless access gateway QoS issues in order to provide homogeneous QoS communication over a heterogeneous service environment.

## 1.6 Cross-domain QoS

The proposed PYLON-Lite model follows a cross-domain (horizontal) view of the QoS problem. As illustrated in Figure 1-1, PYLON-Lite defines and operates on the essential components required to achieve consistency between QoS implementations in the ad-hoc, and in the fixed topology networks.

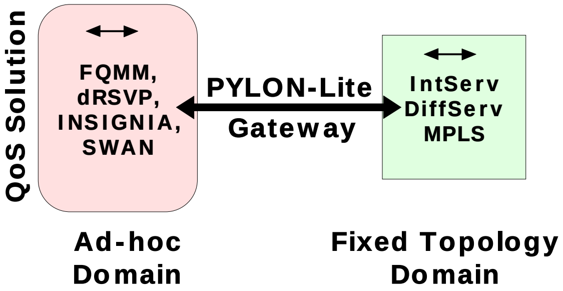


**Figure 1-1:** *PYLON-Lite Problem Domain*

In this sense, PYLON-Lite is very essential to facilitate end-to-end QoS if one communication endpoint is located in the ad-hoc network, and the other endpoint is in the fixed topology network. PYLON-Lite is a pioneering cross-domain QoS solution that is essential for cross-domain QoS support of real-time applications like video-conferencing, voice over IP, and streaming in ad-hoc environment.

## 1.7 Challenges to the Design of QoS Access Gateway

The design challenges for gateways attached to ad-hoc domains reflect the difficulties in performing QoS in the ad-hoc domain itself. The QoS solutions on the access network are likely to be per-class services in order to facilitate scalability. Therefore, another challenge to the gateway design is the mapping between different service granularities. Following is a list of major challenges to the design of gateways attached to ad-hoc domains:

1- The merging of different QoS model employed on both sides of the gateway. This leads to, for example, the cross mapping between per-flow and per-class services. Therefore, the design components of the gateway are essentially heterogeneous to reflect the asymmetric nature of the environment.

2- The lack of a policing authority in the ad-hoc domain leads to the need for extra policing on the gateway that may not be scalable. The compromise between policing and scalability is hard, and must rely ultimately on human intervention.

3- The difficulty in maintaining accounting and billing records for the mobile nodes due to the lack of a central authority. In addition, mobile nodes can not rely on the fixed topology network before building sufficient levels of trust.

4- The difficulty in obtaining effective service provisioning leaves the gateway with a hard optimization decision. Therefore, allocated network resources are expected to be under-utilized.

5- The MANET charter [25] imposes no limits on the ad-hoc network in terms of scalability. This open scalability hits the gateway as a potential service bottle neck.

Those challenges influence the design of PYLON-Lite and its components.

## 1.8 Research Contributions

Our research in the area of ad-hoc QoS and gateway design has resulted in the following contributions.

1- Established ESWAN as an advanced QoS model for ad-hoc networks. The model is described in Appendix C.

2- Established the PYLON model as the first model to address the QoS issues in the gateway to ad-hoc networks. Enhanced PYLON into PYLON-Lite as a lightweight model for QoS gateways to ad-hoc networks. The observed downfalls of full-scale PYLON influenced the design of PYLON-Lite to be reactive and lightweight.

   The experience gained from the PYLON design showed that service provisioning in ad-hoc networks represents a serious challenge to the per-domain service allocation process. PYLON-Lite employs reactive service allocation combined with service ladder policies to avoid service provisioning. The lightweight design of PYLON-Lite aspires to limit the implementation at mobile nodes in order to allow nodes to roam freely into other access gateways that may employ QoS models other than PYLON-Lite.

   The PYLON-Lite uses end-to-end aggregate reservation concept and adapts the use reactive collective aggregated services. In addition, PYLON-Lite introduces the use of limited service policing and Service Ladder Policy.

3- Provided recommendations for securing the PYLON-Lite gateway as illustrated in Appendix E.

## 1.9 Refereed Publications

The research performed here has also resulted in a number of publications that supported and enriched our view for the PYLON-Lite model. The following list represents our publications related to PYLON-Lite up to the current date.

1- Yasser L. Morgan and Thomas Kunz, "*A Proposal for an Ad-hoc Network QoS Gateway*", to appear in the Proceedings of the IEEE International Conference on

Wireless and Mobile Computing, Networking and Communications WiMob-05, Montreal Canada, August 2005.

This paper covers parts of the model design with less focuss on the evaluation and performance measures.

2- Yasser L. Morgan and Thomas Kunz, "*A Design Framework for Wireless MANET QoS Gateway*", to appear in the Proceedings of the 6th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp. 31-37, Towson USA, May 2005.

   This paper focuses on the evaluation and the behavioral characteristics of the model

3- Yasser L. Morgan and Thomas Kunz, *"Enhancing SWAN QoS Model By Adopting Destination-based Regulation (ESWAN)"*, in Proceedings of the 2nd WiOpt'04 Conference for Modeling and Optimization in Mobile Ad-hoc and Wireless Networks, pp. 112-121, Cambridge, UK, March 2004.

   This paper introduces a set of problems related to the SWAN dynamic regulation and presents solutions followed by thorough evaluation. Appendix C covers the ESWAN contribution in details.

4- Yasser L. Morgan and Thomas Kunz, *"PYLON: An Architectural Framework for Ad-hoc QoS Interconnectivity with Access Domains"*, in Proceedings of the 36th International Conference on System Sciences (HICSS-36), pp. 309-318, Hawaii USA, IEEE Computer Society Press 2003, ISBN 0-7695-1874-5, January 2003.

   This paper introduces the PYLON (full scale) model. Parts of the research in PYLON has been enhanced in PYLON-Lite as described in the body of this thesis.

5- Yasser L. Morgan and Thomas Kunz, *"An Architecture Framework for MANET QoS Interaction with Access Domains"*, in Proceedings of the 1st International Conference on Ad-hoc and Wireless Networks, pp 33-47, Toronto, Canada, September 2002.

   This paper illustrates an early model of the major directions that PYLON crystallized later. This paper represent more of a high level model design.

## 1.10 Organization of the Thesis

This chapter introduces the research motivation, the problem domain, and highlights the contribution. Chapter two provides an in-depth analysis of the most dominant QoS models in the fixed topology networks, namely, Integrated Services (IntServ) [25], and Differentiated Services (DiffServ) [9] and [43]. Then, it provides a brief review of the Multi-protocol Label Switching (MPLS) to show its use in engineering DiffServ domains. In order to illustrate the similarities between the proposed PYLON-Lite model and other cross-domain QoS approaches, Chapter two provides a section for discussing IntServ operations over DiffServ domains as a classical IETF solution. Chapter two also illustrates some of the most promising comprehensive independent QoS solutions developed specifically for the ad-hoc domains. It covers, namely, In-band Signaling (INSIGNIA) [62], and Stateless Wireless Ad-hoc Networks (SWAN) [2]. The chapter summarizes and criticizes both solutions, then concludes on their use and limitations.

Chapter three introduces the basic terminologies used in describing PYLON-Lite. The design principles of communication gateways in a cascaded service environment are investigated to address the common design principles in PYLON-Lite. Then, it illustrates the PYLON-Lite design fundamentals, and the special considerations of the model. Chapter three also defines PYLON-Lite basic components and architectural design at both mobile node, and gateway. It illustrates the use of aggregate reservation and details the problems related to traffic policing and model scalability. At the end, the chapter compares PYLON-Lite to the full-scale PYLON model [74], and concludes.

Chapter four introduces the PYLON-Lite test-bed, and investigates the major factors affecting the model performance. Then, Chapter four analyzes the PYLON-Lite behavioral trends in both upstream and downstream scenarios and the important delays related to PYLON-Lite. Chapter four is tightly related to Appendix D that extends the model results by investigating its interoperability with other models.

Chapter five concludes on the adopted design approaches and the results. Chapter five also investigates, and recommends future extensions to the model.

Appendix A reviews the QoS solutions in cellular and WLAN networks. Appendix B reviews the Asynchronous Transfer Mode (ATM) QoS solutions as essential QoS background for current IETF-based solutions. Appendix C introduces ESWAN briefly and highlights the enhancements it offers to the original SWAN model. Appendix D extends the analysis provided in Chapter four by exploring the PYLON-Lite performance when attached to ad-hoc QoS models other than SWAN such as INSIGNIA and ESWAN. Appendix E covers possible security loopholes in the PYLON-Lite design and recommends relevant solutions. Both Appendix D and Appendix E are considered an integral part of the PYLON-Lite model.

## CHAPTER 2

# QoS Models in Fixed and Mobile Domains

The need for differential QoS has been fueled by the continuous evolution in user applications. While some applications can tolerate limited bandwidth and network delays, others require high bandwidth and very limited delays. Different applications also generate different traffic patterns, which in turn impose different QoS parameters. The first part of this chapter analyzes the various types of user applications, categorizes, and then defines the effect of applications on QoS design.

The QoS research efforts started in wire-line networks and have been crystallized, within the IP networks, into two well-known models, namely, *Integrated Services* (IntServ) [14], [98], and *Differentiated Services* (DiffServ) [9], in addition to various related technologies like *Resource Reservation Protocol* (RSVP) [15], and *Multi-Protocol Label Switching* (MPLS) [96], [109]. In the second part, we review and comment on IntServ, DiffServ, and MPLS. Then, we review the IntServ over DiffServ operations, summarize and conclude on our observations.

In the third part of this chapter we focus on QoS models in ad-hoc domains since they deal with fairly different challenges. Unfortunately, early QoS models in the ad-hoc environment, like the *Dynamic QoS* for mobile ad-hoc networks (dQoS) [70], [71], and the *Flexible QoS Model for Mobile* ad-hoc networks (FQMM) [110], propose little more than a mere extension to the wire-line QoS models. Recently, researchers started to realize the peculiar nature of the ad-hoc networks, introducing the *In-band Signaling* (INSIGNIA) [62] approach, and the *Stateless Wireless Ad-hoc Networks* (SWAN) [3] proposal. While dQoS and FQMM suggest different variations to the wire-line inter-layer QoS models, INSIGNIA and SWAN propose a fresh design and inject new fundamentals to the QoS problem domain. The third part of this chapter provides a quick overview of the dQoS and FQMM models, and then elaborates on the design details of both INSIGNIA and SWAN. It comments on each of the four inter-layer QoS models, and focuses on the SWAN model and the enhanced SWAN extension (ESWAN) since it is considered the most promising and advanced model of the four proposals. Appendix C of this thesis extends the focus on ESWAN at a more detailed level. Finally, the chapter ends with thorough conclusive remarks.

## 2.1 Types of QoS Applications

QoS applications can be divided into different classes using different approaches. Applications may be classified into *Real-time* (RT) and *Elastic* applications. RT-applications need packets to arrive within certain time limits, and will disregard packets arriving past that time. Elastic applications can tolerate delays of packet arrival, and can afford to wait for packets.

### 2.1.1 Real-time Applications (Guaranteed & Predictive Services)

One important class of RT-applications is the class of *playback* applications where a source host issues a stream of packets and the packets travel over the network. The network introduces some delay variations to the packets. The destinations receive the packets and try faithfully to regenerate the original stream. In order for the destination to maintain a stable rate for the reproduced stream signal, it introduces a fixed delay. This delay allows the destination to buffer received data, and hides the delay variations from the application user.

The performance of playback applications is measured by latency and fidelity. Some playback applications that require duplex communication such as digital telephony (VoIP) require a sufficient level of interaction between both ends, and hence are more sensitive to latency. Others like streaming of a movie or lecture are not sensitive to latency. Similarly, it is possible to classify playback applications as tolerant or intolerant to loss of fidelity. Intolerant applications need to obtain from the network information about a guaranteed upper bound on the maximum delay of each packet.

Predictive services on the other hand are proposed for tolerant applications since they can cope with some delays. The delay bound is not computed based on maximum delay; instead, it is computed based on conservative predictions about the behavior of the flow. The network may violate the delay bound, and the application performance will certainly be impacted, however, the application users may be willing to accept the statistical possibility of lower application performance in return for lower cost. The deterministic guarantee bound delay and the statistical guarantee bound delay together help achieving higher levels of network utilization [14].

### 2.1.2 Elastic Applications (BE-Service)

Elastic applications are able to wait for data to arrive. This does not mean elastic applications are insensitive to traffic delays. To the contrary, significant delays to packets will often harm application performance. However, this category of applications does not buffer incoming traffic, instead, it uses the data immediately. Therefore elastic applications do not need a priori characterization of data, and the performance of the application depends more on the average packet delay. Examples of elastic applications are: interactive burst (Telnet, X, NFS), interactive bulk transfer (FTP), asynchronous bulk transfer (e-mail, fax), and Hyper-Text Transfer Protocol (HTTP). The delay requirements for these elastic applications vary from rather demanding (as for interactive burst applications), to rather tolerant (as for asynchronous bulk transfer applications), with interactive bulk transfer being somewhere in the middle [14].

Best Effort (BE) service can be used with elastic applications. Traffic of elastic applications does not require admission control; however, BE-service can provide low delays for interactive bursts. Some current or future applications might produce traffic with some characteristics that overlap different traffic classes; the application must decide which service model to follow in this case. Finally, the described classification is merely an attempt to catch reality, however, it is neither exact nor complete, but can be used as a guideline.

## 2.2 QoS Solutions for Fixed Topology Domains

QoS research in fixed topology networks evolved during the 1990s due to the increasing demand on bandwidth since the commercialization of the Internet, and due to the increasing traffic of real-time applications. During the early 1990s, the IntServ model [98] was developed, followed by the DiffServ model [9], [43]. The IETF has adopted both models and presented integration mechanisms such as IntServ over DiffServ domains [8] in order for models to complement each other, and to operate together.

This section provides a brief review of the QoS solutions on fixed topology domains with a focus on the IETF-based solutions. Other QoS solutions like QoS in WLAN networks, MAC QoS, cellular QoS and ATM solutions are considered beyond the main focus of this research, and therefore, are presented in Appendix A and Appendix B.

### 2.2.1 Integrated Service Model (IntServ)

The Integrated Service model is a QoS model that aims to achieve high levels of service guarantee based on end-to-end negotiation. IntServ provides RT-services and shares dynamically the available link capacity in a controlled manner. RT QoS is essential for applications such as interactive video conferencing. IntServ divvies up the available bandwidth (BW) into several classes with a certain minimum assured BW to each of the classes, which require controlled link sharing. Resource reservation is done typically via RSVP. The details of the RSVP operational framework with IntServ are provided in [107].

#### *2.2.1.1 The Use of RSVP with IntServ*

The Resource Reservation Protocol (RSVP) is an end-to-end resource reservation protocol that is commonly used by IntServ to control link sharing [15]. The host starts by initiating a reservation request. The receiving host responds by sending reservation-reply (RESV) messages specifying the class of service it can support. These reservation messages pass through all the network elements in the reverse path until the sending host receives them. RSVP attempts to make a resource reservation at each network element through which the application flow will pass.

Over the past few years, the IETF has moved to reduce the verbose nature of RSVP by introducing the use of tunneling, and by enhancing its granularity to support aggregated traffic flows. An aggregate reservation would carry packets from a large number of flows that belong to the same traffic class or otherwise require similar treatment.

#### *2.2.1.2 Comments on the IntServ Model*

IntServ puts a lot of demands on the routers. IntServ-enabled routers must store state information and this information increases in proportion to the number of flows. All routers must understand RSVP, must make more decisions on accepting flows, and must implement queues in order to classify and provide appropriate services to the flows.

The IntServ model is effective in dealing with RT-flows in fixed topology networks. As the number of IntServ flows grows, signaling and route maintenance of flows raises scalability concerns. To facilitate scalable implementation of IntServ, flows may be grouped in bigger granularities. Enforcing *IntServ over DiffServ Operations* may also decrease the scalability concerns on the core network. In this sense, IntServ is pushed more towards the edges of the Internet, and models like DiffServ are pushed more towards the core networks.

Ad-hoc networks cannot efficiently adopt IntServ. The limited capabilities of the mobile nodes make it very difficult to maintain a reasonable number of flows due to signaling overheads. In other words, IntServ is known to lack scalability, in addition, the idea of maintaining flow guarantees is hard to implement in a mobile and dynamic environment. As nodes move around, QoS flows lose essential resources, and the reconstruction of such resources carries a high cost. The dynamics of the ad-hoc network in terms of unpredictable mobility, bandwidth variation, delays, and packet dropping make it hard to efficiently adopt IntServ to ad-hoc networks.

### 2.2.2 Differentiated Service Model (DiffServ)

DiffServ is designed for scalable services based on per-domain behavior. Scalability is achieved by relying on per-class granularity, and by adopting a reservation-less approach. A predefined *Per-Hop-Behavior* (PHB) provides the vehicle for packet forwarding by facilitating a per-domain behavior. Sophisticated classification, marking, policing, and shaping operations need only to be implemented at network boundaries and hosts. Wide varieties of service types can be implemented based on these building blocks [9].

The DiffServ architecture relies on establishing and maintaining the following agreements:

- The level of service provided to a traffic aggregate.
- The conditioning functions and PHB used to realize services.

- The DiffServ code-point (DSCP) used to mark packets subject to a certain PHB.
- The particular node implementation mechanisms, which realizes a PHB.

### *2.2.2.1 Characteristics of the DiffServ Model*

Service provisioning and traffic conditioning policies are decoupled from the forwarding behaviors. DiffServ exhibits the following characteristics [9]:

- It allows decoupling of the service from the particular application in use since it removes the end-to-end resource negotiation.
- It decouples traffic conditioning and service provisioning functions from forwarding behaviors implemented within the core network nodes.
- It requires only a small set of forwarding behaviors whose implementation complexity does not dominate the cost of a network device, and which will not introduce bottlenecks for future high-speed system implementations.
- It permits simple packet classification implementations in core network nodes (Behavior Aggregate BA classifier).
- It does not depend on hop-by-hop application signaling.
- It accommodates incremental deployment.

The PHB is designed in a concrete yet flexible architecture. A simple PHB example can be to assign X% of a link (over some reasonable period of time) to a behavior aggregate. A PHB may be specified in terms of network resources (e.g., buffer, bandwidth), priority relative to other PHB, or in terms of network relative observable traffic characteristics (e.g., delay, loss).

A DSCP field may hold up to 64 different code-points. The 64 code-points are 32 standard code-points (-8 reserved), 16 locally defined, and 16 configurable code-points. Unmapped code-points map to a default PHB. The DSCP to PHB mapping can be a one-to-one mapping or an n-to-one mapping [9]. The code-points can follow a set of mandatory values as defined in [79], combined with a set of recommended values, or have purely local definitions. Simple administrative controls can be used to configure the interaction between traffic conditioners and interior nodes of the domain. This administration may require operational control through protocols and a control entity.

#### *2.2.2.2 Comments on DiffServ Model*

The DiffServ model provides scalable solutions for many QoS problems. Since DiffServ does not maintain state information, it provides scalability. Even though DiffServ does not provide deterministic guarantees for traffic, it can provide statistical expectations, using enough confidence. For instance, if a DS domain accepts traffic with a certain DSCP, the *Traffic Conditioning Agreement* (TCA) guarantees that the traffic will experience a given bandwidth *W*; delay *D*, losses *L*, and jitter *J* over a certain (long enough) period of time *T*. The TCA holds between domains, not between end users or traffic flows. Engineering a DiffServ domain is challenging and requires human intervention.

The DiffServ model cannot be used in ad-hoc networks without applying massive changes to its main framework. For one reason, DiffServ requires intensive network administration that violates the automatic configuration required by the MANET charter [25]. Node mobility may cause nodes to reroute traffic differently without reservation. Therefore, flows can either lose promised QoS or get dropped.

DiffServ provides a per-domain-behavior, which is essential for cross-domain traffic. In addition, it provides a scalable model, with reasonable guarantees and record keeping that reports actual network behavior. For these reasons, DiffServ suits the core network while IntServ fits the edge of the network. DiffServ commonly employs MPLS in order to facilitate traffic engineering and to provide tighter QoS guarantees. Therefore, a combination of DiffServ and IntServ is likely to co-exist, but in different parts of the network.

### 2.2.3 Multiprotocol Label Switching (MPLS)

Multiprotocol Label Switching (MPLS) is an IETF technology used for speeding up network traffic flow and simplifies its management. MPLS involves setting up a specific path for a given sequence of packets, identified by a label put in each packet, thus saving the time needed for a router to look up the address to the next hop. In addition to moving traffic faster, MPLS simplifies the QoS network management. MPLS plays a vital role in enabling QoS; however, QoS is not a fundamental feature of MPLS.

#### *2.2.3.1 The Use of MPLS with DiffServ*

MPLS is commonly used within DiffServ domains to engineer traffic delivery. MPLS can force packets into specific paths and can guarantee bandwidth for each service class. However, MPLS inherently cannot specify per-class differential treatment of flows. Combining the DiffServ-based classification and PHB with MPLS leads to true QoS in data network backbones. The mechanisms for MPLS support of DiffServ are described in [109].

In summary, MPLS support of DiffServ satisfies two necessary conditions for QoS: guaranteed bandwidth, and differential queue services. MPLS satisfies the first condition, i.e., it forces packet flows into the routers with guaranteed bandwidth; and along these routers, DiffServ satisfies the second condition by providing differential queue services.

#### *2.2.3.2 Comments on MPLS*

MPLS support of DiffServ is simpler and more scalable than IntServ with Standard RSVP. IntServ requires maintaining per-flow signaling and per-flow states in each router. In contrast, MPLS can combine flow aggregations of many flows and thus requires less signaling. Routers do not keep per-flow states; instead, they keep aggregated information on the bandwidth availability for all service classes or for each priority queue.

The IntServ architecture offers hard guarantees, but is not scalable or practical to operate and manage as a cross-domain QoS model. The DiffServ architecture has provided a scalable alternative but it offers merely statistical guarantees that may not be sufficient for commercial use. MPLS uses label fields in its headers to tag traffic flows and direct them into specific connection-oriented links, and thus provides stronger service guarantees. Several research proposals evolved in an effort to define traffic-engineering approaches for managing the use of MPLS over DiffServ domains. Oliveira [81] presents a tool to monitor and manage DiffServ-aware MPLS domains. Recent IETF work on combining the DiffServ and MPLS technologies in a packet network lead to enabling hard QoS assurances; and these guarantees come with better scalability and reduced complexity in comparison with IntServ. These improvements are a result of the stacking hierarchies and aggregations of MPLS networks as well as the aggregated states maintained by the DiffServ-supporting nodes.

Fewer research efforts have been directed to mapping IntServ into MPLS for per-domain services. The reason is the IntServ scalability problem in dealing with flows. However, [102] provides some guidelines for mapping IntServ to MPLS. The scheme is valuable and limits the mapping to the edge routers.

### 2.2.4 IntServ Operations over DiffServ Domains

The early research on QoS for IP networks has led to two distinct approaches: the IntServ architecture, and the DiffServ architecture. IntServ suffers from scalability issues since it maintains per-flow information in intermediate routers, but offers tight QoS guarantees, and is therefore suitable at the edge of the network. DiffServ offers scalable models and, when combined with MPLS, provides tight QoS guarantees, but can operate on a per-domain basis, and therefore is suitable at the core of the network. The IETF has proposed a framework to combine both models by employing each model where it performs better [8].

#### *2.2.4.1 The Use of Aggregate Resource Reservation Protocol ARSVP*

The IETF has extended the use of the *Resource Reservation Protocol* RSVP [15] into the *Aggregate Resource Reservation Protocol* ARSVP [6]. The ARSVP protocol introduces the IP Protocol Number `RSVP-IGNORE (134)` that flags intermediate routers to ignore the ARSVP message, and edge routers to intercept and perform per-domain admission control.

#### *2.2.4.2 Advantages of Aggregate Reservation*

Aggregate resource reservation removes the scalability concerns for the per-domain packet forwarding. Intermediate routers forward marked packets using the predefined policies associated with the PHB defined to serve a specific DSCP, and therefore, allow lighter operations on intermediate routers. Flows can benefit from the statistical variations in bandwidth usage, and hence, network resources are utilized in a better way.

#### *2.2.4.3 The Benefits of Using IntServ over DiffServ Domains*

The primary benefit of DiffServ aggregate traffic control is scalability. However, various other benefits can also be encountered.

- **Resource Based Admission Control:** IntServ uses an explicit setup mechanism to request resources from the network. The network may then accept or reject the requests. The explicit admission control is used by the edge routers and assures optimal use of network resources.
- **Policy Based Admission Control:** In DiffServ domains where RSVP is used, resource requests are intercepted by edge routers and are reviewed against the domain policies.
- **Source Marking:** In the case of source marking, the host operating system marks transmitted packets with desired DSCP. This approach has the benefit of shifting per-flow classification and marking to the source of the traffic, where it scales best.
- **Router Marking:** In the case of router marking, multi-flow classification criteria are configured in the edge routers. This may be done dynamically, per-request, or statically by manual configuration or by automated scripts.
- **Traffic Conditioning:** IntServ-capable routers are able to condition traffic at the per-flow granularity, via combination of shaping and policing. Pre-conditioned traffic enhances the ability of the DiffServ domains to provide quantitative services using aggregate traffic control.

In order for a DiffServ domain to provide services for IntServ connections, the domain has to be equipped with a valid policy that maps various IntServ *Flow Specifications* into domain specific PHBs.

#### *2.2.4.4 Comments on the IntServ Operations over DiffServ Domains*

The IntServ Operations over DiffServ Domains is a cross-domain QoS framework led by the IETF. The framework provides different mechanisms to utilize both IntServ and DiffServ models. In this framework, it is possible to benefit from the IntServ explicit resource reservations, which accurately defines the per-flow requirements, while providing a scalable implementation by relying on the DiffServ class aggregation in the core network.

### 2.2.5 Comments on QoS Models for Fixed Topology Domains

This section introduces the IntServ model followed by comments on its usage for per-flow services; it illustrates the DiffServ model, and comments on the DiffServ usage to provide per-domain QoS. In addition, it illustrates the use of MPLS to reinforce DiffServ in providing tighter QoS guarantees. Both IntServ and DiffServ models are found to be inappropriate for the ad-hoc environment. IntServ focuses on per-flow services, by simulating a dedicated end-to-end connection. In doing that, IntServ implements mechanisms to allocate resources along the route of a specific flow. DiffServ on the other hand focuses on per-domain services by introducing the concept of QoS policy and statistical guarantees.

The section also investigates the IntServ operations over DiffServ domains. This framework illustrates typical cross-domain QoS mechanisms. For instance, the use of ARSVP is commonly adopted for per-domain resource reservation. PYLON-Lite is inspired by the use of default mappings and ARSVP as discussed later in Chapter 3.

## 2.3 QoS Solutions for Mobile Ad-hoc Domains

Researchers in the area of QoS solutions for ad-hoc networks are faced with several fundamental challenges due to the peculiar nature of the ad-hoc networks. These challenges can be summarized as follows.

1- The lack of processing power and storage that can be employed to monitor, track, and store information about specific flows, or node mobility. (Lightweight)

2- The limited capabilities of mobile nodes' batteries, which leads to a tight use of resources to reserve power. (Lightweight)

3- The lack of central authority and human intervention that is needed to manage, to monitor, and to engineer the network resources. (Distributed)

4- The dynamic mobile nature, which leads to possible network partitioning or merger, and make it difficult to maintain resources over a period of time. (Robust, Self-recovering)

Those inherent limitations of the environment fuel the need for distributed, lightweight, robust, and self-recovering approaches. This section reviews QoS solutions that provide sound solutions to the defined challenges. Currently the SWAN model is considered the most promising and is enhanced into the ESWAN model. ESWAN is discussed in greater detail in Appendix C.

### 2.3.1 Dynamic Quality of Service (dQoS)

The Dynamic QoS model focuses on the problem of bandwidth variations in a wireless mobile environment. The dQoS solution extends the classical RSVP to the dynamic RSVP (dRSVP) in order to handle percentile resource reservation. When the bandwidth of a specific wireless link experiences some variations, dQoS attempts to follow a fair approach. Maintaining the same sharing percentage among different flows as the bandwidth varies enforces fairness in dQoS terms [70] and [71].

The dQoS model maintains flow-specific information at intermediate nodes, and therefore suffers from scalability problem. In addition, the dQoS fairness tenet is questionable.

### 2.3.2 Flexible QoS Model for Mobile Ad-hoc Networks (FQMM)

FQMM follows a hybrid approach combining the per-class granularity of DiffServ with the per-flow granularity of IntServ. FQMM adopts DiffServ, but improves the per-class granularity to per-flow granularity for certain classes of traffic. Since the traffic load within ad-hoc networks is limited compared to the backbone traffic, FQMM claims that per-flow treatment of a small number of individual flows cannot harm the overall performance [110].

FQMM falls short of addressing the full scalability demands; instead, it reaches a compromise solution for small to medium size ad-hoc networks (roughly less than 50 nodes). FQMM is built over IntServ, and DiffServ models, hence can operate directly with extranet traffic. FQMM scalability issues limit its application to specific installations and restrict its commercial use.

### 2.3.3 In-band Signaling Model (INSIGNIA)

The INSIGNIA QoS Framework [62] is designed to support the delivery of adaptive services in mobile ad-hoc networks. A key component of the INSIGNIA QoS framework is the in-band signaling system that supports fast reservation, restoration, and adaptation algorithms. In-band signaling is lightweight and highly responsive to changes in network topology, node connectivity, and end-to-end (E2E) QoS conditions. The basic design tenet of INSIGNIA is to create an independent state-based QoS framework in order to operate freely over various MAC layers, and routing protocols.

*2.3.3.1 INSIGNIA Framework*

ISNIGNIA is designed to deal with the dynamics of node mobility by providing a distributed adaptive system with localized adaptation and restoration. INSIGNIA provides soft QoS guarantees as defined in Section 1.1, and uses adaptive services that require the user application to adopt to different levels of services, while it responds to topology changes by providing different levels of services.

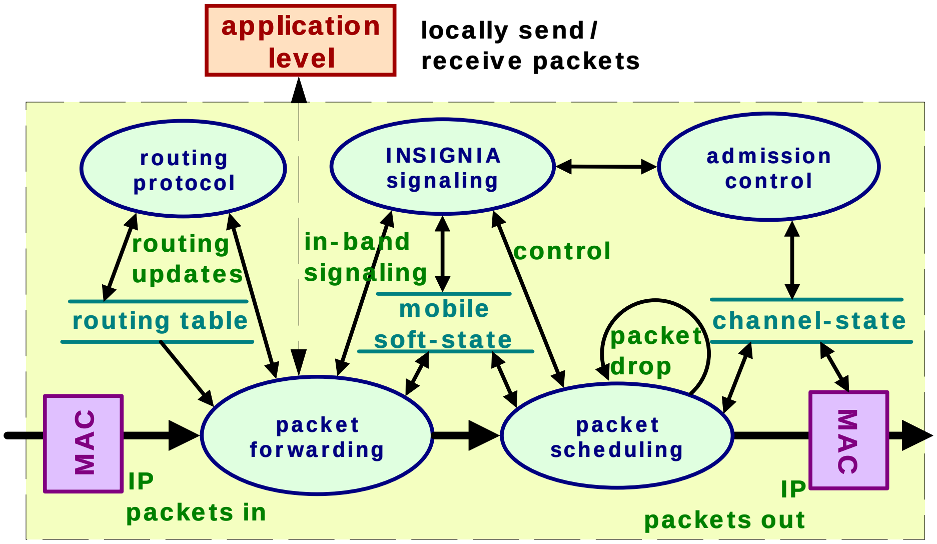


**Figure 2-1:** *The INSIGNIA QoS Framework*

INSIGNIA [62] is built on the architectural components illustrated in Figure 2-1. The INSIGNIA role is to establish, restore, adapt, and tear down adaptive services between source and destination pairs. Based on an approach that explicitly carries control information in the IP packet header, flow sessions can be rapidly established, restored, adapted, and released in response to wireless impairments and topology changes as in Figure 2-1.

The protocol operates over five basic mechanisms: *Fast Reservation* is initiated by a source node to request resource reservation. The destination node initiates *QoS reporting* to report changes in the status of current reservations. *Soft-state Resource Management* is responsible for responding to changes in network topology. The *Restoration* mechanism (in all nodes) is responsible for recovering from possible degradation in services. The *Adaptation* mechanism in the destination node responds to changes made at intermediate nodes [62].

#### *2.3.3.2 Comments on INSIGNIA Framework*

INSIGNIA is a promising QoS approach proposed for ad-hoc networks, however, is not adopted by any standardization community. The INSIGNIA model is an efficient design for lightweight in-band signaling, which is highly responsive to network dynamics. The model is distributed, but if the distance (in terms of number of hops) is high, the mobility scenarios will increase the chances of traffic degradation. Another problem with INSIGNIA resides in its need to synchronize the state of all mobile nodes involved in each flow. The following comments represent serious challenges to INSIGNIA:

1- **Scalability:** All nodes along the route to the destination have to maintain per-flow status. Maintaining per-flow status imposes scalability limitations.

2- **Traffic policing and security challenges:** INSIGNIA considers all security concerns outside the scope of the model.

3- **Discovering optimal QoS route:** INSIGNIA maintains a clear separation from routing algorithms. As a result, finding a route to the destination is a process performed completely by the routing algorithm. Routing algorithms usually use non-QoS search criteria like number of hops for route selection.

4- **Route maintenance:** The INSIGNIA model maintains the route initiated when the session started. Unless the flow experiences service degradation, INSIGNIA has no mechanism to detect whether better routes have become available.

5- **QoS reporting issues:** The INSIGNIA model relies on sending QoS report messages to a source node. This requires finding a reverse path, which is not always possible in wireless environment.

6- **QoS parameters:** INSIGNIA employs bandwidth as the only QoS parameter. The model does not provide any mechanism for considering delay, jitter, or any other QoS parameter.

7- **State-machine based model:** INSIGNIA is a state-machine based model. Updates to nodal states are essential for nodes to switch to their next state. Losing state update messages is quite possible in the ad-hoc wireless environment.

8- **INSIGNIA interaction with fixed QoS models:** INSIGNIA is not equipped with any mechanism to handle QoS traffic exchange with nodes outside the ad-hoc domain. PYLON and PYLON-Lite are the most promising proposals that fulfill cross-domain requirements so far.

The limitations observed in the INSIGNIA model are somewhat reasonable. INSIGNIA is a pioneering scientific research project that remains to be enhanced, and evolve towards a more mature model. The Columbia University team that worked on INSIGNIA has observed many of those limitations. For instance, in the new Columbia University project SWAN [2] the team selected a stateless model to avoid problems observed with the INSIGNIA state machine. SWAN also proposes a scalable model that does not maintain per-flow information at intermediate nodes.

### 2.3.4 Stateless Wireless Ad-hoc Networks (SWAN)

The Stateless Wireless Ad-hoc Network (SWAN) model uses a best-effort MAC layer and employs feedback-based mechanisms to support soft real-time services and service differentiation in mobile ad-hoc networks. SWAN uses rate control for UDP and TCP best-effort traffic and uses source-based admission control for UDP real-time traffic. In addition, SWAN uses the Additive Increase Multiplicative Decrease algorithm to control the rate of a flow. SWAN also uses Explicit Congestion Notification (ECN) [83] to dynamically regulate admitted real-time traffic in response to network dynamics such as mobility or traffic overload. SWAN does not require intermediate nodes to keep per-flow state information. As a result, there is no need for signaling or complex control mechanisms to update, refresh or remove per-flow state information, as is the case with state-based models like INSIGNIA [62].

Changes to topology, network conditions, and even node and link failure do not affect the operation of the SWAN control system. SWAN uses feedback information from the network instead of relying on state information.

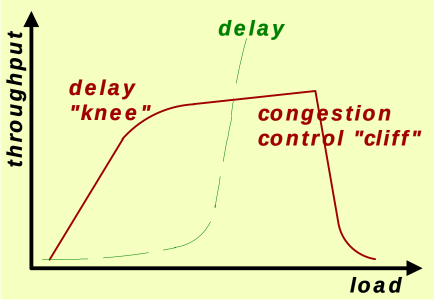


**Figure 2-2:** *General Behavior of AIMD Congestion Controlled System*

SWAN uses AIMD rate control to improve the performance of real-time UDP traffic. TCP attempts to avoid network congestion collapse by using packet loss as feedback. SWAN controls the rate of TCP traffic more conservatively to avoid excessive delays of real-time UDP traffic by using local per-hop packet delays as a feedback to local rate controllers. SWAN uses per-hop MAC delay as a feedback for local control instead of packet loss. The reason for doing this is that loss typically happens well after delays start to increase. The TCP congestion control algorithm operates closer to the throughput cliff as in Figure 2-2 in order to ensure maximum system throughout. SWAN's AIMD control algorithm keeps the system at the delay ″knee″ where the system output is almost as high as at the cliff, but buffers are significantly less loaded, so the delay is closer to minimum. Hence, SWAN follows a conservative approach to avoid excessive queues as shown in Figure 2-2 [3].

### *2.3.4.1 The SWAN Basic Components*

The SWAN model includes a number of building blocks to support rate regulation of BE-traffic, as illustrated in Figure 2-3. A Classifier and a Shaper operate between the IP and BE-MAC layers.

The *Shaper* processes BE-packets only, and represents a simple leaky bucket traffic shaper. The goal of the Shaper is to delay BE-packets in conformance with the rate calculated by the *Rate Controller*. When a session is admitted there is no admission control decision taken at intermediate nodes. Rather, the source node *Admission Controller* tests to determine if a new RT-session should be admitted based on the result of an E2E request-response probe. A key role of the *Admission Controller* is to efficiently estimate local bandwidth availability by analyzing information from the MAC layer.

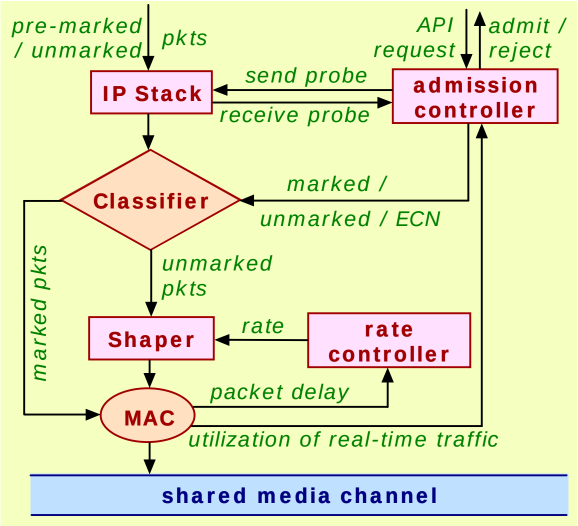


**Figure 2-3:** *The SWAN Model Framework*

*2.3.4.2 Comments on the SWAN Model*

The SWAN model distinguishes itself from other ad-hoc inter-layer QoS models in its pragmatically conservative resource utilization and its stateless design to handle QoS requirements. That pragmatic approach puts SWAN in a vulnerable position to criticism. The key point in evaluating SWAN is to maintain the operational and practical needs of ad-hoc networks.

The SWAN model's biggest advantage is its stateless mechanism, which does not require synchronization of nodal state. In addition, it follows a pragmatic conservative admission control approach in order to leave enough slack of network resources to carry best-effort traffic, to compensate for bursty real-time traffic, and to allow fast recovery from different mobility scenarios.

1- **Best-effort to Real-time Load Balance:** The advantages SWAN offers come at the expense of low resource utilization, especially when best-effort traffic is significantly low. The model provides soft guarantees as defined in Section 1.1 in response to bandwidth variations, node mobility, and false admission [75]. The SWAN model relies on BE-traffic to fill up the gap between the conservative bandwidth usage adopted by Admission Control, and the actual bandwidth available for traffic. In situations where BE-traffic forms a small percentage of transmitted traffic, SWAN presents a less than optimal use of network resources. In the same sense, in situations where BE-traffic forms a significantly high percentage of transmitted traffic, SWAN unnecessarily over-regulates BE-traffic, and therefore, SWAN presents a suboptimal use of network resources.

2- **Dynamic Regulation Issues:** The SWAN model describes the possibility of congestion due to node mobility or false admission [75]. SWAN also presents the dynamic regulation of real-time flows, and introduces two solutions, namely source and network-based regulation algorithms, aiming to provide full congestion recovery. ESWAN extends the dynamic regulation by offering a destination-based regulation combined with preemptive behavior. ESWAN is shown to limit the influence of dynamic regulation issues on SWAN performance [75].

3- **Discovering and Maintaining QoS Routes:** SWAN provides no mechanism for finding optimal QoS routes; instead, it relies on the underlying routing algorithm in this regard. Furthermore, changes to a selected route are controlled by the routing algorithm.

SWAN has some problems related to its design fundamentals. For instance, SWAN operates with implicit assumption of fidelity in all mobile ad-hoc nodes. Therefore, SWAN has no traffic policing and is vulnerable to many security attacks. Other problems in SWAN like weak QoS reporting can be enhanced, and ESWAN [75] provides some insights into this issue. SWAN views QoS in terms of single parameter, namely bandwidth, for operational reasons. Like all other ad-hoc domain QoS initiatives, the problem of extending QoS support across the access domain is not discussed nor provisioned by the model.

However, the SWAN model offers faster resource reservation than INSIGNIA when tested using the network simulator NS-2 [123]. The SWAN model is not viewed as a generic solution, rather, a pragmatic one that fits specific operational scenarios, mainly when there is almost equivalent amount of RT and BE-traffic expected in the ad-hoc network.

The stateless approach pioneered by SWAN offers a lot to QoS model robustness, and scalability. In this research, we used SWAN as a test model to evaluate the proposed PYLON-Lite model.

### 2.3.5 The Enhanced SWAN Extension (ESWAN)

SWAN is found vulnerable to problems related to mobility and false admission. The original SWAN model discusses the two problems as part of a dynamic regulation of real-time flows, and introduces two solutions, namely source and network-based regulation algorithms, aiming to provide full congestion recovery. Both SWAN solutions provide random or almost random selection of victim flows, and therefore add little value to the model. In addition, SWAN lacks a mechanism to deal with expired RT-packets, which can consume network resources.

ESWAN introduces a new mechanism to enhance the congestion recovery of real-time flows using a destination-based regulation, which applies a biased rule to select victim flows. In addition, ESWAN employs preemptive behavior to decrease the frequent occurrence of expired RT-packets, and to trigger QoS route maintenance [75]. A thorough description of the ESWAN model is provided in Appendix C.

### 2.3.6 Comments on QoS Models for Ad-hoc Domains

This section provides an overview of both dQoS and FQMM as inter-layer QoS alternatives for ad-hoc networks. Then the section presents the two major research QoS models for ad-hoc networks, namely INSIGNIA and SWAN. INSIGNIA offers an attractive and easy to implement solution, but lacks scalability, and has no mechanisms for finding optimal QoS routes, or switching to newer better routes when they become available. Therefore, INSIGNIA may not be widely adopted.

In an effort to remove scalability concerns, SWAN adopts a stateless solution where the per-flow information is not maintained at intermediate nodes. ESWAN enhances SWAN dynamic regulation, and adds a preemptive behavior to trigger route maintenance. However, SWAN presumes operation in a network that is, more or less, equally loaded with RT and BE-traffic; otherwise, SWAN provides a suboptimal solution. Like INSIGNIA, SWAN has no mechanisms for finding optimal QoS routes, or switching to newer better routes.

## 2.4 Conclusion

This chapter provides a brief survey of QoS solutions in fixed topology wire-line domains and mobile ad-hoc environment. The QoS solutions on fixed topology domains are inappropriate for the mobile ad-hoc environment. Other QoS solutions like QoS solutions on WLAN, cellular networks, MAC QoS, and ATM all provide valuable ideas, but are considered outside our main research focus. An extended review of those technologies is provided in Appendix A and Appendix B.

As the QoS research evolves, more solutions will become available. The IEEE (802.11e / 802.11g) guidelines [28] provide a per-link scheduling at the MAC layer. The growing QoS-aware routing approaches [63], [77] and [86] provide solutions for finding optimal QoS routes. We believe that a sound QoS approach in ad-hoc networks should provide useful integration with both routing and MAC layers, while maintaining layer separation. This vertical view of the QoS problem is essential to solving compound problems like finding and maintaining optimal QoS routes. The design of a cross-domain QoS model for ad-hoc domains has to consider the evolution of QoS models on the ad-hoc side as described in this section. This can be done by offering generic solutions that accommodate various ad-hoc QoS implementations.

Due to the differences between QoS solutions provided on both sides of the gateway, i.e. the fixed topology domains and mobile ad-hoc domain, the design of a generic QoS gateway is quite hard. For instance, the QoS gateway has to compensate for the absence of service provisioning in the ad-hoc domain. The difficulties in designing a QoS gateway are illustrated in the next chapter along with the PYLON-Lite solutions.

CHAPTER 3

# The PYLON-Lite QoS Model

PYLON-Lite is a cross-domain QoS model focused on solving the QoS problems experienced between access network QoS models on one side, and ad-hoc QoS models on the other. PYLON-Lite expects comprehensive support from the fixed topology access network. A full-scale PYLON QoS model was initially presented in [73] and detailed in [74]. This chapter focuses on a scaled-down version that demonstrates a practical approach, and is presented as PYLON-Lite. PYLON-Lite is a lightweight model that implements minimum necessary functionalities to facilitate interaction between QoS models in the ad-hoc and access networks.

In this chapter, Section 3.1 presents the basic definitions and terminologies required to understand the PYLON-Lite model design. Section 3.2 reviews common gateway design principles. Section 3.3 lists the major environmental limitations affecting the PYLON-Lite design. Section 3.4 provides the design fundamentals of the PYLON-Lite model. Section 3.5 illustrates some of the special design considerations. Then, Section 3.6 presents the PYLON-Lite design architecture. Section 3.7 covers the use of aggregate resource reservation (ARSVP for short) in the model. Section 3.8 covers the traffic policing issues in PYLON-Lite, and Section 3.9 covers the PYLON-Lite gateway scalability and complexity issues. Section 3.10 illustrates the full-scale PYLON model and major differences from the presented PYLON-Lite model. Section 3.11 summarizes the model design and operations.

## 3.1 Basic Definitions and Model Terminologies

This section describes some basic definitions in addition to identifying the PYLON-Lite model terminologies.

**Ad-hoc Intranet Traffic:** It is the traffic that is initiated on a node in the wireless ad-hoc network, and targets a node that resides in the same ad-hoc network. The data packets travel only through the same ad-hoc wireless network.

The QoS of ad-hoc Intranet traffic represents the kind of QoS that is studied in INSIGNIA [62], SWAN [2] [3], FQMM [110], and dQoS [71] models as described in Chapter 2 and illustrated in Figure 3-1.

**Ad-hoc Extranet Traffic:** It is the traffic traveling through access networks as well as wireless ad-hoc networks as illustrated in Figure 3-1. The challenge of this type of traffic is that it has to deal with different QoS models running in different types of networks, yet provide a consistent view of the QoS. Broadcast traffic that requires QoS (like in voice conferencing) is considered as ad-hoc extranet traffic if at least part of the traffic travels through an ad-hoc network and through access networks. In terms of direction, extranet traffic can be divided into two categories.

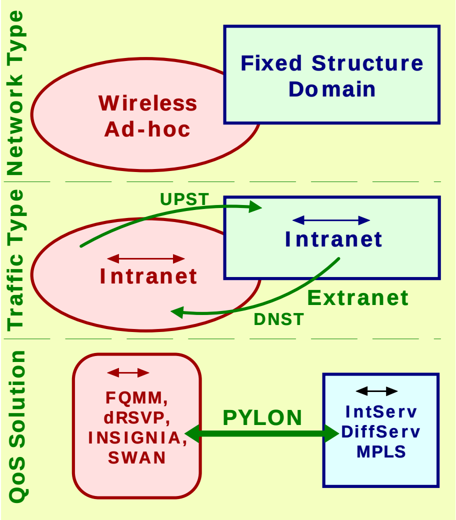


**Figure 3-1:** *The Relation between Network Type, Traffic Type, and QoS Solutions*

1- **Upstream Extranet Traffic (→UPST)** which is the extranet traffic initiated by a source node located in the ad-hoc domain.

2- **Downstream Extranet Traffic (←DNST)** which is the extranet traffic targeting a destination node located in the ad-hoc domain. In this sense, the terms upstream and downstream are relative to the ad-hoc domain in reference to extranet traffic direction as illustrated in the middle part of Figure 3-1.

The ad-hoc extranet traffic requires some cooperation between the ad-hoc and the access network in order to maintain QoS requirements. Access networks are, typically, fixed topology networks that can provide ad-hoc networks with accessibility to the Internet. The access (hosting) network may be willing to provide many services on behalf of the ad-hoc network. The access domains can be classified as follows.

**Un-friendly Access Domain:** A domain that is attached to an ad-hoc domain, but unwilling to provide any services or resources to the ad-hoc domain. The reason for denying services can be lack of trust, agreement, or resources. However, basic signaling can be performed between the ad-hoc domain and the access domain in order to exchange basic information like domain name and the reason for service denial.

**Friendly Access Domain:** A domain that is attached to an ad-hoc domain, and is willing to host some of the administrative process on behalf of the ad-hoc domain. This friendship is based on differentiated levels of mutual trust, and is expressed in the friendly domains' ability to host extranet traffic or configuring QoS parameters. Friendly access domains may demonstrate different levels of friendship as follows:

> **Public (Basic) Domain:** A domain that is willing to support any ad-hoc domain and is willing to provide basic services (namely, best effort services with no service differentiation). An example of public (basic) domain is a public library or shopping malls, where access gateways can connect any ad-hoc network to the access domain, and provide a basic set of services. A low level of trust can exist between the two domains, yet some services are provided.
>
> Public (basic) best effort access domains do not support any QoS requirements, but may require authentication. This does not necessarily mean there is no cost associated with providing services, rather, it means there is no cost or service differentiation. A Public (Basic) Friendly Domain does not provide any QoS challenges that need to be resolved, simply, the ad-hoc domain forwards packets to the Basic Friendly Domain, which in turn forwards packets to the Internet using merely best-effort services. Neither QoS guarantees are provided, nor are costs differentiations required for specific services. Instead, the public domain provides one level of service for all users, in return for a single cost (which can be null).
>
> **Private Friendly Domain:** A domain that is willing to provide different levels of services to a clinging ad-hoc domain. An example can be a corporate network that is willing to host an ad-hoc domain formed in a meeting room, or a hotel that is willing

to host an ad-hoc meeting or a conference. A private friendly domain provides various services, and illustrates different levels of trust. This trust is achieved by exchanging security parameters, and services may be associated with differentiated costs. Services can be guaranteed for a specified period of time, and the ad-hoc domain has to renew each service request before it expires. In addition, an ad-hoc domain may require a promotion or demotion of services. The private friendly domain is committed to provide the same level of services during the lifetime of the request placed by the ad-hoc domain.

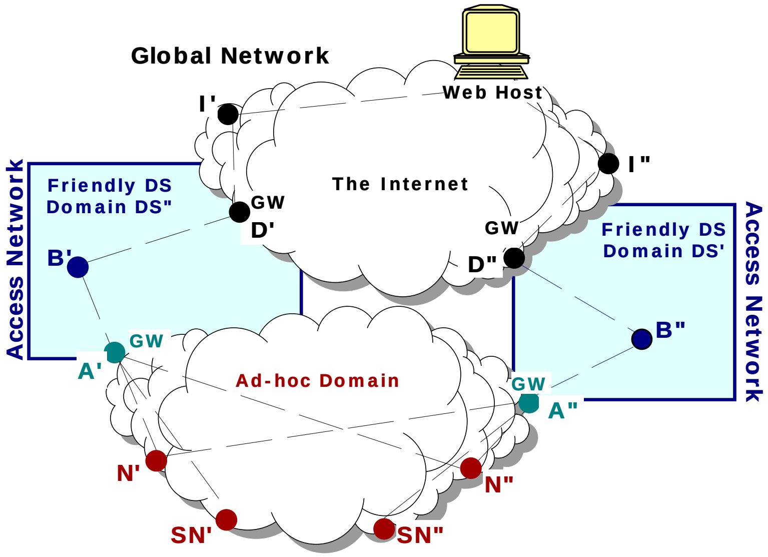


**Figure 3-2:** *Typical Ad-hoc Network Connected to the Internet*

Note that the illustrated classification does not mean a private friendly domain cannot provide best effort services. On the contrary, a great percentage of the extranet ad-hoc traffic can be satisfied by merely a best-effort service, and private friendly domains typically provide a best-effort service. Even if in a specific situation a private friendly domain provides only best-effort services, the distinction between a basic and a private friendly domain remains in its ability to support QoS requirements, while Public Friendly Domains do not have any QoS differentiation mechanism.

Unfriendly Domains cannot deal with ad-hoc wireless networks due to lack of trust, agreements, or resources. As soon as the condition changes and the Unfriendly Domain becomes willing to support the ad-hoc domain, the domain automatically falls into one of the Friendly Domain categories.

**Gateways (GW):** Access domains connect to ad-hoc domains usually through IEEE 802.11, Bluetooth, or cellular gateways as in Figure 3-2. A GW is any node that connects to access and ad-hoc domains simultaneously. In this research the terms access points, hot spot, base station, and gateways are used alternatively and with no technical differences.

**Sponsor Node (SN):** Sponsor Nodes are nodes on the ad-hoc domain side that have the authority and willingness to allocate resources with a friendly access domain, and sponsor the use of those resources by other ad-hoc nodes and users as illustrated in Figure 3-2.

Since service differentiation requires essentially marking packets with the desired DSCP, the following two points discuss the mechanisms for selecting the DSCP set.

**Dominant DSCP Set:** If an ad-hoc domain has established a private relationship with an access domain, it can adopt the private domains' set of DSCP codes. Every mobile node in the ad-hoc domain becomes aware of the adopted DSCP set, and uses it to mark its extranet traffic. This adopted set of DSCP values is called the dominant DSCP set. When the ad-hoc domain gets attached to another access domain, which has a different DSCP set, the ad-hoc domain does not switch to the new DSCP, only the boundary nodes (GW) have to perform the DSCP mapping function.

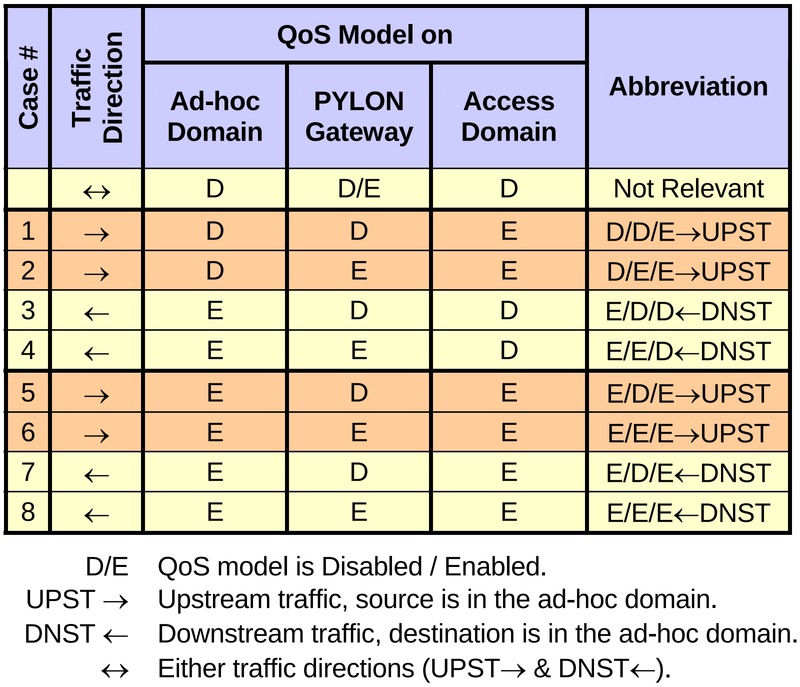
However, if the ad-hoc domain gets completely disconnected from any access network, the dominant DSCP set looses its relevance. At some later time, the ad-hoc domain may re-adopt a DSCP set of the first access domain it finds, and consider it the new dominant DSCP set.

**Native DSCP Set (NDSCP):** An ad-hoc domain may use a predefined DSCP set that is called a Native DSCP set. The NDSCP set is a dominant DSCP set that can never change. Boundary nodes of the ad-hoc domain perform the NDSCP mapping function based on predefined Traffic Conditioning Agreement (TCA) with the specific SN [80].

The use of an NDSCP set is highly recommended. An NDSCP set is expected to be highly detailed. When the ad-hoc domain clings to a private access domain that adopts a less detailed (smaller) set of DSCP codes, it is rather easy to do the mapping by combining slightly similar NDSCP codes into one access domain DSCP code, and use the same PHB. The reverse is quite challenging. The PYLON-Lite model adopts the use of an NDSCP set.

**PYLON-Lite Model Cases:** In order to illustrate the operations of the PYLON-Lite model, it is important to show the possible cases where the model may operate.

Table 3-1 illustrates the possibilities of having a QoS model running in either the ad-hoc or the access domains, and then shows the possibilities of running PYLON-Lite as well. The first possibility listed on Table 3-1 is simple; no QoS model is implemented in the ad-hoc or access domains, and therefore, there is no point in implementing PYLON-Lite.



| Case # | Traffic Direction | QoS Model on | | | Abbreviation |
|---|---|---|---|---|---|
| | | Ad-hoc Domain | PYLON Gateway | Access Domain | |
| | ↔ | D | D/E | D | Not Relevant |
| 1 | → | D | D | E | D/D/E→UPST |
| 2 | → | D | E | E | D/E/E→UPST |
| 3 | ← | E | D | D | E/D/D←DNST |
| 4 | ← | E | E | D | E/E/D←DNST |
| 5 | → | E | D | E | E/D/E→UPST |
| 6 | → | E | E | E | E/E/E→UPST |
| 7 | ← | E | D | E | E/D/E←DNST |
| 8 | ← | E | E | E | E/E/E←DNST |

D/E QoS model is Disabled / Enabled.
UPST → Upstream traffic, source is in the ad-hoc domain.
DNST ← Downstream traffic, destination is in the ad-hoc domain.
↔ Either traffic directions (UPST→ & DNST←).

**Table 3-1:** *PYLON-Lite Possible Model Cases*

Model cases 1 and 2 show the situation when the ad-hoc domain does not implement any QoS model, but the access domain does. In such a case, only upstream extranet traffic can be regulated by the use of PYLON-Lite. Those two cases are abbreviated as D/D/E→UPST and D/E/E→UPST, where PYLON-Lite may be disabled, or enabled. Model cases 3 and 4 illustrate the inverse of cases 1 and 2 as shown in Table 3-1.

If both the ad-hoc and the access domain implement QoS, then both traffic directions become of interest. When including the possibility of PYLON-Lite being disabled or enabled, the cases 5, 6, 7 and 8 become clear as illustrated on Table 3-1. Obviously, comparing cases 1 and 2 shows the effect of enabling PYLON-Lite, and so does the comparison of pairs 3 and 4, 5 and 6, and 7 and 8. However, the most challenging cases are 5 and 6, and 7 and 8, while cases 3 and 4 are commercially common.

## 3.2 Common Gateway Design Principles

Most of the proposed PYLON-Lite implementation takes place at the ad-hoc network gateway. The principles of gateway design have been discussed and established more than a decade ago. Research initiatives started by focusing on the problem of interconnectivity between inconsistent networks and provided different mechanisms for protocol conversion, as in [11], [40], and [42]. By the end of the 1980s the major principals of gateway design were established.

### 3.2.1 Properties of Cascaded Service Networks

To review the End-to-End (E2E) services, it is important to explore the situation of cascaded networks as illustrated in Figure 3-3. PYLON-Lite deals with the problem of cascaded services which is similar to the cascaded networks problem.

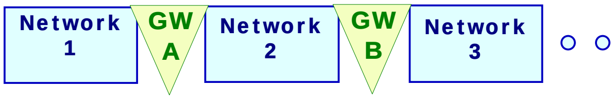


**Figure 3-3:** *View of Cascaded Service Networks with Connecting Gateways*

Gien [40] studied cascaded networks and defined two important properties. First is the cumulative concatenation property which describes the cumulative addition of some service properties like per-domain packet delay to calculate the E2E delay. Second is the concept of least common property that describes limiting some of the service properties to the least offered. For example the E2E service guarantees follow the concept of the least common property. Thus the E2E guarantee is the least granted service provided by all cascaded service networks. Since the ad-hoc domain merely provides a soft guarantee at best, any E2E service is either soft guaranteed or less.

### 3.2.2 QoS Measures in Cascaded Service Networks

In this study, the concept of QoS is discussed as a cross-domain concept. This means services are viewed on an E2E-basis across multiple domains (i.e. cascaded service networks). From the E2E viewpoint, service quality can be measured in terms of three comprehensive parameters, namely E2E bandwidth, E2E delay, and E2E jitter. Those parameters can be used to evaluate PYLON-Lite performance. However, some QoS domain models, like SWAN, may use only bandwidth to drive its operations.

Suppose the E2E communication service is obtained by the cascading of *n* service networks. Then, the E2E delay follows the concept of cumulative property concatenation, and can be described as in Equation 3-1.

$$E2E\ Delay = \sum_{i=1}^{n} (domain_i\ \ delay) \qquad \ldots (3\text{-}1)$$

In contrast the E2E bandwidth follows the concept of the least common service provided by the *n* networks. Thus, the minimum bandwidth can be formulated as in Equation 3-2.

$$E2E\ Bandwidth = \min\{domain_i\ \ bandwidth\}_{i=1}^{n} \qquad \ldots (3\text{-}2)$$

Now, assume that the per-domain jitter is defined as the average deviation of the delay curve from its average delay. Then, Equation 3-3 can be easily derived.

$$E2E\ Jitter = \sum_{i=1}^{n} (domain_i\ \ jitter) \qquad \ldots (3\text{-}3)$$

These simple formulas can be used to analyze QoS performance on an E2E-basis of any linearly cascaded service networks as the case with PYLON-Lite [10].

### 3.2.3 PYLON-Lite Common Service Subset

Most common gateway designs accommodate a service subset that fits our view for the PYLON-Lite QoS model. A good elaborate view of the generic gateway design which facilitates a Common Service Subset is provided in [12]. The design principles of communication gateways as described in [12] are used throughout this chapter in a top-down approach to formally define the PYLON-Lite model solutions.

*"The inter-connection of two service networks must be based on a common communication service provided in both networks"* [40]. As illustrated in Figure 3-4, assume that the ad-

hoc domain is represented by the oval and the access domain is represented by the rectangle. If *service-x* and *service-y* follow different service models, then the gateway will have to connect both services through *service-z* as a common service layer. For efficiency reasons, *service-z* should be implemented as low as possible. However, if the common service does not exist at a lower layer, *service-z* may ultimately exist at the application layer. In the case where *service-z* does not exist on either sides of the gateway, the proposed gateway model should define and generate it [40]. In PYLON-Lite, *service-y* is represented by DiffServ with ARSVP, and one example of *service-x* can be the SWAN model with the probing messages.

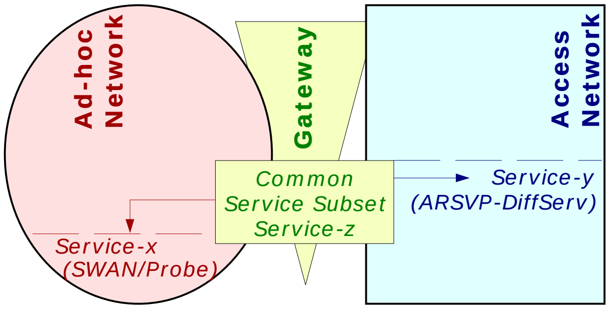


**Figure 3-4:** *PYLON-Lite Service Complementation*

In addition, the service adaptation layer defines the service conversion and mapping. It is important to realize that the mapping of the Common Service Subset as illustrated in Figure 3-4 is asymmetric. If *service-*$x_1$ requested by the ad-hoc domain maps to *service-*$y_1$ on the access domain side, that does not necessarily mean that *service-*$y_1$ also maps to *service-*$x_1$. The access domain provides aggregated services that deliver the same services to all packets belonging to the same service class. But the ad-hoc domain may provide only per-flow services that require explicit resource allocation. The Common Service Subset deals only with service protocol messages. Actual services are ultimately provided by the underlying QoS model, DiffServ in the access domain and SWAN, for instance, in the ad-hoc domain.

### 3.2.4 PYLON-Lite Service Adaptation Function

PYLON-Lite uses ARSVP as the Common Service Subset. The Service Adaptation Function transforms the common services into service specific requests as illustrated in

Figure 3-5. The Compatibility Module carries out the task of protocol conversion to the specific protocol implemented on the ad-hoc side and to accommodate future QoS model changes. In order to enforce negotiated services, data packets are re-marked at the gateway.

This approach provides great flexibility in defining operational QoS models on the ad-hoc domain, but imposes challenges in designing the Service Adaptation Function, and various other conversion policies. One challenging problem is the design of the Compatibility Module which should evolve as PYLON-Lite defines interfaces to emerging QoS models on the ad-hoc domain.

The Compatibility Module operates by maintaining two association tables between the two mapped protocols, one for each flow direction. The association table is linked to the specific process that has to be invoked on the Admission Controller. In addition, a set of rules are defined to respond to different interaction messages from the Admission Controller and to administrate timeout mechanisms.

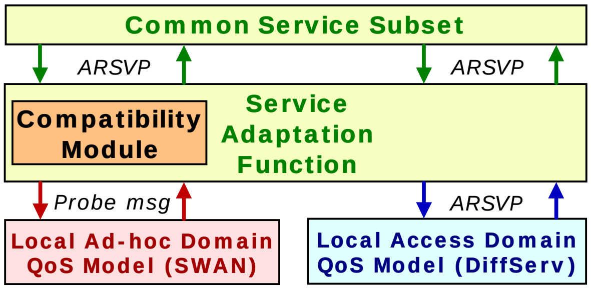

Common Service Subset
ARSVP
ARSVP
Compatibility Module
Service Adaptation Function
Probe msg
ARSVP
Local Ad-hoc Domain QoS Model (SWAN)
Local Access Domain QoS Model (DiffServ)

**Figure 3-5:** *Description of the PYLON-Lite Service Adaptation Function*

The use of both Common Service Subset and Service Adaptation Function leads to some similarities between PYLON-Lite and the IntServ over DiffServ framework as specified in [8]. For instance, PYLON-Lite relies on the ARSVP protocol on an E2E-basis in the same way as specified in the IntServ over DiffServ framework in order to perform the E2E admission for downstream flows.

### 3.2.5 PYLON-Lite Asymmetric Environment

It is also important to revisit the fact that PYLON-Lite operates in an asymmetrical environment. Services provided for the downstream flows are different from services provided for the upstream and PYLON-Lite employs different policies and requirements as well. Due to the vast differences between the network capabilities on both sides of the gateway, the environment is inherently asymmetric, and therefore, PYLON-Lite uses heterogeneous components. The provided services depend on the specific QoS implementation on both sides of the gateway, and on the enforced policies. Table 3-2 summarizes the PYLON-Lite characteristics and highlights the differences in QoS models implemented on both sides of the gateway.

| Description | Ad-hoc Domain QoS Model | PYLON-Lite | Access Domain QoS Model |
|---|---|---|---|
| **Adapted QoS Model** | SWAN, ESWAN, INSIGNIA, or others | PYLON-Lite on the access gateway | DiffServ |
| **Service Registration Protocol** | Various probe messages that follow non-standard protocols | Converts probe messages to the Common Service Subset using (ARSVP) as a common protocol. Thus, the E2E protocol is ARSVP, but a protocol conversion takes place to match the per-domain needs. PYLON-Lite assumes that ← DNST traffic follow the IntServ over DiffServ framework [8]. | ARSVP |
| **Service Accounting** | Mobile nodes and gateways can sponsor services, but no service accounting is available in the ad-hoc domain. | PYLON-Lite gateway validates and authenticates sponsorship via AAA server, then maintains per-domain service accounting. | Edge routers validate and authenticate ARSVP messages via AAA server, then maintain per-domain service accounting. |
| **Service Policing** | No service policing is currently available, thus, nodes can abuse | In → UPST scenarios, services are policed only at a per-class granularity. A limited per-flow policing is optional in PYLON-Lite. | Edge routers police services at a per-class granularity. Model relies on source |

| Description | Ad-hoc Domain QoS Model | PYLON-Lite | Access Domain QoS Model |
|---|---|---|---|
| | and drain network resources. | In ← DNST scenarios, services are policed only at a per-class granularity. Per-flow service policing takes place at source domain. | domain for per-flow policing. |
| **Packet Marking** | Most models do not rely on packet marking. | Re-marking is enforced at the PYLON-Lite gateway. | Re-marking is essential and is enforced at edge routers. |
| **Classes of Service (CoS) and Mapping of Service Equivalence** | Most models provide only two classes (real-time and best-effort). | Supports 8 service classes via the Type of Service (ToS) field. Can be extended to match more elaborate classes as in DiffServ.<br>Service Equivalence mapping policy can be designed and stored in AAA server; the PYLON-Lite gateway enforces it via traffic re-marking. | Uses the ToS field to define various Classes of Services (CoS). |
| **Collective Aggregate Reservation** | Currently, not available | In an → UPST scenario, it converts per-flow into per-class aggregation.<br>In a ← DNST scenario, it converts per-class into per-flow granularity. | Possible |
| **Service Granularity** | Typically, per-flow | Per-class | Per-class |
| **Service Guarantees** | Only soft service guarantees are supported due to the network dynamics. | Using the least common service concept, the E2E services supports soft service guarantees at best. | Can support statistical service guarantees. |

| Description | Ad-hoc Domain QoS Model | PYLON-Lite | Access Domain QoS Model |
|---|---|---|---|
| **Service Provisioning and Resource Allocation** | Currently, there is no defined mechanism. | Uses reactive approach to allocate required services in order to avoid service provisioning. Relies on Service Ladder policies to accommodate or upgrade to future services. | Rely on statistical model to provision services. |
| **Scalability** | Some scalability concerns in ad-hoc networks. From QoS view point, there are concerns about the amount of signaling and load balancing. | Fully scalable, no limits on the number of gateways an ad-hoc network can cling to. No boundaries on the number of flows that can pass through a specific PYLON-Lite gateway. | The DiffServ model is highly scalable bounded by the physical limitations. |

**Table 3-2:** *Asymmetric Characteristics of the PYLON-Lite Gateway Environment*

In the absence of a cross domain model like PYLON-Lite, extranet traffic is simply downgraded to best-effort traffic. Thus, the E2E quality of any extranet flow improves significantly with the use of PYLON-Lite.

## 3.3 Major Limitations Affecting PYLON-Lite Design

The QoS gateway to the ad-hoc networks essentially inherits some of the QoS challenges described for the ad-hoc domains in Section 1.3 in addition to having other challenges. As a result, the design of PYLON-Lite is found to be fairly challenging, and the limitations influencing its design can be summarized as follows.

1- Difficulties provisioning services in the ad-hoc domain. PYLON-Lite uses reactive service allocation combined with service ladder policies to alleviate the need for service provisioning. (Reactive collective resources)

2- The complexity of dealing with different QoS models employed on each side of the gateway. The design components of PYLON-Lite are, therefore, heterogeneous. The design of the Service Adaptation Function and its Compatibility Module is fairly challenging. Gateway administrators also have to define difficult hybrid policies that can, for instance, map per-class services into per-flow services. This should be, however, acceptable as it reflects the complexity of the problem domain. (Asymmetric, Heterogeneous)

3- The lack of per-flow policing from the ad-hoc network side. Flow policing can take place either at a distributed level, or at the gateway. The former solution raises security concerns, while the later raises scalability concerns. PYLON-Lite follows a flexible design that can perform either solution. The PYLON-Lite gateway administrator can reach a reasonable human compromise based on knowledge about the ad-hoc network characteristics. (Secure - Scalable)

4- The difficulties in generating and maintaining accounting records for mobile nodes. Following the same reasoning used with flow policing, the PYLON-Lite gateway administrator can reach reasonable human compromise based on knowledge about the ad-hoc network characteristics. (Secure - Scalable)

5- The lack of processing power, storage, and the limited capabilities of mobile nodes forces the design of PYLON-Lite to limit the processing that takes place at the mobile nodes to the minimum. (Lightweight)

6- The dynamics of the ad-hoc network in terms of node mobility, and unpredictability of wireless links enforces the design of a robust, self recovering model. (Robust – Self-recovering)

Those inherent limitations of the environment fuel the need for a reactive, heterogeneous, secure, scalable, lightweight, robust, and self-recovering model. PYLON-Lite uses reactive collective resource allocation to alleviate the need for service provisioning. The model architecture and its Compatibility Module deal with the asymmetry of the QoS problem. PYLON-Lite security issues are discussed in details in Appendix E, and scalability issues are discussed in Section 3.9. The model is lightweight and requires mobile nodes, in the worst case, to merely be RSVP-aware. PYLON-Lite uses a timeout mechanism to deal with network dynamics and node mobility. It applies timeout mechanisms for implicit resource de-allocation in order to facilitate self-recovery from adverse situations.

## 3.4 PYLON-Lite Design Fundamentals

The PYLON-Lite model relies on a pragmatic view of the cross-domain QoS problem; PYLON-Lite views the ad-hoc domain as a parasite domain that needs to cling to a hosting access domain in order to gain access to the Internet and other services. PYLON-Lite expects access domains to support DiffServ functionalities; therefore, gateways to access domains are expected to be DiffServ and RSVP-aware edge routers. PYLON-Lite also relies on the existence of one or more Sponsor Nodes (SN) in the ad-hoc domain. Since the gateways are part of the ad-hoc as well as the access domains, a gateway node can act as a Sponsor Node (SN). PYLON-Lite is based on a range of assumptions and fundamental tenets that are listed and discussed below.

### 3.4.1 Access Network as a DiffServ Domain

It is important to provide sufficient consistency between QoS models running on both ad-hoc and access domains. While QoS models on the ad-hoc side are still in experimental design stages, the QoS models on the fixed topology access domains have been settled into the known DiffServ and IntServ models [9], [97] and [106].

If a specific flow is required to explicitly follow an IntServ approach by booking E2E resources, it is rather easy to show that such an approach will essentially fail to run in the ad-hoc domain due to the dynamic nature of mobile nodes as shown in Section 2.2.1.2. Now assume that an enhanced approach implements E2E IntServ reservations with efficient local route repairs. Then assume that the new approach is able to provision and utilize bandwidth fluctuations efficiently. The IntServ per-flow nature will essentially raise scalability concerns, and will bound the implementation to limited size ad-hoc network installations.

The PYLON-Lite model views the access domains' DiffServ QoS model as a closer match to ad-hoc QoS models. Enabling DiffServ with collective resource allocation on the access domain provides scalability and better use for the elasticity of bandwidth reservation on the ad-hoc side. At the same time, optimal QoS solutions can be applied on the ad-hoc side without any enforced limitations.

### 3.4.2 Access Network as a Private Friendly Domain

Unfriendly access domains, as illustrated in Section 3.1, are not covered by PYLON-Lite since they do not provide any services to the ad-hoc domain. Public Friendly (Basic) Access Domains provide merely one set of services that has no service differentiation, and therefore, represent no QoS challenges. Hence, the PYLON-Lite model framework, and operations are geared towards Private Friendly Domains only.

### 3.4.3 Reactive Resource Allocation

Services are requested only when needed, and are maintained via a periodical service refresh message. If the PYLON-Lite gateway does not receive the service refresh message for a certain period of time, it issues a solicit ARSVP message to find a new sponsor for the admitted flows. If the gateway cannot establish new services for the ad-hoc domain, it denies priority services to the related flows, and releases allocated resources. This can happen, for example, due to ad-hoc partitioning. Source nodes that require new services have to issue flow service request, which trigger the gateway to solicit an ARSVP message. The reactive resource allocation approach must not be confused with the routing protocol approaches; in fact PYLON-Lite operates independent of the routing protocol. The reactive resource allocation alleviates the need for service provisioning.

### 3.4.4 Collective Aggregated Resource Allocation

PYLON-Lite uses collective aggregated resource allocation in order to allow ad-hoc nodes to share resources. PYLON-Lite adopts the use of the Aggregate RSVP (ARSVP) [6] protocol, which is particularly efficient for cross-domain reservations. Any sponsor node can perform service allocation with an access domain on behalf of the entire ad-hoc domain. All mobile nodes may benefit from the allocated resources without having any service agreement with the access domain. A timeout mechanism is used to free allocated resources in case of network partitioning. The collective aggregated resource allocation plays part in alleviating the need for service provisioning.

## 3.5 PYLON-Lite Design Considerations

The PYLON-Lite design highlights certain considerations that define the model in general, and are essential to convey the main ideas behind PYLON-Lite.

### 3.5.1 Periodical QoS Reporting

The PYLON-Lite gateway generates QoS reports based on the amount and duration the services have been provided. The Sponsor node SN receives the QoS reports and issues a periodic service refresh message. The gateway replies by acknowledging the refresh message. The absence of a gateway acknowledgement is used by SN to mark the end of a service, and can be used for billing purposes. From the gateway side, the absence of a refresh message indicates a network partitioning, and forces the gateway to solicit a new sponsor for the ongoing flows before the granted services expire.

### 3.5.2 Limited Policing

PYLON-Lite implements limited policing on extranet flows and uses a simple hash function based on the source and destination IP and port addresses to distinguish each flow. Admitted flows cannot upgrade the granted level of service without performing a new admission. If the used bandwidth of a specific service class grows higher than the burst limits agreed upon in the TCA, the PYLON-Lite gateway detects the out-of-context packets and downgrades excessive packets to lower services. PYLON-Lite implements limited policing using two distinct levels, mandatory per-class policing to enforce the TCA, and optional per-flow policing via the PYLON-Lite Flow Policing Controller (FPC) component to enforce the flow admission parameters.

Limited policing is designed to police service levels of legitimate mobile nodes. Nodes can initiate service requests only after joining the ad-hoc network. However, using the current ad-hoc architecture, un-authorized nodes can still join the ad-hoc network, to listen, intercept, or send traffic. This is one of the many security issues in ad-hoc networks that have gained increasing research attention. Another problem is reading the IP header of an IP-secured flow for example. PYLON-Lite relies on a multi-fence security solution to defend the ad-hoc network from various types of attacks, and remains focused on QoS issues and traffic policing. PYLON-Lite assumes the fidelity of all nodes forming the ad-hoc network either due to the nature of the network application or due to several security fences added to various implementation layers. However, we discuss all issues related to security in Appendix E, and provide recommendations in addition to illustrating different possible solutions.

### 3.5.3 Service Ladder Policy

In an upstream scenario, the access domain provides services on a per-class basis. PYLON-Lite recommends a simple Service Ladder Policy that is consistent with its reactive collective resource allocation mechanism and facilitates the aggregation of flows into a single service class. The Service Ladder Policy employs a pre-configured set of services that are used in an incremental pattern when upgrading services, and in a decremental pattern when downgrading services.

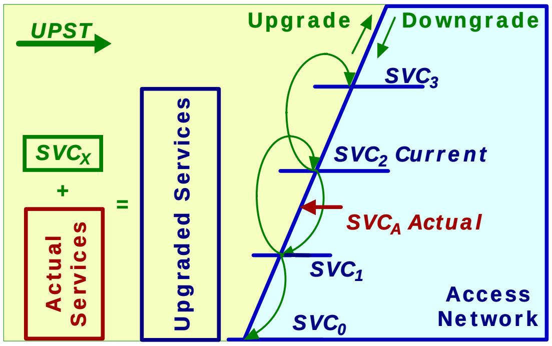


**Figure 3-6:** *Service Upgrade Process Using Service Ladder Policy*

Figure 3-6 illustrates a simple example, suppose the PYLON-Lite gateway runs a Service Ladder Policy for service levels $SVC_1$, $SVC_2$, and $SVC_3$ where $SVC_1 < SVC_2 < SVC_3$, the actual service level is $SVC_A$, and the currently allocated service level is $SVC_2$ where $SVC_1 < SVC_A < SVC_2$. Then, assume the PYLON-Lite gateway admits a new flow that requires additional services $SVC_X$, where $SVC_2 < SVC_A + SVC_X$. In this case, the Service Ladder Policy requires that the PYLON-Lite gateway upgrades the services for the ad-hoc domain to $SVC_3$. In the same way, suppose the actual service level is $SVC_A$, and the currently allocated service level is $SVC_2$ where $SVC_1 < SVC_A < SVC_2$. Then assume the PYLON-Lite gateway decides to terminate a flow that used to have $SVC_X$ amount of services, where $SVC_1 < SVC_A – SVC_X$. In this case, the Service Ladder Policy requires that the PYLON-Lite gateway maintains the services for the ad-hoc domain at the $SVC_2$ level.

The problem with the Service Ladder Policy is that it is less than optimal. In a Service Ladder Policy, allocated services are more than, or equal to, actual services. In fact, for this specific reason the Service Ladder Policy accommodates the ad-hoc network dynamics. The level of services required by the ad-hoc domain fluctuates significantly, reflecting the dynamic nature of the network. An optimum service policy will lead to frequent adjustments to allocated services in response to minor variations on the ad-hoc side. The critical decision is to configure the service ladder levels such that the difference between subsequent levels is not too high that it wastes resources, yet not too small that the gateway requires to process service upgrades or downgrades too frequently. In the absence of a sound service provisioning mechanism for the ad-hoc domain, PYLON-Lite leaves the policy decision to the gateway administrator to benefit from information about each specific ad-hoc network application.

The use of reactive resources combined with the use of collective aggregated resources and a Service Ladder Policy limit the influence of service variations on the model performance. Part of the service variation is absorbed by the slack of resources, and therefore, removes the complication of performing admission in response to minor variations in the ad-hoc network topology for instance. It is important to realize that the Service Ladder Policy is not part of the PYLON-Lite design; it is merely a recommended policy that can simplify the QoS problem in the absence of a service provisioning mechanism for ad-hoc networks.

### 3.5.4 Other Miscellaneous Considerations

It is important to highlight some other miscellaneous considerations as part of the PYLON-Lite design. For instance, the model does not require any gateway discovery mechanism. Instead, it relies on the underlying routing algorithm to carry the traffic to the "*closest*" gateway. Also, the process of admission control takes place at the gateway, and packets are re-marked at the gateway.

A single mobile node may deliver part of a specific flow to one gateway, then, due to mobility, submits another part of the traffic to a different gateway as in Figure 3-2. Since gateways may, potentially, be connected to different access domains, packets with the same DSCP value may experience slightly different PHB when transmitted by different access domains. This problem is inescapable due to mobility scenarios, and it exists in most mobile environments.

## 3.6 PYLON-Lite Design Architecture and Building Blocks

PYLON-Lite uses a predefined fixed NDSCP in order to distinguish classes of services. The model defines a limited set of operations that take place at mobile nodes, and defines those operations when the mobile node acts as source, or sponsor node. The model also defines three techniques essential to its operations in a zero-mobile-implementation mode where no implementation is required at mobile nodes. Then the model defines the PYLON-Lite gateway architecture and operations in both upstream and downstream scenarios.

### 3.6.1 PYLON-Lite NDSCP

The PYLON-Lite model adopts eight service classes as its NDSCP. The eight classes provide a reasonably detailed DSCP and are consistent with the 802.11e/802.11g guidelines [28], [118]. Table 3-3 shows the proposed NDSCP.

| Service Class | Designation |
|---:|---|
| 0 | Best Effort |
| 1 | Best Effort |
| 2 | Best Effort |
| 3 | QoS Probing & Signaling |
| 4 | Audio / Video |
| 5 | Audio / Video |
| 6 | Audio / Video |
| 7 | Audio / Video |

**Table 3-3:** *PYLON-Lite NDSCP Priority Table*

The application selects the proper service class. For instance, a video streaming application may select any service class, knowing that classes 4 to 7 provide increasingly better resources. The application may select class 6, and an online video application may select class 7. The cost associated with having higher services forces applications to perform wiser class selection.

In the NDSCP set, class number 3 is reserved for QoS probing and signaling. Classes 0, 1, and 2 provide relative service for non real-time traffic. Some of the proposed QoS models like SWAN, and INSIGNIA provide only two classes of services (namely real-time RT and best-effort BE). Even with this limitation, it is possible to map BE-services to one lower class (0-2) and RT-services to one higher class (4-7). This study refers to flows demanding high services as RT-flows, and packets that require no service differentiation as BE-packets. Since INSIGNIA, SWAN, and ESWAN all divide traffic into RT and BE-traffic only, this research follows the same notion. However, PYLON-Lite is not limited to the two classes.

The PYLON-Lite gateway re-marks packets to reflect the service level required for the payload. Limited policing is implemented at the gateway to filter out greedy and selfish use of resources.

### 3.6.2 PYLON-Lite at Mobile Node

In the PYLON-Lite framework, mobile nodes may operate as a source, a destination, or a sponsor node. The gateway is a special type of mobile node that has both wired, and wireless interfaces.

#### *3.6.2.1 The Source / Destination Node Operations*

In an upstream scenario → UPST, a mobile node may initiate an extranet flow managed by PYLON-Lite. The source node must form and submit a service request to the destination node. The gateway intercepts the request and starts the admission process. The source node then waits for a flow probe reply to start transmission with the required, or a possibly lower DSCP. The PYLON-Lite gateway hides the ad-hoc QoS specific implementation from the destination node in upstream scenario → UPST.

The same approach is used in the downstream scenario ← DNST. Since the source node is located outside the ad-hoc domain, the gateway to the ad-hoc domain intercepts the source RSVP message and initiates the service request on behalf of the source node, and then, it follows the ad-hoc QoS model specifics in order to submit RT-flows to the ad-hoc domain. The gateway must perform all necessary processes on behalf of the original source node. The PYLON-Lite gateway hides the ad-hoc QoS specific implementation from the source node in the downstream scenario ← DNST.

#### *3.6.2.2 The Sponsor Node Operations*

When a mobile node receives a solicit ARSVP message, it ignores the message if the node cannot sponsor services with the associated access domain. Otherwise, it checks the soliciting gateway and issues a service refresh message if the node is already sponsoring services with that gateway. Otherwise, it forms and issues an ARSVP initiate message to the gateway. The details on how the gateway can locate a sponsor node, and solicit a service request follows [74].

#### *3.6.2.3 The Zero-mobile-Implementation Mode*

PYLON-Lite has one built-in, and two optional techniques that together facilitate zero-implementation at mobile nodes. First, the PYLON-Lite gateway, as described in Section 3.6.2.1, acts on behalf of destination nodes in an upstream scenario → UPST, and on behalf of the source nodes in a downstream scenario ← DNST, by performing the ad-hoc specific QoS probing, regulating, and reporting on relevant extranet traffic. This built-in technique hides the QoS differences from mobile nodes, and provides mobile nodes with a global network view that virtually employs the same QoS model the mobile nodes have.

Second, since the PYLON-Lite gateway is part of the ad-hoc domain, it can adopt the role of SN, and therefore, the role of mobile node as SN becomes optional. Third, PYLON-Lite, like many other QoS models, expects the source application to specify the required NDSCP and to mark packets submitted to the network using the selected DSCP. However, in a practical sense, source nodes may not be able to accurately use the NDSCP set. The PYLON-Lite gateway is equipped with an admission control mechanism that intercepts (→ UPST), or initiates (← DNST) extranet service requests, translates the service requests into the corresponding DSCP, then registers the flows with the selected DSCP and enforces flow re-marking.

When applying the three zero-mobile-implementation techniques, PYLON-Lite can be deployed merely by implementing the PYLON-Lite gateway and without any specific implementation or configuration at mobile nodes. This characteristic is very powerful since it enables mobile nodes to roam around different access domains freely without worrying about the type of QoS implementation and without even having any domain specific relation. The use of PYLON-Lite zero-mobile-implementation is highly recommended and will be used throughout this research as a default configuration. The ability to initiate new services in response to a request from mobile sponsor node is viewed as an additional feature that may be utilized in specific situations.

### 3.6.3 PYLON-Lite Gateway Architectural Components

This section illustrates the PYLON-Lite gateway architecture in both downstream and upstream scenarios.

*3.6.3.1 PYLON-Lite Gateway in Downstream Scenario*

The basic PYLON-Lite gateway architectural components in a downstream scenario are illustrated in Figure 3-7. Some of the components in Figure 3-7 exist in most current gateway architectures. The blocks numbered 1' to 5' represent the PYLON-Lite specific components.

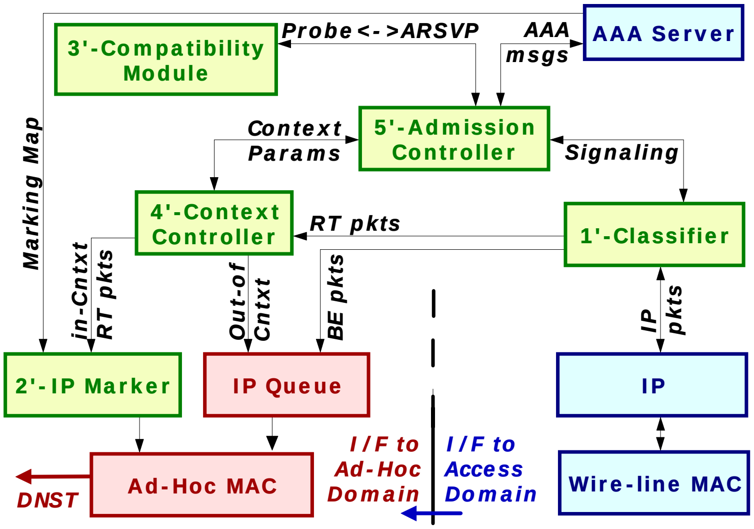


**Figure 3-7:** *PYLON-Lite Gateway Architecture in Downstream Scenario*

**1'- The Classifier:** The function of the Classifier is to direct relevant signaling packets to the PYLON-Lite Admission Controller, to deliver the RT-packets to the PYLON-Lite Context Controller, and to deliver BE-packets to the IP-queue on the interface to the ad-hoc domain.

**2'- The IP Marker:** The IP Marker re-marks in-context packets into the correct marking to match the notion used by the QoS model in the ad-hoc domain.

**3'- The Compatibility Module:** The Compatibility Module performs necessary protocol conversion between the ARSVP as a Common Service Subset, and the specific message protocol implemented by the ad-hoc QoS model (e.g. probe messages in SWAN).

**4'- The Context Controller:** The Context Controller maintains the levels of the aggregated rate of flow in order to comply with the rates agreed upon in the Traffic Conditioning Agreement (TCA), and initiates upgrade request when the node demands more services. An upgrade request may be granted by the Admission Controller if, for instance, the SN is willing to sponsor more traffic. The Context Controller operates based on the context parameters provided by the Admission Controller. In-context packets are forwarded to the IP Marker downstream for re-marking and forwarding. Out-of-context packets can be downgraded to BE-packets, and then send to the IP-queue for forwarding.

The Context Controller may operate at two distinct levels. A first level is the per-class context control that validates the use of aggregate services against the defined service class context parameters. A second level is the per-flow context control that validates the flow use of services against the admitted flow context parameters. In both cases, out-of-context packets are handled according to the pre-defined policies. The different levels of Context Controller are discussed in detail in Section 3.8.3.

**5'- The Admission Controller:** The Admission Controller validates the authenticity of service requests and sponsorship via the AAA Server, and deals with service upgrades and downgrades. In addition, it updates the AAA Server with the actual resource usage for accounting purposes. The Admission Controller performs additional generic functions in the downstream scenario. It performs the necessary signaling with the ad-hoc domain using the selected QoS model. In the same way, it accepts signaling from the access domain side on behalf of the mobile destination node. PYLON-Lite assumes that downstream RT-traffic follows the IntServ over DiffServ model as defined in [8] and discussed in Section 2.2.4.

When the Context Controller detects a new downstream RT-flow, it informs the Admission Controller which in turn has to submit a service request on behalf of the source host. The Admission Controller assumes that the new flow belongs to a RT-flow that has been admitted by another gateway elsewhere. The Admission Controller uses the Compatibility Module to perform the admission in the ad-hoc domain using the correct probe messages.

There is an implicit assumption of fidelity in all nodes forming the ad-hoc domain as discussed in Section 3.5.2.

The AAA server illustrated in Figure 3-7 for downstream scenarios represents a logical entity that may not necessarily exist on the PYLON-Lite gateway. Instead, the gateway is expected to have access to the AAA server. The AAA server maintains authentication, authorization and accounting information in addition to information about the TCA.

### *3.6.3.2 PYLON-Lite Gateway in Upstream Scenario*

The components defined for the downstream scenario are reused for an upstream scenario and are illustrated in Figure 3-8. Most components perform the same functionality but in almost a reverse sense. For instance, the classifier in → UPST scenario (1′) filters probe messages but the classifier in ← DNST scenario (1″) filters RSVP messages. The IP Marker (2′) illustrated in Figure 3-7 is different from the IP Marker (2″) illustrated in Figure 3-8. The former IP Marker re-marks in-context RT-packets using the selected NDSCP value, while the latter IP Marker uses the local notion of the QoS model running in the ad-hoc domain. This difference between → UPST and ← DNST scenarios reflects the heterogeneous nature of the PYLON-Lite environment.

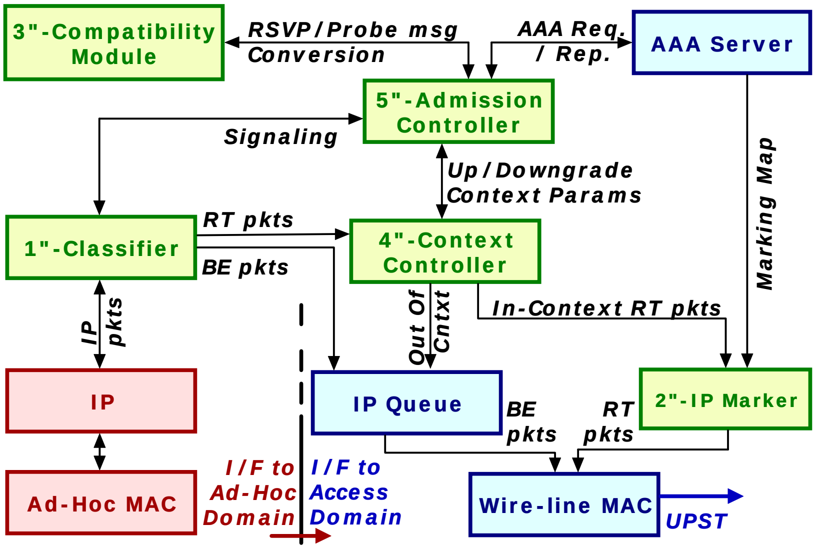


**Figure 3-8:** *PYLON-Lite Gateway Architecture in an Upstream Scenario*

The Admission Controller performs a reverse function in the upstream scenario as well. It accepts signaling from the ad-hoc domain side on behalf of the destination host. Upstream flows do not have to follow the IntServ over DiffServ operations.

While PYLON-Lite assumes fidelity in all nodes forming the ad-hoc domain, it is important to discuss the per-flow policing. Based on the PYLON-Lite design illustrated so far, it is possible for a greedy node to request a certain level of service and then to consume a significantly higher service level. PYLON-Lite per-flow policing issues are discussed in Section 3.8, and a Flow Policing Controller (FPC) component is added as an optional component. Therefore, the architecture described in Section 3.6.3 can be viewed as PYLON-Lite architecture in FPC-Disabled mode.

## 3.7 The Use of Aggregate RSVP

ARSVP [6] is used by Sponsor Nodes to apply collective resource reservations. ARSVP is known to be efficient particularly for cross-domain reservations. Assume one aggregate is associated with one service class. All E2E reservations that employ RSVP use the same aggregate if they belong to the same class. In other words, all same class reservations share resources reserved by a single ARSVP. This raises the problem of dealing with bursty traffic, since some flows may consume the resources of other flows. Clark [22] proved that the performance degradation due to bursty flows comes with performance enhancements in the form of reductions of delay in the tail of the delay distribution (e.g. 99% percentile delay) for the flows.

### 3.7.1 Size of Aggregate Reservation

There is a range of options for determining the size of the aggregate reservation presenting a tradeoff between simplicity and scalability. Simplistically, the size of the aggregate reservation needs to be greater than or equal to the sum of the bandwidth of all flows it aggregates, and its burst capacity must be greater than or equal to the sum of all burst capacities. However, if followed religiously, this leads to changes in the bandwidth of the aggregate reservation each time a new flow is admitted or changed, which looses one of the key benefits of aggregation. If the GW has a significant amount of bandwidth reserved but has very little probability of using it, the policy may be to release the excess resources [6].

PYLON-Lite recommends the use of Service Ladder Policy as described in Section 3.5.3 which enables some slack in the amount of bandwidth reserved to accommodate expected network dynamics. The gateway administrator can set up the values for the policy ladder steps to maintain a reasonable balance between accommodating network dynamics and wasting network resources. Unused resources are automatically released following a timeout mechanism.

The PYLON-Lite model polices the size of aggregation at gateways, and facilitates a flexible service upgrade or downgrade without referring to an SN in order to decrease traffic signaling overheads. Hence, the GW monitors the actual traffic utilization of the friendly domain and forwards the actual resource utilization to the SN using QoS reporting. Then the SN accumulates the actual resource utilization reports for accounting purposes.

Estimating the size of an aggregate reservation is a growing research topic, with variant approaches like in [33] [34] and [38]. While the PYLON-Lite framework provides the mechanism to increase or decrease the size of the aggregation, PYLON leaves the decision of defining the ideal aggregate size for vendor-specific approaches, but recommends the use of the Service Ladder Policy. PYLON-Lite merely defines the GW as the decision point. An advantage of this framework is the fact that it accommodates innovative vendor solutions without introducing changes to the basic framework.

### 3.7.2 ARSVP Message Handling

In response to a solicit message issued by the PYLON-Lite gateway, a sponsoring node SN generates an ARSVP initiate message describing the traffic aggregate (service class) based on Table 3-3, and loads the message with the authentication record that can be verified with the AAA server. The gateway receives the ARSVP initiate message and uses standard procedures as described in Section 3.6.3 to allocate resources in the access domain.

The ARSVP initiate message is forwarded using the IP Protocol Number *RSVP-IGNORE* (*currently 134*). This message is forwarded by any RSVP router not performing aggregation [15], [45]. Therefore, the ARSVP initiate message traverses the ad-hoc domain, without being intercepted by intermediate nodes; instead, intermediate nodes simply forward it as a normal IP packet until it reaches the PYLON-Lite gateway.

If this admission is granted, the GW simply confirms by replying to the SN with IP Protocol Number set to *RSVP-CONF (15)*. Alternatively, if the access domain is unwilling to support required services, the gateway communicates the reservation failure by sending an ARSVP reply message, with IP Protocol Number set to *RSVP-ERR (6)*, to the SN.

### 3.7.3 Removal of Aggregate Reservation

Following Section 2.9 of RFC 3175 [20], an aggregate reservation is removed upon expiration. A GW receives a periodic refresh message from the SN to ensure the continuous existence of the SN within the ad-hoc domain. Otherwise the GW presumes that the SN is dead, or unreachable. It is also possible to make the reservation for a specified limited or unlimited lifetime. In all cases, policies are predefined at the AAA server.

## 3.8 PYLON-Lite Policing Issues

The cascaded networks environment illustrated in Figure 3-9 and described earlier in Section 3.2 represents the operational environment for PYLON-Lite gateway. In a cascaded service networks, two types of service policing can take place, namely, per-class policing and per-flow policing. This section describes PYLON-Lite's per-flow policing solution, and discusses the impact on scalability associated with flow policing.

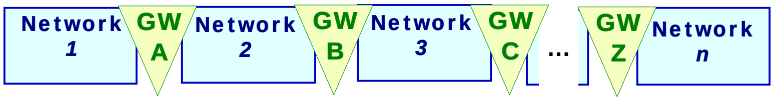


**Figure 3-9:** *View of n Cascaded Service Networks*

For example, assume that a host from network *1* in Figure 3-9 is the source of a specific RT-flow that targets another host located in network *n*. Service agreements between intermediate networks (e.g. networks *2* and *3*) are likely to be a per-class agreement, and the gateway between the two networks (e.g. gateway B) can enforce such an agreement. Enforcing a per-class service policing keeps the edge routers (gateways) scalable since they do not need to maintain per-flow information. This edge-routers' per-class service policing is popular in current networks and gateways.

### 3.8.1 The Per-flow Service Policing

The second type of service policing is the per-flow policing. Following the same example, if a RT-flow is initiated in network *1* and targets a host in network *n*, then network *1* has to limit the admitted RT-flow to its negotiated service parameters and prevent any greedy use of resources that may violate the admission constrains. The per-flow policing process requires maintaining per-flow information and a user profile. Since only the source network (i.e. network *1*) is expected to have access to the user profile, the per-flow policing is commonly performed at the source network. The Authentication, Authorization, and Accounting server (AAA server) is used to provide information about user profiles. Current networks enforce per-flow policing usually at the first hop following the source host in order to maintain a lower volume of traffic within its infrastructure (e.g. within network *1*). Distributed per-flow policing is commonly used in fixed-topology network.

Enforcing the per-flow policing closer to the source host and the per-class policing at edge routers leads to tightly regulated use of network resources. The absence of per-flow policing may not cause traffic overflow in intermediate networks for example, but will allow unfair use of resources at the source network. For example, some flows may be able to use higher bandwidth than their agreements while other flows will be denied services granted in their agreements. Similarly, the absence of the per-class policing simply enables violations to the service agreements. The per-flow and per-class policing together maintain the volume of traffic within the designed and desired levels.

### 3.8.2 The Enforcement Point for Per-flow Policing

In an upstream scenario, the source domain is in charge of the per-flow policing. Unfortunately, ad-hoc networks have an inherent problem performing a distributed per-flow policing. Because the ad-hoc network can get divided into fragmented clusters, some mobile nodes may not have access to the AAA server to validate user profiles. Assuming that, somehow, the user profiles are accessible within an ad-hoc network, and the hop next to the source node can enforce the per-flow policing, since the source node may move to a different cluster, the per-flow information must be re-generated, and the per-flow policing must restart all over. In a dynamic environment like ad-hoc networks, the regeneration of user profiles and re-initiation of the per-flow policing information can consume considerable resources and compromise network security.

The alternative to the distributed per-flow policing is to perform the policing at the gateway. PYLON-Lite proposes a Flow Policing Controller (FPC) to operate on the gateway where it is safer to acquire user profile information, and to perform authentication. PYLON-Lite recommends the use of the gateway as an enforcement point for the per-flow traffic policing. The architecture described in Section 3.6.3 represents the PYLON-Lite gateway in FPC-Disabled mode and can not perform per-flow policing. The next section describes the PYLON-Lite operations in FPC-Enabled mode and focuses on its per-flow policing components.

### 3.8.3 The PYLON-Lite Gateway Architecture in the FPC-Enabled Mode

The PYLON-Lite architecture in the FPC-Enabled mode is illustrated in Figure 3-10 and is briefly discussed here to review its operations. Since the per-flow policing takes place in an upstream scenario, the discussion here is limited to the upstream PYLON-Lite architecture.

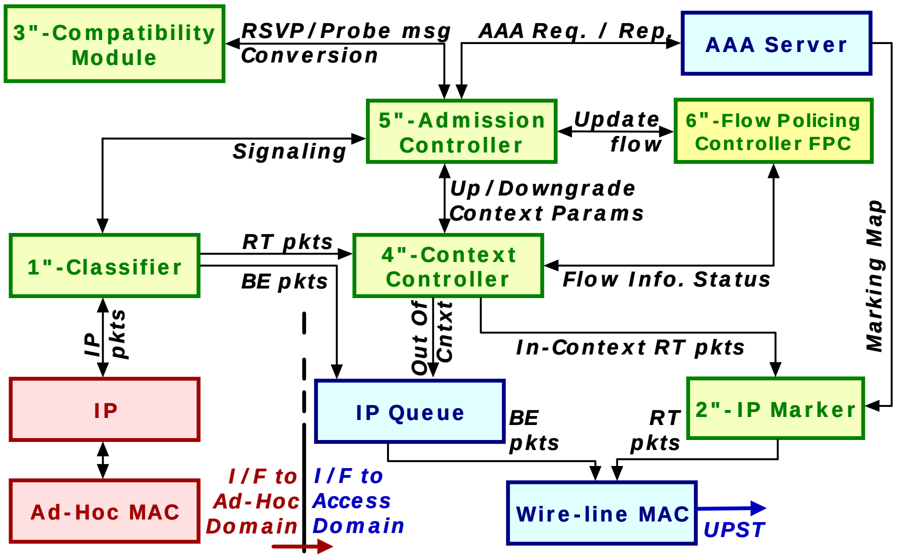


**Figure 3-10:** *PYLON-Lite Architecture in the FPC-Enabled Upstream Gateway*

The PYLON-Lite gateway in → UPST FPC-Enabled mode contains the 6 distinct components numbered 1″ to 6″ in Figure 3-10. The operational function of most of the unnumbered components is the same as in any generic gateway. The functions of the

Classifier, IP Marker, and Compatibility Module are identical to the description in Section 3.6.3. The operations of the other three components are illustrated hereafter.

**4″- The Context Controller:** The Context Controller monitors the per-class use of resources, and ensures it complies with the Traffic Conditioning Agreement TCA. In an FPC-Enabled mode, the Context Controller periodically performs the same function on a per-flow as well. The FPC provides the per-flow context information. Out-of-context flow packets can be downgraded to BE-packets. If a specific RT-flow continues to send more packets than its designated bandwidth for a certain period of time, the Context Controller informs the Admission Controller to process an upgrade request for this RT-flow. Upgrade requests follow a process similar to the process of admission requests. If the Context Controller realizes the arrival of a new RT-flow that can not be associated with an existing flow in the FPC table, it assumes that the flow has been admitted by a gateway elsewhere then moved to the current gateway, and informs the Admission Controller to process a new admission request for this RT-flow.

**5″- The Admission Controller:** The Admission Controller performs the exact functionality as describer earlier in Section 3.6.3 except that it updates the FPC with new RT-flow parameters for each admitted RT-flow.

**6″- The Flow Policing Controller (FPC):** The Flow Policing Controller maintains a lookup table for the flow-specific information. This information is updated by the Admission Controller, and is removed following a timeout mechanism. The sorting and searching algorithms are considered integral part of the FPC.

The PYLON-Lite FPC-Enabled mode enhances the traffic policing of the model, but raises scalability concerns. Maintaining per-flow information at the gateway imposes significant overhead, consuming the gateway resources. These concerns are addressed and evaluated in Section 3.9.

## 3.9 PYLON-Lite Scalability Issues

An integral part of the PYLON-Lite design is the analytical investigation of its scalability. Gateway scalability is commonly viewed in horizontal and vertical dimensions. The horizontal scalability refers to the possible restrictions on the number of PYLON-Lite gateways that can be attached to the same ad-hoc domain, or the same access domain. PYLON-Lite imposes no design restrictions on either domain. Therefore, PYLON-Lite fully scales horizontally.

The vertical scalability on the other hand investigates limitations that may influence the performance of the gateway as the amount of traffic grows. For instance, in a large-scale ad-hoc network with a large number of mobile nodes, the number of extranet RT-flows is expected to be high as well. PYLON-Lite responds to scalability concerns by operating in FPC-Disabled mode. Thus, it is important to review the trade-off between PYLON-Lite FPC-Enabled mode, and FPC-Disabled mode and the impact of FPC-Enabled mode on model scalability.

### 3.9.1 Components Scalability Concerns in FPC-Enabled Mode

In order to spot the PYLON-Lite components with limited scalability in an FPC-Enabled mode, the UPST architecture illustrated in Figure 3-10 is reused in this section. The six PYLON-Lite components are revisited with a focus on their scalability.

**1″- The Classifier:** The classification process is independent of the number of RT-flows; instead, it depends on the amount of transmitted packets which in turn is a function of the bandwidth. In particular, the wireless bandwidth is limited to a few tens of Megabytes per second even with current and future radio technologies such as IEEE 802.11 and IEEE 802.16. Since the amount of used bandwidth is bound by the gateway physical limitations, the classification process is scalable.

**2″- The IP Marker:** The marking process is independent of the number of RT-flows; it depends on the number of RT-packets which in turn is a function of the RT-bandwidth. Since the amount of used RT-bandwidth is bound by the gateway physical limitations, the marking process is scalable.

**3″- The Context Controller:** In FPC-Disabled mode, the Context Controller operates on the aggregate level only (i.e. per-class Context Controller). Since the number of traffic aggregates is limited, the Context Controller imposes no scalability concerns in FPC-Disabled mode. But when PYLON-Lite runs in FPC-Enabled mode, the Context Controller operates in a per-flow policing as well, and therefore, the Context Controller imposes scalability concerns. The complexity of the Context Controller in FPC-Enabled mode is discussed in Section 3.9.2.

**4″- The Admission Controller:** The Admission Controller does not keep track of a per-flow admission; instead, admission is performed on a per-class aggregate. The basic design idea behind PYLON-Lite is to convert the per-flow aggregation to a per-class aggregation, and to use the collective resource reservation to serve traffic. Therefore, the PYLON-Lite Admission Controller imposes no scalability concerns in either FPC operational modes.

However, if the PYLON-Lite gateway receives too many service requests in a specific time span, that can cause performance degradation. This case is considered a security risk, and is discussed in Appendix E.

**5″- The Compatibility Module:** The processing load of the Compatibility Module has a direct relation to the amount of signaling traffic. The Compatibility module handles messages arriving from the Admission Controller only, and if the Admission Controller is dealing with a high volume of signaling traffic, PYLON-Lite may experience performance degradation. This case is considered a security risk, and is discussed in Appendix E.

**6″- Flow Policing Controller (FPC):** The FPC component operates only in FPC-Enabled mode to maintain a lookup table on the flow-specific information. The size of the FPC table is directly related to the number of RT-flows admitted by the PYLON-Lite gateway, and therefore, the FPC component imposes scalability concerns. These concerns are discussed and evaluated in Section 3.9.2.

When PYLON-Lite operates in FPC-Disabled mode, greedy mobile nodes may abuse the network services and consume more resources than originally requested. But the FPC-Disabled mode allows the model to run without scalability concerns. On the other hand, when PYLON-Lite operates in FPC-Enabled mode, RT-flows can be policed and are forced to adhere to negotiated service parameters. But as the number of mobile ad-hoc nodes increases, the number of RT-flows grows, and the FPC lookup table size grows linearly. Two components of the model raise scalability concerns, namely the Context Controller, and the FPC. By focusing on those two components, Section 3.9.2 evaluates the complexity of the PYLON-Lite in FPC-Enabled mode.

### 3.9.2 The Complexity of the FPC-Enabled mode

The complexity of PYLON-Lite operation in a FPC-Enabled mode is essential supporting information for the gateway administrators. Typically, when the gateway admits a new RT-flow, it registers the flow information with the FPC. Then the Context Controller monitors the RT-flow use of resources, and periodically, validates the resource usage against the registered flow context information. Therefore, throughout the lifetime of a typical flow, it is ideally registered once, and is validated many times.

Precisely, if the lifetime of the flow is $L$ and the gateway performs $V$ validations in unit time, the RT-flow is validated $L\ V$ times. However, due to expected network dynamics, the RT-flow may need to re-register with a new gateway. Therefore, the PYLON-Lite gateway performs $r$ registrations and $V$ validations, and it is expected that $V >>> r$ during the lifetime $L$ of the flow. Also assume that $R$ is the number of flows registered in FPC at a specific time.

Assume for a specific gateway, every flow registers only once with the FPC, and assuming that the FPC uses any efficient sorting algorithm to maintain a sorted flow information table, then the complexity can be computed as in Equation 3-4):

$$registration\ complexity = O(\log R) \qquad \ldots (3\text{-}4)$$

Then, assuming the FPC implements a simple binary search algorithm to retrieve flow information for validation, the complexity of the validation process can be computed as in Equation 3.5:

$$validation\quad complexity = O(V \log R) \qquad \ldots (3.5)$$

In addition to the time complexity, storage complexity can be calculated as in Equation 3.6:

$$storage\ complexity = O(R) \qquad \ldots (3.6)$$

In order to put this complexity in perspective, a table of one million flows can be searched by performing less than 20 comparisons in the worst case [46]. Storage complexity on the other hand is a less alarming factor as the cost of storage is very small.

### 3.9.3 PYLON-Lite Scalability Note

In small to medium sized ad-hoc networks there is no scalability concerns in applying PYLON-Lite. However, the PYLON-Lite FPC-Enabled mode is presented as an optional part of the model. In installations where there is a huge emphasis on scalability, like the situation of limited resources on the gateway or the case of satellite ad-hoc networks where the number of RT-flows can grow beyond manageable limits. The gateway administrator can switch PYLON-Lite to run in FPC-Disabled mode. The gateway administrator should balance the alternatives between trusting mobile nodes and enforcing the per-flow policing option, and then, pursue a reasonable compromise. The complexity formulas provided in Section 3.9.2 help the administrator evaluate the value of the *"manageable limit "* for a specific gateway. We argue that the compromise in valuing scalability versus per-flow policing is better left to human judgment, and therefore, we provide the pros and cons of each operational mode, and support the gateway administrator with the required complexity formulas.

## 3.10 The Full-scale PYLON Model

The Full-scale PYLON model has been presented in [73] and detailed in [74]. The model follows a proactive approach that enforces a gateway discovery mechanism. Gateways proactively adopt an *Authentication Node* (AN) registration system, then solicit, and maintain (AN) information. Mobile nodes adopt a gateway registration system, and perform GW selection based on the metric information maintained on each GW. Due to the amount of data maintained at both GW and mobile nodes, the Full-scale PYLON model is perceived as a heavyweight model compared to PYLON-Lite.

Full-scale PYLON requires a comprehensive contribution from the access domain towards service provisioning following [72]. In contrast, PYLON-Lite follows a simple on-demand approach. Gateways can increase or decrease aggregate resource reservations in response to network demands. The PYLON-Lite on-demand approach comes at the expense of some delays when increasing or decreasing resources. The PYLON-Lite use of reactive service allocation is a less controversial approach than the Full-scale PYLON use of service provisioning, as it is quite hard to provision services for ad-hoc domains.

## 3.11 PYLON-Lite Design Conclusion

The PYLON-Lite design presents a set of essential terminologies required to communicate the fundamental ideas of the model. PYLON-Lite operates over eight different model cases utilizing network resources in both upstream and downstream traffic directions. The model benefits from the classical gateway design principles by employing Aggregate Resource Reservation Protocol ARSVP as a Common Service Subset, and by designing a Compatibility Module as a Service Adaptation Function. The model responds to the limitations imposed by the ad-hoc environment by providing a reactive heterogeneous, scalable, lightweight, robust, and self-recovering design.

The PYLON-Lite model assumes a DiffServ friendly private access domain. Then the model adopts a reactive collective resource allocation approach with per-class traffic policing to maintain its lightweight nature. The model also recommends the use of a service ladder policy, which is more suitable for the environment dynamics.

The PYLON-Lite model relies on the underlying ad-hoc routing mechanism to perform gateway discovery. The model uses an NDSCP set which is consistent with the IEEE 802.11e/g guidelines [28], [118], and provides component-based design and operations for both mobile nodes and gateways. The model provides a zero-mobile-implementation mode to limit the PYLON-Lite implementation to the gateway for faster and easier field deployment. In addition, the model provides aggregate reservation message handling.

PYLON-Lite proposes the use of a Flow Policing Controller (FPC) and illustrates its use for per-flow traffic policing. The model can operate in either FPC-Enabled or FPC-Disabled modes. PYLON-Lite illustrates the impact of the FPC modes on the model scalability and provides formulas for the added complexity as a decision supporting tool for gateway administrator. Finally, the model provides a brief comparison to the Full-scale PYLON, and describes the reasons for shifting the design towards reactive services.

PYLON-Lite is a pioneering cross-domain QoS model that links ad-hoc domains QoS to the fixed topology QoS solutions. The model is designed in a collective reactive manner to compensate for network dynamics, to accommodate the possibility of ad-hoc partitioning, and to increase model robustness. The model employs timeout mechanisms to facilitate self-recovery from adverse situations, and is easy to implement and deploy.

## CHAPTER 4

# The PYLON-Lite Performance Evaluation

Testing and evaluating network models represent a great challenge due to the difficulties in gathering all possible network scenarios into a single laboratory environment. Network simulators provide a handy tool for testing different algorithmic proposals and new technology mechanisms. However, the results of a simulation test need to be viewed carefully. There are potentially large differences between simulations and real network systems, specifically in a wireless environment that is vulnerable to environmental influences. One key evaluation factor is to provide a detailed description of the simulation assumptions, and used configuration parameters. Thus, before deploying a specific model, it is possible to retest and modify the simulator environment to closely match the real environment.

In this chapter, Section 4.1 illustrates the major factors affecting the PYLON-Lite model and test-bed performance based on our observations. Section 4.2 describes the PYLON-Lite test-bed and lists the used configuration parameters. Then Section 4.3 investigates the PYLON-Lite behavior in both upstream and downstream scenarios when SWAN is employed as QoS model in the ad-hoc domain. The section ends with a conclusion on the PYLON-Lite behavioral trends, and this section is tightly related to Appendix D which provides comparable results when INSIGNIA or ESWAN are employed in the ad-hoc domain. Section 4.4 introduces the average bandwidth efficiency ratio and uses it to measure the QoS provided to RT-flows. Section 4.5 shows the PYLON-Lite performance in extreme traffic load conditions. Section 4.6 highlights the validity of simulator time delays, and compares it to possible real-life delays. Then the section investigates the service initiation delay and the delay caused by a change in the access gateway. Finally, Section 4.7 concludes and comments on the model performance and test results.

## 4.1 Major Factors Affecting PYLON-Lite and Test-bed Performance

PYLON-Lite is a cross-domain QoS model that employs other QoS models in both the ad-hoc and the access domains. Thus, factors affecting PYLON-Lite performance are composed of all factors affecting any of the QoS sub-models. Instead of listing the factors affecting all possible combinations of QoS models, the following subsections illustrate the most significant factors that are found through intensive testing.

### 4.1.1 Hop Count (HC)

The flow hop count is a measure of the average travel distance of specific flow packets. Enabling PYLON-Lite causes less processing delays to RT-packets with higher HC. To illustrate this fact, assume two RT-flows, *F1* travels only two hops, one hop on each side of the gateway, and flow *F2* travels eleven hops, one hop in the access network, and ten hops in the ad-hoc network. Then assume a downstream scenario where RT-flows experience limited delays *d* at each hop equally. It is clear that flow *F2* will experience ten times more delays than flow *F1* considering that the propagation time of a packet is, comparatively, a negligible value. Enabling PYLON-Lite magnifies this difference because removing the queuing delays decreases the value *d*.

Therefore, when testing PYLON-Lite, it is important to decrease the HC effect by using the average of different flows with different HC. Extreme HC values lead to extreme views of the model performance.

### 4.1.2 Volume of Ad-hoc Intranet Traffic

Since the wireless ad-hoc nodes use shared bandwidth, the amount of intranet traffic limits the bandwidth available for extranet traffic. This general rule cannot be applied to the access domain since the bandwidth is always engineered to provide cross-domain traffic with the predefined statistically guaranteed resources as defined by the relevant policy. The problem with QoS models in an ad-hoc domain is that the traffic is treated based on its class and not the source-destination pairs. Therefore, higher intranet traffic simply consumes the same resources available to extranet traffic. For this reason, testing PYLON-Lite is done while carefully monitoring the intranet traffic with an objective to limit the per-link traffic capacity. This is a hard iterative process and requires carefully customizing the traffic generated for each specific mobility scenario. The objective is to generate sufficiently high traffic capacity while preventing congestion or packet dropping due to the volume of intranet traffic. Section 4.5 discusses the extreme traffic load situations such as when the ad-hoc network is under-loaded or over-loaded.

### 4.1.3 Level of Node Mobility

The level of mobility is measured by the node speed and pause time. In highly mobile networks, a high node speed introduces a connectivity problem and makes it harder to maintain routing tables (or to process route discoveries). If mobile nodes move too fast, the routing tables (or route requests) expire so fast and the network consumes considerable resources towards locating newer routes. On the other hand, if mobile nodes move too slowly, the mobility problem becomes hidden for the test-bed observer. Therefore, the test-bed implements reasonable node mobility in order to alleviate the possibility of loosing connectivity.

The listed three major factors have been found to influence the test-bed results significantly if not chosen carefully. Therefore, it is important to monitor those factors and ensure they remain at reasonable values at all times during performance testing. That does not imply in any sense lower interest in other factors like fading of the wireless media for example. Instead, the listed three factors are not only highly influential; they can also be tuned in the simulation environment.

## 4.2 PYLON Test-bed Description

The PYLON test-bed uses the Network Simulator version two (NS-2 [123]) to simulate the mobile ad-hoc network, the access domain, and the core network. The test-bed defines the wire-line links and wireless channels between mobile nodes that move within a predefined virtual testing field. The test-bed also defines the characteristics of QoS models (SWAN in the ad-hoc, and DiffServ in the access domain), then defines PYLON-Lite to run on the gateways. The test-bed uses supplementary tools to simulate traffic generating applications. It uses a comprehensive approach to collect traffic information, consolidate, analyze, and generate statistical conclusions.

### 4.2.1 The Use of NS Simulator

NS-2 is commonly used to test and evaluate new protocols for the ad-hoc domain [17]. NS-2 is a discrete event simulator designed for networking research; it provides substantial support for simulation of routing, TCP, IP, and multicast protocols over wired and wireless (local and satellite) networks.

NS began as a variant of the REAL network simulator in 1989 [123] and has evolved substantially over the past few years. NS-2 has always included substantial contributions from researchers, including wireless code from the UCB Daedalus project [120], CMU Monarch project [121] and SUN Microsystems. Currently, NS-2 provides implementation for many ad-hoc routing protocols like AODV [87], DSR [53], OLSR [23], and others. It is also possible to find implementations for QoS models like DiffServ, and SWAN. The test-bed provided here is implemented in NS version 2.1b9a.

### 4.2.2 PYLON Test Field Topology

The geographical view of the PYLON test field is illustrated in Figure 4-1. Corner nodes $C_a$-$C_d$ are core nodes where every core node forms a distinct domain. Core nodes represent core network hosts. They have only wire-line interfaces, and their geographical location is irrelevant. The test field is a two dimensional square of 700m each side.

Nodes $G_a$-$G_d$ are edge routers (geographically fixed gateways); they have a wire-line interface to communicate with core nodes, and a wireless interface to communicate with ad-hoc mobile nodes. Gateways are located at a distance *d* from the borders of the PYLON test field. The distance *d* is calculated to be at most one-fourth the test field length in order to provide reasonable coverage to mobile nodes.

Nodes $N_0$-$N_x$ are mobile ad-hoc nodes that move in a square field at a speed of 1.0 m/s, and pause time 2 sec. All fixed and mobile nodes are assigned IP addresses that define their distinct domains. In addition, the bandwidths and delays of both wire-line and wireless links are all illustrated in Figure 4-1. The reason for selecting relatively limited bandwidths is to be able to test the model using fairly limited amounts of traffic. When increasing the amount of traffic, the NS-2 runtime increases exponentially. In a multi-domain environment like the one shown in Figure 4-1, it is advisable to decrease the simulation runtime as described at the end of Section 0.

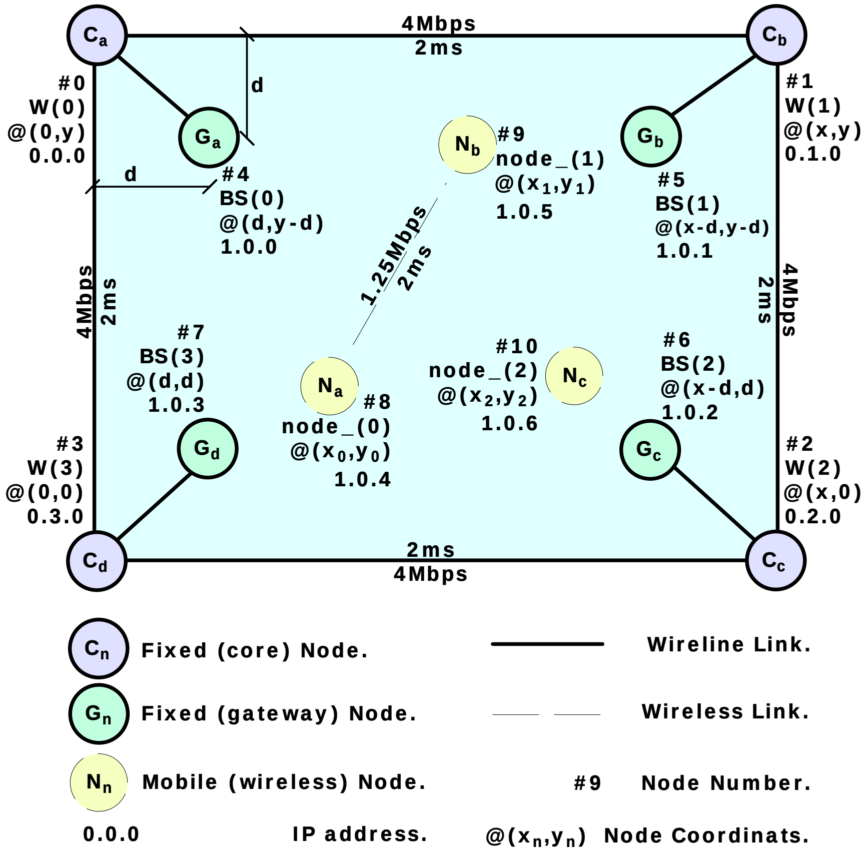


**Figure 4-1:** *Geographical View of PYLON Test Field Showing Bandwidths and Delays*

The test-bed follows a hierarchical scheme for assigning IP addresses. The general form for an IP address is *(d.c.a)*. The first digit *(d)* represents the domain number, and routers having the same *(d)* belong to the same domain. The second digit *(c)* represents the cluster number, an arbitrary sub-classification within a domain, and used here to split a single domain. The third digit *(a)* is a random address given to each router. The IP addresses are designed to be unique and to identify node hierarchy as well.

### 4.2.3 Traffic Generation

Various traffic-generating applications are used to feed the test-bed with the required traffic loads. One of the traffic generating applications uses constant bit rate (CBR) to simulate Voice over IP RT-flows. For variable bit rate RT-flows the test-bed uses a simulated video streaming (MPEG) application. Another application uses TCP connections that simulate greedy FTP with fixed packet size (1460 bytes). TCP connections generate BE-traffic that does not require service quality. The test-bed executes as many applications as required for every specific test. The amount of generated traffic is carefully monitored in all tests to maintain a specific (desired) network load as described in Section 4.1.2. From a high-level perspective, the test-bed employs Voice over IP, MPEG video streaming, and FTP.

The test-bed uses voice and video flows to simulate real-time traffic. Voice flows are generated based on CBR traffic with constant packet size of 80 bytes and frequency of 20 packets per second. The video flows are based on VBR with frequency of 20 packets per second, and variable packet size that follows uniform distribution with 512 bytes minimum and 2kbytes maximum.

The test-bed also uses FTP and HTTP connections to simulate best-effort traffic. FTP packets are generated based on fixed packet size of 1kbyte and a uniform distribution of its frequency. HTTP connections are simulated by generating random shortlived FTP connections (5-10 seconds each). Since the randomization process is seeded by the same value, each run generates the same HTTP traffic pattern. The randomization process follows the Gaussian distribution.

### 4.2.4 Wireless Media

The simulation uses parameters from the Lucent Wave-LAN card to set up radio communication using the IEEE 802.11 as MAC layer where every mobile node has a transmission range of 250 m. Mobile nodes use omni-directional antennas that are centered above the mobile node.

Mobile nodes have a carrier sense range of 550 meters [113], and interference range that follows Equation 4-1 as in Rappaport [92]:

$$R_i = d_s\,(CPThresh)^{1/4} = 1.78\,d_s \qquad \ldots (4\text{-}1)$$

In Equation 4-1, $R_i$ is the interference range, $d_s$ is the closest one-hop distance, and *CPTresh* is the ratio between the signal power, and interference power (=10).

The IEEE 802.11 standards provide data transmission rates of up to 11 Mbps. This rate has been reduced to 1.25 Mbps in the test-bed in order to test the model using fairly limited amounts of traffic. This reduction in the bandwidth and, therefore, the used amount of traffic leads to a drastic decrease in simulation runtime.

### 4.2.5 QoS Models and Routing Protocols

The PYLON test-bed runs DiffServ [9] in the access domains for the reasons described in Section 3.4.1. The test-bed configures DiffServ to run the premium service known as Expedited Forwarding (EF) [51] because it provides low delay, low loss, and low jitter all at a fixed rate (i.e. 650 Kbps). RT-packets in excess of the predefined rate are downgraded to best-effort service, and therefore, magnify the effect of the out-of-context packets.

The PYLON-Lite gateway implements a mapping function from the NDSCP eight service classes into the SWAN binary service classes, and vise versa. To prevent BE-packets from starving, the test-bed configures SWAN to support a minimum BE-bandwidth of 100 Kbps. The test-bed is set up to use the Ad-hoc On-demand Distance Vector (AODV) protocol for routing [87].

The test-bed runs SWAN [2] merely as an arbitrary example of the QoS models in the ad-hoc domain. However, the same set of tests is repeated using an ad-hoc domain running INSIGNIA and ESWAN. The results extracted when running both INSIGNIA and ESWAN are listed and analyzed in Appendix D for readers interested in this level of details. The following is a list of parameters used in the test-bed. The displayed parameter list is written in *tcl* syntax:

```
set opt(adhocRouting)  AODV       ;# AODV, DSDV, or DSR.
set opt(ifq_Len)       50         ;# max packet in ifq.
set opt(initialEnergy) 100        ;# init energy Joules.
set opt(rx_Power)      0.30       ;# receive power.
set opt(tx_Power)      0.60       ;# transmit power.
set opt(acrate)        "200Kbps"  ;# adm. control rate.
set opt(thrate)        "700Kbps"  ;# threshold rate.
set opt(band)          "1250kbps" ;# initial rate.
set opt(minband)       "100kbps"  ;# minimum rate.
set opt(ssthresh)      "100kbps"  ;# slo strt threshold.
set opt(segment)       "20Kbps"   ;# increment segment.
set opt(ds_ef)         "650kbps"  ;# DiffServ EF limit.
set opt(nAp)             4        ;# # access points.
set opt(nFn)             8        ;# # fixed nodes.
set opt(nMn)            20        ;# # mobile nodes.
```

The primary objective of running the test-bed simulation is to validate the model assumptions, and the operational difficulties in its application. All parameters described in this section are basic guidelines that may vary to fulfill specific test requirements. The test-bed can be extended or altered since it is available in source code.

## 4.3 PYLON-Lite Behavioral Trends

This section provides a detailed investigation and comparative analysis between model cases 7 and 8 to illustrate PYLON-Lite behavior in downstream scenarios. Then, it repeats the analysis to compare between model cases 5 and 6 in order to illustrate PYLON-Lite behavior in an upstream scenario. The objective is to show the behavioral trends that PYLON-Lite follows. To review the difference between PYLON-Lite model cases, please check Section 3.1 and specially Table 3-1.

In this section, the test-bed re-uses the environment parameters listed in Section 4.2. The extranet traffic is created by simulating three video connections and three voice connections. In addition, the test-bed uses one FTP and one HTTP extranet connections.

The intranet traffic is simulated by using both real-time and best-effort connections. Four video connections and four voice connections are used to simulate real-time extranet flows. Two FTP and two HTTP connections are used to simulate best-effort extranet connections. The traffic patterns used for downstream scenario is reversed and re-used for upstream scenario. The traffic generation is detailed in Section 4.2.3.

### 4.3.1 PYLON-Lite Downstream Behavior

This section discusses the PYLON-Lite behavior towards downstream traffic as defined in Section 3.1. Model cases 7 and 8 are used as a vehicle to illustrate the PYLON-Lite behavioral characteristics in a downstream scenario.

#### *4.3.1.1 Bandwidth Perspective*

The first test cases are shown in Figure 4-2 and Figure 4-3; both RT and BE-traffic are monitored at a specific gateway. The test is performed on a selected gateway with 1.25 Mbps wireless interface bandwidth and subject to both RT and BE-traffic that are controlled to produce high traffic capacity in order to apply sufficient stress testing. The bandwidths shown in Figure 4-2 and Figure 4-3 represent the total bandwidth observed at destination nodes receiving flows passing through the same gateway.

When running model case 7 (E/D/E🡐DNST), as illustrated in Figure 4-2, the system starts with low BE-bandwidth usage due to the TCP slow-start. The RT-flows benefit from this situation and consume about 590 Kbps until the BE-bandwidth increases to 305 Kbps. At that point, the RT-bandwidth shrinks to 410 Kbps. After 38 seconds of simulation time, the mobility scenario causes a sudden decrease in the RT-bandwidth that is recovered within 20 seconds. The SWAN model consumes part of the available bandwidth by serving more BE-packets up to a threshold value dynamically calculated by the SWAN Rate Controller.

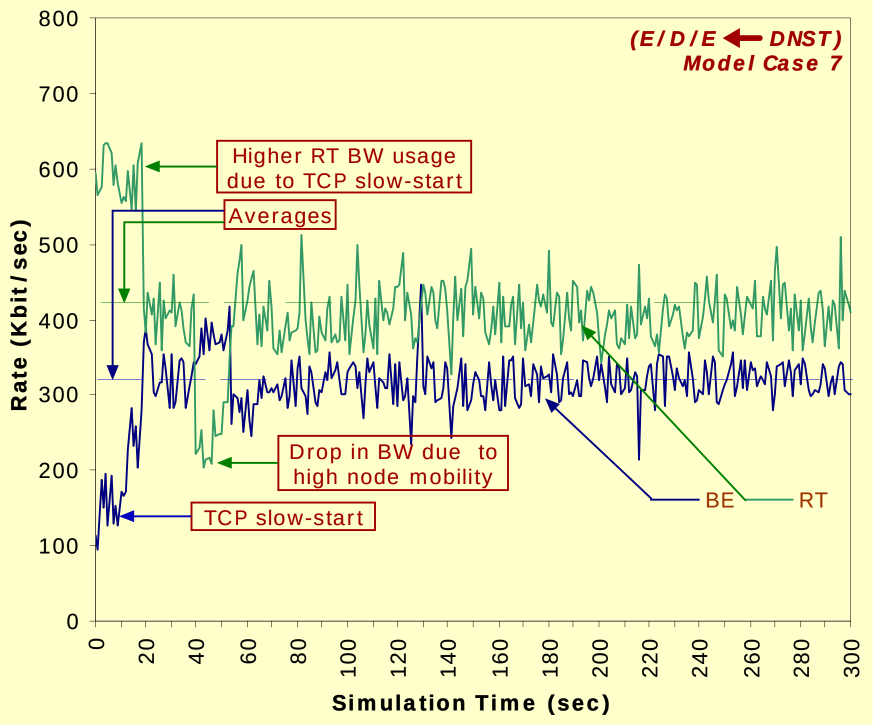


**Figure 4-2:** *Aggregated DNST Bandwidth Chart -Model Case 7- Attached to SWAN*

Model case 7 (E/D/E🡐DNST) shows the situation of downstream traffic going to a destination node in the ad-hoc domain. Since PYLON-Lite is disabled, the downstream RT-flows cannot benefit from the SWAN QoS model running in the ad-hoc domain.

Both monitored RT and BE-packets share the bandwidth that the SWAN model assigns to non-real-time traffic. Therefore, the total consumed bandwidth remains low, and dependant on the volume of ad-hoc intranet traffic as described in Section 4.1.2. The gateway bandwidth is virtually divided into 410 Kbps for RT-bandwidth, and 305 Kbps for BE-bandwidth in model case 7 (E/D/E🡐DNST), for a total of 715 Kbps. The small RT-bandwidth increase over BE-bandwidth is due to the use of DiffServ in the access domain.

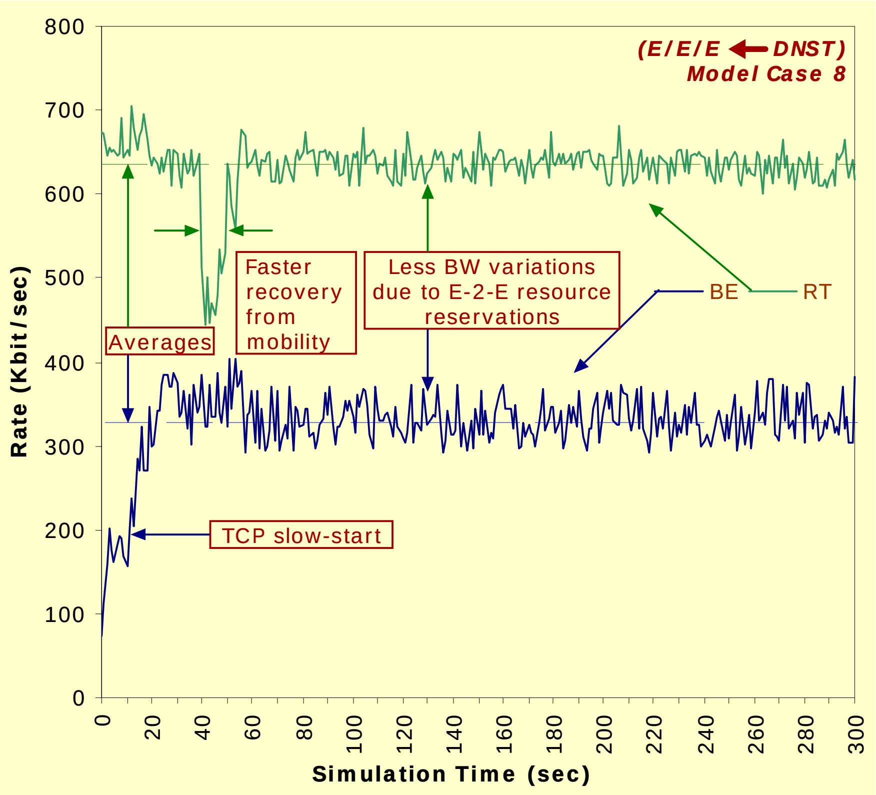


**Figure 4-3:** *Aggregated DNST Bandwidth Chart -Model Case 8- Attached to SWAN*

Studying model case 8 (E/E/E🡐DNST), as shown in Figure 4-3, illustrates the advantage of using PYLON-Lite. RT-bandwidth does not benefit much from the TCP slow-start since the RT-bandwidth is almost at its full capacity. In the same way, the drop in RT-bandwidth at 38 seconds of simulation time does not lead to a comparable increase in BE-bandwidth since the SWAN Rate Controller module controls the BE-bandwidth rate.

The gateway bandwidth is virtually divided into 630 Kbps for RT-bandwidth, and 320 Kbps for BE-bandwidth in model case 8 (E/E/E🡐DNST), for a total of 850 Kbps. Compared to model case 7 (E/D/E🡐DNST) shown in Figure 4-2, enabling PYLON-Lite provides about 54% more RT-bandwidth ( $\frac{630-410}{410}$ ), and almost the same BE-bandwidth.

Since PYLON-Lite is disabled in model case 7, RT-packets arriving to the gateway cannot gain any access to the differential treatment of RT-packets within SWAN. As illustrated in Figure 4-4 (A), SWAN downgrades both RT and BE-packets by processing them through the bandwidth assigned to non-real-time traffic only. No resource reservations or traffic marking take place. Therefore, both RT and BE-packets share the limited bandwidth dynamically defined and regulated by the SWAN Rate Controller.

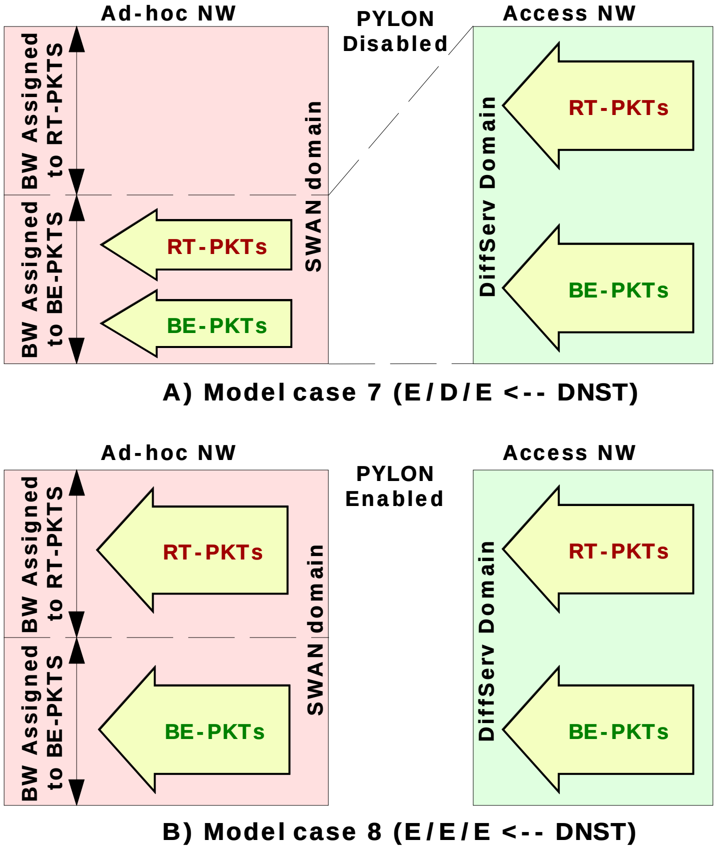


**Figure 4-4:** *PYLON-Lite Downstream Operations with SWAN.*

The dotted line separating bandwidth assigned to RT-packets from the bandwidth assigned to BE-packets in Figure 4-4 (A and B) is a little deceiving. In fact, the SWAN Rate Controller enforces a dynamically defined bandwidth threshold on a per-link basis. This threshold also varies in response to the link status by applying the AIMD algorithm as described in Section 2.3.4.

Therefore, the difference is found between the 715 Kbps (model case 7 E/D/E←DNST), and the 850 Kbps (model case 8 E/E/E←DNST). The increase in bandwidth in model case

8 is attributed to the packets being subject to smaller jitter queues, and fewer packet drops, loss, or delay as in Figure 4-4 (B).

By comparing Figure 4-2 to Figure 4-3, variations in bandwidth can be observed. Enabling PYLON-Lite results in a smoother bandwidth graph and decreases bandwidth variations from 85 Kbps to 45 Kbps when ignoring extreme bandwidth spikes.

### *4.3.1.2 Delay Perspective*

Comparing model case 7 with model case 8 from the delay perspective provides the results illustrated in Figure 4-5.

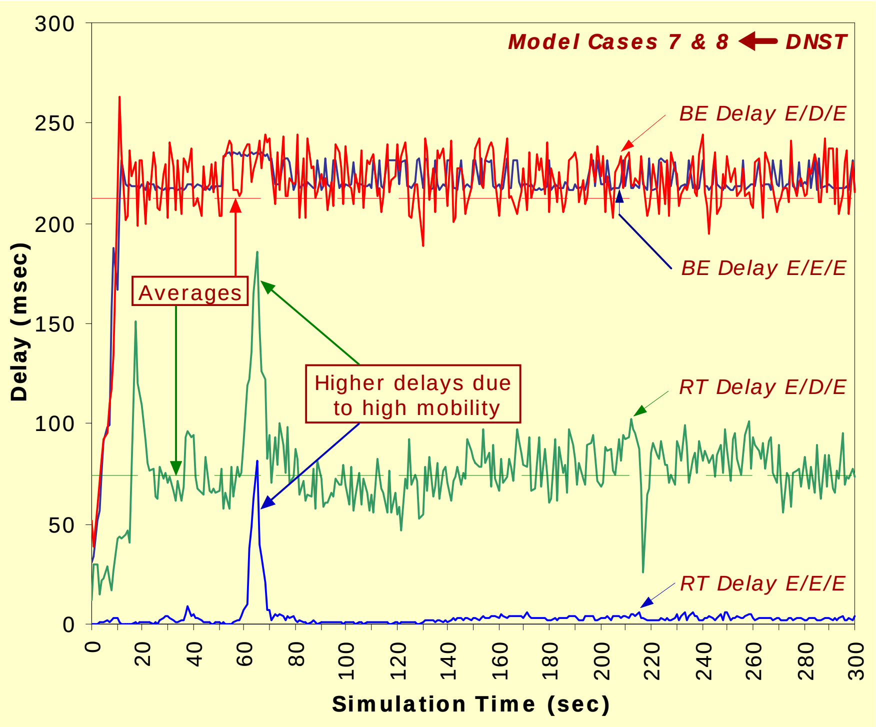


**Figure 4-5:** *Aggregated DNST Delay Chart -Model Cases 7 & 8- Attached to SWAN*

At the simulation start-up time, the TCP slow-start causes generally less traffic generated, and therefore, lowers packet delays. Ignoring the start up period, RT-delays decrease from an average of 80 msec to less than 8 msec when PYLON-Lite is enabled. In addition, average BE-delays decrease from 230 msec to 220 msec.

In other words, enabling PYLON-Lite causes a 90% decrease in RT-delays ($\frac{8-80}{80}$); this decrease is associated with a limited decrease of about 4% in BE-delays. The delay differences observed between model cases 7 and 8 can grow higher or lower depending on many factors as listed in Section 4.1. Furthermore, enabling PYLON-Lite results in much less delay variations, which leads to smaller jitter.

### 4.3.2 PYLON-Lite Upstream Behavior

Upstream extranet traffic is the traffic initiated in the ad-hoc network, and destined to a host in or beyond the access network as defined in Section 3.1. Model case 5 is compared to model case 6 to illustrate the behavioral characteristics of PYLON-Lite in an upstream scenario.

*4.3.2.1 Bandwidth Perspective*

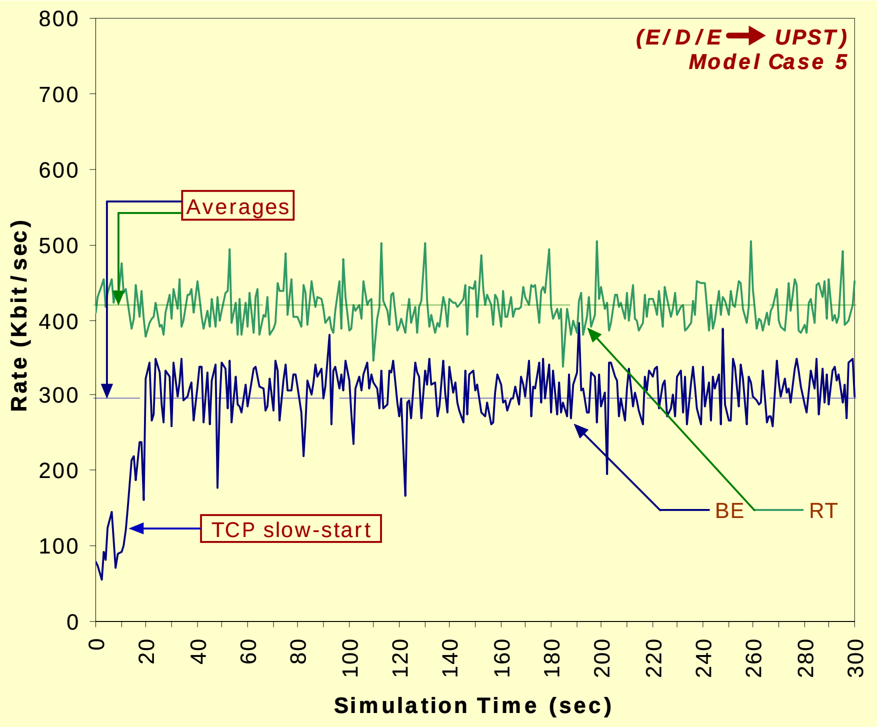


**Figure 4-6:** *Aggregated UPST Bandwidth Chart -Model Case 5- Attached to SWAN*

Figure 4-6 and Figure 4-7 illustrate the test results for model cases 5 and 6 respectively. The test follows the same guidelines used when testing model cases 7 and 8. The test uses high traffic load and maintains control on the volume of generated traffic as described in Section 4.1.2. The bandwidths shown in Figure 4-6 and Figure 4-7 represent the total of bandwidths observed at destination nodes receiving flows passing through the same gateway.

Figure 4-6 illustrates model case 5 (E/D/E→UPST), where the system experiences a slow TCP start that results in low BE-bandwidth usage. Since PYLON-Lite is disabled, no resource reservation takes place on the access domain, and therefore, RT-packets get the same treatment as BE-packets on the access domain. Both RT and BE-packets are subject to queuing delays and jitter in the access domain. Therefore, RT-flows do not benefit from the TCP slow-start, instead, RT-flows maintain about 430 Kbps until the BE-bandwidth increases to 310 Kbps.

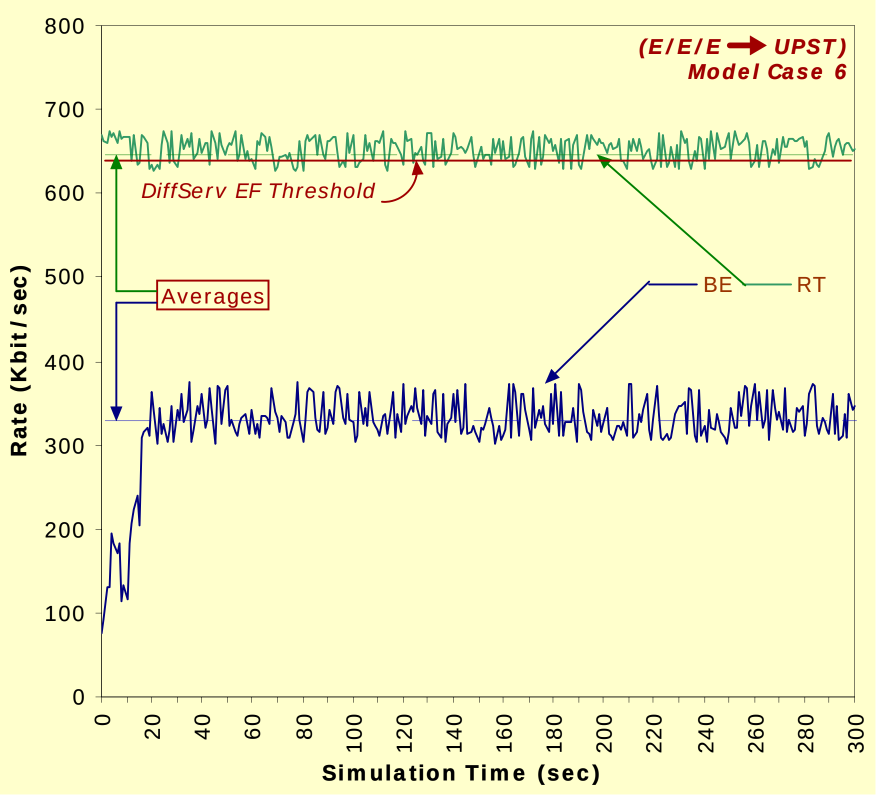


**Figure 4-7:** *Aggregated UPST Bandwidth Chart -Model Case 6- Attached to SWAN*

The relatively higher bandwidth consumed by RT-flows is attributed to the better service that SWAN provides to RT-flows. However, RT-flows suffer from a significant bandwidth drop in the access domain, and the gateway to the access domain shows high jitter and packet drop.

RT-packets get the same treatment as BE-packets over the access domain. In other words, both monitored RT and BE-packets share the bandwidth that DiffServ assigns for non real-time traffic, and therefore the total consumed bandwidth remains low. The gateway bandwidth is virtually divided into 415 Kbps for RT-bandwidth, and 300 Kbps for BE-bandwidth for model case 5 (E/D/E→UPST), for a total of 715 Kbps.

The bandwidth chart of model case 6 (E/E/E→UPST) shown in Figure 4-7 illustrates the advantage of using PYLON-Lite. Since RT-bandwidth is at almost its full capacity, it does not benefit much from the TCP slow-start. RT-flows consume slightly more bandwidth than the bandwidth assigned by the DiffServ EF threshold (i.e. 650 Kbps). PYLON-Lite downgrades the excessive bandwidth (i.e. 20 Kbps) and re-marks it as BE-packets. However, due to the relatively limited amount of downgraded packets, the RT-flows are hardly impacted in terms of average delays as illustrated in Figure 4-8.

The gateway bandwidth is virtually divided into 650 Kbps for RT-bandwidth, and 325 Kbps for BE-bandwidth for model case 6 (E/E/E→UPST), for a total of 975 Kbps. Compared to the 715 Kbps of model case 5 (E/D/E→UPST), enabling PYLON-Lite provides about 57% more RT-bandwidth ( $\frac{650-415}{415}$ ), and almost the same BE-bandwidth.

Comparing Figure 4-6 to Figure 4-7 shows a decrease in bandwidth variations for both RT and BE-traffic. Enabling PYLON-Lite results in a smoother bandwidth graph and decreases bandwidth variations from 70 Kbps to 40 Kbps when ignoring extreme bandwidth spikes.

#### *4.3.2.2 Delay Perspective*

Figure 4-8 shows the delay chart in an upstream scenario represented by comparing model cases 5 to 6. When the simulation begins, the TCP slow-start causes generally less traffic generated, and therefore, lowers packet delays. Ignoring the startup period, enabling PYLON-Lite decreases both RT and BE-delays.

In comparison, enabling PYLON-Lite causes RT-delays to drop from an average of 140 to less than 7 msec. Furthermore, BE-delays decrease from 225 msec to 210 msec. In other words, PYLON-Lite causes a 95% drop in RT-delays ( $\frac{7-140}{140}$ ); this drop is associated with a limited decrease of about 7% in BE-delays. The delay differences observed when comparing model cases 5 and 6 can grow higher or lower depending on many factors as listed in Section 4.1. In addition, enabling PYLON-Lite is associated with a decrease in delay variation, and hence, less jitter.

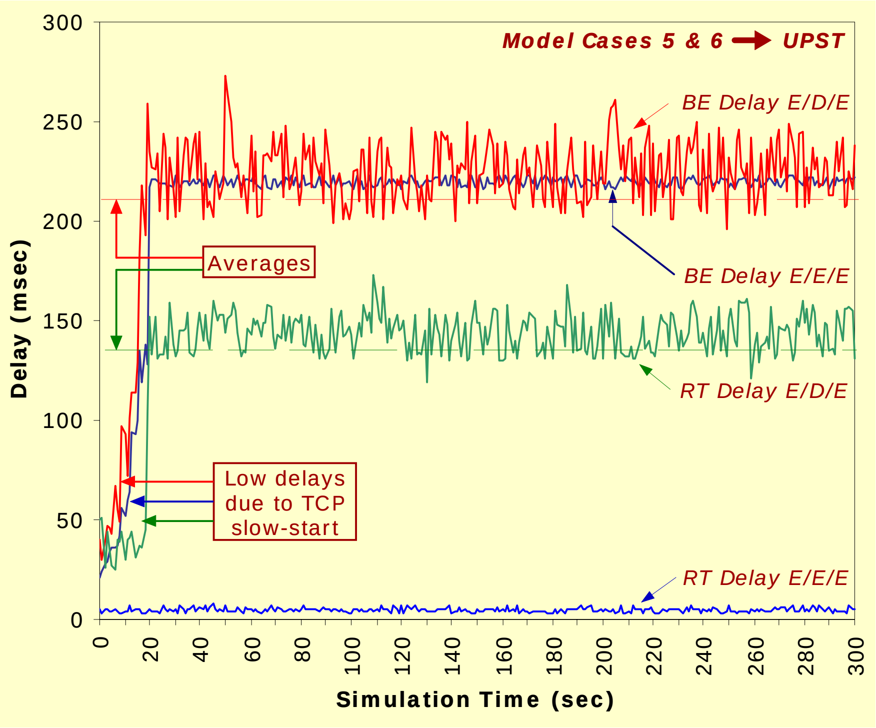


**Figure 4-8:** *Aggregated UPST Delay Chart of -Model Cases 5 & 6- Attached to SWAN*

In conclusion, this section illustrates the behavioral trends of PYLON-Lite. The performance of PYLON-Lite depends on many operational factors related to the underlying models employed on both sides of the gateway. Overall, the use of PYLON-Lite results in three consistent behavioral characteristics.

- First, PYLON-Lite increases the amount of bandwidth assigned to RT-flows while leaving the BE-bandwidth almost unchanged. PYLON-Lite achieves this by making use of the bandwidth assigned to RT-traffic by the relevant QoS model.
- Second, PYLON-Lite decreases the average RT-delay by a high factor while maintaining the average BE-delay around the same level.
- Third, PYLON-Lite decreases the variations in bandwidth and delays for both RT and BE-traffic, which leads to smaller jitter.

These three behavioral trends are observed when PYLON-Lite is tested with an ad-hoc network running SWAN model.

Due to space limitations, we will not illustrate, and analyze all the eight model cases in full detail. Two downstream cases and two upstream cases are shown here to indicate the PYLON-Lite behavioral trends. Tests of model-cases 1-4 have shown the same three behavioral trends. The testing of PYLON-Lite when attached to INSIGNIA or ESWAN also shows the same behavioral trends, and is discussed in details in Appendix D.

## 4.4 Average Bandwidth Efficiency Ratio (ABER) & ($\sigma_B$)

In order to evaluate the performance of a specific QoS mechanism, it is important to use a common normalized measure that can be used in comparative reviews and in other environments. PYLON-Lite uses the Average Bandwidth Efficiency Ratio as a common measure. ABER is the ratio of the bandwidth submitted by a source node to the bandwidth delivered to the destination node over a short period of time. ABER can be calculated as in Equation 4-2, and is introduced here as a QoS performance measure for PYLON-Lite.

$$ABER = \frac{Total\,bandwidth\,delivered\,to\,destination}{Total\,bandwidth\,submitted\,by\,source}\Big|\,over\ period\ of\ time\,T \qquad \ldots\ (4\text{-}2)$$

In lightly loaded networks, ABER equals to one. Contrary, in dynamic operational situations of highly-loaded networks, this value becomes smaller. Occasionally, the ratio may grow bigger than one due to packet de-queuing. An average ABER value closer to one indicates better network services than a smaller value.

The variations in ABER values (called $\sigma_B$) from its average over the lifetime of a RT-flow indicate the amount of used queues (jitter) and can be calculated as in Equation 4-3:

$$\sigma_B = \sqrt{\frac{\sum_{i=0}^{n}(bw_i - bw_a)^2}{n}}\Big|\,over\ period\ of\ time\,T \qquad \ldots\ (4\text{-}3)$$

Where: $n$ = Number of observations in time $T$

$bw_i$ = Bandwidth at time $i$ $bw_a$ = Average bandwidth during time $T$

Higher $\sigma_B$ values indicate higher variations in the provided service, hence, longer buffer queue. This may be tolerated by some RT-streaming applications, but cannot be accepted by interactive RT-streaming applications. The ABER and $\sigma_B$ charts together provide the normalized measures for model performance, and facilitate quantitative comparison to other models.

The test-bed and the parameters described in Section 4.3 are reused to evaluate the PYLON-Lite ABER. In the ABER test, a CBR RT-traffic is generated in a downstream scenario (← DNST) and the values of ABER are calculated throughout the simulation time in a downstream scenario when PYLON-Lite is either disabled or enabled. The results are based on a time period $T$ of 2 seconds, and are illustrated in Figure 4-9.

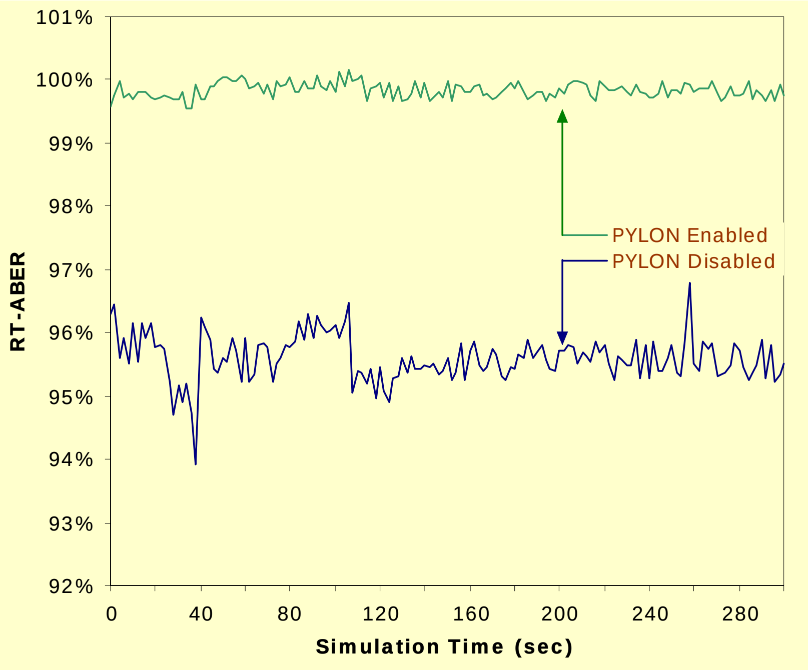


**Figure 4-9:** *RT-ABER Chart in ←DNST Scenario*

Figure 4-9 shows that the RT-flows receive higher service quality when PYLON-Lite is used. RT-ABER values are closer to 1 when PYLON-Lite is enabled. On the other hand, the RT-ABER chart shows higher variations when PYLON-Lite is disabled, mainly due to mobility on the ad-hoc side. When PYLON-Lite is disabled, RT-traffic is treated like BE-traffic, and no service differentiation takes place.

As illustrated in Figure 4-10, enabling PYLON-Lite leads to better ABER values for BE-traffic compared to a scenario when PYLON-Lite is disabled. The reason for this enhancement is that PYLON-Lite uses the bandwidth that SWAN assigns for RT-flows as well as the bandwidth that SWAN assigns for BE-traffic. This situation is described earlier in Section 4.3.1 and is illustrated in Figure 4-4.

Enabling PYLON-Lite causes the average $\sigma_B$ value of RT-flows to drop from 0.00229 to 0.00088; this decrease of 60% in RT-ABER variation leads to smaller jitter. Also the $\sigma_B$ value of BE-traffic drops from an average of 0.0028 to 0.0016 when PYLON-Lite is enabled; this decrease in the BE-ABER variation of 43% similarly leads to smaller jitter.

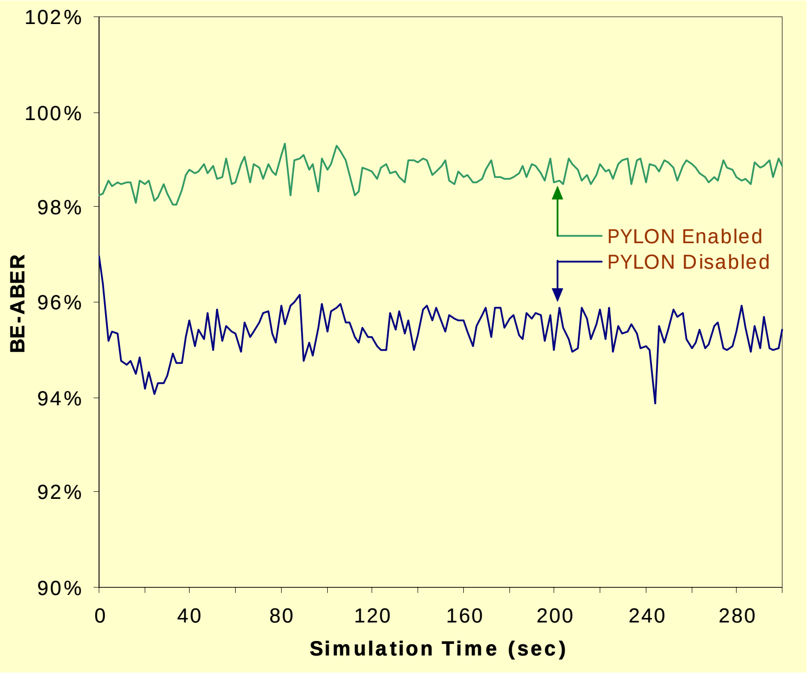


**Figure 4-10:** *BE-ABER Chart in ←DNST Scenario*

In conclusion, the use of PYLON-Lite provides better QoS as measured by the ABER values closer to unity. In addition, the use of PYLON-Lite leads to smaller jitter as measured by $\sigma_B$ values.

## 4.5 Analysis of PYLON-Lite in Extreme Traffic Conditions

PYLON-Lite has been tested in extreme traffic load conditions in order to investigate its design limitations.

### 4.5.1 PYLON-Lite in Over-loaded and Congested Traffic Conditions

Section 4.4 investigates the PYLON-Lite behavior in loaded network scenario. PYLON-Lite is shown to improve the ABER values of RT-flows in loaded traffic conditions. This section revisits the ABER chart analysis with a focus on congested scenarios in order to evaluate the behavior of PYLON-Lite in extreme situations. This section complements the analysis of Section 4.4, and reuses the same test-bed configuration, mobility, and traffic patterns in a comparison between model cases 7 and 8 (DNST). However, background traffic is increased in this test to force congestion situation at the 38th second of the simulation time. This is done by adding three short-lived voice connections (20 seconds each) to the intranet traffic load.

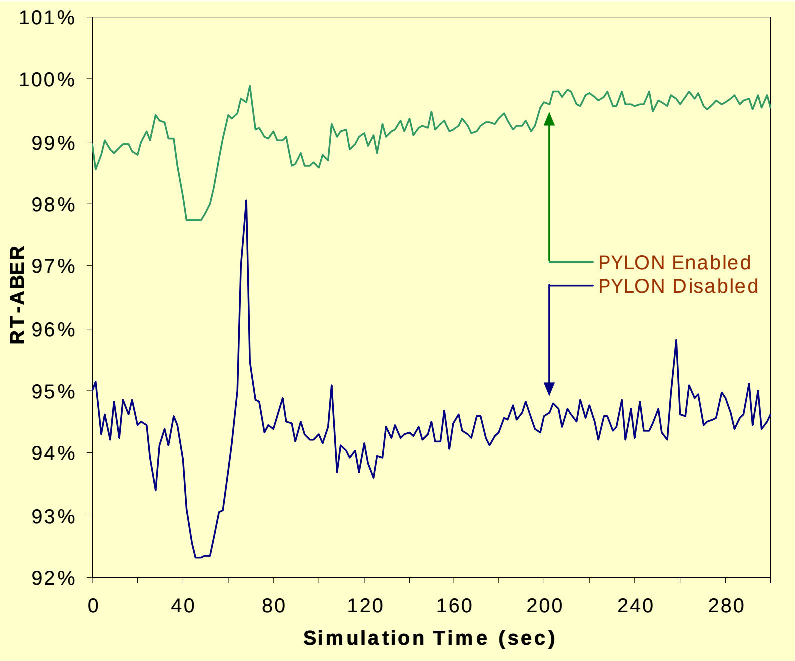


**Figure 4-11:** *RT-ABER Chart in a Congested ← DNST Scenario*

The same test procedures used in generating the results of Figure 4-9 are repeated while enforcing a congestion condition at about 38 seconds of simulation time. The CBR RT-traffic is monitored again and ABER values are calculated based on a time period *T* of 2 seconds as well. Figure 4-11 illustrates the ABER chart in the congested situation. The addition of some background traffic, in order to impose congestion, causes the average RT-ABER values to drop from almost 1.0 as in Figure 4-9 to about 0.99 as in Figure 4-11.

The situation of congested gateway illustrated in Figure 4-11 is quite different from the congestion free scenario illustrated in Figure 4-9. Figure 4-11 shows that the congestion imposed at about 38 seconds after the simulation-start causes RT-bandwidth usage to shrink in tests where PYLON-Lite is disabled or enabled. However, the recovery was a little faster when PYLON-Lite was enabled. After the congestion is resolved, the RT-ABER chart when PYLON-Lite is disabled shows a big increase followed by a drop towards its average values. The reason for the increase is the utilization of queued packets, followed by a drop in queue sizes; then the network balances again. When PYLON-Lite is disabled, the test-bed shows an average ABER close to 0.94, but also shows great variations in the RT-ABER values. The average RT-ABER value grows to a value of 0.99 when PYLON-Lite is enabled.

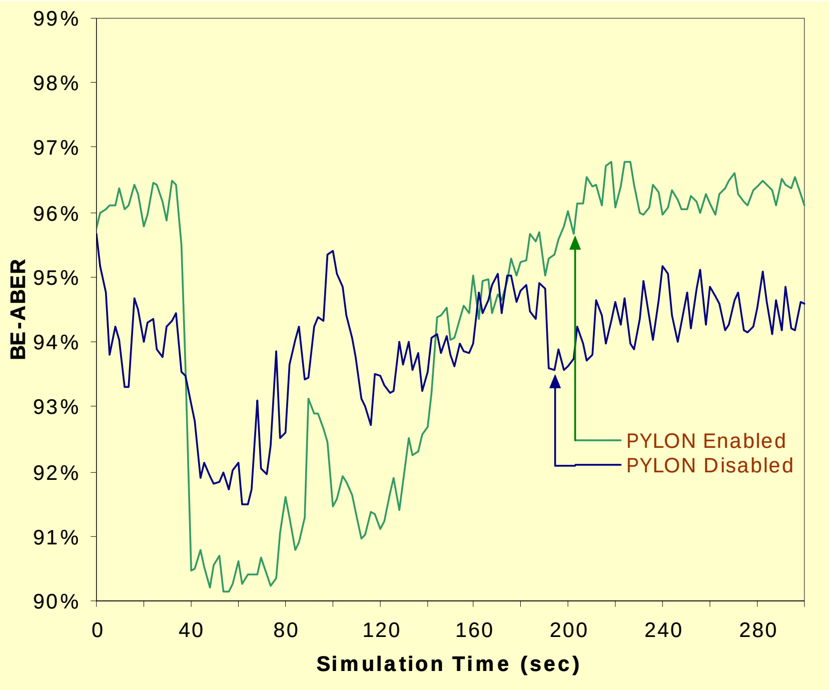


**Figure 4-12:** *BE-ABER Chart in a Congested ← DNST Scenario*

Figure 4-12 illustrates the BE-ABER chart in the same congested downstream scenario used in Figure 4-11. The congestion experienced after 38 seconds of simulation time causes the BE-ABER value to drop when PYLON-Lit is enabled or disabled. However, the BE-ABER value recovers faster when PYLON-Lite is disabled. The reason is that PYLON-Lite provides better services for RT-packets, and therefore, RT-flows recover before BE-traffic. The average value of BE-ABER when PYLON-Lite is enabled remains low, until the full impact of congestion is removed, and then, it climbs up again to a value higher than the BE-ABER values when PYLON-Lite is disabled.

The deterioration in the services provided to BE-traffic when PYLON-Lite is enabled in scenarios where the PYLON-Lite gateway experiences congestion is, relatively, acceptable. The highest levels of BE-ABER are still below the levels shown in the ABER chart in Figure 4-10. The reason is that the addition of some background traffic, in order to impose congestion, causes the average BE-ABER values to drop from almost 0.98 as in Figure 4-10 to about 0.96 as in Figure 4-12.

When the ad-hoc network experiences congestion; PYLON-Lite shows an average RT-$\sigma_B$ of 0.00129, while the same test-bed with PYLON-Lite disabled shows an average RT-$\sigma_B$ of 0.00280. In the same way, the test-bed shows an average BE-$\sigma_B$ of 0.0034 when PYLON-Lite is disabled and this value becomes 0.0030 when PYLON-Lite is enabled. This means that even in congested scenarios, implementing PYLON-Lite leads to smaller jitter buffer.

### 4.5.2 PYLON-Lite in Under-loaded Traffic Conditions

The term under-loaded condition indicates a scenario where transmitted traffic experience, potentially, limited queuing delays. This scenario is applied while comparing model case 5 (E/D/E → UPST) to model case 6 (E/E/E → UPST). The described scenario is ideal for studying the PYLON-Lite added signaling or processing overhead because the scenario removes possible background traffic influences.

In this scenario, model case 6 shows higher levels of average E2E delay compared to model case 5. The average E2E delay observations when running model case 5 are found to be *0.305711, 0.497012, 0.393403, 0.384903, 0.437181, 0.436886, 0.328938, 0.350124, 0.301903, 0.391999* msec in 10 different tests. The corresponding average E2E delay observations when running model case 6 are *0.305830, 0.497012, 0.393411, 0.384984, 0.437120, 0.436898, 0.329043, 0.350137, 0.302001, 0.392019* msec. The confidence interval for the percentage of increase in average E2E delays when using PYLON-Lite is calculated based on the 10 observation samples, and using Equation (4-4).

$$\textit{Confidence Interval} = \overline{X} \pm t_{(\frac{\alpha}{2})} \frac{\sigma}{\sqrt{n}} \qquad \ldots (4\text{-}4)$$

Where:

$\overline{X}$ = *The mean difference between the two delay observations.*

σ = *The standard deviation of the difference between the two delay observations.*

$t_{(\frac{\alpha}{2})}$ = *The upper critical value of the **t** distribution (=2.23)*

*n = Number of samples (n = 10)*

*(1-$\alpha$) = Confidence Level ($\alpha$ = 0.05)*

The 95% confidence interval is [-0.00804%, 0.00014%]. This interval show that the increase in the average E2E delays when PYLON-Lite is enabled in an under-loaded environment is statistically insignificant.

Three PYLON-Lite modules contribute to increasing E2E delays in under-loaded scenarios; namely, the Classifier, the IP-marker, and periodically the Context Controller. It is important to realize that the three processing delays depend highly on the detailed implementation and the gateway processing power. For instance, the marking process causes limited delay on a per-packet basis, and the Context Controller causes delay depending on its validation frequency. There are various optimization techniques to enhance the run-time of critical processes that can decrease the PYLON-Lite delay overhead. We argue that the advantage of using PYLON-Lite out weighs the limited processing delay increase in this specific scenario.

## 4.6 PYLON-Lite Delays

Before investigating the details of PYLON-Lite delays, it is important to review the relation between actual delay values in real life networks and simulation delays. Simulators are invaluable tools for testing algorithmic mechanisms, but provide little guidance on issues like provisioning delay values in relation to actual delays.

In the NS2 simulation environment, the simulator usually runs on a single processor. Events are fed into a scheduler that executes events in sequence. NS2 can use an approximate approach to simulate execution delays. The NS2 scheduler provides a way to attach certain processing time values to the execution of each process, and therefore, the start of the next event can be stamped with a more realistic time stamp. However, the NS2 approximate approach may still generate unrealistic expectations. A great percentage of the simulated network is implemented by simplifying reality, and therefore, it is difficult to predict the processing time of each process with sufficient accuracy. As processes cumulate to form bigger tasks, the simulated processing time grows differently from actual time.

The simulator inaccuracy influences the delay values that depend mainly on processing delays. Fortunately, in most network scenarios, the processing delays are negligible compared to queuing delays. In conclusion, it is inaccurate to rely on the exact delay values extracted from a simulation environment, specially, when the delay values depend mainly on processing delays.

In the following subsections, we provide the observed average delay values produced by the NS2 simulator as a guideline, but recommend the reader to rely on the provided formulas in each subsection to gain an accurate expectation of each delay value.

### 4.6.1 Analysis of Service Initiation Delay *($\Delta_i$)*

*Service Initiation Delay (SID) $\Delta_i$* is the time required to initiate services with a private access domain. PYLON-Lite uses Sponsor Nodes (SN) to establish services with the access domain. SID is an important performance measure, and may impact the model performance in general, especially when service upgrades and downgrades are frequent enough. In the PYLON-Lite test-bed, the value of $\Delta_i$ is found to be between 3-12 milliseconds in an upstream scenario, and between 3-7 milliseconds in a downstream scenario. As described in the introduction of Section 4.6, this value must not be taken as a reference to the delays in actual time; instead, it provides a mere guideline.

The analysis of $\Delta_i$ in this section assumes an upstream scenario because the upstream scenario associates with bigger values of $\Delta_i$. The service initiation delay $\Delta_i$ value can be divided into five basic time delays as follows:

1- Service Request Delay (SRQD)

2- Service Sponsor Solicitation Delay (SSSD)

3- Aggregate Service Reservation Delay (ASRD)

4- Service Reply Delay (SRPD)

5- Service Mechanism Setup Delay (SMSD)

The maximum $\Delta_i$ is the sum of the five delays as in Equation 4-5:

$$\max \Delta_i = SRQD + SSSD + ASRD + SRPD + SMSD \qquad \dots (4\text{-}5)$$

The minimum $\Delta_i$ shows the situation when the gateway already has an active aggregated service and the registered service can accommodate the newly requested service. In that case, minimum $\Delta_i$ can be calculated as in Equation 4-6.

$$\min \Delta_i = SRQD + SRPD \qquad \dots (4\text{-}6)$$

Each of the five delay components is detailed in the following subsections.

#### *4.6.1.1 Service Request Delay (SRQD)*

When a mobile node decides to initiate a service, it sends a service request message. The Service Request Delay is the time delay to form and deliver this message to the PYLON-Lite access gateway. The ***SRQD*** value can be formulated as in Equation 4-7.

$$SRQD = p_0 + \sum\nolimits_{0,SRCHC} (LD_x + QD_x) \qquad \dots (4\text{-}7)$$

Where:

***SRCHC*** is the hop-count from the source to access gateway.

***$p_0$*** is the processing delay required by the source node to form the *Service Request* message.

***$LD_x$*** is the delay of link *x*.

***$QD_x$*** is the delay at queue *x*.

The processing time is considered almost negligible compared to propagation time. In other words, Equation 4-7 can be approximated by removing the first term.

#### *4.6.1.2 Solicit Service Sponsor Delay (SSSD)*

The time delay used to solicit a service sponsor depends mainly on the location of the sponsor. The PYLON-Lite gateway broadcasts a solicit sponsor message, and waits for any SN to respond and sponsor the required services. Since the gateway waits for messages to take a round trip to SN, the delay can be approximated to twice the one-way delay, in addition to the delay spend in forming and processing the Solicit Service Sponsor (SSS) message and its reply. The ***SSSD*** value can be formulated as in Equation 4-8:

$$SSSD = p_1 + 2\sum\nolimits_{0,SNHC} (LD_y + QD_y) + p_2 \qquad \dots (4\text{-}8)$$

Where:

***SNHC*** is hop-count between the gateway and the SN.

***$p_1$*** is the processing delay required by the GW to form and process the *Solicit Service Sponsor* message.

***$p_2$*** is the processing delay required by the SN (GW) to form and process the *Solicit Service Reply* message.

***$LD_y$*** is the delay of link *y*.

***$QD_y$*** is the delay at queue *y*.

Since PYLON-Lite gateways always perform the role of a SN as well, the value of ***SSSD*** is limited to ***$p_1$*** and ***$p_2$*** as in Equation 4-9.

$$SSSD = p_1 + p_2 \qquad \ldots (4\text{-}9)$$

*4.6.1.3 Aggregate Service Reservation Delay (ASRD)*

The Aggregate Service Reservation Delay (ASRD) is the delay time required to perform service reservation. This delay is divided into the time required to form and process the relevant messages on the GW and the AAA server, in addition to the time required to exchange those messages between the GW and the AAA server in order to setup the QoS and authentication parameters. The ***ASRD*** value can be formulated as in Equation 4-10:

$$ASRD = p_3 + 2\sum\nolimits_{0,AAAHC}(LD_z + QD_z) + p_4 \qquad \ldots (4\text{-}10)$$

Where:

***AAAHC*** is hop-count from the gateway to the AAA server.

***$p_3$*** is the processing delays required by the GW to form and process relevant messages.

***$p_4$*** is the processing delays required by the AAA server to form and process relevant messages.

***$LD_z$*** is the delay of link *z*.

***$QD_z$*** is the delay at queue *z*.

*4.6.1.4 Service Reply Delay (SRPD)*

The Service Reply Delay (SRPD) is the delay time required to form the service reply message at the GW, to send the service message to the source node, and to process the service message at the source node. Equation 4-11 shows the ***SRPD*** formula:

$$SRPD = p_5 + \sum\nolimits_{0,SRCHC}(LD_x + QD_x) + p_6 \qquad \ldots (4\text{-}11)$$

Where:

***SRCHC*** is hop-count from the gateway to the source node.

***$p_5$*** is the processing delay required by GW to form the *Service Reply* message.

***$p_6$*** is the processing delay required by the source node to process the *Service Reply* message.

***$LD_x$*** is the delay at link *x*.

***$QD_x$*** is the delay at queue *x*.

### *4.6.1.5 Service Mechanism Setup Delay*

The Service Mechanism Setup Delay is the delay required to establish the actual service across the access network. This delay depends on the detailed implementation of the QoS model employed in the network. For example, the DiffServ model may require the establishment of MPLS tunnels, parameters propagation between ingress and egress, and configuring data collection for accounting purposes. The ***SMSD*** represents the delay required to establish those kinds of mechanisms. Optimizing ***SMSD*** is part of the administration and optimization of the specific QoS model and falls beyond PYLON-Lite focus.

### *4.6.1.6 Formula for Δi*

Using Equations 4-5 to 4-11, it is easy to put together Equation 4-12

$$\begin{aligned}\max\Delta_i &= p_0 + p_1 + p_2 + p_3 + p_4 + p_5 + p_6 \\ &\quad + SMSD + 2\sum\nolimits_{0,SRCHC}(LD_x + QD_x) + 2\sum\nolimits_{0,AAAHC}(LD_z + QD_z)\end{aligned} \quad \ldots (4\text{-}12)$$

| **Where:** | **Depend on:** |
|---|---|
| $p_0, p_6$ | The processing power of the source node. |
| $p_1, p_2, p_3, p_5$ | The processing power of the PYLON-Lite access gateway. |
| $p_4$ | The processing power of the AAA Server, the size of its directory, and the length of queued requests. |
| ***SMSD*** | The used QoS mechanism and the network configuration. |
| $\sum_{0,SRCHC}(LD_x + QD_x)$ | The hop count distance between source node and the gateway. |
| $\sum_{0,AAAHC}(LD_z + QD_z)$ | The hop count distance between the gateway and the AAA server. |

Equation 4-12 illustrates the major factors affecting the max *Δ<sub>i</sub>*. Using this equation, to decrease $\Delta_i$, it is clear that the processing delay is always much smaller compared to network and queuing delays. Therefore, max $\Delta_i$ can be approximated to Equation 4-13:

$$\max\,\Delta_i = SMSD + 2\sum\nolimits_{0,SRCHC}(LD_x + QD_x) + 2\sum\nolimits_{0,AAAHC}(LD_z + QD_z) \qquad \ldots (4\text{-}13)$$

The first term of Equation 4-13 depends on the engineering of the DiffServ mechanisms within the access domain and the network administrator can use any of the DiffServ recommendations to decrease it. For instance, the network administrator of the access domain may define MPLS tunnels to decrease the value of *SMSD*. Techniques for minimizing the value of *SMSD* fall beyond the scope of this research. The second term is hard to minimize. Since the hop-count between the source node and the PYLON-Lite gateway is controlled by the normal user mobility, it is clear that the second term is also beyond the control of the PYLON-Lite service model. In contrast, the third term in Equation 4-13 can be optimized using various techniques.

Using the same reasoning, it is easy to approximate the value of min $\Delta_i$ as in Equation 4-14.

$$\min \Delta_i = 2\sum\nolimits_{0,SRCHC} (LD_x + QD_x) \qquad \ldots (4\text{-}14)$$

In order to decrease the service initiation delay, it is important to decrease both min $\Delta_i$ and max $\Delta_i$. Typically, max $\Delta_i$ takes place when new services are initiated, and when services are upgraded. But min $\Delta_i$ takes place when existing services can accommodate the new flow. The occurrence frequency of max $\Delta_i$ vs. min $\Delta_i$ depends on the size of aggregation as defined by the Service Ladder Policy. Therefore, the steps of the Service Ladder Policy should be designed to decrease the frequency of max $\Delta_i$ as possible. On the other hand, the frequency and the value of min $\Delta_i$ are harder to control. The min $\Delta_i$ value depends on the hop count between the gateway and the source node, and this factor depends mainly on the source node location and mobility.

Designing smaller max $\Delta_i$, on the other hand, is quite possible. It is advisable that gateway administrators install the AAA server closer to the PYLON-Lite gateway, or at least create a replicated copy of the AAA server on the gateway. If both solutions are not feasible, the gateway administrator can build an MPLS tunnel between PYLON-Lite gateways and the AAA server in order to guarantee a pre-defined maximum limit on the *ASRD*. A less economic and less efficient way would be to increase the processing power of the gateway and the AAA server; however, more factors may contribute to such a decision. The analysis of the Service Initiation Delay $\Delta_i$ is essential in engineering the PYLON-Lite gateway. Gateway administrators can use this analysis to understand the factors affecting $\Delta_i$ , and to provide customized solutions for the engineering of the access network.

### 4.6.2 Delay Due to the Change of Access Gateway ($\Delta_{GW}$)

One stressful scenario to the PYLON-Lite design is when a RT-flow changes its access gateway. It is important to investigate the disruption in service quality when a RT-flow switches to a new gateway. The new gateway can be attached to the same or a different access domain. The RT-flow may experience some level of disruption as a result of changing the access gateway. This disruption can be expressed as the amount of time until RT-services at the gateway return to the required levels ($\Delta_{GW}$).

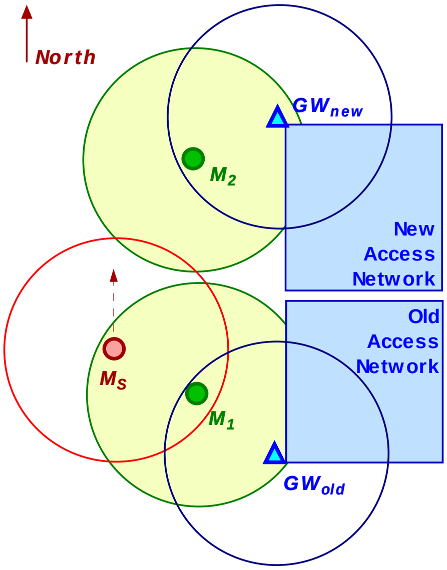


**Figure 4-13:** *Change of Access Gateway Due to Mobility*

The PYLON-Lite test-bed has generated $\Delta_{GW}$ values between 9-15 milliseconds. As described in the introduction of Section 4.6, this value must not be taken as a reference; instead, it provides a mere guideline. This section investigates the time period of service disruption during which the mobile node changes its access gateway.

For simplicity, assume a single mobile node roaming in a linear direction from being serviced by one gateway to another. As illustrated in Figure 4-13 the mobile node $M_S$ is connected to the gateway $GW_{OLD}$ possibly through an intermediate node like $M_1$. As the mobile node $M_S$ moves north, it loses connection with $M_1$ and remains disconnected from access to any gateway for a while. Then, it discovers $M_2$ which offers access to a new gateway $GW_{NEW}$. Assuming that all of $M_1$, $GW_{OLD}$, $M_2$, and $GW_{NEW}$ are willing to provide the services required by the roaming node $M_S$, and assuming that services provided by old and new access networks are identical, then we remain focused on the consequences caused merely by the mobility of $M_S$. Using such a general setup, it is easy to observe the bandwidth chart illustrated in Figure 4-14.

#### *4.6.2.1 Types of Delays*

In Figure 4-14, the CBR RT-flow is served by $GW_{OLD}$, then a disruption is caused by the mobility of $M_S$. After a short delay, the service resumes at the same CBR but through $GW_{NEW}$. The period of service disruption ($\Delta_{GW}$) can be divided into five types of time delays as displayed in Figure 4-14, Figure 4-15, and Figure 4-16.

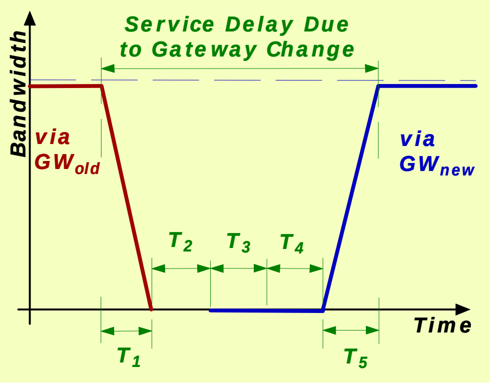


**Figure 4-14:** *Bandwidth Chart in General Mobility Scenario Causing Gateway Change*

***$T_1$*** is the time period for $M_S$ connection with $GW_{OLD}$ to completely fade. As $M_S$ moves away from $M_1$, the radio connection fades gradually, and as a result, the amount of bandwidth fades gradually as illustrated in Figure 4-14 and Figure 4-15. The value of $T_1$ depends merely on the relative speed of $M_S$ and is beyond the service model control.

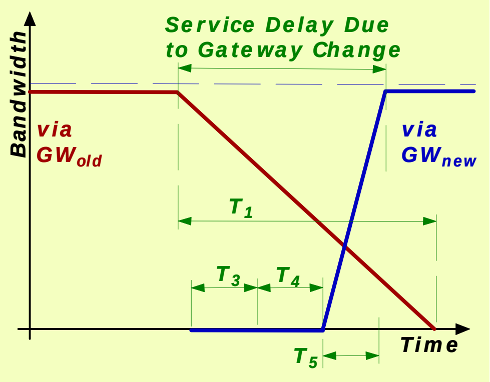


**Figure 4-15:** *Bandwidth Chart in Alternative Mobility Scenario Causing GW Change*

***$T_2$*** is the time period when the roaming node $M_S$ has no connection with any gateway. The value of $T_2$ can be zero if the ad-hoc network is highly connected, has no clusters, and has sufficient redundant links. Figure 4-15 illustrates this scenario. $T_2$ depends merely on the network connectivity and redundancy and is beyond the control of the service model.

***$T_3$*** is the time required by the routing protocol to locate a route to $GW_{NEW}$. $T_3$ depends on the routing algorithm and the hop distance between the new location of $M_S$ and $GW_{NEW}$.

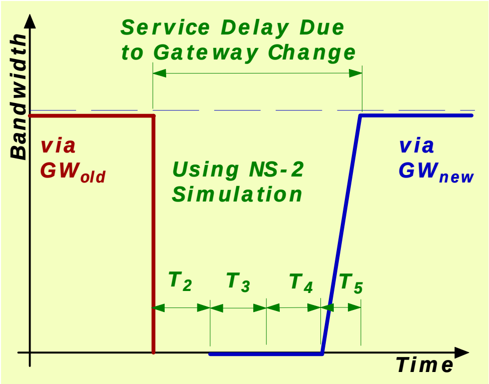


**Figure 4-16:** *Bandwidth Chart in a Mobility Scenario Causing GW Change Using NS2*

***$T_4$*** is the time required for $M_S$ to initiate new services on $GW_{NEW}$. This time is equivalent to $\Delta_i$ *(SID)* and is discussed in Section 4.6.1.

***$T_5$*** is the time delay until the first packets send through $GW_{NEW}$ resume original levels of bandwidth as illustrated in Figure 4-14, Figure 4-15 and Figure 4-16.

### *4.6.2.2 The Formula for $\Delta_{GW}$*

The analysis of the time periods $T_1$, $T_2$, $T_3$, $T_4$ and $T_5$ does not mean they have to occur in the sequence illustrated in Figure 4-14. In fact, they may take place in various sequences as illustrated also in Figure 4-15 for instance. Another example is the case when running NS2 simulations as illustrated in Figure 4-16. In that case $T_1$ is equivalent to zero since NS2 does not simulate the channel fading phenomenon. However, the maximum delay in services caused by gateway change remains the algebraic sum of all five delays as in Equation 4-15.

$$\max \Delta_{GW} = T_1 + T_2 + T_3 + T_4 + T_5 \qquad \ldots (4\text{-}15)$$

The values of time periods $T_1$, $T_2$, $T_3$ and $T_5$ can be engineered by mechanisms outside the scope of PYLON-Lite. For instance, the values of time period $T_3$ depend on the routing protocol. In contrast $T_4$ depends on the PYLON-Lite configuration as illustrated and discussed in Section 4.6.1. One advantage of the aggregated services approach in PYLON-Lite is the possibility of decreasing $T_4$.

## 4.7 Conclusion on PYLON-Lite Performance

This chapter focuses on evaluating the PYLON-Lite model. Extensive testing of different model cases revealed that the Hop Count (HC), the volume of ad-hoc intranet traffic, and the level of node mobility are the most influential factors on model performance. In order to study PYLON-Lite behavioral characteristics, a test-bed is set up based on the network simulator NS-2. The test-bed defines the PYLON test field topology, size, and geographical node locations. In addition, the test-bed defines bandwidths of wire-line links and various characteristics of the wireless channel. Traffic generating applications are installed and added to the test-bed to provide the desired level of traffic loads. The test-bed uses the SWAN implementation for NS-2 as an arbitrary QoS model running in ad-hoc networks. The test-bed also uses DiffServ with strict EF running in the access network in order to provide the access domain QoS model as defined in Section 3.4.1.

The analysis of the aggregated bandwidth charts and the aggregated delay charts in various operational scenarios identifies the following three behavioral trends.

1- Enabling PYLON-Lite increases the amount of bandwidth assigned to RT-flows by a large margin. Associated with the RT-bandwidth increase, the BE-bandwidth experiences some change, however, the amount of BE-bandwidth change is always limited. In other words, enabling PYLON-Lite leads to significant increase in RT-bandwidth associated with a limited effect on BE-bandwidth.

2- Enabling PYLON-Lite decreases the average RT-delays by a large margin associated with limited change in BE-delays.

3- Enabling PYLON-Lite results in a smoother bandwidth and delay charts for both RT and BE-traffic. The smoother bandwidth and delay represent smaller jitter, and therefore, user applications at the destination nodes receive highly consistent behavior from the network.

In further testing, detailed in Appendix D, PYLON-Lite is shown to consistently illustrate these three behavioral trends regardless of the operational scenarios, or the specific QoS implementation on the ad-hoc side. It is rather simplistic to associate the use of PYLON-Lite with certain percentages of improvements since the final performance depends on the used QoS models and on many other operational factors like mobility and network dynamics. Instead, the three behavioral characteristics show the trends that PYLON-Lite follows while the figures provide mere guidelines.

This chapter also introduces ABER as a normalized measure for the quality of delivered traffic. The enhancement in the quality of RT-flows is discussed using the ABER chart as a measuring factor. Then, PYLON-Lite is studied in extreme traffic conditions, and the ABER chart is used to investigate the deterioration in the quality of the BE-traffic in congested scenarios.

This chapter also provides a thorough analysis of important delay values in PYLON-Lite. The analysis provides useful formulas for estimating delay values, and recommendations to enhance the real-time performance of the PYLON-Lite gateway.

CHAPTER 5

# Notes on PYLON-Lite

PYLON-Lite is a pioneering QoS model that operates on the gateway between the ad-hoc and the access networks to provide homogeneous QoS to extranet traffic. This thesis provides a brief view of the related models, protocols and technologies in both ad-hoc and access domains. The thesis introduces the PYLON-Lite model and details the different components of the model, in addition to illustrating the model performance and providing a thorough analysis of the model. The PYLON-Lite research is extended to investigate various security aspects of the model. An enhancement to the classical SWAN model came out of this research and has been published to improve the use of SWAN by relying on destination based regulation.

The PYLON-Lite QoS model represents a corner stone in providing homogeneous services to extranet traffic in a heterogeneous environment. No comparable models surfaced in the literature up to this point in time. Therefore, PYLON-Lite sheds light on a problem that has been ignored to date.

We believe that PYLON-Lite is an important achievement in its area, and advances in this field will follow shortly. Section 5.1 summarizes the PYLON-Lite QoS model from a higher perspective. Section 5.2 provides a list of some important notes on PYLON-Lite and its implementation. Section 5.3 introduces future enhancements that can improve PYLON-Lite functionality.

## 5.1 Summery of PYLON-Lite QoS Solutions

The PYLON-Lite model is a QoS model that operates on the ad-hoc network gateway to the fixed topology access network. PYLON-Lite uses the principals of gateway design in concatenating service networks in order to provide homogeneous services for cross-domain traffic in a heterogeneous environment. The design of PYLON-Lite is faced with the following challenges:

1- Difficulties provisioning services in the ad-hoc domain. PYLON-Lite follows a reactive service allocation mechanism to avoid service provisioning.

2- The complexity of dealing with different QoS models employed on each side of the gateway. PYLON-Lite uses aggregate resource reservation as a common service subset.

3- The lack of per-flow policing from the ad-hoc network side. PYLON-Lite proposes the use of a Flow Policing Controller in situations where scalability is a minor concern.

4- The difficulties in generating and maintaining accounting records for mobile nodes. PYLON-Lite uses aggregated services to group the per-flow accounting into a per-class accounting.

5- The lack of processing power, storage, and the limited capabilities of mobile nodes. The PYLON-Lite default implementation is limited to the access gateway.

6- The dynamics of the ad-hoc network in terms of node mobility, and unpredictability of wireless links. PYLON-Lite employs timeout mechanisms to recover from adverse situations.

PYLON-Lite relies on the gateway administrator to implement a set of policies and rules in order to enhance the model performance. For instance, PYLON-Lite recommends the use of a Service Ladder Policy and the gateway administrator is required to make decisions about the exact service levels. In addition, the gateway administrator must make an educated compromise between policing flows and increasing scalability.

The set of decisions delegated to the gateway administrator reflect the complexity of the problem. It also adds flexibility to the model and makes it highly adaptable to various situations.

## 5.2 Notes on PYLON-Lite QoS Model

The PYLON-Lite model covers most operational scenarios and supports different QoS models on the ad-hoc side. However, it is important to strengthen PYLON-Lite with sufficient support for alternative ad-hoc network configurations and operational scenarios. The following list illustrates some areas that PYLON-Lite design may cover in the future.

- **Ad-hoc Networks That Run QoS-aware Routing:** The PYLON-Lite design has been focused on utilizing various QoS models in the ad-hoc network. Recently, the QoS research within the ad-hoc network has been a shifted towards QoS-aware routing solutions. Currently, there exist less than a handful of proposed QoS-aware routing solutions. The PYLON-Lite architectural design is flexible enough to deal with QoS-aware routing. For instance, QoS-aware routing solutions are expected to embed the probe messages within the route request messages. PYLON-Lite may use the Compatibility Module to handle the new route request messages in the same way it handles the probe request messages. However, this will require higher integration with the routing protocol running on the gateway. That level of integration is important for the future PYLON-Lite.

- **Dealing with Performance Enhancing Proxies (PEP):** A PEP [RFC 3135] is used to enhance the performance of the TCP connection over the wireless links. The PEP can enhance the user perception of the service given to TCP traffic by applying various mechanisms such as the TCP acknowledgement handling and spacing or the use of local acknowledgement. PYLON-Lite can be extended to trigger and support the use of PEPs. Due to the controversial use of PEP, this kind of extension must be controlled by the gateway administrator.

## 5.3 Areas for Future QoS Gateway Research

This section looks at future research areas that are considered major extensions to PYLON-Lite. In fact, we consider the following future research areas separate from the PYLON framework and they may result in other initiatives within the QoS gateway. The following list illustrates those research areas and highlights the importance of each one.

- **The Problem of Defining Application Needs:** This problem is common between wired and wireless networks. The deployment of most QoS solutions hinges at the end on the ability of user applications to specify their needs in a quantitative terms. Most applications are unable to provision their QoS requirements and, at best, can define a short term requirement only.

The user interaction with the application usually plays a role in specifying the actual QoS parameters. However, users expect to get the services as they interact with the application. That level of synchronization between defining user and application requirements, and performing resource allocation, is hard to achieve. A better solution can be to insert a middleware application that can monitor user behavior and traffic patters, and then provisions the QoS requirements ahead of time based on intensive statistics about user behavior.

- **Non-DiffServ Access Domains:** PYLON-Lite relies on the existence of DiffServ in the access domain in order to support aggregated services, and therefore, maintain scalability as illustrated in Section 3.4.1. However, this assumption may not always be true. There are many other types of access networks that require QoS gateways. The following list illustrates other possible types of access networks.

  - **MESH Networks:** The evolving MESH networks are very similar to ad-hoc networks in principal, but typically rely on multiple wireless interfaces within selected mobile nodes. Due to this similarity, solutions developed for ad-hoc networks can be adopted by MESH networks after applying minor modifications. However, the PYLON-Lite design needs more than just a modification in order to fit special cases of MESH networks, like the case when access gateways are also moving.

  - **Cellular Networks:** The cellular networks are built to serve mainly voice traffic. The major challenge in designing a QoS gateway for cellular networks is the fact that the design has to deal with an already built network. QoS solutions for cellular networks may rely largely on the MAC-QoS since they serve a single wireless link. The major challenge may be to design a QoS mechanism that can interact with the classical Public Switches Telephone Network (PSTN).

  Other important types of non-DiffServ networks include WLAN, ATM and many other types of wireless networks. The future QoS research areas here are not meant to be exclusive; instead, they merely illustrate our current research interests, and may change over time.

# Appendix A

# Review of QoS for Cellular and WLAN Networks

This appendix provides a brief review of the QoS approaches in both cellular and WLAN networks. The appendix focuses specifically on the cross-domain issues in both environments.

## A.1 QoS Support for Current and Future PCS Systems

*Personal Communication Services* (PCS) networks are different from the mobile ad-hoc networks partly because the PCS mobile *User Equipments* (UE) do not route traffic and UE are always one hop away from their *Base Transceiver Station* (BTS). The evolution of the cellular QoS architecture from the *General Packet Radio Service* (GPRS) to *Universal Mobile Telecommunication Systems* (UMTS) is defined by *Third Generation Partnership Project* (3GPP) release 1999 standards. Even though the air interface technology is different, both *Code Division Multiple Access* (CDMA), and *Time Division Multiple Access* (TDMA) networks can benefit from this evolution to the common QoS based networks on the GPRS model. Indeed, the future third generation cellular systems will use various radio technologies and a common access network to bring a variety of packet data services.

### A.1.1 QoS Approach in the Current GPRS Systems

GPRS associates a set of QoS parameters, referred to as a QoS profile, to each *Packet Data Protocol* (PDP) context. The QoS profile consists of five attributes: *delay*, *service precedence*, *reliability*, *mean throughput*, and *peak throughput*. The delay attribute indicates the acceptable transfer time of a packet from one edge of the GPRS system to the other edge. Whereas the delay attribute affects the scheduling order of data packets belonging to different PDP contexts, the precedence attribute indicates the drop preference during network abnormalities, and the reliability attribute indicates the tolerance for error rates and subsequent level of support needed for packet re-transmission. The mean throughput and peak throughput attributes specify the average rate and the maximum rate, respectively, of data transfer during the remaining lifetime of an active PDP context [116].

### A.1.2 QoS Approach Limitations in Current GPRS Systems

There are several limitations of the QoS approach in the current GPRS system that makes it infeasible to support real-time traffic [90]:

- Only one QoS profile can be used for a given PDP address. All the application flows sharing the same PDP context experience the same QoS defined for the PDP context, and no per-flow prioritization is possible. Furthermore, GPRS Phase 1 specifications do not allow QoS re-negotiation to be initiated by the mobile node or the *Gateway GPRS Support Nodes* (GGSN).
- The parameters of the QoS profile are vaguely specified in the standards, forcing ambiguity in interpreting implementations and thus raising inter-operability concerns. Moreover, since the GPRS radio is designed for BE-traffic, some of the parameters, such as the delay attribute, cannot be practically implemented on an E2E-basis.
- In GPRS Phase 1, the *Base Station Subsystem* (BSS) does not possess sufficient information to perform resource management for the data flows, or simply reserving resources for the higher priority flows. GPRS radio is capable of supporting BE-traffic only.

These limitations fuel the need for UMTS QoS model development.

### A.1.3 QoS Approach in UMTS

In the UMTS architecture, a bearer service defines the characteristics and functionalities established between communicating end-points. UMTS uses control plane signaling to set up an appropriate bearer that complies with the E2E QoS application requirements. Once the appropriate bearer is established, the actual bearer service support is formed by user plane transport and QoS management functions. A single mechanism is needed to bind together the QoS offered by different bearer services. The GPRS PDP context, with necessary enhancements, provides this important functionality [119].

UMTS solves the QoS problems observed in GPRS such as using per-aggregate traffic classification and the use of multiple QoS profiles for a single PDP.

### A.1.4 Cross-domain QoS in UMTS

The UMTS QoS model is designed to be independent of any external network and external QoS mechanisms. However, to ensure sufficient E2E support for QoS across domains using the existing IETF-defined QoS schemes, there is a need to provide a mapping from the external QoS to UMTS internal QoS concepts. For example, it should be possible to provide support for RSVP and Differentiated Services as defined by the IETF within the UMTS QoS model. In the UMTS architecture, this objective is met by burdening the *User Equipment* (UE) with the necessary QoS parameter mapping responsibility [57].

The UE is responsible for identifying requirements when it establishes new PDP contexts on demand with the required QoS. The basic assumption here is that the UE understands common IETF QoS mechanisms, such as RSVP, or DiffServ, and houses various applications that use them. In addition, it is easier to upgrade a UE in both software and hardware. On the other hand, the UMTS network elements do not have to understand IETF QoS mechanisms and be upgraded often. Thus, it is important that the UE be able to represent the IETF QoS requirements in a form suitable for the UMTS QoS model [57].

The UE selects the QoS parameters to comply with the E2E requirements for an application flow. In addition the UE performs all necessary mapping mapped to IETF QoS parameters. The UMTS layer in the UE terminal can then map the specified parameter values in the signaling protocol, such as RSVP, to those understood in the UMTS network. The details of mapping are left open for vendor specific implementations. These mapped values are then used as parameters in the PDP context QoS profile specification. A UMTS network can reverse the QoS mapping at the other edge of the network [57].

### A.1.5 Comments on GPRS and UMTS QoS Support

The UMTS QoS promises wider support for different classes of flows compared with the GPRS support. The UMTS QoS implementation shifts much of the mapping burdens towards the UE, which facilitates cheaper QoS implementation, and faster UMTS deployment. However, as the user traffic increases, and the user applications become more demanding to QoS support, the UMTS architecture is expected to face serious challenges. Another problem with the UMTS proposal is overloading the UE with defining and negotiating required QoS demands. This design can be too demanding when considering the limited capabilities of UE. UMTS networks need to be supported with comprehensive QoS monitoring tools to facilitate challenging and complex traffic engineering. There is some skepticism over the capabilities of the UMTS proposal in fulfilling such demands. UMTS provides QoS support for RT-traffic, and follows the classical IETF models for cross-domain QoS. GPRS does not provide better than BE-support for RT-traffic, and therefore, no cross-domain QoS model is proposed [57].

## A.2 QoS Support for Wireless Local Area Networks (WLAN)

The common architecture of a WLAN is a fixed topology infrastructure network that extends over a single hop wireless link. Users gain access to the fixed topology network via Access Points or Hot Spots.

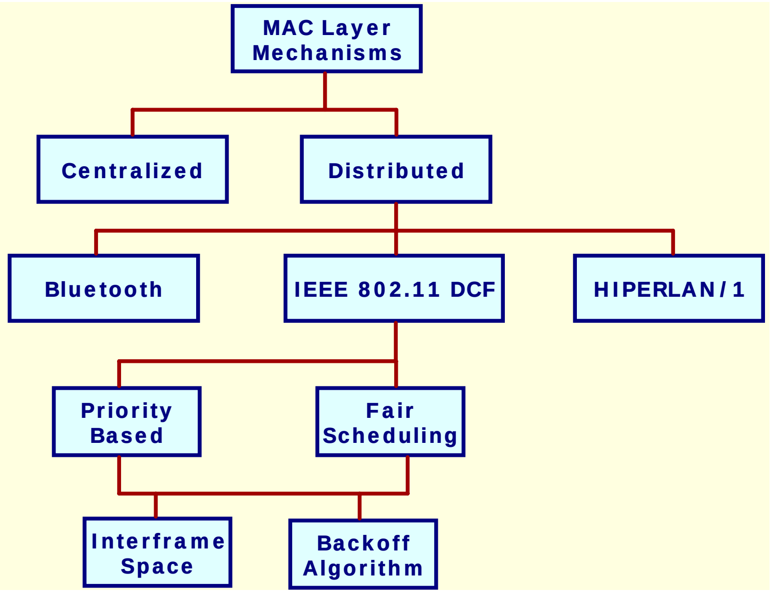


**Figure A-1:** *QoS Approaches in WLANs*

The mere use of per-link priority scheme satisfies QoS demands in WLANs. This per-link priority is typically implemented at the MAC layer. A common QoS implementation in the wireless link of the WLAN is applied over IEEE 802.11, HIPERLAN/1, and BLUETOOTH. QoS in the fixed topology WLAN follows the classical IETF approach of either IntServ or DiffServ. The problem of extending the WLAN QoS support to attached networks follows the classical IETF framework. In that sense, the QoS challenges in WLANs are consolidated to the different technologies that can apply to the wireless link.

QoS mechanisms in a WLAN can be categorized as in Figure A-1. Researchers have given the IEEE 802.11 mechanisms a higher emphasis due to its wide popularity and efficiency. Centralized QoS schemes are losing ground to distributed schemes. The popular Distributed Coordination Function (DCF), which is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) technology, is gradually taken over by the Enhanced Distributed Coordination Function (EDCF) as mentioned in Section 1.4.1. Figure A-1 illustrates the different possible technologies and focuses on the IEEE 802.11 with DCF, which is used extensively in this study.

The research literature has been flooded with various proposals for WLAN QoS in an attempt to fulfill the immediate market demand for WLANs. As an example, [30] proposes a priority scheme using IEEE 802.11 DCF, [7] proposes the use of *Tiered Contention Multiple Access* (TCMA), [58] introduces distributed fair queuing, and [52] presents a scheduling algorithm for Bluetooth. Those studies and many others can be categorized as MAC QoS approaches in addition to being WLAN QoS researches.

In conclusion, WLAN QoS approaches are focused on MAC layer solutions. WLANs do not exhibit a need for new cross-domain QoS solution; instead they follow the classical IETF framework.

# Appendix B

# QoS Support in Asynchronous Transfer Mode Protocol (ATM)

This appendix provides a brief review of the Asynchronous Transfer Mode (ATM) protocol use in the core network. The appendix illustrates the QoS support in ATM networks and the common implementation of IP over ATM. Finally, the appendix comments on the ATM interconnectivity with IP networks and reviews its QoS support.

## B.1 ATM for Data Network Back Bones

ATM was originally envisioned in the mid 1980s to operate over optical fiber facilities; the *International Telecommunication Union* (ITU-T) specifically refers to *Synchronous Optical Network* (SONET)/ *Synchronous Digital Hierarchy* (SDH). ATM is a dedicated-connection switching technology that organizes digital data into 53-byte cell units and transmits them over a physical medium using digital signal technology. Individually, a cell is processed asynchronously relative to other related cells and is queued before being multiplexed over the transmission path.

Because ATM is designed to be easily implemented by hardware, faster processing and switch speeds are possible. Speeds on ATM networks can reach 10 Gbps and higher over SONET and several other technologies. ATM is a key component of broadband ISDN (BISDN) [67].

## B.2 Principle Characteristics of ATM

The ATM protocol suite exhibits the following characteristics:

1- The ATM standards define a full suite of communication protocols from an application level to the physical layer. However, ATM operates independent of the physical layer.

2- The ATM service model includes constant bit rate service (CBR), variable bit rate service (VBR), available bit rate service (ABR), and unspecified bit rate service (UBR).

3- ATM uses packet switching with fixed-length packets (cells) of 53 bytes. Five bytes of each cell are used by the ATM cell header. The fixed-length cells with simple header facilitate higher speed.

4- ATM provides no retransmission on a per-link basis; instead, it implements an error correction scheme. When correction is not possible, the cell is simply dropped.

## B.3 ATM Protocols and Classes of Services

Figure B-1 shows the ATM protocol stack. The bottom-most layer is the Physical Layer, responsible for moving cells within the ATM network. The Physical Layer performs medium-specific functions (such as bit timing, medium characteristics, and physical connectors) as well as medium-independent transmission functions (such as framing and signaling).

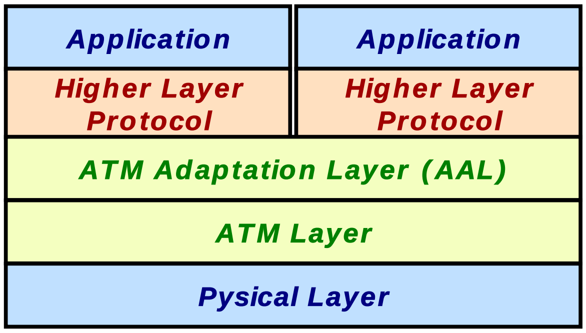


**Figure B-1:** *ATM Protocol Stack*

The ATM Layer provides a connection-oriented cell switching service, meaning that a logical connection between two ATM hosts must be in place before information may be exchanged between those two hosts. The ATM Layer is primarily responsible for the generation of the cell Header and the functions associated with the Header, including the switching and routing of cells, flow control, congestion notification, bit error detection in the Header, and cell delineation.

The Physical Layer and ATM Layer, taken together, provide the facilities for the connection-oriented transport of cells. These two protocol layers must be implemented in every ATM device, including end-user hosts and broadband switching systems

The ATM Adaptation Layer (AAL) is an end-to-end protocol that provides the interface between the ATM Layer and higher layer protocols and applications. The AAL is responsible for accepting messages from higher layer protocols and fragmenting them into smaller entities for transport in cells. It is also responsible for providing any necessary additional services that might be expected by the higher-layer application, such as timing, synchronization, sequencing, and error detection and/or correction.

| A | B | C | D |
|---|---|---|---|
| Contention-oriented | | | Contentionless |
| Constant | Variable | | |
| Delay-sensitive | | Delay-insensitive | |
| Circuit Emulation (Voice) | Video Packet | Frame Relay | SMDS, IP |
| 1 | 2 | 3/4, 5 | 3/4, 5 |

**Figure B-2:** *AAL Service Classes and Types*

It is the capabilities of the ATM Adaptation Layer that makes ATM able to support a wide variety of services. The services are defined by the ITU-T in terms of their service classes (Figure B-2), which are distinguished by a variety of communications parameters, including:

1- *Class A:* Connection-oriented, delay-sensitive, constant bit rate services, such as circuit emulation, voice, or constant bit rate video.

2- *Class B:* Connection-oriented, delay-sensitive, variable bit rate services, such as variable bit rate (compressed) video.

3- *Class C:* Connection-oriented, delay-insensitive, variable bit rate data services, such as X.25 or frame relay.

4- *Class D:* Connectionless, delay-insensitive, variable bit rate data services, such as *Switched Multi-megabit Data Service* (SMDS).

Figure B-2 also denotes the AAL type, which describes the format of the user data in the cell Payload. This format will differ based on the service class being supported. AAL Type 1 (AAL1), for example, is defined for Class A service. An AAL1 Payload contains one octet of overhead for clocking and sequencing plus 47 octets of user data. AAL2 has not yet been defined; currently, it is a place-holder for packet video services.

## B.4 Comments on the ATM Protocol

During the 1990s the Internet evolved into a huge network that facilitates commercial use. The evolution of the Internet fueled by the view of the Internet as an all-IP network resulted in limiting ATM to the core network. Currently ATM is viewed as a core network protocol that provides QoS support. The IETF influences and supports that view. IETF has issued multiple RFCs to define the IP operation over ATM networks (RFCs [13], [24], [36], and [85] and others).

ATM has a few drawbacks, for instance, it does not provide a multicast approach [67]. ATM also has a security problem when carrying IP secure packets over an ATM link. Another problem is the absence of ATM implementations on popular operating systems like Windows and Linux. In dynamic wireless environment, the ATM approach has shown less than favorable results. In contrast the ATM implementation in a less dynamic or fixed wireless networks like satellite communications shows encouraging results.

The IETF reflects the view of ATM as a core network solution and has issued RFC 1932 and RFC 1755 in support of this view. ATM provides an excellent platform for DiffServ management and QoS network engineering. The absence of per-flow controls is consistent with the DiffServ traffic class granularity. Due to all those reasons, and due to the successful marriage between the ATM and the SONET in optical networks, ATM has been pushed to the core network.

# Appendix C

# The ESWAN Destination-based Model

SWAN is a flexible QoS model for ad-hoc networks and can run over any routing protocol or MAC layers. SWAN provides some advantages over competitive models such as INSIGNIA [62], FQMM [110], or dQoS [70]. The major advantage of SWAN is implementing a distributed stateless model, which allows operation in a fully decentralized manner. However, SWAN is vulnerable to problems related to mobility and false admission. The original SWAN model discusses the two problems as part of a dynamic regulation of real-time flows, and introduces two solutions, namely source and network-based regulation algorithms, aiming to provide full congestion recovery. Both SWAN solutions apply random (or almost random) selection to victim flows, and therefore add little value to the model. ESWAN, on the other hand, improves SWAN by employing a destination-based mechanism to enhance the congestion recovery of real-time flows rather than using the source or network-based regulations. The destination-based regulation uses a biased rule to select victim flows, and adds a preemptive behavior to decrease the frequent occurrence of congestion.

Section C.1 introduces the disadvantages of the SWAN model. The appendix focuses on the problem of dynamic regulation of real-time flows in Section C.2 and the reader can refer to Section 2.3.4 for information on the SWAN components. Then, Section C.3 shows the two proposals provided within the original SWAN framework, namely source-based and network-based algorithms, and criticizes both proposals. Section C.4 defines terminologies essential to discussing the dynamic regulation and introduces the destination-based algorithm. Then, it illustrates the rational behind the destination-based algorithm, and the reason it selects a better set of victim flows. Section C.5 analyzes, and evaluates the ESWAN enhancements. Finally Section C.6 concludes the research and suggests possible future work. This appendix relies on [75] in defining and analyzing the ESWAN model.

## C.1 Introduction

QoS researchers have recognized that QoS models for the fixed Internet are inappropriate for networks with highly dynamic topology like ad-hoc networks. Researchers have proposed a number of QoS solutions for ad hoc networks. Among those proposed is the SWAN model [2] that operates over Best Effort (BE) MAC layers such as IEEE 802.11 Distributed Coordination Function (DCF) [118], and uses a stateless distributed approach to solve the dynamic QoS issues. SWAN operates in a fully decentralized manner in order to deal with the ad-hoc network dynamics. SWAN uses source admission control to limit the amount of admitted real-time flows. In response to network dynamics, which leads to occasional congestion, SWAN uses Explicit Congestion Notification (ECN) to dynamically regulate real-time traffic. Since intermediate nodes do not maintain per-flow state information, solving congestion scenarios is a bit challenging. However, maintaining the tenet of stateless model keeps the system simple, lightweight, robust, and scalable.

However, SWAN has ignored the dimension of real-time end-to-end (E2E) flow delays as a QoS measure. The SWAN framework uses per-link delays merely to detect congestion, and uses bandwidth estimates to perform admission control. E2E delays have not been considered in measuring the quality of real-time flows in SWAN. When source and destination nodes are far from each other (in terms of number of hops), real-time packets experience higher E2E delays. When the E2E delay of a real-time flow approaches a certain threshold, the flow becomes highly sensitive to network dynamics. But the destination nodes can make better judgment on the quality of received real-time flows by using E2E delays. Therefore, ESWAN proposes a destination-based algorithm to solve the dynamic regulation problems.

## C.2 Dynamic Regulation Issues

SWAN introduces dynamic regulation mechanisms in response to conditions raised by network dynamics like node mobility, and false admission. It is important to illustrate the impact of both issues on network resources.

### C.2.1 Mobility

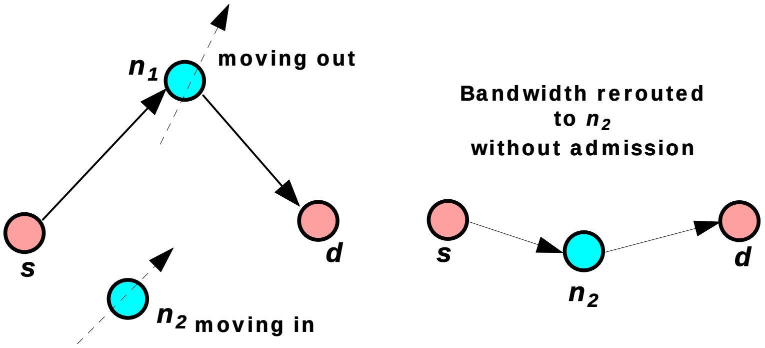


**Figure C-1:** *Congestion/Overload Due to Mobility*

As illustrated in Figure C-1, real-time flows between nodes *s* and *d* can be redirected from node $n_1$ to node $n_2$ due to mobility, and the underlying routing algorithm will perform the necessary rerouting. However, node $n_2$ will experience an increase in real-time traffic even though it did not perform any admission process to allow the new flow. If this rerouting causes congestion, it is called congestion due to mobility.

### C.2.2 False Admission

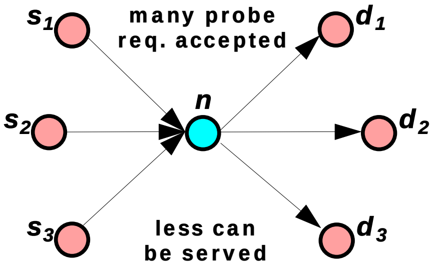


**Figure C-2:** *Congestion/Overload Due to False Admission*

As illustrated in Figure C-2, nodes $s_1$, $s_2$, and $s_3$ may initiate a probe request to send real-time flows to nodes $d_1$, $d_2$, and $d_3$ (respectively) through node *n*. If node *n* processes the three requests within a short time (i.e. before real-time packets start arriving at *n*), the admission controller at node *n* will accept the three flows even if it practically has room for only one flow. This is due to the lack of resource reservation in SWAN. Until real-time packets consume available bandwidth, node *n* will always admit new real-time flows. This situation is referred to as congestion due to false admission.

It is important to realize that mobility, and false admission conditions merely represent two issues among other issues related to network dynamics. For instance, deterioration in radio link quality may occur due to interference, introduction of a barrier, or due to diminishing battery life. The term network dynamics commonly refers to issues related to mobility, radio quality, and distributed operation. Network dynamics can cause congestion condition at mobile nodes.

SWAN adopts the ECN regulation algorithm to recover from congestion conditions caused by network dynamics. Since nodes are continuously (and independently) monitoring their bandwidth utilization, nodes can detect violations. Congested nodes use the ECN bits in the IP header of the real-time packets to inform destination(s) of the existence of congestion. Each destination node issues a regulate message to the relevant source node. Then, source nodes re-initiate new probe requests to locate newer, possibly better, routes to the destination or terminate the flow due to lack of resources.

## C.3 Common Dynamic Regulation Solutions

The decision of congested nodes to mark packets with ECN is very critical because flows that get marked with ECN may lose their QoS privileges. SWAN has proposed two regulations, namely source and network-based regulations. Both approaches mark ECN packets differently, but follow the same consequences afterwards.

1- **Source-based Regulation:** In source-based regulation, a congested node marks all RT-flows with (congestion experienced) CE using the ECN bits. When destination nodes encounter packets with CE bit marked, they send a regulate message to the associated source nodes. Source nodes immediately perform multiplicative decrease on relevant RT-flows. As a result, the congested node experiences a gradual decrease in the amount of RT-traffic until the congestion condition is removed, at that point, the intermediate (previously congested) node stops marking CE bits. If the used bandwidth of a specific RT-flow is unsatisfactory to its source node, it has to backup a random amount of time, and then re-initiate a probe request to re-establish the desired level of service, possibly on a different route. Source nodes have to stagger the re-initiation in order to avoid a flash-crowd condition where nodes may fall into false admission again, therefore, the random backup time is essential. A source node may perform a biased re-initiate flow towards newly admitted real-time flows if it can keep information about newly admitted flows.

Source-based regulation forces all RT-flows going through one congested node to be regulated. This approach seems to be too aggressive. It forces too many flows to be regulated even if the amount of bandwidth violation is limited. In addition, it does not discriminate between different RT-flows. Following that approach, some limited-quality RT-flows might maintain connectivity, while other higher-quality RT-flows might be denied service re-initiation or unnecessarily get disrupted.

2- **Network-based Regulation:** In a network-based regulation, a congested node selects a subset of all its real-time flows to be a ″congestion set″ or ″victim flow set″. The congested node marks packets associated with victim flows only. It is possible for a congested node to distinguish a specific set of RT-flows by applying a simple hash function without any need to keep flow information. Packets of victim RT-flows will reach relevant destination nodes marked with CE, and then the network-based approach follows the same process as described for source-based regulation. If a congested node does not experience any decrease in the amount of real-time traffic after a period of time *T* seconds, it calculates a new set of victim flows. SWAN suggests applying some intelligence at the congested node in order to select the set of victim flows. For instance, if source nodes inject RT-flows with RT-old or RT-new, using the IP-ToS field, congested nodes can use a biased function to form the set of victim flows out of newer flows, hoping to decrease false admission.

Network-based regulation selects the victim flows set randomly, and in the best case, discriminates against newly admitted flows. However, the idea of loading the IP-ToS field may conflict with flows that need to use this field, especially when the flow extends over the Internet as in [74].

## C.4 Destination-based Regulation

The Enhanced SWAN with destination-based regulation (ESWAN for short) evolved from our observation about SWAN's behavior in response to above-average traffic load. By above-average traffic load we mean at least one third of available bandwidth is consumed by real-time traffic; and one third by best-effort traffic. The delay histogram chart in Figure C-3 shows that the majority of RT-packets experience a delay of less than 35 msec in the above-average traffic load condition.

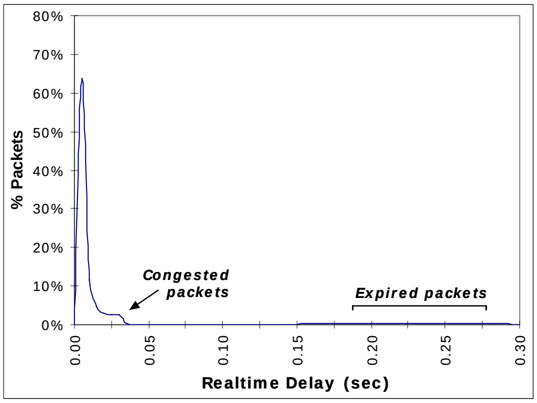


**Figure C-3:** *SWAN RT-delay Histogram*

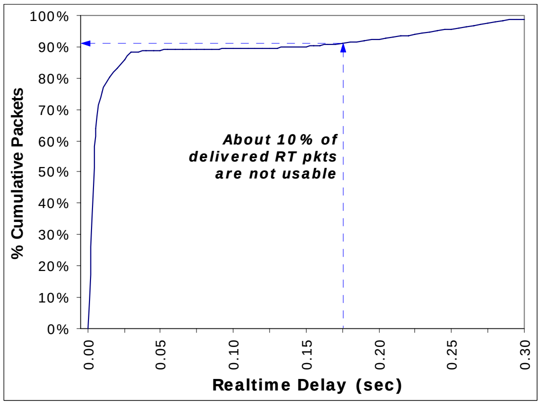


**Figure C-4:** *Cumulative RT-delay %*

However, a considerable percentage of packets appear to experience delay higher than 175 msec. This is clear on the cumulative graph in Figure C-4, where about 9% of RT-packets experience delays beyond the 175 msec. Interactive VoIP flows, for instance, will disregard packets with delays beyond a certain threshold (150 msec) (called expired packets for short). When repeating the same test for various mobility scenarios, the SWAN model consistently caused about 9% of the RT-packets to expire (9.04%, 9.64%, 9.97%, 9.72%, 10.03%, and 9.87%). Therefore, the bandwidth consumed by highly delayed packets (expired bandwidth for short) is a bandwidth that unnecessarily consumes network resources, and degrades services provided to other RT-flows. Optimizing network resource utilization requires decreasing the amount of expired bandwidth.

### C.4.1 Basic Definitions

Destination-based regulation relies on destination nodes to detect an increase in expired bandwidth, and regulate each flow accordingly. This preemptive behavior is followed to prevent wasting network resources. In case of congestion, the destination-based regulation selects a subset of the congested flows in order to regulate. This subset is selected based on flow quality starting with the lowest quality flows first. The quality of each flow is measured as a function of its packet delays.

**Maximum acceptable packet delay (MAPD)** is the threshold packet delay value (in sec) that will certainly result in the destination application ignoring packets of a specific flow. Therefore, MAPD is a flow specific value, known to the destination node, and can be compared to the E2E packet delay.

**Expired packets** are real-time flow packets that exhibit a delay more than MAPD. In the same way, **expired bandwidth** is the bandwidth consumed by expired packets. Also (*effective bandwidth = received bandwidth – expired bandwidth*) Therefore, **effective bandwidth** is the bandwidth realized at a destination node and that can be used by a destination application to replay the real-time flow over a period of time *T*.

**Limited QoS** is the QoS a network provides to a flow where the required bandwidth is more or less delivered, but the effective bandwidth perceived by a destination node over a period of time *T* is hardly sufficient for the application to effectively replay the RT-flow.

**Effective bandwidth ratio (EBR $\beta$)** is the percentage of effective to received real-time bandwidth at a destination node for a specific real-time flow over a period of time *T*. Since the bandwidth ratio ($\beta$) is based on effective bandwidth $0 \leq \beta \leq 1$. Figure C-5 and Equation C-1 illustrate the relationship between $\beta$, and limited QoS definitions.

$$EBR\ \beta = \left.\frac{effective\ BW}{received\ BW}\right| over\ time\ T \qquad \textbf{… (C-1)}$$

*EBR* ($\beta$) measures the quality of a RT-flow where values closer to *1* indicate high flow quality, and values closer to *0* indicate limited flow quality and inefficient bandwidth usage. The RT-flow-specific EBR values ($\beta_H$, $\beta_L$) represent desired measures of real-time flow quality, where values lower than $\beta_L$ represent a waste in network resources that require regulation. The probe request message is used to communicate both $\beta_H$, and $\beta_L$ values to the destination node.

**Effective delay ratio (EDR $\delta$)** is the percentage of average effective packet delays at a destination node to MAPD, over a period of time *T*. Since delay ratio ($\delta$) is based on effective bandwidth only, $0 \leq \delta \leq 1$.

$$EDR\ \delta = \left.\frac{avg.\ eff.\ pkt\ delay}{MAPD}\right| over\ time\ T \qquad \textbf{… (C-2)}$$

Effective delay ratio measures the quality of the flow. *EDR* values closer to *0* indicate higher QoS, while *EDR* values closer to *1* indicate a flow that is suffering from high delay averages, but the quality is still acceptable, as described in Equation C-2.

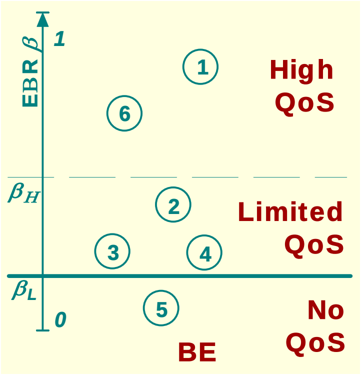


**Figure C-5:** *(EBR $\beta$) Service View in a Loaded Intermediate Node*

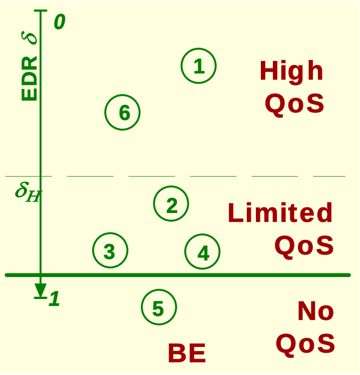


**Figure C-6:** *(EDR $\delta$) Service View in a Loaded Intermediate Node*

As illustrated in Figure C-5 and Figure C-6, a real-time flow like flow 1 (denoted by circle 1) is a flow that is getting better than required bandwidth, and therefore has high QoS. Flows 2, 3, and 4 are having limited QoS since they are getting the required bandwidth, but their effective bandwidth is hardly at the required limit. Flow 5 is a flow demoted to non-QoS (i.e. best-effort), and is not seen by intermediate nodes as a distinct RT-flow any more. Also flow 1 has a better quality than flow 6, and the difference can be expressed by the flow values of $\delta$ and $\beta$.

### C.4.2 Preemptive Behavior

The destination-based algorithm is based on two behaviors. First is the preemptive behavior, which monitors the provided service quality by the network, and requests service upgrades if the provided service is unsatisfactory. Second is the recovery behavior, which is triggered when intermediate nodes experience congestion, by regulating limited QoS flows before regulating higher QoS flows.

Destination nodes perform preemptive behavior on flows experiencing limited QoS without detecting a congestion condition. Simply, if a sufficient number of packets arrive at a destination node with packet delays higher than *MAPD*, over a period of time *T* the destination node detects a limited QoS condition ($\beta < \beta_L$), and issues a regulate message to the relevant source node. The source node then triggers a re-initiate procedure to locate, possibly, another route with better quality.

### C.4.3 Recovery Behavior

Recovery behavior is also performed by a destination node and implements the following mechanism:

1- If an intermediate node is experiencing congestion, it marks all RT-packets with CE using the ECN bits. The marking of packets will continue until the intermediate node realizes a sufficient decrease in the arriving bandwidth.

2- Destination nodes with $\delta \geq \delta_1$ will have to issue a regulate message immediately.

3- Other destination nodes will wait for time $T$, if packets keep arriving marked with CE, then destination nodes with $\delta \geq \delta_2$ will have to issue a regulate message immediately.

4- Over time, congestion gets resolved by removing flows with higher relative delays $\delta$ first. Values of $\delta_i$ are constants for the network, and have to be selected such that $\delta_i > \delta_{i+1}$.

This mechanism enables different destination nodes to independently regulate related real-time flows by regulating lower quality flows first. If congestion is not resolved, flows having slightly better quality are regulated, until congestion is resolved.

## C.5 Evaluation and Analyses

This section provides a general view of the ESWAN performance evaluation and analysis. A full evaluation of the ESWAN is available in [75].

### C.5.1 Cumulative Packet Delay

Figure C-7 and Figure C-8 show the histogram and cumulative distribution of RT-packet delay in ESWAN and can be compared to Figure C-3 and Figure C-4. In ESWAN less than 1.2% of the delivered RT-packets were found expired. When repeating the same experiment for various mobility scenarios, the ESWAN model consistently caused less than

1.2% of RT-packets to expire (1.13%, 1.16%, 1.17%, 1.15%, 1.16%, and 1.14%). Destination nodes use this limited percentage to monitor the services provided by the network and force regulation when necessary.

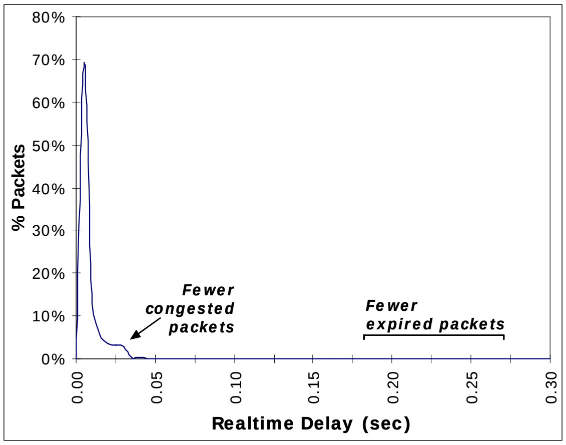


**Figure C-7:** *RT-delay Histogram in ESWAN*

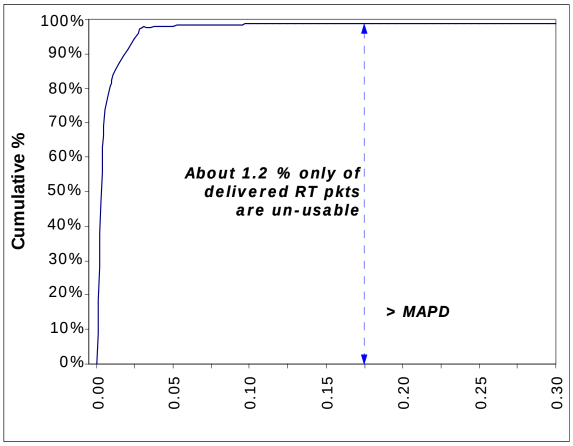


**Figure C-8:** *Cumulative RT-delay% -ESWAN*

Comparing the figures observed in this test with the corresponding results from Section C.4, the ESWAN model has consistently decreased the percentage of expired RT-packets by (7.91%, 8.48%, 8.80%, 8.57%, 8.87%, and 8.73%). In order to calculate the Confidence Interval for the series, we apply Equation C-3.

$$\textit{Confidence Interval} = \overline{X} \pm t_{(\frac{\alpha}{2})}\frac{\sigma}{\sqrt{n}} \qquad \ldots \textbf{(C-3)}$$

Where:

$\overline{X}$ = *The mean difference between SWAN and ESWAN observations.*

σ = *The standard deviation of the difference SWAN & ESWAN observations.*

$t_{(\frac{\alpha}{2})}$ = *The upper critical value of the* ***t*** *distribution (=2.45)*

*n = Number of samples (n = 6)*

*(1-α) = Confidence Level (α = 0.05)*

The confidence interval for the percentage of decrease in expired RT-packets when using ESWAN compared to when using SWAN is calculated based on the 6 observation samples. The 95% confidence interval is [8.21%, 8.91%]. As this interval does not include 0, the performance improvement by ESWAN is statistically significant, even with our somewhat limited sample size of 6. This percentage gain in RT-packets has a significant impact on the delivered RT-quality and on the user perception.

### C.5.2 Effect of Mobility

Node mobility is an important factor in the design and evaluation of ad-hoc based technologies. The speed of mobile nodes and their pause time are commonly used attributes to define mobility. The test-bed uses a pause time of 2 seconds, and when changing the pause time, both SWAN and ESWAN showed little changes in observed behavior. When running the same test-bed with node speeds of 10, …, 50 meter per second, both SWAN and ESWAN maintained the same level of average packet loss as illustrated in Figure C-9 up to a node speed of about 35 meter per second.

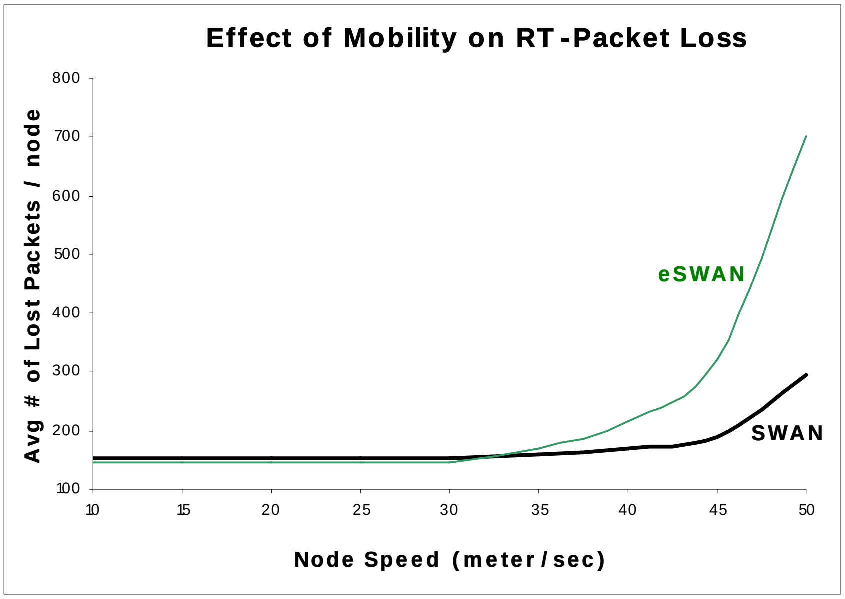


**Figure C-9:** *The Effect of Node Mobility on Average Packet Loss*

When mobile nodes move faster than 35 meter per second, deterioration in radio link quality takes effect. ESWAN shows a higher number of packet losses and the losses grow faster compared to SWAN. The reason is the preemptive behavior in ESWAN, which responds to the limited QoS perceived at destination nodes by forcing too many re-initiate probe requests flooding the relevant routes and causing congestion, and packet loss. By contrast, SWAN relies on re-routing, which is sufficient in high mobility scenarios.

Therefore, ESWAN is recommended in installations involving limited mobility (i.e. ≤ 35 m/s). This is a minor restriction since the threshold speed here is beyond vehicular speed limits (i.e. ≤ 125 km/hr).

### C.5.3 Overall Evaluation

In order to investigate the behavior of EBR ($\beta$), and EDR ($\delta$) ratios, an increasing traffic load is applied to a five mobile nodes test-bed, and the total consumed bandwidth is measured, normalized over a period of time *T* sec. The mobile nodes are forced to a no mobility condition, and the values of $\beta$ and $\delta$ ratios of a VoIP flow are observed against the increasing RT-traffic load of the network.

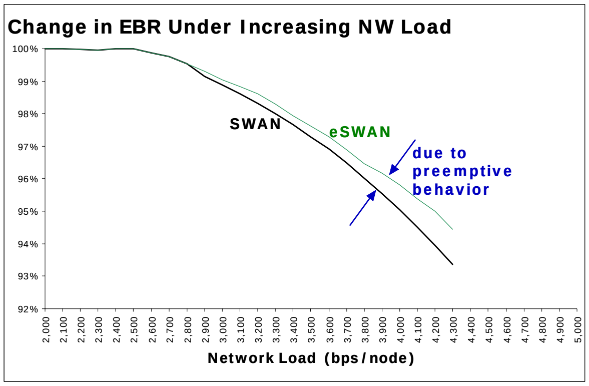


**Figure C-10:** *Network Load Effect on EBR*

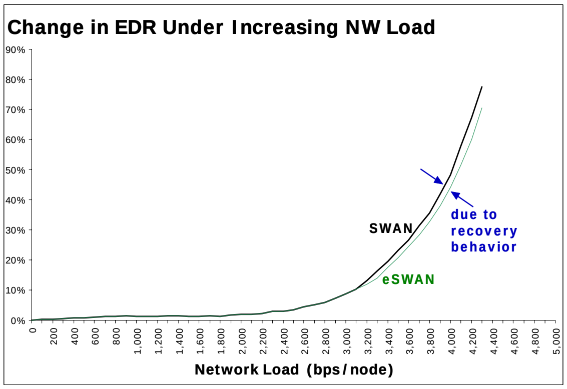


**Figure C-11:** *Network Load Effect on EDR*

Figure C-10 and Figure C-11 represent the results under these conditions. In both figures, the horizontal axes (average load per node) represent the normalized collective bandwidth consumed by all five nodes for RT-flows. Therefore, the exact values of the network RT-load will vary based on the test-bed topology, flow directions, setup, and configurations; however, the shape of the curves will remain the same.

Figure C-10 illustrates the impact of increasing overall network RT-load on the EBR $\beta$. Due to the preemptive behavior, ESWAN tends to show higher EBR $\beta$ values than SWAN. EBR $\beta$ values lower than 95% are regulated by ESWAN, and the re-initiation of RT-flows provides either higher EBR $\beta$ value, or the RT-flow will be denied service, and hence, have no EBR $\beta$.

Figure C-11 illustrates the impact of increasing overall network RT-load on the EDR $\delta$. When network RT-load increases, the effective average packet delay increases, hence the EDR $\delta$. There are relatively lower values for EDR $\delta$ on ESWAN than SWAN due to the recovery behavior. High values of $\delta$ (> 70%) are commonly associated with congestion, while low values (< 10%) are associated with healthy RT-flows.

## C.6 Comments and Conclusion

Packets traveling over larger ad-hoc networks are likely to experience longer delays since they travel over more hops. Enforcing a MAPD threshold, using EDR $\delta$, enables the network to limit expired bandwidth, which releases part of the traffic load, and ultimately increases both bandwidth availability and effective use of RT-bandwidth. This enhancement comes at the expense of BE-traffic that experiences relatively higher average delays, but has only a minor influence on the BE-bandwidth. The preemptive behavior smoothes the resource utilization over time and decreases chances of congestion. In addition, it enables destination nodes to monitor the actual level of service, and request a service upgrade when the provided service is unsatisfactory. The recovery behavior of the destination-based approach provides slower recovery when compared to the source-based approach introduced by SWAN. A gradual recovery is effective in disrupting a lower number of RT-flows, and provides less average variation in RT-delays. In addition, ESWAN decreases the amount of expired RT-packets by about 8.5%, which represents a significant improvement compared to the original SWAN implementation.

ESWAN is used in this research as an alternative model to SWAN and INSIGNIA [62]. Appendix D extends the evaluation of PYLON-Lite and investigates its inter-operability with ESWAN. The full details of the ESWAN model, and its evaluation can be found in [75] for interested readers.

# Appendix D

# The PYLON-Lite Performance and Behavioral Characteristics When Attached to Alternative Ad-hoc QoS Models

In order for PYLON-Lite to operate with the current ad-hoc QoS models and to accommodate future ad-hoc QoS development, PYLON-Lite is designed as a generic model that runs attached to any ad-hoc QoS model. PYLON-Lite does not adopt a specific resource allocation mechanism in the ad-hoc network; instead, it uses its Compatibility Module to trigger the specific admission control of the QoS model running in the ad-hoc network and to perform resource reservation. The test-bed presented in Chapter 4 uses SWAN merely to illustrate the PYLON-Lite performance and behavioral characteristics.

The PYLON-Lite architecture uses the Compatibility Module to limit the efforts required when configuring the model to operate with alternative QoS models on the ad-hoc side. The implementation of the Compatibility Module involves many details, however, the experience gained when working with the first model (SWAN) has been useful and sometimes reusable towards the second model (INSIGNIA).

To show the PYLON-Lite generic nature, it is important to repeat the performance analysis using alternative ad-hoc QoS models. ESWAN and INSIGNIA provide a competitive alternative to SWAN and can be used as a vehicle to demonstrate the PYLON-Lite generic nature. This appendix presents the PYLON-Lite performance and behavioral characteristics when attached to ESWAN or INSIGNIA as alternative QoS solutions to SWAN.

To facilitate easy comparison with Section 4.3, the PYLON-Lite performance analysis in this appendix is organized in the same manner used in Section 4.3. In addition, the same network loads and mobility scenarios employed in Section 4.3 are reused in this appendix to analyze a PYLON-Lite gateway attached to ESWAN and INSIGNIA.

This appendix is divided into two major sections, the first section investigates the PYLON-Lite behavior when attached to ESWAN; the second part investigates the PYLON-Lite behavior when attached to INSIGNIA. Each of the two parts is organized in the same way. The two sections start by investigating the results collected from a downstream scenario using model cases 7 and 8. We compare the bandwidth charts resulting from the two model cases, and then we compare delay charts and comment on the RT and BE-traffic services. The same investigations performed on downstream traffic are repeated on upstream traffic using model cases 5 and 6. Finally the appendix summarizes the results in a tabular form and it ends with observations and conclusions.

## D.1 PYLON-Lite Attached to ESWAN Ad-hoc Domain

ESWAN is an extension of the SWAN model that adds two behaviors, namely preemptive and recovery behaviors, as illustrated in Appendix C. The preemptive behavior monitors the service provided by the ad-hoc network and uses the QoS reporting mechanism to respond to service degradation. The recovery behavior enhances the SWAN dynamic regulation mechanism by improving congestion recovery. As a result of the two behaviors, ESWAN is relatively less vulnerable to network dynamics compared to SWAN [75]. Due to the algorithmic similarities between SWAN and ESWAN, the PYLON-Lite behavior depicted in this appendix matches closely with the observations on the PYLON-Lite behavior when attached to the SWAN model as illustrated earlier in Section 4.3. This section reuses the same parameters illustrated in Section 4.2 and used in Section 4.3.

### D.1.1 Downstream Behavior

Downstream traffic is the extranet traffic initiated by a host located outside the ad-hoc network and destined to a node within the ad-hoc network as defined in Section 3.1. Model cases 7 and 8 are used as an example of a downstream scenario to illustrate the PYLON-Lite behavioral characteristics.

*D.1.1.1 Bandwidth Perspective*

The test of model cases 7 and 8 followed the same guidelines used in Section 4.3. The resulting bandwidth charts are shown in Figure D-1 and Figure D-2 and represent the total of bandwidths observed at destination nodes receiving flows passing through the same gateway.

Figure D-1 illustrates the bandwidth chart depicted when running model case 7 (E/D/E←DNST). Similar to our analysis in Section 4.3, the BE-bandwidth usage starts low due to the TCP slow-start. The RT-flows benefit from this situation and consume about 590 Kbps until the BE-bandwidth grows to 313 Kbps. At that point, the RT-bandwidth shrinks to 420 Kbps. After 38 seconds of simulation time, the mobility scenario causes a sudden decrease in the RT-bandwidth that is recovered within 20 seconds. The ESWAN model uses part of the available bandwidth towards more BE-packets up to a threshold value dynamically calculated by the ESWAN Rate Controller. This behavior matches the PYLON-Lite behavior when running attached to the SWAN model at the ad-hoc side and the results shown here can be compared to the results illustrated in Section 4.3.

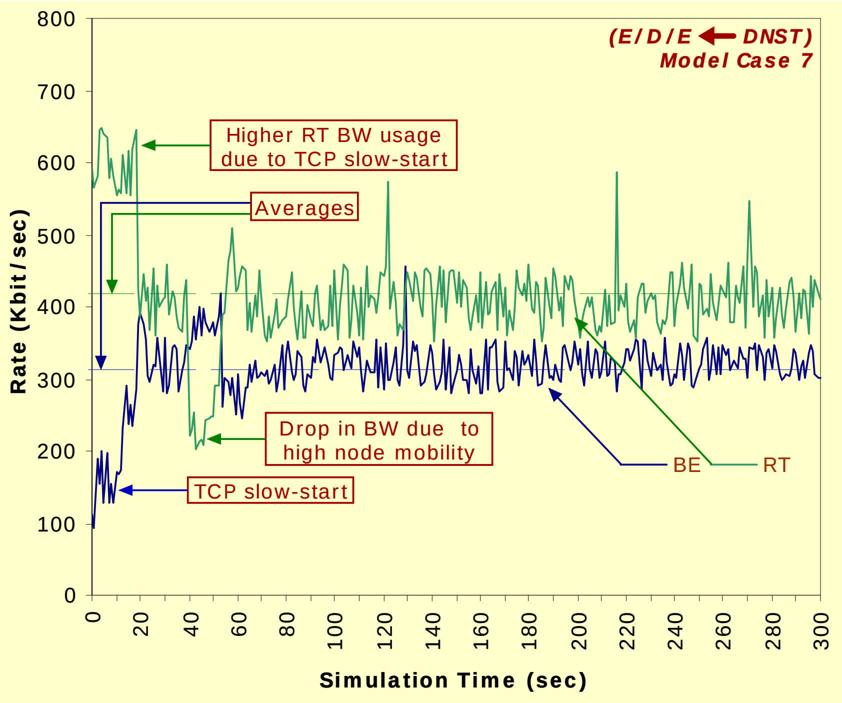


**Figure D-1:** *Aggregated DNST Bandwidth Chart -Model Case 7- Attached to ESWAN*

ESWAN assigns a limited amount of bandwidth for non-real-time traffic. Both monitored RT and BE-flows share the bandwidth assigned to non-real-time traffic when PYLON-Lite is disabled. Therefore, the total used bandwidth remains low, and depends on the volume of ad-hoc intranet traffic as described in Section 4.1.2.

The gateway bandwidth is practically divided into 420 Kbps for RT, and 313 Kbps for BE-bandwidth in model case 7 (E/D/E←DNST), for a total of 733 Kbps. The small RT-bandwidth increase over BE-bandwidth is due to the use of DiffServ in the access domain.

Figure D-2 shows model case 8 (E/E/E←DNST) in which the RT-bandwidth does not benefit much from the TCP slow-start since the RT-bandwidth is almost at its full capacity. In the same way, the drop in RT-bandwidth at 38 seconds of simulation time does not lead to a comparable increase in BE-bandwidth since the ESWAN Rate Controller module regulates the BE-bandwidth.

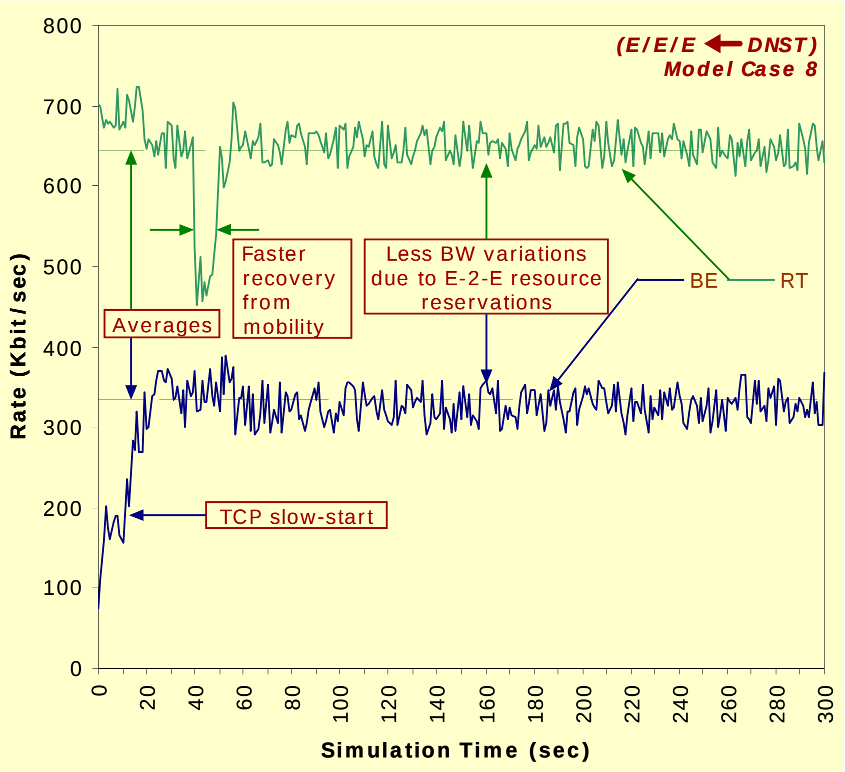


**Figure D-2:** *Aggregated DNST Bandwidth Chart -Model Case 8- Attached to ESWAN*

The gateway bandwidth is practically divided into 650 Kbps for RT-bandwidth, and 320 Kbps for BE-bandwidth in model case 8 (E/E/E←DNST), for a total of 970 Kbps. Compared to model case 7 (E/D/E←DNST) shown in Figure D-1, enabling PYLON-Lite provides about 55% more RT-bandwidth ( $\frac{650-420}{420}$ ), and almost the same BE-bandwidth.

The same concept described for SWAN and illustrated in Section 4.3.1.1, applies to the ESWAN model as well. In a downstream scenario, when PYLON-Lite is disabled, both RT and BE-packets share the bandwidth ESWAN assigns for non-real-time-traffic. By comparing Figure D-1 to Figure D-2, variations in bandwidth can be observed. Enabling PYLON-Lite results in a smother bandwidth graph and decreases bandwidth variations from 85 Kbps to 40 Kbps when ignoring extreme bandwidth spikes.

### *D.1.1.2 Delay Perspective*

Figure D-3 illustrates the results of comparing model case 7 with model case 8 from the delay perspective.

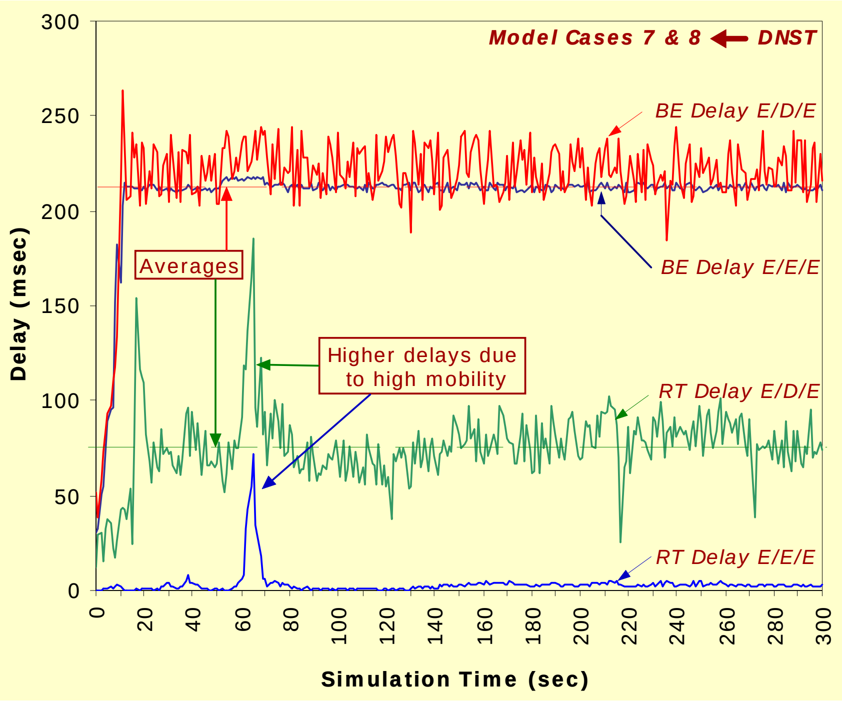


**Figure D-3:** *Aggregated DNST Delay Chart -Model Cases 7 & 8- Attached to ESWAN*

When the simulation starts, the TCP slow-start causes less BE-traffic generated, and therefore, smaller RT and BE-delays. Ignoring the start up period, RT-delays decrease from an average of 75 msec to less than 6 msec and BE-delays decrease from 220 msec to 210 msec when PYLON-Lite is enabled. In other words, enabling PYLON-Lite causes a 92% decrease in RT-delays ($\frac{6-75}{75}$); this decrease is associated with a limited decrease of about 4% in BE-delays. The delay differences observed between model cases 7 and 8 can grow higher or lower depending on many factors as listed in Section 4.1. Also enabling PYLON-Lite provides much less delay variations, which leads to smaller jitter.

### D.1.2 Upstream Behavior

Upstream extranet traffic is the traffic initiated in the ad-hoc network, and destined to a host in or beyond the access network as defined in Section 3.1. Model cases 5 and 6 are used as an example to illustrate the behavioral characteristics of PYLON-Lite in an upstream scenario.

*D.1.2.1 Bandwidth Perspective*

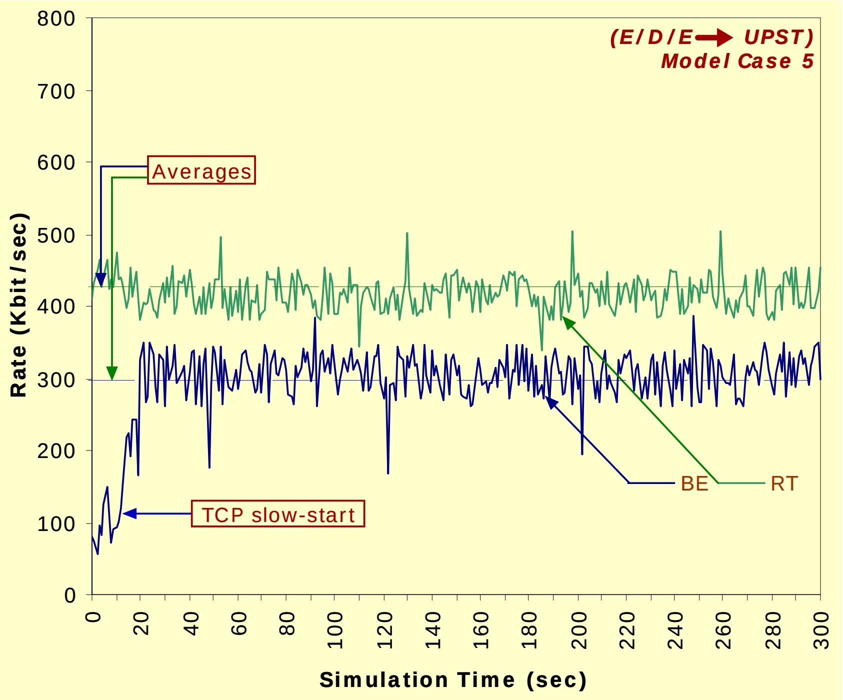


**Figure D-4:** *Aggregated UPST Bandwidth Chart -Model Case 5- Attached to ESWAN*

The test of model cases 5 and 6 followed the same guidelines used when testing model cases 7 and 8. The resulting aggregated bandwidth charts are shown in Figure D-4 and Figure D-5 and represent the total of bandwidths observed at destination nodes receiving flows passing through the same gateway.

Similar to the analysis in Section 4.3, Figure D-4 illustrates model case 5 (E/D/E→UPST), where the system experiences a slow TCP start that results in low BE-bandwidth usage.

Because PYLON-Lite is disabled, no resources are allocated on the access domain. Hence, RT-packets are treated like non-real-time-packets through the access domain. Both RT and BE-packets experience queuing delays and jitter through the access domain. Therefore, RT-flows do not benefit from the TCP slow-start; instead, RT-flows maintain bandwidth of about 430 Kbps until the BE-bandwidth increases to 310 Kbps.

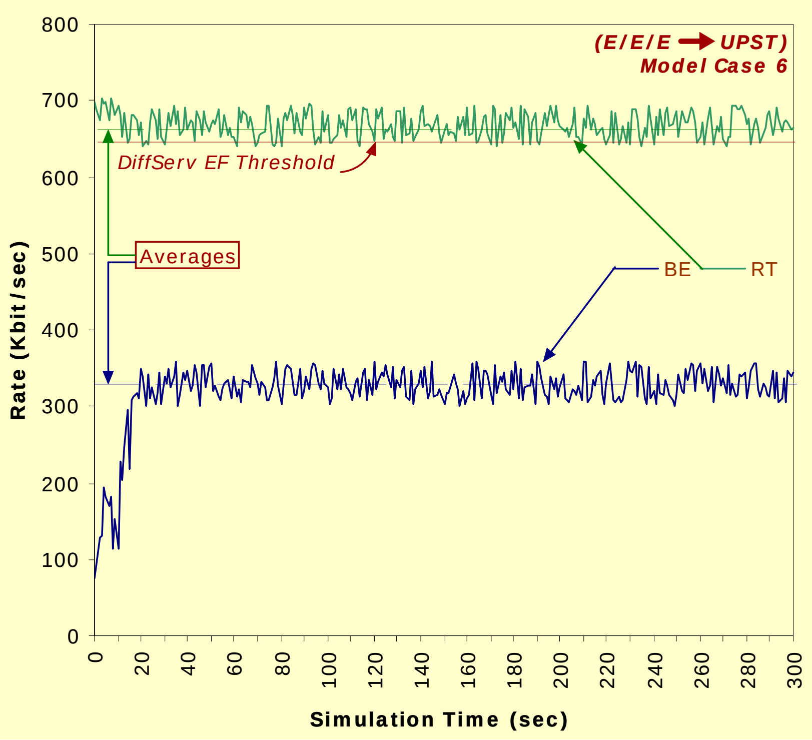


**Figure D-5:** *Aggregated UPST Bandwidth Chart -Model Case 6- Attached to ESWAN*

Due to the ESWAN differential treatment of RT-flows, the bandwidth chart in Figure D-4 shows slightly higher RT-bandwidth than BE-bandwidth usage. However, the gateway to the access domain shows high jitter and packet loss which causes RT-flows to suffer from bandwidth drop in the access domain. The gateway bandwidth is practically divided into 420 Kbps for RT-bandwidth, and 300 Kbps for BE-bandwidth in model case 5 (E/D/E→UPST), for a total of 720 Kbps.

Figure D-5 shows the bandwidth chart of model case 6 (E/E/E→UPST). RT-bandwidth is at almost its full capacity, and therefore, can not benefit much from the TCP slow-start. RT-flows occupy slightly more bandwidth than assigned by the DiffServ EF threshold (i.e. 650 Kbps). DiffServ downgrades the excessive bandwidth (i.e. 20 Kbps) in response, and re-marks it as BE-packets. Since the amount of downgraded packets is relatively limited, the average delay of RT-flows remains almost unaffected as illustrated in Figure D-6.

The gateway bandwidth is practically divided into 670 Kbps for RT-bandwidth, and 320 Kbps for BE-bandwidth for model case 6 (E/E/E→UPST), for a total of 990 Kbps. Compared to the 720 Kbps of model case 5 (E/D/E→UPST), enabling PYLON-Lite provides about 60% more RT-bandwidth ( $\frac{670-420}{420}$ ), and almost the same BE-bandwidth.

Comparing Figure D-4 to Figure D-5 shows a decrease in bandwidth variations for both RT and BE-traffic. Enabling PYLON-Lite results in a smother bandwidth graph and decreases bandwidth variations from 70 Kbps to 37 Kbps when ignoring extreme bandwidth spikes.

### *D.1.2.2 Delay Perspective*

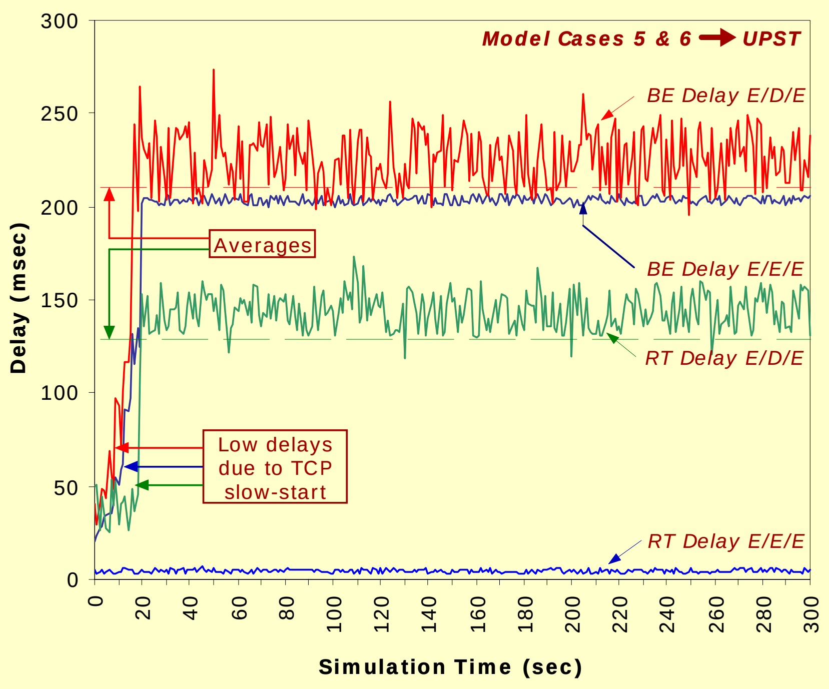


**Figure D-6:** *Aggregated UPST Delay Chart -Model Cases 5 & 6- Attached to ESWAN*

The delay chart in an upstream scenario represented by model cases 5 and 6 is shown in Figure D-6. At the beginning of the simulation, the TCP slow-start causes generally low BE-traffic loads, and therefore, smaller delays. If the startup period is ignored, enabling PYLON-Lite decreases both RT and BE-delays.

In comparison, enabling PYLON-Lite causes RT-delays to drop from an average of 140 to less than 6 msec. Furthermore, BE-delays decrease from 220 msec to 195 msec. In other words, PYLON-Lite causes a 96% drop in RT-delays ($\frac{6-140}{140}$); this drop is associated with a limited decrease of about 13% in BE-delays. In addition, enabling PYLON-Lite results in a decrease in delay variation, and hence, less jitter. Factors influencing the delay reduction observed between model cases 5 and 6 are listed in Section 4.1.

In conclusion, PYLON-Lite maintains its common behavior characteristics when attached to ESWAN rather than SWAN. PYLON-Lite raises the amount of bandwidth assigned to RT-flows while leaving the BE-bandwidth almost unchanged. PYLON-Lite decreases the average RT-delay by a high factor while maintaining average BE-delay around the same level. PYLON-Lite also decreases the variations in bandwidth and delays for both RT and BE-traffic, which leads to smaller jitter.

## D.2 PYLON-Lite Attached to INSIGNIA Ad-hoc Domain

In order to facilitate easy comparison with the results illustrated in Section 4.3, the PYLON-Lite performance and behavioral characteristics here follow the same sequence used in Section 4.3. In addition, the same network loads, and mobility scenarios used when testing SWAN and ESWAN are used in testing INSIGNIA. INSIGNIA implements a state-full QoS model that requires admission control and uses allocated resources for admitted flows. This section reuses the same parameters illustrated in Section 4.2 and used in Section 4.3. However, the DiffServ EF threshold is set to 550Kpbs in this section. The reason is that INSIGNIA limits the traffic rate arriving to the gateway compared to SWAN and ESWAN. In order to facilitate fair comparison, the EF threshold limit must be adjusted.

```
set opt(ds_ef)         "550kbps"  ;# DiffServ EF limit.
```

### D.2.1 Downstream Behavior

This section shows the PYLON-Lite behavior towards downstream traffic as defined in Section 4.1. Model cases 7 and 8 are used as an example to illustrate the behavioral characteristics.

### *D.2.1.1 Bandwidth Perspective*

The same PYLON-Lite behavior found when running model case 7 (E/D/E←DNST) while attached to SWAN and ESWAN is also repeated while attached to INSIGNIA. As illustrated in Figure D-7, the system starts with low BE-bandwidth usage due to the TCP slow-start. The RT-flows benefit from this situation and consume about 620 Kbps until the BE-bandwidth increases to 355 Kbps. After 38 seconds of simulation time, the mobility scenario causes a sudden decrease in RT-bandwidth until it reaches 245 Kbps, which is recovered within 20 seconds. The mobile nodes react by serving more BE-packets, and BE-bandwidth soars to 510 Kbps during this short period.

RT-flows suffer from node mobility and RT-bandwidth remains low until INSIGNIA repairs local QoS routes. BE-traffic benefits from this period and more BE-packets are served. However, after INSIGNIA repairs local QoS routes, BE-bandwidth shrinks back, and RT-bandwidth maintains its levels.

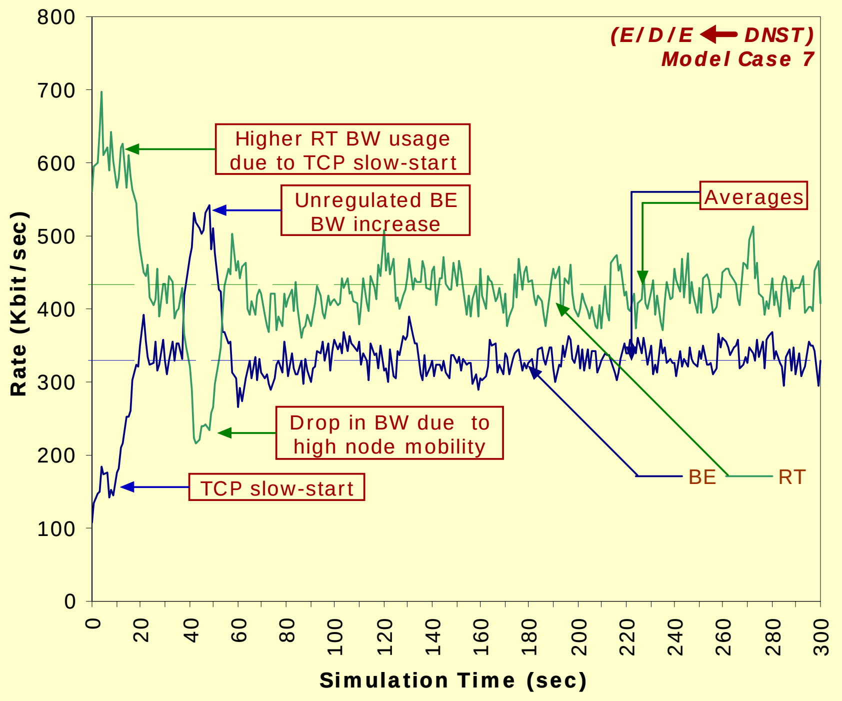


**Figure D-7:** *Aggregated DNST Bandwidth Chart -Model Case 7- Attached to INSIGNIA*

The gateway bandwidth is practically divided into 430 Kbps for RT-bandwidth, and 330 Kbps for BE-bandwidth in model case 7 (E/D/E←DNST), for a total of 760 Kbps. The small RT-bandwidth increase over BE-bandwidth is due to the use of DiffServ in the access domain.

The advantages of using PYLON-Lite becomes clear when studying model case 8 (E/E/E←DNST), as illustrated in Figure D-7. RT-bandwidth does not benefit much from the TCP slow-start since the RT-bandwidth is almost at its full capacity. In the same sense, the drop in RT-bandwidth at 38 seconds of simulation time leads to limited increase in BE-bandwidth until INSIGNIA performs local repairs to fix broken QoS routes. TCP retransmission causes BE-bandwidth to suffer until the load balances again through rerouting.

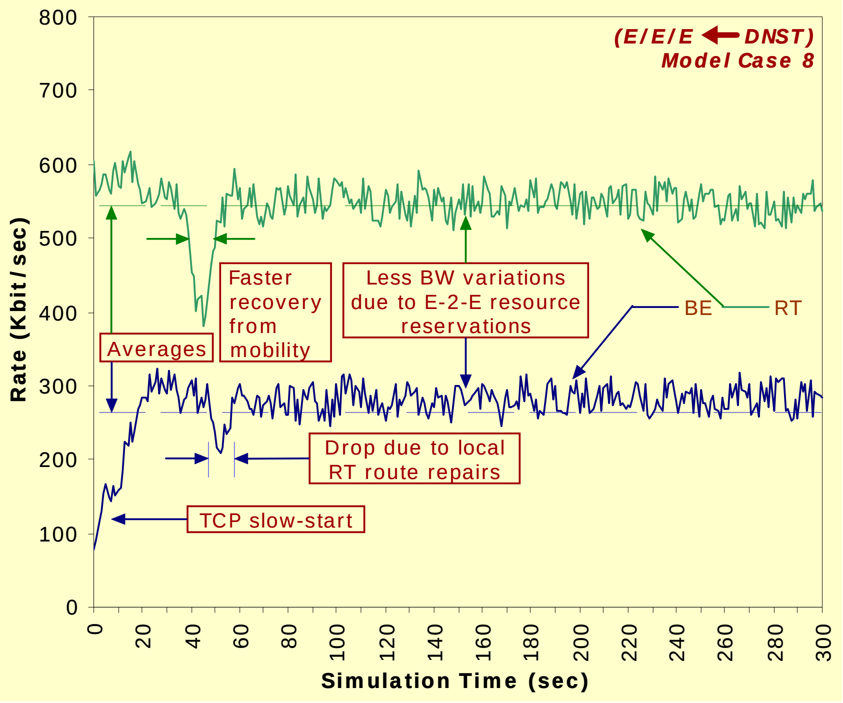


**Figure D-8:** *Aggregated DNST Bandwidth Chart -Model Case 8- Attached to INSIGNIA*

In a practical sense, the gateway bandwidth is divided into 545 Kbps for RT-bandwidth, and 275 Kbps for BE-bandwidth in model case 8 (E/E/E←DNST), for a total of 810 Kbps. Compared to model case 7 (E/D/E←DNST) shown in Figure D-7, enabling PYLON-Lite provides about 27% more RT-bandwidth ($\frac{545-430}{430}$), and 17% less BE-bandwidth.

Enabling PYLON-Lite leads to smoother bandwidth delivery and smaller bandwidth variations from 70 Kbps to 50 Kbps when ignoring extreme bandwidth spikes.

### *D.2.1.2 Delay Perspective*

Comparing model case 7 with model case 8 from the delay perspective provides the results illustrated in Figure D-9.

TCP slow-start causes less BE-traffic when simulation starts, and therefore, smaller delays. Comparing the delay chart when PYLON-Lite is enabled to when PYLON-Lite is disabled throughout the rest of the simulation, PYLON-Lite causes RT-delays to decrease from an average of 62 msec to an average of less than 7 msec. In return, average BE-delays increase from 215 msec to 225 msec.

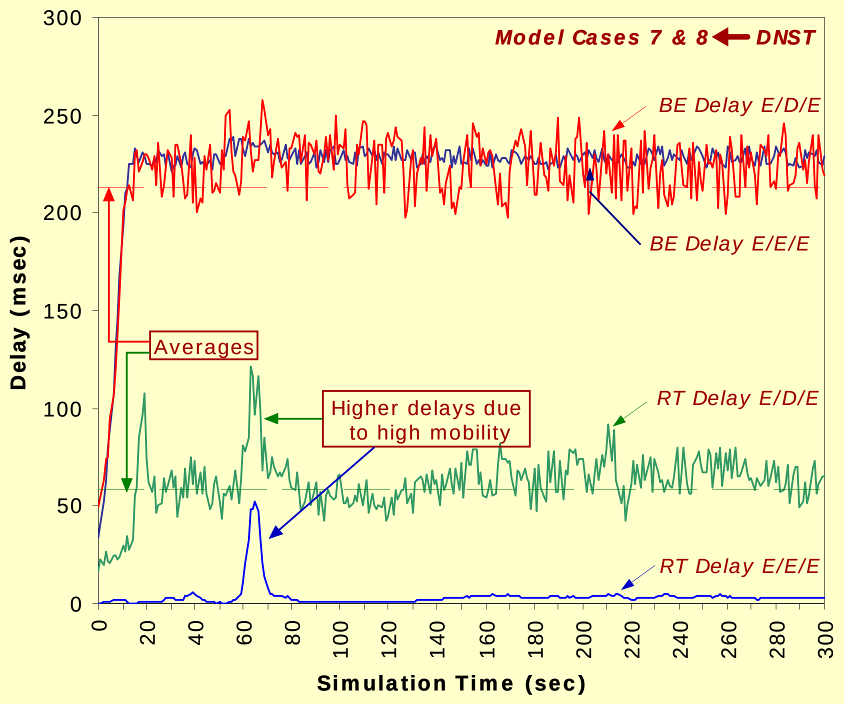


**Figure D-9:** *Aggregated DNST Delay Chart -Model Cases 7 & 8- Attached to INSIGNIA*

Enabling PYLON-Lite causes an 88% decrease in RT-delays ($\frac{7-62}{62}$); this decrease is associated with a limited increase of about 5% in BE-delays. The delay differences observed between model cases 7 and 8 can grow higher or lower depending on many factors as listed in Section 4.1. Also enabling PYLON-Lite provides much less delay variations, which leads to smaller jitter.

### D.2.2 Upstream Behavior

Model cases 5 and 6 are used as an example to illustrate the behavioral characteristics of PYLON-Lite in upstream scenario.

*D.2.2.1 Bandwidth Perspective*

The test of model cases 5 and 6 follows the same guidelines used when testing model cases 7 and 8. The resulting bandwidth charts are shown in Figure D-10 and Figure D-11 and represent the total of bandwidths observed at destination nodes receiving flows passing through the same gateway.

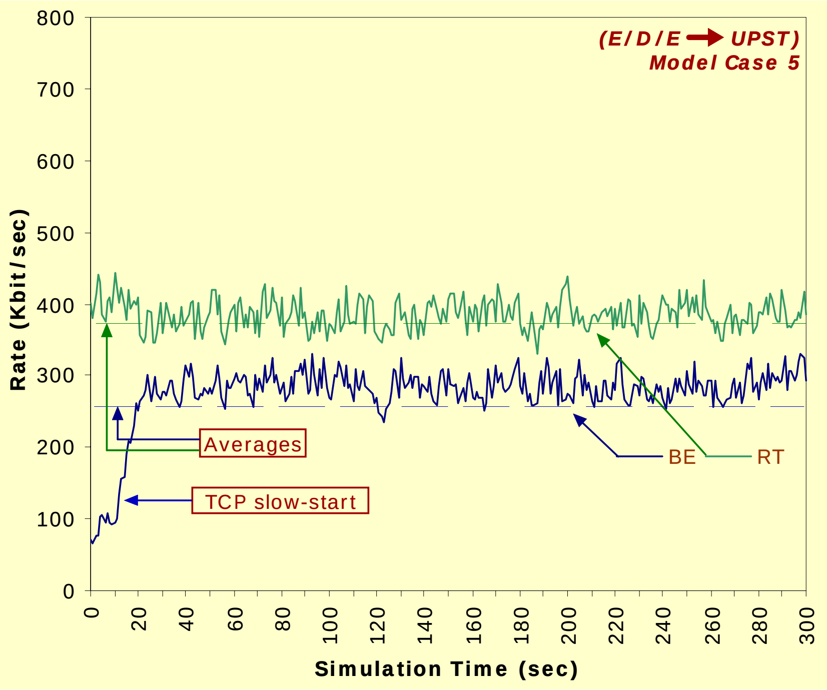


**Figure D-10:** *Aggregated UPST Bandwidth Chart -Model Case 5- Attached to INSIGNIA*

Similar to the analysis in Section 4.3, Figure D-10 represents model case 5 (E/D/E→UPST). At the simulation start, the system experiences a slow TCP start that causes low BE-bandwidth usage. Since PYLON-Lite is disabled, no resource reservation takes place on the access domain, and therefore, RT-packets get the same treatment as BE-packets on the access domain. Both RT and BE-packets are subject to queuing delays and jitter in the access domain. Therefore, RT-flows do not benefit from the TCP slow-start, instead, RT-flows maintain about 420 Kbps until the BE-bandwidth increases to 265 Kbps.

The relatively higher bandwidth consumed by RT-flows is attributed to the better service that INSIGNIA provides for RT-flows. However, RT-flows suffer from a significant bandwidth drop in the access domain, and the gateway to the access domain shows high jitter and packet drop.

Both RT-packets and BE-packets get the same treatment over the access domain. Monitored RT and BE-flows share the bandwidth the DiffServ model assigns for non real-time traffic, and therefore the total consumed bandwidth remains low. The gateway bandwidth is practically divided into 390 Kbps for RT-bandwidth, and 275 Kbps for BE-bandwidth for model case 5 (E/D/E→UPST), for a total of 665 Kbps.

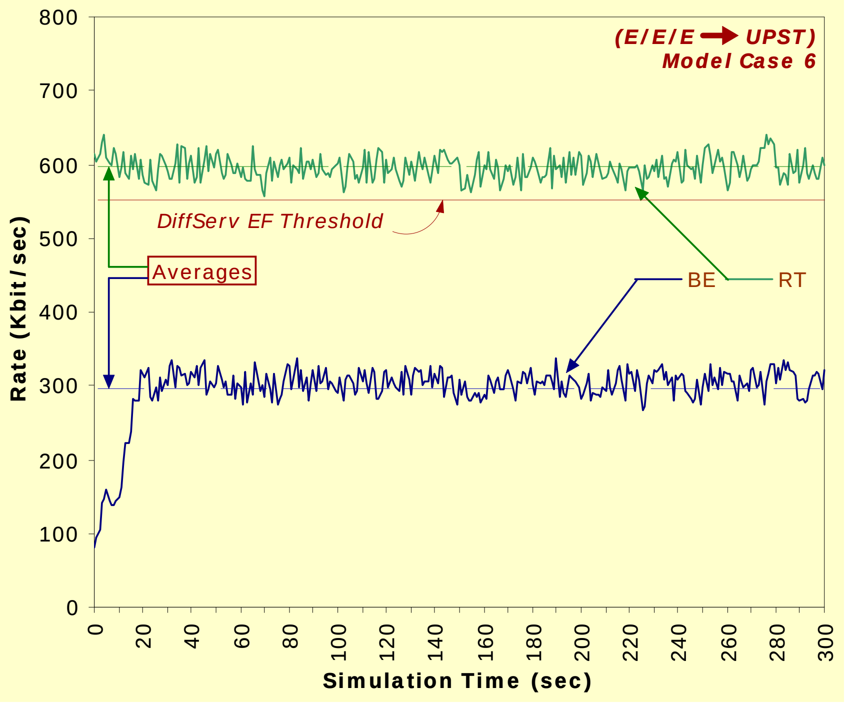


**Figure D-11:** *Aggregated UPST Bandwidth Chart -Model Case 6- Attached to INSIGNIA*

The bandwidth chart of model case 6 (E/E/E→UPST) shown in Figure D-11 illustrates the advantage of using PYLON-Lite. Since RT-bandwidth is at almost its full capacity, it does not benefit much from the TCP slow-start. RT-flows consume slightly more bandwidth than the bandwidth assigned by the DiffServ EF threshold (i.e. 550 Kbps). DiffServ downgrades the excessive bandwidth (i.e. 20 Kbps) and re-mark it as BE-traffic. However, due to the relatively limited amount of downgraded packets, The RT-flows are not highly impacted in terms of delays as illustrated in Figure D-12.

In a practical sense, the gateway bandwidth is divided into 595 Kbps for RT-bandwidth, and 295 Kbps for BE-bandwidth for model case 6 (E/E/E→UPST), for a total of 890 Kbps. Compared to the 665 Kbps of model case 5 (E/D/E→UPST), enabling PYLON-Lite provides about 53% more RT-bandwidth ( $\frac{595-390}{390}$ ), and almost the same BE-bandwidth.

### D.2.2.2 Delay Perspective

Figure D-12 shows the delay chart in an upstream scenario represented by model cases 5 and 6. At the simulation start, the TCP slow-start causes generally less BE-traffic, and therefore, smaller delays. Ignoring the startup period, enabling PYLON-Lite decreases both RT and BE-delays.

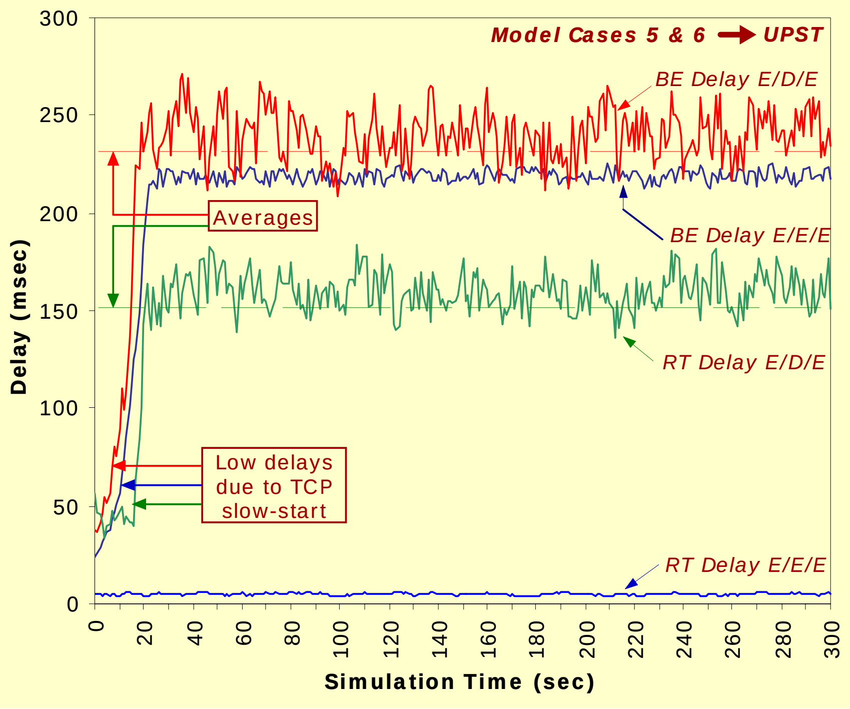


**Figure D-12:** *Aggregated UPST Delay Chart -Model Cases 5& 6- Attached to INSIGNIA*

In comparison, enabling PYLON-Lite causes RT-delays to drop from an average of 153 to less than 7 msec. Furthermore, BE-delays decrease from 230 msec to 210 msec. In other words, PYLON-Lite causes a 95% drop in RT-delays ( $\frac{7-153}{153}$ ); this drop is associated with a limited decrease of about 9% in BE-delays. The delay differences between model cases 5 and 6 can grow higher or lower depending on many factors as listed in Section 4.1. In addition, enabling PYLON is associated with a decrease in delay variation, and hence, less jitter.

In conclusion, PYLON-Lite is shown to maintain its common behavior characteristics when attached to INSIGNIA rather than SWAN. PYLON-Lite raises the amount of bandwidth assigned to RT-flows while leaving the BE-bandwidth almost unchanged. PYLON-Lite decreases the average RT-delay by a high factor while maintaining average BE-delay around the same level. PYLON-Lite also decreases the variations in bandwidth and delays for both RT and BE-traffic, which leads to smaller jitter.

## D.3 Conclusion on PYLON-Lite Behavior

In order to draw a conclusion for the results presented in Section 4.3 and in this appendix, it is useful to view the results in a tabular form. Table D-1 illustrates the average bandwidths, delays, and ABER observed throughout various PYLON-Lite simulations when attached to SWAN, ESWAN, and INSIGNIA QoS models in the ad-hoc network. The averages are divided into downstream (← DNST) and upstream (→UPST) traffic direction, then into RT and BE-traffic. The rows illustrate comparative cases between the situations when PYLON-Lite is disabled, or enabled. The percentage of change in bandwidth, delay, and ABER is added in each third row.

Reviewing Table D-1, three major behavioral characteristics of PYLON-Lite can be crystallized as follows:

1- Enabling PYLON-Lite always increases the amount of bandwidth assigned to RT-flows. The percentage of RT-bandwidth increase varies but remains significantly high (between 25 and 60%). Associated with the RT-bandwidth increase, the BE-bandwidth experiences some change, increasing in the case of SWAN and ESWAN (or decreasing in case of INSIGNIA). However, the amount of BE-bandwidth change is always limited (between -16% and 8%). In other words, enabling PYLON-Lite causes significant increase in RT-bandwidth and limited effect on BE-bandwidth.

2- Enabling PYLON-Lite always decreases the average RT-delays by a large margin (between 89% and 96%). Enabling PYLON-Lite also causes BE-delays to decrease (or increase in case of INSIGNIA). However, the amount of change in BE-delays is very limited (between -5 and 11%). In other words, enabling PYLON-Lite causes a large decrease in RT-delays while having a very limited effect on BE-delays.

3- Enabling PYLON-Lite always results in a smoother bandwidth and delay charts for both RT and BE-traffic. The smoother bandwidth and delay represent smaller jitter buffer, and therefore, user applications at the destination nodes receive highly consistent behavior from the network.

| Average | QoS Model | | SWAN | | | | ESWAN | | | | INSIGNIA | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Scenario | | ← DNST | | → UPST | | ← DNST | | → UPST | | ← DNST | | → UPST | |
| | | | RT | BE | RT | BE | RT | BE | RT | BE | RT | BE | RT | BE |
| Bandwidth | E/D/E | Kbps | 410 | 305 | 415 | 300 | 420 | 313 | 420 | 300 | 430 | 330 | 390 | 275 |
| | E/E/E | | 630 | 320 | 650 | 325 | 650 | 320 | 670 | 320 | 545 | 275 | 595 | 295 |
| | **Increase** | | 54% | 5% | 57% | 8% | 55% | 2% | 60% | 6% | 27% | -16% | 53% | 7% |
| Delay | E/D/E | msec | 80 | 230 | 140 | 225 | 75 | 220 | 140 | 220 | 62 | 215 | 153 | 230 |
| | E/E/E | | 8 | 220 | 7 | 210 | 6 | 210 | 6 | 195 | 7 | 225 | 7 | 210 |
| | **Decrease** | | 90% | 4% | 95% | 6% | 92% | 5% | 96% | 11% | 89% | -5% | 95% | 9% |
| ABER | E/D/E | % | 90 | 94 | 93 | 93 | 93 | 98 | 95 | 93 | 79 | 94 | 85 | 85 |
| | E/E/E | | 94 | 97 | 97 | 99 | 97 | 99 | 99 | 97 | 81 | 84 | 89 | 90 |
| | **Increase** | | 4.4% | 3.5% | 4.6% | 5.6% | 4.5% | 1.4% | 4.9% | 4.2% | 2.2% | -10% | 4.4% | 4.9% |

**Table D-1:** *Summery of Bandwidth and Delay Simulation Results*

The three behavioral characteristics listed here are shown to be consistent, and PYLON-Lite always illustrates those characteristics regardless of the specific QoS implementation on the ad-hoc side. It is rather simplistic to associate the use of PYLON-Lite with certain percentages of improvements since the final performance depends on the QoS sub-models used and on many other operational factors like mobility and network dynamics. Instead, the three behavioral characteristics show the trends that PYLON-Lite follows while the figures provide mere guidelines.

# Appendix E

# Security Issues in Ad-hoc Networks and PYLON-Lite

The architecture of the ad-hoc networks as defined in [25] imposes serious challenges to its security due to:

1- The dynamic nature of the topology.

2- The limited capabilities of wireless devices.

3- The limited physical protection of wireless devices.

4- The vulnerability of wireless links and the sporadic nature of connectivity.

5- The lack of a central monitoring point and the absence of a certificate authority.

Those challenges are related to the nature of wireless ad-hoc networks and impose serious difficulties to security researchers. While ad-hoc network research in areas like routing or energy management has shown significant advances in the last five years, the security of the ad-hoc network remains relatively untouched. This Appendix defines the common vulnerable network targets in Section E.1, and Section E.2 introduces different types of malicious attacks. Section E.3 summarizes the common attacks on the ad-hoc networks. Section E.4 illustrates the tenet of multi-fence security and Section E.5 reviews the fundamental cryptographic approaches. The Appendix illustrates the security of basic network mechanisms in Section E.5. Then, Section E.6 illustrates the vulnerabilities directly related to PYLON-Lite and proposes solutions. Section E.7 investigates the operations of PYLON-Lite with the *Virtual Private Network* (VPN) protocols. Finally, the Section E.8 ends with a conclusion on PYLON-Lite security.

## E.1 Common Vulnerable Targets

Most security attacks target the basic vulnerabilities of the network. Those vulnerabilities are described follows:

1- **Availability:** If the network connection ports are unreachable, or the routing, forwarding, and services mechanisms are out of order, the network would cease to exist.

2- **Confidentiality:** Confidentiality describes the need to protect the data roaming in the network from being understood by unauthorized parties.

3- **Privacy:** Privacy can be viewed in terms of location, identity, and existence. While confidentiality deals with the protection of data packets, privacy deals with the protection of the infrastructure information and meta-data.

4- **Authenticity:** Network elements, entities, and services must represent what they claim to represent.

5- **Integrity:** Information delivered to destinations must, genuinely, represent the information submitted by sources.

6- **Non-repudiation:** Messages can be traced back to their senders, without the sender being able to deny having sent them.

A secure ad-hoc network must protect all the listed vulnerabilities. Research efforts in the security area have covered those vulnerabilities and many standards are already in place for wire-line networks. Solutions for the ad-hoc environment are still evolving with an attempt to consider the peculiar nature of the ad-hoc networks. This appendix focuses on attacks with direct relevance to the service layer.

## E.2 Types of Attacks

Attacks on networks come in many varieties and can be grouped based on different characteristics. The least studied attacks are attacks on physical devices (hijacking). If a trusted ad-hoc wireless node is hijacked, it can virtually bypass all other security fences since it is a trusted node. One defensive approach is to enforce the use of smart cards or the continuous existence of a specific fingerprint to allow the device operation. Proposed solutions must take into account the limited capabilities of wireless devices.

Another class of attacks is the attack on the security mechanism. As mentioned in [68], the goal of a good cryptographic design is to reduce complex problems to the proper safe-keeping and management of a small number of cryptographic keys. Due to the distributed nature of an ad-hoc network, this objective is difficult to accomplish. For instance, assume that the ad-hoc network hosts a distributed trusted server, if the server falls under the control of malicious party, the entire security is compromised. However, threats to the security mechanisms are not specific to ad-hoc networks and many solutions have been proposed, but the solutions must consider the peculiarities of the ad-hoc network.

A third class of attacks is the selfishness attacks on basic cooperation mechanisms. In this class of attacks the attacker can be viewed as selfish or dishonest but somewhat innocent. The attacker can be a user, application or an entity that tries to get access to more resources by compromising the cooperation assumption. For instance, an intermediate node may drop packets related to other flows in order to offload the network, and hence, grant itself access to more resources.

A fourth class of attacks is malicious attacks on basic cooperation mechanisms like routing or service mechanisms. This class of attacks is very similar to the selfishness attacks; the only difference is the reason behind the attack. One solution to this class of attacks is to implement basic mechanisms over a tamper resistant hardware [89]. Both selfishness and malicious attacks on basic cooperation mechanisms are directly related and can be treated in the same manner, regardless of the objective of the attacker.

From a different view, attacks can be seen as active or passive attacks as well. Passive attacks do not involve any disruption to the network. Instead, the main objective of passive attacks is to copy information, therefore, passive attacks rely on sniffing information, eavesdropping, or listening. Passive attacks are harder to discover since they do not cause alarming symptoms in the network. Active attacks on the other hand alter data, obstruct operations, or cause denial of service. The objective of active attacks is to disrupt and to cause the network operations to fail.

## E.3 Common Attacks on Ad-hoc Networks

There are several common attacks on ad-hoc networks. Table E-1 describes some of the popular attacks. A malicious party may form an attack using various combinations of the listed attacks. Table E-1 shows a sample of the common attacks related to the service layer, but this table is not intended to cover all possible attacks.

| Name of Attack | Description |
|---|---|
| **System Imprinting** | The malicious node listens to the initialization of the network and takes advantage of the fact that the system must be told (one way or the other) about its identity and users. |
| **Man-In-The-Middle** | The malicious node is positioned between two nodes. It listens, to learn the used IP and MAC addresses of the communicating nodes. Then use it to impersonate one of the nodes. |
| **Impersonation or Spoofing** | The malicious node injects false routing information to impersonate one of the nodes. Collect essential information, then, may launch another attack. |
| **Eavesdropping** | The malicious node listens to various traffic exchanges until it can resolve the security key. Then it can keep listening to copy the conversation or proceed with another attack. The malicious node requires relatively high processing power and storage. General information about the network may speed up key decryption. |
| **Sinkhole** | After the malicious node joins the network (using one of the previous mechanisms), it fakes and injects a new routing table so that most traffic will go through it. The malicious node may alter the traffic before forwarding to the next node, or merely copy all data. |
| **Wormholes** | The malicious nodes inject new routing tables promoting a path outside the network as a shorter path. The new path goes through a compromised network that may alter or copy all data. |
| **Sybil Attack** | The malicious nodes impersonate several authentic nodes. Then provide data redundancy making fake information seem authentic. |
| **Rushing Attack** | The malicious node intercepts a ROUTE REQUEST of any reactive routing protocol. Adds itself to the table, then, rush its ROUTE REPLY. Since routing protocols process the first reply, the malicious node is guaranteed to receive all the traffic of the victim node. |
| **Selfish or Dishonest Attack** | A selfish or dishonest attacker may tamper with one or more of the co-operation mechanisms to take advantage of network resources. For example, an attacker may deny services passing through its node, or publish false routing tables to prevent others from getting services. Then the attacker gets the services denied to others. An attacker can be a user, an application or other network entities. |
| **Sleep Deprivation Torture** | The malicious node(s) request services repeatedly. The network gets consumed in responding to those requests. Mobile nodes may lose energy, and the network may lose availability. |
| **Ghost Packet Attack** | The malicious node uses compromised routing information to inject a ghost packet that keeps looping around. Many ghost packets will degrade network performance. |

| Name of Attack | Description |
|---|---|
| Denial of Service and Flooding (DoS) | Any attack that aims to consume network resources, and as a result, deny services to legitimate nodes. Sleep deprivation and ghost packet are two examples of a DoS attack. |

**Table E-1:** *Common Security Attacks on Wireless and ad-hoc Networks*

The listed attacks may take place in a variety of combinations, and may lead to different consequences. However, attacks leading to Denial-of-Service (DoS) and the Selfish attack are of particular interest to the PYLON-Lite gateway, and will be discussed further in this Appendix.

## E.4 Multi-fence Security Tenet

The latest security proposals for ad-hoc networks can be classified into two major approaches and one policy; namely proactive and reactive approaches and response policy. The proactive approach attempts to protect the network against security threats proactively. Typically, proactive mechanisms follow various cryptographic techniques. In contrast, the reactive approach detects threats after they take place. Finally a response policy follows predefined regulation in dealing with the ongoing threat. Each approach has its own merits and is suitable for addressing different vulnerabilities within the entire domain. For instance, secure routing protocols may adopt the proactive approach to protect routing messages, while a reactive approach is used to detect attacks on packet forwarding mechanisms.

Ad-hoc networks are inherently vulnerable to many security attacks. Therefore, ad-hoc network security solutions must integrate proactive prevention, reactive detection, and response policy to form a multi-fence comprehensive security solution. The prevention component, significantly, increases the difficulty of penetrating the system. The prevention component uses a proactive approach and is likely to be implanted in the routing mechanism for instance.

The detection component uses a reactive approach to discover, potential, misbehavior. As soon as a malicious node is defined, the response component takes a decision on the best policy to follow. The response component is likely to be policy driven. PYLON-Lite relies on the multi-fencing tenet to protect the service layer from various attacks.

## E.5 Cryptographic Notes

This section reviews the typical cryptographic mechanisms used in all IP networks and commonly adapted by the IETF.

### E.4.1 Common Cryptographic Primitives

There are three cryptographic mechanisms widely used to authenticate the content of messages exchanged among nodes. The following subsections describe the three mechanisms. Most of this section follows [111].

*E.4.1.1 Hashed Message Authentication Codes (HMAC)*

If two nodes share a secret symmetric key *K*, they can use a cryptographic one-way hash function *h* to generate and verify a message authenticator $h_K(\cdot)$. The computation is relatively simple, and therefore, affordable for low-end devices. HMAC, however, fails in broadcasting message authentication. In addition, establishing pair-wise keys is a difficult process that goes beyond the capabilities of mobile nodes, especially when the number of mobile nodes is high. In such case, a total of $(\frac{1}{2}\, n\, (n-1))$ keys must be maintained in a network with *n* nodes. Following this approach, [82] proposed a solution for secure Dynamic Source Routing (DSR) using pair-wise shared keys.

*E.4.1.2 Digital Signature*

Another proposal is the digital signature which is based on asymmetric key cryptography. All nodes can communicate efficiently knowing the public key of the signing node. This public key is used by the digital signature algorithm to verify the message of the signing node. The digital signature is scalable since only a total number of *n* public/private key-pairs need be managed in a network of *n* nodes. However, the computation overhead in signing/decrypting and verifying/encrypting operations is quite high. Digital signatures are vulnerable to DoS attacks. For instance, the attacker can feed the victim node with a large number of bogus signatures and exhaust the victim node's computational resources. In addition, each node needs to keep a Certificate Revocation List of revoked certificates which imposes a storage challenge. This approach is followed by the *Secure Ad-hoc On-demand Distance Vector routing protocol* SAODV [114] and Authenticated Routing for Ad-hoc Networks (ARAN) [95].

#### *E.4.1.3 One-way HMAC Key Chain*

A better proposal may be to apply a one-way HMAC key chain. It is possible to use a one-way function $f(x)$ such that, given the output value of $f(x)$ it is computationally infeasible to find the input $x$. If the function $f(\cdot)$ is applied repeatedly on an initial input $x$, it generates a chain of outputs $f^{\,k}(x)$. These outputs are used in the reverse order to authenticate the messages. The receiver keeps a message $f^k(x)$ until the sender reveals $f^{k-1}(x)$, and then, the receiver can authenticate the message [59]. The Timed Efficient Stream Loss-tolerant Authentication (TESLA) [88] uses a one-way HMAC key to authenticate broadcast messages. Another example is the *Secure Efficient Ad-hoc routing* (SEAD) for Destination Sequenced Distance Vector (DSDV) [47]. Ariadne [48] based on DSR, is highly efficient and uses symmetric cryptographic primitives. The packet leashes idea [49] is created in response to wormhole attacks and uses the one-way HMAC key chains.

The computational footprint of the one-way-key chain-based authentication is lightweight, and is suitable for broadcasting scenarios. In return for such flexibility, it is easy to observe some disadvantages. For instance, the authentication of hashed key-chain requires accurate clock synchronization that may need special hardware support. In addition, receivers must buffer messages until the arrival of the relevant hashed chain. This delay imposes challenges to the responsiveness of the routing protocol. If immediate authentication is desired, very tight clock synchronization and large storage are necessary (as in [49]). Also the release of the key requires essentially a second round of communication. Therefore, the timer synchronization must be accurately maintained [111].

### E.4.2 Common Message Digest Algorithms

There are many proposed message digest algorithms that compress the content of messages. This section presents two widely used message digest algorithms that follow the IETF guidelines describing the IP security architecture [55].

#### *E.4.2.1 Secure Hash Algorithm SHA*

SHA version 1 computes a condensed representation of a message or data file. When a message $M_1$ of any length < $2^{64}$ bits is input, the SHA1 produces a 160 bit output $O_1$ called a message digest. The message digest can then, for example, be input to a signature algorithm which generates or verifies the signature for the message. Signing the message digest rather than the message often improves the efficiency of the process because the message digest is usually much smaller in size than the message. The same hash algorithm must be used by the verifier of a digital signature as was used by the creator of the digital signature.

The SHA1 is secure because it is computationally infeasible to find a message which corresponds to a given message digest, or to find two different messages which produce the same message digest. Any change to a message in transit will, with very high probability, result in a different message digest, and the signature will fail to verify [31].

#### *E.4.2.2 Message Digest MD*

The MD version-5 Message Digest algorithm takes as input a message $M_1$ of arbitrary length and produces as output a 128-bit message digest $O_1$ that represents the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given pre-specified target message digest. One application of the MD5 algorithm is when a large file must be *"compressed"* in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem. The MD5 algorithm is quite fast on 32-bit processors and does not require any large substitution tables; the algorithm can be coded quite compactly [93].

When HMAC combined with either SHA or MD5 is applied to the IP header, it generates the Encapsulated Security Payload (ESP) as described in [21] and [55]. The ESP can provide authenticity of the sender. The two message digest technologies mentioned here represent a basic part of the IETF security proposal highlighted in [31].

In August 2004, successful attempts to crack the SHA and MD algorithms were published. SHA-0 is found to be vulnerable to differential cryptanalysis attack that looks for some type of characteristic masks that can be added to an input word with non-trivial probability of generating the same output of the compression function [19]. Wang [105] found many real collisions in MD which are composed of two 1024-bit messages with the original initial value. It takes about one hour on an IBM P690 to resolve the initial message, and the attack works for any given initial value. While those attacks represent a serious risk to the IETF based security, the attacks are not unique to PYLON-Lite. Future enhancements to message digest algorithms can be adapted by PYLON-Lite at the time.

### E.4.3 Summary on Cryptographic Notes

The different cryptographic primitives and message digests included in this section provide a short review of commonly adopted algorithms. Different algorithms can fit into securing different components of the network. The IETF has presented RFC-2401 to provide an architectural view of various possible cryptographic solutions, and where each solution may fit. PYLON-Lite is consistent with the IETF security architecture, and therefore, can adopt the IETF RFC-2401 recommendations to secure different components of its model. The one-way HMAC key chain is commonly used to secure routing messages due to its ability to deal with broadcasting messages. The HMAC can be combined with either SHA or MD5 as described in [21] to provide a ciphered digital signature.

The advantages of a one-way HMAC key chain come at a certain cost as discussed in Section E.4.1.3. We argue that the cost of using a one-way HMAC key chain can be tolerated within the context of PYLON-Lite signaling messages exchange. For instance, the buffering of PYLON-Lite service messages and the delay in their processing to wait for the release of the key adds a little delay. In addition the release of the key involves a second round of communication which may add to the Service Initiation Delay (i.e. only when a new service is initiated). Unlike routing algorithms, the delay in starting a service in a secure environment can be tolerated and justified to most users. Finally, the timers of communicating nodes can be carefully gauged. For example [35], [94], and [100] have shown different mechanisms to achieve synchronization between ad-hoc nodes.

In conclusion, the one-way HMAC key chain fits the security needs of PYLON-Lite. PYLON-Lite recommends the use of one-way HMAC to broadcast secure sponsorship messages, the probe messages, and message exchanges with the AAA server. PYLON-Lite is less stringent in its Service Initiation Delay compared to routing protocols. The relatively lightweight computational footprint of the one-way HMAC provides a good defense against DoS attacks.

## E.5 Security of Basic Mechanisms

This section slightly extends the survey into three important security fences, namely, the link layer, the routing protocol and the packet forwarding. Those three fences are built to secure the basic network mechanisms and are closely related to securing the PYLON-Lite gateway. Solutions offered to secure the routing messages are particularly related to PYLON-Lite and provide a valuable set of solutions that PYLON-Lite can adopt.

### E.5.1 Link-layer Security

The MAC layer vulnerabilities of the wireless channels have been the subject of research for decades, and many solutions and standards exist nowadays. For instance, the *Wired Equivalent Privacy* (WEP) [118] is a popular mechanism that can satisfy the security needs of low-end networks. WEP is found to be vulnerable to message privacy, message integrity attacks [53], and probabilistic cipher key recovery [101] attacks such as the Fluhrer-Mantin-Shamir attack [37]. The recently proposed IEEE 802.11i WiFi Protected Access (WPA) [117] has mended all obvious loopholes in WEP. Currently a new proposal called *Robust Security Network* (RSN) requires new hardware and software drivers to support its Advanced Encryption Standard (AES) [117]. DoS attacks on IEEE 802.11 have been identified only recently. The attacker may exploit the 802.11x binary exponential back-off scheme to launch DoS attacks as in [44] and [60]. The MAC layer vulnerabilities are not unique to ad-hoc networks and have been extensively studied in the context of wireless LANs.

### E.5.2 Routing Protocol Security

Research proposals to secure the ad-hoc routing protocols apply a proactive security approach to the existing MANET routing protocols. Each mobile node signs its routing messages using selected cryptographic authentication mechanism. Collaborative nodes can efficiently detect legitimate messages and therefore achieve a protected routing. But because authenticated nodes can be hijacked and may fall under the control of a malicious attacker, routing messages are also validated against the fundamental compliance with the routing algorithm.

Ariadne provides a secure extension to Dynamic Source Routing (DSR) in [48] that relies on a key chain algorithm. The main challenge for the Ad-hoc On-demand Distance Vector routing protocol (AODV) is to securely advertise hop count. Hop count is also expected to be increased by one only, and never decrease. A hop count hash chain is proposed in [47] and [114] such that an intermediate node cannot decrease the hop count of a routing update message. Hu [49] follows a sophisticated mechanism called hash tree chain to ensure monotonically increasing hop count when the routing update traverses the network. A recent security proposal is published for routing protocols based on link state like the Optimized Link State Routing (OLSR) [83] that makes the protocol less vulnerable to DoS attacks. In [50], Hu proposes a mechanism to protect various ad-hoc routing protocols from rushing attacks.

In all the described attacks and defenses, the Byzantine Generals [61] hypothesis is an embedded assumption that relaxes the complicated security dilemma. The hypothesis assumes that one General commands each division of the army, and some of the Generals, who communicate via messenger, are traitors. All loyal Generals must decide upon the same plan of action. Since a small number of traitors, or malicious nodes, cannot cause loyal nodes to adopt routing tables that violate the algorithm fundamentals, the network will not fail. A proposal to increase the resilience of the AODV protocol to Byzantine failures can be found in [4].

### E.5.3 Packet Forwarding Security

Packet payload can be ciphered by the application layer, using various technologies. That can protect the content of the packet but does not guarantee its delivery. For instance, a malicious intermediate node may drop a stream of perfectly ciphered packets altogether. Therefore, the ad-hoc networks need to adopt a reactive detection algorithm to discover problems in packet forwarding. A local detection mechanism can simply listen to the forwarding process as in [66] and [112]. Assume that node X is forwarding a packet to node Y, and node Y is supposed to forward it to node Z, furthermore, assume symmetric bidirectional radio connectivity. After X forwards a packet to Y, it can listen to verify that

Y forwards the packet to Z, and this is called positive acknowledgement. Awerbuch [4] enforces the use of an explicit acknowledgement mechanism.

## E.6 PYLON-Lite Vulnerabilities and Solutions

PYLON-Lite follows the tenet of multi-fencing security in order to provide comprehensive protection to the service layer. PYLON-Lite recommends the use of multi-fencing components, namely, proactive prevention, reactive detection, and response policy. In a proactive sense, PYLON-Lite recommends a secure message exchange to protect all signaling messages. A one-way HMAC key chain [59] can be used by either using MD5 or SHA1 for condensing the signaling messages as described in [21]. The process is simple and has a small computational footprint.

Another solution can follow a simple algorithm. The HMAC can be combined with MD5 or SHA1 condensing functions then used with fixed length ESP to provide source authentication in addition to data integrity. The generated message digest is a short fixed-length summary of the original message. To generate an HMAC, the secret key is XOR'ed with an inner pad value, then hashed with the message text to produce a keyed hash value. Next, the secret key is XOR'ed with an outer pad value, then hashed with the first hash value. This produces a second keyed hash that is carried within the ESP packets. The recipient uses the same algorithm, secret key, and message text to generate his own HMAC. Then it compares its HMAC to the HMAC carried in the arriving packet. If the two values match, the packet is considered authentic. If the two values differ, the packet is discarded because the message was either modified in transit or the HMAC was generated with the wrong key or algorithm. PYLON-Lite can import the use of HMAC without any change to its design architecture. In a secure PYLON-Lite operational mode, all of the sponsoring messages, probe messages, and AAA server messages must be authenticated to provide the proactive security fence.

Other vulnerabilities related to the PYLON-Lite architecture fall into the reactive detection component. PYLON-Lite is vulnerable in specific to Denial-of-Service and Selfish attacks. Other attacks are expected to be handled by different components of the network in a multi-fencing security environment.

The Denial-of-Service attacks can take place if one or many nodes flood the PYLON-Lite gateway with service requests. PYLON-Lite recommends the use of filtration mechanism to limit the number of active requests a single node can make, and the total number of requests the PYLON-Lite gateway can handle during a specific period of time. Thresholds like this must be configurable, and the gateway administrator can be provided with log files describing the service requests, and the over-the-threshold service requests. Configuring PYLON-Lite to handle too few service requests may impose unnecessary limitations on the gateway services, and configuring PYLON-Lite to handle too many service requests may make it vulnerable to DoS attacks. The acceptable amount of service requests from the same node, or from the entire network depends on the PYLON-Lite gateway resources, and on realistic traffic expectations from the connected networks.

The Selfish or Dishonest attack can be experienced when a single flow increases its traffic eating up resources that could be allocated to other flows. PYLON-Lite offers a Flow Policing Controller FPC component that facilitates the monitoring of RT-flows. RT-packets arriving at the PYLON-Lite gateway are validated against the context parameters of its flow. If some RT-packets are found to be in excess of the original flow limitations, packets can either be downgraded to BE-packets or dropped altogether depending on the policy. However, a greedy node can tamper with its packets header, for instance, and use the header of packets belonging to other flows in order to gain more bandwidth. The PYLON-Lite gateway recommends the use of a hash function that combines the source address, source port, destination address, and destination port in order to identify individual RT-flows. PYLON-Lite relies on securing the packet forwarding mechanism also as defined in Section E.5.3 to limit the possibilities of Selfish or Dishonest attacks.

## E.7 PYLON-Lite Operations with VPN Protocols

The Virtual Private Network VPN Tunnel is merely a logical data path that provides privacy through security procedures and tunneling protocols such as the *Layer Two Tunneling Protocol* (L2TP) [103] or the IPSec's Encapsulated Security Payload (ESP) [56].

Typically, VPN uses ESP alone if either ends of the tunnel can not support L2TP.

The problem with VPN protocols is the fact that they hide the original IP header. Essential

IP header information like source address, destination address, communication port, payload protocol type, and DSCP become inaccessible to gateways and intermediate routers. Figure E-1 illustrates the IP packet structure in ESP tunnel mode which results in the intermediate routers and gateways losing access to essential IP header information. This problem is not limited to ESP, in fact it exists with L2TP, IP Authentication Header (AH), and other VPN protocols; and influences the operation of *Network Address Translation* (NAT), *Port Address Translation* (PAT) as well as DiffServ ingress gateways. Many proposals have been introduced to provide some access to the IP-header essential information as in [1] and [16]. This section presents one of the solutions as an example, and refers to [1] for various other solutions.

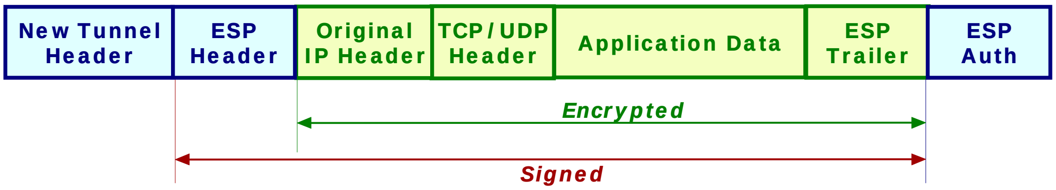


**Figure E-1:** *IP-packet Structure in ESP Tunnel Mode*

Assume that the tunnel starts at the source host, and ends at the destination host. PYLON-Lite can rely on the source host to mark packets and add the DSCP field to the new tunnel header since it is not encrypted. In this case, PYLON-Lite does not change packet marking; instead, it provides the negotiated service for the marked packets based on the TCA. However, PYLON-Lite will view all tunneled packets coming from the same source as one flow. Therefore, during admission, the source node has to compensate for future flows when forming its bandwidth request. Another problem with this solution is the possibility of a selfish node marking all the tunneled packets, even BE-packets, as high priority.

The problem of ciphered IP-header information is not unique to PYLON-Lite. DiffServ, NAT, and PAT face the same difficulties as well. PYLON-Lite can adopt any of the proposed solutions in [1], and the selected solution should generally consider network configuration, the level of required security, and the unique nature of ad-hoc networks.

## E.8 Conclusion on PYLON-Lite Security

This appendix illustrates the security challenges related to the ad-hoc environment, and describes the increased security risks of the wireless environment compared to the wire-line. The appendix also identifies common vulnerable targets, and classifies attacks, then describes the common attacks on ad-hoc networks, and defines both DoS and Selfish attacks as closely related to the PYLON-Lite model. Since security attacks may take place in different forms and at different layers, the multi-fence security tenet is introduced as essential defense strategy and can be divided into proactive prevention, reactive detection, and response policy. The appendix describes three of the most common cryptographic primitives in addition to two message digest algorithms, and then it reviews the security of three basic network mechanisms.

The vulnerabilities of the PYLON-Lite gateway are defined as well, and a possible proactive solution to protect the PYLON-Lite message exchange is proposed. PYLON-Lite proposes a solution that follows a reactive detection mechanism to detect attacks like DoS or Selfish. The defined vulnerabilities are not unique to the PYLON-Lite model, thus, common solutions are presented. In addition, a brief discussion of the operations of PYLON-Lite model with other security protocols like VPN, L2TP, and ESP is included. The proposed solutions to PYLON-Lite vulnerabilities are not considered part of the model. Instead, they are viewed as add-ons, and other solutions may apply as well.

The PYLON-Lite model relies on the multi-fencing tenet in its defense against most security hazards, hence, remains focused on security issues directly related to its design. Security can certainly be brought to decentralized self-organizing networks. Ad-hoc networks are subject to the wireless channel vulnerability in addition to the dynamics and decentralization of its environment. But ad-hoc security is achievable and the growing research initiatives reflect the interest in this promising subject.

# Abbreviations and Acronyms

## A

| | |
|---|---|
| **AAA Server** | *Authentication, Authorization, and Accounting server* |
| **AAL** | *ATM Adaptation Layer* |
| **AAL1, AAL2** | *AAL type 1 and 2* |
| **ABER** | *Average Bandwidth Efficiency Ratio*<br>*(PYLON-Lite term)* |
| **ABR** | *Available Bit Rate* |
| **AES** | *Advanced Encryption Standard* |
| **AIMD** | *Additive Increase Multiplicative Decrease* |
| **AODV** | *On Demand Distance Vector ad-hoc routing protocol* |
| **ARAN** | *Authenticated Routing for Ad-Hoc Networks* |
| **ARSVP** | *Aggregated Resource Reservation Protocol* |
| **ASR** | *Aggregate Service Reservation* |
| **ASRD** | *Aggregate Service Reservation Delay*<br>*(PYLON-Lite term)* |
| **ATM** | *Asynchronous Transfer Mode* |

## B

| | |
|---|---|
| **BA** | *Behavioral Aggregate* |
| **BE** | *Best Effort* |
| **BISDN** | *Broadband Integrated Services Digital Network* |
| **BSS** | *Base Station Subsystem* |
| **BTS** | *Base Transceiver Station* |
| **BW** | *Bandwidth* |

## C

| | |
|---|---|
| **CBR** | *Constant Bit Rate* |
| **CDMA** | *Code Division Multiple Access* |
| **CE** | *Congestion Experienced* |
| **CoS** | *Class of Service* |
| **CSMA/CA** | *Carrier Sense Multiple Access with Collision Avoidance* |

## D

| | |
|---|---|
| **DCF** | *Distributed Coordination Function* |
| **DiffServ** | *Differentiated Service Model* |
| **DNST** | *Downstream (destination is a node in the ad-hoc network)*<br>*PYLON-Lite term* |
| **dQoS** | *dynamic Quality of Service* |
| **DSCP** | *Differentiated Service Code Point* |
| **DSR** | *Dynamic Source Routing* |

## E

| | |
|---|---|
| **E2E** | *End-to-End* |
| **EBR** | *Effective Bandwidth Ratio*<br>*PYLON-Lite term* |
| **ECN** | *Explicit Congestion Notification* |

| | |
|---|---|
| **EDCF** | *Enhanced Distributed Coordination Function* |
| **EDR** | *Effective Delay Ratio*<br>*PYLON-Lite term* |
| **EF** | *Expedited Forwarding* |
| **ESP** | *Encapsulated Security Payload* |
| **ESWAN** | *Enhanced SWAN QoS model for ad-hoc networks* |

**F**

| | |
|---|---|
| **FPC** | *Flow Policing Controller*<br>*PYLON-Lite term* |
| **FPRP** | *Five-Phase Reservation Protocol* |
| **FQMM** | *Flexible QoS Model for Mobile ad-hoc networks* |
| **FTP** | *File Transfer Protocol* |

**G**

| | |
|---|---|
| **3GPP** | *Third Generation Partnership Project* |
| **GGSN** | *Gateway GPRS Support Nods* |
| **GPRS** | *General Packet Radio Service* |
| **GW** | *Gateway*<br>*PYLON-Lite term* |

**H**

| | |
|---|---|
| **HC** | *Hop Count* |
| **HMAC** | *Hashed Message Authentication Codes* |

**I**

| | |
|---|---|
| **IEEE 802.11** | *IEEE Wireless Specifications* |
| **IESG** | *Internet Engineering Standard Group* |
| **INSIGNIA** | *In-band Signaling QoS model for ad-hoc networks* |
| **IntServ** | *Integrated Services QoS model* |
| **ISDN** | *Integrated Services Digital Network* |
| **ITU-T** | *International Telecommunication Union, Telecommunication Standardization Sector* |

**L**

| | |
|---|---|
| **L2TP** | *Layer Two Tunneling Protocol* |

**M**

| | |
|---|---|
| **MAC** | *Media Access Control* |
| **MANET** | *The Mobile Ad-hoc Networking group is part of the IETF* |
| **MAPD** | *Maximum Acceptable Packet Delay*<br>*PYLON-Lite term* |
| **MD or MD5** | *The Message Digest algorithm (version 5)* |
| **MPEG** | *The Moving Picture Experts Group digital video format* |
| **MPLS** | *Multi-Protocol Label Switching* |

**N**

| | |
|---|---|
| **NAT** | *Network Address Translation* |
| **NDSCP** | *Native DSCP Set*<br>*(PYLON-Lite term)* |
| **NS2** | *The Network Simulator version 2* |

**O**

| | |
|---|---|
| **OLSR** | *Optimized Link State Routing* |

| | |
|---|---|
| **OSPF** | *Open Shortest Path First* |

**P**

| | |
|---|---|
| **PAT** | *Port Address Translation* |
| **PCS** | *Personal Communication Services* |
| **PDP** | *Packet Data Protocol* |
| **PHB** | *Per-Hop-Behavior* |
| **PTN** | *Public Telephone Network* |

**Q**

| | |
|---|---|
| **QoS** | *Quality of Service* |

**R**

| | |
|---|---|
| **RSN** | *Robust Security Network* |
| **RSVP** | *Resource Reservation Protocol* |
| **RT** | *Real-time* |

**S**

| | |
|---|---|
| **SAODV** | *Secure Ad-hoc On-demand Distance Vector routing protocol* |
| **SDH** | *Synchronous Digital Hierarchy* |
| **SEAD** | *Secure Efficient Ad-hoc routing or Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks* |
| **SEEDEX** | *MAC Protocol for Ad-hoc Networks* |
| **SHA** | *Secure Hash Algorithm* |
| **SID** | *Service Initiation Delay*<br>*PYLON-Lite term* |
| **SMDS** | *Switched Multi-megabit Data Service* |
| **SONET** | *Synchronous Optical Network* |
| **SRPD** | *Service Reply Delay*<br>*PYLON-Lite term* |
| **SRQD** | *Service Request Delay*<br>*PYLON-Lite term* |
| **SSSD** | *Service Sponsor Solicitation Delay*<br>*PYLON-Lite term* |
| **SWAN** | *Stateless Wireless Ad-hoc Networks* |

**T**

| | |
|---|---|
| **TCA** | *Traffic Conditioning Agreement* |
| **TCMA** | *Tiered Contention Multiple Access* |
| **TDMA** | *Time Division Multiple Access* |
| **TELSA** | *Timed Efficient Stream Loss-tolerant Authentication* |
| **ToS** | *Type of Service* |

**U**

| | |
|---|---|
| **UBR** | *Unspecified Bit Rate* |
| **UE** | *User Equipments* |
| **UMTS** | *Universal Mobile Telecommunication Systems* |
| **UPST** | *Upstream (source is a node in the ad-hoc network)*<br>*PYLON-Lite term* |

**V**

| | |
|---|---|
| **VBR** | *Variable Bit Rate* |
| **VPN** | *Virtual Private Network* |

**W**

| | |
|---|---|
| **WEP** | *Wired Equivalent Privacy* |

# References

**[1]** B. Aboba and W. Dixon, *"IPsec Network Address Translation (NAT) Compatibility Requirements"*, IETF RFC-3715, March 2004.

**[2]** G. Ahn, A. Campbell, A. Veres and L. Sun, *"SWAN: Service Differentiation in Stateless Wireless Ad-hoc Networks"*, in the Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM-02), vol. 2, pp. 457-466, New York NY-USA, June 2002.

**[3]** G. Ahn, A. Campbell, A. Veres and L. Sun, *"Supporting Service Differentiation for Real-time and Best Effort Traffic in Stateless Wireless Ad-hoc Networks (SWAN)"*, in the IEEE Transactions on Mobile Computing (TMC-02), vol. 1, issue 3, pp. 192-207, September 2002.

**[4]** B. Awerbuch, D. Holmer and H. Rubens, *"Swarm Intelligence Routing Resilient to Byzantine Adversaries"*, in the Proceedings of the International Seminar on Communications, pp. 160-163, Zurich Switzerland, February 2004.

**[5]** H. Badis, A. Munaretto, K. Al Agha and G. Pujolle, *"QoS for Ad hoc Networking Based on Multiple Metrics: Bandwidth and Delay"*, in the Proceedings of the 5th IEEE International Conference on Mobile and Wireless Communications Networks (MWCN-03), Singapore, October 2003.

**[6]** F. Baker, C. Iturralde, F. Faucheur and B. Davie, *"Aggregation of RSVP for IPv4 and IPv6 Reservation"*, IETF RFC-3175, September 2001.

**[7]** M. Benveniste, *"Tiered Contention Multiple Access (TCMA), A QoS-based Distribution MAC Protocol"*, in the Proceedings of the 13th IEEE International Symposium on Personal Indoor and Mobile Radio Communications, vol. 2, pp. 598-604, Lisbon Portugal, September 2002.

**[8]** Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski and E. Felstaine, *"A Framework for Integrated Services Operation over Diffserv Networks"*, IETF RFC-2998, November 2000.

**[9]** S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, *"An Architecture for Differentiated Services"*, IETF RFC-2475, December 1998.

**[10]** G. Bochmann and A. Hafid, *"Some Principles for Quality of Service Management"*, in the Transactions of the Distributed Systems Engineering, vol. 4, pp. 16-27, March 1997.

**[11]** G. Bochmann and C. Sunshine, *"Formal Methods in Communication Protocol Design"*, in the IEEE Transactions on Communications, vol. COM-28, no. 4, pp. 624-631, April 1980.

**[12]** G. Bochmann and P. Monval, *"Design Principles for Communication Gateways"*, in the IEEE Journal on Selected Areas in Communication, vol. 8, no. 1, pp. 12-21, January 1990.

**[13]** M. Borden, E. Crawley, B. Davie and S. Batsell, *"Integration of Real-time Services in an IP-ATM Network Architecture"*, IETF RFC-1821, April 1995.

**[14]** R. Braden, D. Clark and S. Shenker, *"Integrated Services in the Internet Architecture: an Overview"*, IETF RFC-1633, July 1994.

**[15]** E. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, *"Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification"*, IETF RFC-2205, September 1997.

**[16]** T. Braun, M. Guenter and I. Khalil, *"Management of Quality of Service Enabled VPNs"*, in the IEEE Communication Magazine, vol. 39, no. 5, pp. 90-98, May 2001.

**[17]** J. Broch, D. B. Johnson, Y. Hu and J. Jetcheva, *"A Performance Comparison of Multi-hop Wireless Ad-hoc Networking Routing Protocols"*, in the Proceedings of the ACM/IEEE 4th International Conference on Mobile Computing and Networking (MOBICOM-98), pp. 85-97, Dallas TX-USA, October 1998.

**[18]** L. Burgstahler, K. Dolzer, C. Hauser, J.Jahnert, S. Junghansm and C. Macian, *"Beyond technology: the missing pieces for QoS success"*, in the Proceedings of the ACM/SIGCOMM WorkShop on Revisiting IP QoS (RIPQOS-03), pp. 115-120, Karlsruhe Germany, August 2003.

**[19]** F. Chabaud and A. Joux, *"Differential Collisions in SHA-0"*, in the Proceedings of the 24th Annual International Cryptology Conference (IACR-CRYPTO-04), Santa Barbara CA-USA, August 2004.

**[20]** S. Chen and K. Nahrstedt, *"Distributed Quality-of-Service Routing in Ad-hoc Networks"*, in the IEEE Journal on Selected Areas in Communication, vol. 17, no. 8, pp. 1-18, August 1999.

**[21]** P. Cheng and R. Glenn, *"Test Cases for HMAC-MD5 and HMAC-SHA-1"*, IETF RFC-2202, September 1997.

**[22]** D. Clark, S. Shenker and L. Zhang, *"Supporting Real-time Applications in an Integrated Services Packet Network: Architecture and Mechanism"*, in the Proceedings of the Annual Technical Conference on Communications Architecture and Protocols (SIGCOMM-92), vol. 22, pp. 14-26, Baltimore MD-USA, October 1992.

**[23]** T. Clausen and P. Jacquet, *"Optimized Link State Routing Protocol (OLSR)"*, IETF RFC-3626, October 2003.

**[24]** R. Cole, D. Shur and C. Villamizar, *"IP over ATM: A Framework Document"*, IETF RFC-1932, April 1996.

**[25]** S. Corson and J. Macker, *"Mobile Ad-hoc Networks: Routing Protocol Performance Issues and Evaluation Considerations"*, IETF RFC-2501, January 1999.

**[26]** E. Crawley, R. Nair, B. Rajagopalan and H. Sandick, *"A Framework for QoS-based Routing in the Internet"*, IETF RFC-2386, August 1998.

**[27]** J. Crowcroft, S. Hand, R. Mortier, T. Roscoe and A. Warfield, *"QoS's Downfall: At the bottom, or not at all!"*, in the Proceedings of the ACM/SIGCOMM Workshop on Revisiting IP QoS (RIPQOS-03), pp. 109-114, Karlsruhe Germany, August 2003.

**[28]** G. Daqing and J. Zhang, *"Evaluation of EDCF Mechanism for QoS in IEEE 802.11 Wireless Networks"*, in the Proceedings of the World Wireless Congress (WWC-03), URL: <http://www.delson.org/wwc03/>, San Francisco CA-USA, May 2003.

**[29]** A. Das, A. Ghose, A. Razdan, H. Saran and R. Shorey, *"Enhancing Performance of Asynchronous Data Traffic over the Bluetooth Wireless Ad-hoc Network"*, in the Proceedings of the 20th Annual Joint Conference of Computer and Communications Societies, IEEE (INFOCOM-01), vol. 1, pp. 591-600, Anchorage AK-USA, April 2001.

**[30]** J. Deng and R. Chang, *"A Priority Scheme for IEEE 802.11 DCF Access Method"*, in the IEICE Transactions on Communication, vol. E82-B, no. 1, pp. 96-102, January 1999.

**[31]** M. Do, Y. Park and J. Lee, *"Channel Assignment with QoS Guarantees for a Multi-class Multi-code CDMA System"*, in the IEEE Transactions on Vehicular Technology, vol. 51, issue 5, pp. 935-948, September 2002.

**[32]** D. Eastlake and P. Jones, *"US Secure Hash Algorithm 1 (SHA1)"*, IETF RFC-3174, September 2001.

**[33]** J. Ehrensberger, *"Resource Demand and Loss Probabilities of Aggregated Resource Reservation"*, in the Proceedings of the 5th International Conference on Information System Analysis and Synthesis (SCI-ISAS-99), vol. 4, Orlando FL-USA, August 1999.

**[34]** J. Ehrensberger, *"Resource Demand of Aggregated Resource Reservations"*, in the Proceedings of the 1st European Conference on Universal Multiservice Networks (ECUMN-00), pp. 56-61, Colmar France, October 2000.

**[35]** J. Elson, L. Girod and D. Estrin, *"Fine-grained Network Time Synchronization using Reference Broadcasts"*, in the Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI-02), vol. 36, issue SI, pp. 147-163, Boston MA-USA, December 2002.

**[36]** J. Eriksson, *"An Experimental Encapsulation of IP Data-grams on Top of ATM"*, IETF RFC-1926, April 1996.

**[37]** S. Fluhrer, I. Mantin and A. Shamir, *"Weakness in the Key Scheduling Algorithm of RC4"*, in the Proceedings of the 8th Annual Workshop on Selected Areas in Cryptography (SAC-01),pp. 1-24, Toronto Canada, August 2001.

**[38]** H. Fu and E. Knightly, *"Aggregation and Scalable QoS: A Performance Study"*, in the Proceedings the 9th International Workshop on Quality of Service (IWQOS-01), pp. 307-324, Karlsruhe Germany, June 2001.

**[39]** K. Geihs, *"Analysis of Adaptation Strategies for Mobile QoS-aware Applications"*, in the Proceedings of the 5th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Networking (MASWMN-02), pp. 90-97, Atlanta GA-USA, September 2002.

**[40]** M. Gien and H. Zimmermann, *"Design Principles for Network Interconnection"*, in the Proceedings of the 4th IEEE/ACM Data Communication Symposium, vol. VI, pp. 109-119, November 1979.

**[41]** J. Gomez and A. Campbell, *"Havana: Supporting Application and Channel Dependent QoS in Wireless Packet Networks"*, in the ACM/Kluwer Journal on Wireless Networks (WINET-03), vol. 9, no. 1, pp. 21-35, January 2003.

**[42]** P. Green, *"Protocol Conversion"*, in the IEEE Transactions on Communications, vol. COM-34, no. 3, pp. 257-268, March 1986.

**[43]** D. Grossman, *"New Terminology and Clarifications for DiffServ"*, IETF RFC-3260, April 2002.

**[44]** V. Gupta, S. Krishnamurthy and M. Faloutsos, *"Denial of Service Attacks at the MAC Layer in Wireless Ad-hoc Networks"*, in the Proceedings of the IEEE Military Communications Conference (MILCOM-02), vol. 2, no. 1, pp. 1118-1123, Anaheim CA-USA, October 2002.

**[45]** S. Herzog, *"RSVP Extensions for Policy Control"*, IETF RFC-2750, January 2000.

**[46]** C. Hoare, *"Quicksort"*, in the Transactions of the Computer Journal, vol. 5, no. 1, pp. 10-15, January 1962.

**[47]** Y. Hu, D. Johnson and A. Perrig, *"SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad-hoc Networks"*, in the Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA-02), pp. 3-13, Callicoon NY-USA, June 2002.

**[48]** Y. Hu, A. Perrig and D. Johnson, *"Ariadne: A Secure OnDemand Routing Protocol for Ad-hoc Networks"*, in the Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MOBICOM-02), pp. 12-23, Atlanta GA-USA, September 2002.

**[49]** Y. Hu, A. Perrig and D. Johnson, *"Packet Leashes: A Defense Against Wormhole Attacks in Wireless Networks"*, in the Proceedings of the IEEE Conference on Computer Communications (INFOCOM-03), vol. 22, no. 1, pp. 1976-1986, San Francisco CA-USA, May 2003.

**[50]** Y. Hu, A. Perrig and D. Johnson, *"Rushing Attacks and Defense in Wireless Ad-hoc Network Routing Protocols"*, in the Proceedings of the ACM Workshop on Wireless Security (WISE-03), pp. 30-40, San Diego CA-USA, September 2003.

**[51]** V. Jacobson, K. Nichols and K. Poduri, *"An Expedited Forwarding PHB"*, IETF RFC-2598, June 1999.

**[52]** N. Johansson, U. Korner and P. Johansson, *"Performance Evaluation of Scheduling Algorithm for Bluetooth"*, in the Proceedings of the IFIP Broadband Communications, pp. 139-150, Hong Kong, November 1999.

**[53]** D. Johnson, D. Maltz and Y. Hu, *"The Dynamic Source Routing Protocol for Mobile Ad-hoc Networks (DSR)"*, IETF draft, July 2004.

**[54]** V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi and E. Knightly, *"Distributed Multi-hop Scheduling and Medium Access with Delay and Throughput Constraints"*, in the Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, pp. 200-209, Rome Italy, July 2001.

**[55]** S. Kent and R. Atkinson, *"Security Architecture for the Internet Protocol"*, IETF RFC-2401, November 1998.

**[56]** S. Kent and R. Atkinson, *"IP Encapsulating Security Payload (ESP)"*, IETF RFC-2406, November 1998.

**[57]** Rajeev Koodli and Mikko Puuskari, *"Supporting Packet Data QoS in Next Generation Cellular Networks"*, in the IEEE Communications Magazine, pp. 180-188, February 2001.

**[58]** A. Kopsel and A. Wolisz, *"Voice Transmission in an IEEE 802.11 WLAN Based Access Network"*, in the Proceedings of the 4th ACM International Workshop on Wireless Mobile Multimedia (WOW-MOM-01), pp. 24-33, Rome Italy, July 2001.

**[59]** H. Krawczyk, M. Bellare and R. Canetti, *"HMAC: Keyed-hashing for Message Authentication"*, IETF RFC-2104, February 1997.

**[60]** P. Kyasanur and N. Vaidya, *"Detection and Handling of MAC Layer Misbehavior in Wireless Networks"*, in the Proceedings of the Dependable Computing and Communications Symposium (DCC), the International Conference on Dependable Systems and Networks (DSN-03), pp. 173-182, San Francisco CA-USA, June 2003.

**[61]** L. Lamport, R. Shostak and M. Pease, *"The Byzantine Generals Problem"*, in the ACM/IEEE Transactions on Programming Languages and Systems, Advanced Topics in Ultra Dependable Distributed Systems, IEEE Computer Society Press, vol. 4, no. 3, pp. 382-401, July 1982.

**[62]** S. Lee, G. Ahn, X. Zhang and A. Campbell, *"INSIGNIA: An IP-based QoS Framework for Mobile Ad-hoc Networks"*, in the Journal of Parallel and Distributed Computing, vol. 60, no. 4, pp. 374-406, April 2000.

**[63]** R. Leung, J. Liu, E. Poon, A. Chan and B. Li, *"A QoS-aware Multi-path Dynamic Source Routing Protocol for Ad-hoc Networks"*, in the Proceedings of the 26th IEEE Annual Conference on Local Computer Networks (LCN-01), pp. 132-141, Tampa FL-USA, November 2001.

**[64]** W. Liao, Y. Tseng and K. Shih, *"A TDMA-based Bandwidth Reservation Protocol for QoS Routing in a Wireless Mobile Ad-hoc Network"*, in the Proceedings of the IEEE International Conference on Communications (ICC-02), vol. 25, no. 1, pp. 3186-3190, New York NY-USA, May 2002.

**[65]** S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz and L. Stibor, *"IEEE 802.11e Wireless LAN for Quality of Service"*, in the Proceedings of the European Wireless, vol. 1, pp. 32-39, Florence Italy, February 2002.

**[66]** S. Marti, T. Giuli, K. Lai and M. Baker, *"Mitigating Routing Misbehavior in Mobile Ad-hoc Networks"*, in the Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM-00), pp. 255-265, Boston MS-USA, August 2002.

**[67]** D. McDysan and D. Spohn, *"ATM Theory and Application"*, McGraw-Hill Series on Computer Communications, ISBN: 007-060-3626, September 1994.

**[68]** A. Menezes, P. Oorschot and S. Vanstone, *"Handbook of Applied Cryptography"*, 5th, Edition, CRC Press, ISBN: 084-938-5237, August 2001.

**[69]** N. Milanovic, M. Malek, A. Davidson and V. Milutinovic, *"Routing and Security in Mobile Ad-hoc Networks"*, in the Proceedings of the IEEE Computer Society, pp. 69-73, February 2004.

**[70]** M. Mirhakkak, N. Schult and D. Thomson *"Dynamic QoS and Adoptive Applications for variable Bandwidth Environment"*, MITRE-DoD project paper, <URL: http://www.mitre.org/work/tech_papers/tech_papers_00/thomson_bandwidth/index .html>, April 2000.

**[71]** M. Mirhakkak, N. Schult and D. Thomson, *"Dynamic QoS for Mobile Ad-hoc Networks"*, MITRE-DoD project paper, <URL: http://www.mitre.org/work/tech_papers/ tech_papers_00/thomson_mp_adhoc/index.html>, January 2000.

**[72]** B. Moon and H. Aghvami, *"DiffServ Extensions for QoS Provisioning in IP Mobility Environments"*, in the IEEE Journal of Wireless Communications, vol. 10, issue 5, pp. 38–44, October 2003.

**[73]** Y. Morgan and T. Kunz, *"An Architecture Framework for MANET QoS Interaction with Access Domains"*, in the Proceedings of the 1st International Conference on Ad-hoc and Wireless Networks, pp 33-47, Toronto ON-Canada, September 2002.

**[74]** Y. Morgan and T. Kunz, *"PYLON: An Architectural Framework for Ad-hoc QoS Interconnectivity with Access Domains"*, in Proceedings of the 36th International Conference on System Sciences (HICSS-36), pp. 309-318, Hawaii USA, IEEE Computer Society Press 2003, ISBN 0-7695-1874-5, January 2003.

**[75]** Y. Morgan and T. Kunz, *"Enhancing SWAN QoS Model By Adopting Destination-based Regulation (ESWAN)"*, in the Proceedings of the 2nd Conference for Modeling and Optimization in Mobile Ad-hoc and Wireless Networks (WIOPT-04), pp. 112-121, Cambridge UK, March 2004.

**[76]** J. Moy, *"OSPF Version 2"*, IETF RFC-2328, April 1998.

**[77]** A. Munaretto, G. Pujolle, H. Badis and K. Agha, *"QoS-enhanced OLSR Protocol for Mobile Ad-hoc Networks"*, in the Proceedings of the 1st International Workshop (ANWIRE-03), pp. 171-175, Glasgow Scotland, April 2003.

**[78]** Q. Ni, L. Romdhani, T. Turletti and I. Aad, *"QoS Issues and Enhancements for IEEE 802.11 Wireless LAN"*, INRIA Research Report No. 4612, November 2002.

**[79]** K. Nichols, S. Blake, F. Baker and D. Black, *"Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers"*, IETF RFC-2474, December 1998.

**[80]** K. Nichols and B. Carpenter, *"Definition of Differentiated Services per Domain Behaviors and Rules for their Specification"*, IETF RFC-3086, April 2001.

**[81]** J. Oliveira, C. Scoglio, T. Anjali, L. Chen, I. Akyildiz and J. Smith, *"Design and Management Tools for a DiffServ-aware MPLS Domain QoS Manager"*, in the Proceedings of the International Society for Optical Engineering (SPIE-ITCOM-02), vol. 4868, pp. 43-54, Boston MA-USA, August 2002.

**[82]** P. Papadimitratos and Z. Hass, *"Secure Routing for Mobile Ad-hoc Networks"*, in the Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS-02), San Antonio TX-USA, January 2002.

**[83]** P. Papadimitratos and Z. Hass, *"Secure Link State Routing for Mobile Ad-hoc Networks"*, in the Proceedings of the IEEE Symposium on Applications and the Internet Workshop on Security and Assurance in Ad hoc Networks (SAINT-03), pp. 379-383, Orlando FL-USA, January 2003.

**[84]** S. Paskalis, A. Kaloxylos, E. Zervas and L. Merakos, *"An Efficient RSVP Mobile IP Interworking Scheme"*, in the Journal of Mobile Networks and Applications, vol. 8, issue 3, pp. 197-207, June 2003.

**[85]** M. Perez, F. Liaw, A. Mankin, E. Hoffman, D. Grossman and A. Malis, *"ATM Signaling Support for IP over ATM"*, IETF RFC-1755, February 1995.

**[86]** C. Perkins, E. Belding-Royer and S. Das, *"Quality of Service for Ad-hoc On-Demand Distance Vector Routing"*, IETF draft, <URL:http://people.nokia.net/~charliep/txt/aodvid/qos.txt>, July 2000.

**[87]** C. Perkins, E. Belding-Royer and S. Das, *"Ad-hoc On-Demand Distance Vector (AODV) Routing"*, IETF RFC-3561, July 2003.

**[88]** A. Perrig, R. Canetti, J. Tygar and D. Song, *"The TESLA Broadcast Authentication Protocol"*, in the Journal of the RSA CryptoBytes, vol. 5, no. 2, pp. 2-13, May 2002.

**[89]** A. Pfitzmann, B. Pfitzmann, M. Schunter and M. Waidner, *"Trusting Mobile User Devices and Security Modules"*, in the Proceedings of the IEEE Computer, vol. 30, no. 2, pp. 61-68, February 1997.

**[90]** M. Puuskari, *"Quality of Service Framework in GPRS and Evolution Towards UMTS"*, in the Proceedings of the 3rd European Personal Mobile Communications Conference (EPMCC-99), Paris France, March 1999.

**[91]** K. Ramakrishnan, S. Floyd and D. Black, *"The Addition of Explicit Congestion Notification (ECN) to IP"*, IETF RFC-3168, September 2001.

**[92]** T. Rappaport, *"Wireless Communications: Principle and Practice"*, Prentice Hall, Second Edition, ISBN: 013-042-2320, December 2001.

**[93]** R. Rivest, *"The MD5 Message Digest Algorithm"*, IETF RFC-1321, April 1992.

**[94]** K. Römer, *"Time Synchronization in Ad-hoc Networks"*, in the Proceedings of the International Symposium on Mobile Ad-hoc Networking and Computing (MOBIHOC-01), pp. 173-182, Long Beach CA-USA, October 2001.

**[95]** K. Sanzgiri, B. Dahill, B. Levine, C. Shields and E. Belding-Royer, *"A Secure Routing Protocol for Ad-hoc Networks"*, in the Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP-02), pp. 78-89, Paris France, November 2002.

**[96]** V. Sharma and F. Hellstrand, *"Framework for Multi Protocol Label Switching (MPLS) Based Recovery"*, IETF RFC-3469, February 2003.

**[97]** S. Shenker, C. Partridge and R. Guerin, *"Specification of Guaranteed Quality of Service"*, IETF RFC-2212, September 1997.

**[98]** S. Shenker and J. Wroclawski, *"General Characterization Parameters for Integrated Service Network Elements"*, IETF RFC-2215, September 1997.

**[99]** J. Sobrinho and A. Krishnakumar, *"Quality-of-Service in Ad-hoc Carrier Sense Multiple Access Networks"*, in the IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pp. 1353-1368, August 1999.

**[100]** E. Sourour and M. Nakagawa, *"Mutual Decentralized Synchronization for Intervehicle Communications"*, in the Transactions on Vehicular Technology, vol. 48, no. 6, pp. 2015-2027, November 1999.

**[101]** A. Stubblefield, J. Ioannidis and A. Rubin, *"Using the Fluhrer, Mantin and Shamir Attack to Break WEP"*, in the Proceedings of the Network and Distributed Systems Security Symposium (NDSS-01), pp. 17-22, San Diego CA-USA, August 2001.

**[102]** F. Tommasi, S. Molendini and A. Tricco, *"Mapping of IntServ/RSVP Reservations into MPLS Domains"*, in the Proceedings of the IEEE International Conference on Software Telecommunications and Computer Networks (SOFTCOM-2002), pp. 8-11, Split Croatia, October 2002.

**[103]** W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn and B. Palter, *"Layer Two Tunneling Protocol L2TP"*, IETF RFC-2661, August 1999.

**[104]** A. Veres, A. Campbell, M. Barry and L. Sun, *"Supporting Service Differentiation in Wireless Packet Networks Using Distributed Control"*, in the IEEE Journal on Selected Areas in Communications, Special Issue on Mobility and Resource Management in Next Generation Wireless Systems, vol. 19, no. 10, pp. 2094-2104, October 2001.

**[105]** X. Wang, D. Feng, X. Lai and H. Yu, *"Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD"*, in the Proceedings of the 24th Annual International Cryptology Conference (IACR-CRYPTO-04), Santa Barbara CA-USA, August 2004.

**[106]** J. Wroclawski, *"Specification of the Controlled Load Network Element Service"*, IETF RFC-2211, September 1997.

**[107]** J. Wroclawski, *"The Use of RSVP with IETF Integrated Services"*, IETF RFC-2210, September 1997.

**[108]** W. Wu, S. Das, A. Misra and S. Das, *"QoS Framework for Supporting Intra-domain Mobility"*, in the Proceedings of Mobile Computing and Communications Review (SIGMOBILE-03), vol. 7, issue 1, pp. 25-27, Atlanta GA-USA, September 2003.

**[109]** L. Wu, B. Davie, S. Davari, P. Vaananen, R.Krishnan and P. Cheval, *"Multi-Protocol Label Switching (MPLS) Support of Differentiated Services"*, IETF RFC-3270, May 2002.

**[110]** H. Xiao, W. Seah, A. Lo and K. Chiang, *"Flexible QoS Model for Mobile Ad-hoc Networks"*, in the Proceedings of the IEEE 51st Vehicular Technology Spring Conference (VTC-00), vol. 1, pp. 445-449, Tokyo Japan, May 2000.

**[111]** H. Yang, H. Luo, F. Ye, S. Lu and L. Zhang, *"Security in Mobile Ad-hoc Networks: Challenges and Solutions"*, in the Proceedings of the IEEE Wireless Communications, vol. 11, issue 1, pp. 38-47, February 2004.

**[112]** H. Yang, X. Meng and S. Lu, *"Self-organized Network Layer Security in Mobile Ad-hoc Networks"*, in the Proceedings of the ACM Workshop on Wireless Security (WISE-02), pp. 11-20, Atlanta GA-USA, September 2002.

**[113]** F. Ye, S. Yi and B. Sikdar, *"Improving Spatial Reuse of IEEE 802.11 Based Ad-hoc Networks"*, in the Proceedings of the IEEE Global Communications Conference (GLOBECOM-03), vol. 2, pp. 1013-1017, San Francisco CA-USA, December 2003.

**[114]** M. Zapata and N. Asokan, *"Securing Ad-hoc Routing Protocols"*, in the Proceedings of the ACM Workshop on Wireless Security (WISE-02), pp. 1-10, Atlanta GA-USA, September 2002.

**[115]** C. Zhu and M. Corson, *"QoS Routing for Mobile Ad-hoc Networks"*, in the Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM-02), New York NY-USA, June 2002.

**[116]** Digital Cellular Telecommunications System (Phase 2+), General Packet Radio Service (GPRS), Service Description, Stage Two (GSM 03.60 version 7.0.0 Release 98).

**[117]** Draft Supplement to Standard for Telecommunications and Information Exchange between Systems LAN/MAN Specific Requirements, *"Part II: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Specification for Enhanced Security"*, IEEE Standards for 802.11i/D30, August 2002.

**[118]** IEEE Computer Society, LAN/WAN Standards Committee, *"Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications"*, IEEE Standard 802.11 Draft Supplement for Telecommunications and Information Exchange, June 2003.

**[119]** Universal Mobile Telecommunications System (UMTS), *"QoS Concept"*, TS 23.107 version 3.1.0.

**[120]** Daedalus is a research project initiated by university of Berkeley, check <URL: http://daedalus.cs.berkeley.edu/>.

**[121]** Monarch is a research project initiated by university of Rice, check <URL: http://www.monarch.cs.rice.edu/>.

**[122]** The Internet Engineering Standards Group is an engineering community driving Internet standards; check <URL: http://www.ietf.org/iesg.html>.

**[123]** The Network Simulator version II is a software simulation for networks and is described in <URL: http://www.isi.edu/nsnam/ns/>.