TRITA-IMIT-LECS AVH 03:02
ISSN 1651-4070
ISRN KTH/IMIT/LECS/AVH-03/02SE

KTH

Performance Analysis of Wireless LAN Access Points

Iyad Al Khatib

Department of Microelectronics and Information Technology
Royal Institute of Technology

Stockholm, Sweden, 2003

Performance Analysis of Wireless LAN Access Points

Iyad Al Khatib

A thesis submitted to
the Royal Institute of Technology
in partial fulfillment of the requirements for
the degree of Licentiate of Technology

May 27, 2003

TRITA-IMIT-LECS AVH 03:02
ISSN 1651-4070
ISRN KTH/IMIT/LECS/AVH-03/02SE

Department of Microelectronics and Information Technology
Royal Institute of Technology
Stockholm, Sweden

ABSTRACT

Technological progression in data communications is occurring rapidly. The future of information technology features users enjoying easier and probably ubiquitous communications. However, the quality of their connections plays a significant role in the success of a communications technology, whether wired or wireless. In this thesis we evaluate the performance of wireless Local Area Networks (WLANs), which utilize access points (APs). Users show remarkable interest in getting connected without being tethered by a wire, but without degradation in the performance of services. Delay is one of the key factors behind lowering quality, thus leading to users' dissatisfaction. Minimizing delay should be a major objective in improving the performance of any networking device. Because delay plays a big role in users' technology preference, it should not be surprising that the most successful wireless data technology is the one providing the fastest speed, namely WLAN technology. Nevertheless, there seems to be a lack of understanding of the real delays in WLANs. There is talk of improving protocols, which are, firstly, hard to realize, and secondly, wouldn't necessarily improve total delay for infrastructure based WLANs. The access point is the part of the communications path that plays a central role in end-to-end performance by imposing relatively significant delay. Moreover, in wireless communications, wireless link connectivity is fundamental to other layers and their services. Wireless connectivity in WLANs is provided by access points, so the WLAN encompasses not only a link, but also a connecting node, which performs processing on the packet, hence consuming time. In this thesis, I have considered the WLAN AP as a system to be investigated and mathematically modeled. The major contribution in this research work is introducing analytic models that can be used for the enhancement of the quality of services over WLANs. The WLAN AP is firstly modeled as a queuing system, whose parameters can be calculated by analyzing experimental data of specifically designed tests. The queuing model of the AP is used for further modeling the AP as a data communications link. This link model of the AP enables an analytic formula for the throughput of WLAN APs. A key result is that the throughput of a WLAN AP is an increasing linear fractional transformation of payload. I further analyze the throughput formula of WLAN APs to model AP throughput using a feedback control system. The resulting throughput formula shows good correlation with real measurements. These results could form a basis for further simulation and traffic shaping.

ACKNOWLEDGMENTS

Firstly, I would like to thank my advisors, professor Rassul Ayani and professor Gerald Q. Maguire Jr.

Special thanks to Daimler Sweden AB for supporting me with different WLAN APs. I thank all the people who helped me and supported me throughout the work, in the good time and the bad time. In particular, I would like to thank Nicolas Baron, whose support has been beyond description, even when I needed money to rent or buy equipment for the research, Nicolas was there. In the same way, I thank Dave and Sue Peterson, who were always there for me when I needed support. I extend grand thanks to Jad Attalah, who always was there when I needed to talk about experimentation. I especially thank Jad for helping me remember the *truncate* function in C++, which helped me get better resolution in my results of the *ap-analyzer* program. Thanks to Tamer El-Nady for helping me draw some figures during the first phase of my work. I remain grateful to Adil El-Hassan for his help in searching for related work. Many thanks go to Sermed Al-Abbasi for his help with WLAN APs, and to Dr. Luc Onana for the fruitful discussions. I am grateful to Katherine Stuart, whose help in all aspects could not be forgotten. Thanks to Dr. Nil Tarim for her help in layout issues. I would like also to acknowledge my friends for their continual support and encouragement. In particular, I would like to thank Guy Davis, Khaled Khalil, and Ian Marsh. I also thank Enrico Pelletta, system admin at IMIT, KTH for supporting me with measured data on WLAN AP throughput. Thanks to Professor Axel Jantsch at IMIT, KTH, for the fruitful discussions on buffer size estimation. I also thank Hans Berggren, system admin at KTH, for his support.

Finally, I remain grateful to my family, who never hesitated to help me. I thank my father, who always supported me even at the stressful time when publishing his new books of poetry and plays. I also thank my mother, who was an endless source of information and advice throughout the phases of this work. Knowing how hard she had worked for her Ph.D. and endless publications made me always feel I had to do more. I thank my brother for showing willingness to help at any time and expense! Your winning of the best United Nations (UN) research paper award out of that large number of countries in year 2002 made me so happy and enthusiastic to work, I thank you for that and I always congratulate you for that. I thank my sister for her beautiful writings to me that made me feel so good and encouraged me to work more and more.

"All things began in order, so shall they end, and so shall they start again, by the will of the Ordainer of order and the mystical mathematics of the city of heaven."

Sir Thomas Browne

I dedicate this

To my family

and my advisors,

and I apologize for not having written, visited, nor socialized!

I have been so busy, as they taught me to be; as they tried to convince me how one should look like. Having raised and advised me, you probably guessed by now that I, on this to-be-quickly-flipped page, will use this not for what I've academically digested, but what is righteous and straight, as a scientist must be

Wondering is my core whether you could believe me had I stated that my self found no time for writing you letters, even the shortest there could be! But instead, I wrote you this long one, with a longer one promised to be

CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

ABBREVIATIONS

ACK	Link layer Acknowledgment in IEEE 802.11 standard
AP	Access Point
BSS	Basic Service Set in WLANs
CCK	Complementary Code Keying in IEEE 802.11 standard
CRC	Cyclic Redundancy Check
DBPSK	Differential Binary Phase Shift Keying
DIFS	DCF-IFS; Distributed Coordinated Function Interframe Space in IEEE 802.11 standard
DSSS	Direct-Sequence Spread Spectrum radio transmission technology in IEEE 802.11 standard
EDGE	Enhanced Data for Global Evolution
ETSI	European Telecommunications Standards Institute
FCFS	First Come First Served
FCS	Frame Check Sequence in MAC frame
FHSS	Frequency Hoping Spread Spectrum
FIFO	First In First Out
GFSK	Gaussian Frequency Shift Keying
GPRS	General Packet Radio Service
HIPERLAN	High Performance European Radio LAN, a set of WLAN communication standards primarily used in European countries
HomeRF	Home Radio Frequency protocol
IBSS	Independent Basic Service Set in WLANs
IEEE	Institute of Electrical and Electronics Engineers

IEEE 802.11	IEEE Standard for Information Technology- Telecommunications and information exchange between systems- Local and metropolitan area networks- Specific requirements
IEEE 802.11a	Supplement to IEEE 802.11, for Higher-Speed extension, version a
IEEE 802.11b	Supplement to IEEE 802.11, for Higher-Speed extension, version b
IEEE 802.11g	Supplement to IEEE 802.11, for Higher-Speed extension, version g
IFS	Interframe Space
IP	Internet Protocol
IPv4	Internet Protocol version 4
IR	Infrared radio transmission technology in IEEE 802.11 standard
ISM	Industrial, Scientific, and Medical frequency bands
LAN	Local Area Network
LCLS	Last Come Last Served
LIFO	Last In First Out
LSA	Link State Advertisement
MAC	Medium Access Control
MIPS	Million Instructions Per Second
MPDU	MAC Protocol Data Unit in IEEE 802.11 standard
NOP	No Operation *opcode* in a processor
OFDM	Orthogonal Frequency Division Multiplexing
OSPF	Open Shortest Path First routing protocol used for intra-domain routing for IP based networks
PC	Personal Computer

PDU	Protocol Data Unit in IEEE 802.11 standard
PHY	Physical layer in IEEE 802.11 specifications
PLCP	Physical Layer Convergence Protocol
PPDU	PLCP Protocol Data Unit in IEEE 802.11 standard
QoS	Quality of Service, a performance set of metrics
RTS/CTS	RTS stands for Request to Send, and CTS stands for Clear To Send. RTS/CTS is an access method used to solve the hidden node problem in IEEE 802.11 standard
SIFS	Short Interframe Space in IEEE 802.11 Standard
STA	Station in WLANs (Source or Destination)
SWAP	Shared Wireless Access Protocol defined for a broad range of interoperable consumer devices
TCP	Transmission Control Protocol, a transport layer protocol
UDP	User Datagram Protocol, a transport layer protocol
WLAN	Wireless Local Area Network

Software Abbreviations

C++	C++ computer programming language
MATLAB	MATLAB software and computer programming language
MGEN	The Multi-GENerator; an open source software by the Naval Research Laboratory (NRL) PROTocol Engineering Advanced Networking (PROTEAN) Research Group. Used to send UDP datagrams in testbed
tcpdump	TCP-dump program used on Unix operating systems to record network frames

Unit abbreviations

µs	microseconds
bps	bits per second
G	Giga, equals 10^9, but for bits and bytes (binary data storage) it is equivalent to 1K*1K*1K
Hz	Hertz, unit of frequency
K	Kilo, equals 10^3, but for bits and bytes (binary data storage) it is equivalent to 1024
M	Mega, equals 10^6, but for bits and bytes (binary data storage) it is equivalent to 1K*1K = 1024*1024
ms	milliseconds
nm	nanometer
s, sec.	seconds

Definitions

Bandwidth	Amount of data that can be sent through a given communications circuit per second
De	Deterministic (fixed) inter-arrival time
Downlink	The data communication path from Ethernet to WLAN is defined as Down
E_k	Erlang distribution of order k
Ge	General probability distribution
GI	General and Independent (inter-arrival time) distribution.
Jitter	Variation in latency
Latency or delay	Time for a packet to arrive at the destination or round trip

Ma	Memory-less or Markovian process, which mathematically means a *Poisson* process
Reliability	Bit error rates, bit loss, packet loss; Note that these terms are often mixed up, and the term QoS is sometimes used to refer to what we would rather call service guarantees or quality in general
Throughput	Amount of user data successfully moved from sender (source) to receiver (sink) per unit time
Uplink	The data communication path from WLAN to Ethernet is defined as Up

NOMENCLATURES

t_i^k	Time when the k^{th} packet is out on the i^{th} link along a path in the multi-packet test
∂b_a	Differential adaptive bandwidth in bits per second; partial incremental
∂P	Differential IP payload in bytes; partial incremental
q_l^k	Queuing delay due to other packets than the test packets in multi-packet tests
α	Access Point brand
ν	Fraction of *successfully received* user data bits to totals (overhead and data) transmitted on a link
η	Number of IEEE 802.11 link frames transmitted per second
$\tau(u)$	Time (in seconds) to transmit one IEEE 802.11 link frame with IP payload of u bytes
b	Bandwidth of the bottleneck (in bits per second) in packet pair queuing analysis
b_a	Adaptive bandwidth in Mbps
b_i	Bandwidth of the ith link on a path in the multi-packet test
B_j	Size of buffer (in bytes) at the j^{th} occurrence of loss; $j \in \mathbf{N}$
$b_{min(l)}$	Bandwidth of the smallest-bandwidth link l
BSE_loss_counter	Index in BSE algorithm to count number of losses each time an occurrence of loss happens
Buffer_test_success	Boolean variable, has value zero when BSE fails and value 1 when BSE succeeds

C	Output of feedback control model in bits
d	Nomenclature used to resemble destination in packet pair queuing (and to resemble departure in relationship to the queuing model of the AP). It also resembles the number of links on the path
dB	Unit of Decibel, used for the value of noise in GPRS experiments
d_i	Latency of the i^{th} link on a path in multi-packet tests
$diff_i$	Difference between two consecutive buffer-size estimates in the BSE algorithm; $i \in \mathbf{N}$
$DST(\alpha, x)$	Downlink Service Time of AP, α, for IP payload x
E	Error of the feedback control loop
EPCm	Ethernet PC of index m in testbed; $m \in N^*$
$f(u)$	Size (in bytes) of the link frame of IEEE 802.11 WLAN
G_f	Feed-forward function in feedback control system
H	Feedback transfer function in feedback control system
h	Constant parameter resembling the size of the IP header in bytes
H_u	Hurst Parameter used to quantify the self-similarity attribute
i, j, k, m, n, q, z	Indices; i, j, k, m, n, q and $z \in \mathbf{N}$
$INFO$	Boolean Variable informing BSE whether there is enough information to estimate allocated buffer size when loss occurs (value 1 means there is enough information, and value zero means there is no information)
IP header	Header control bits in the IP packet
IP payload	Amount of user data in IP packet
l	Link on a path in the multi-pair queuing tests

Last_before_loss	Index ($\in$ **N**) for the packet that just departed before the occurrence of losses in SSTP-1.3
L_j	Number of lost packets at the j^{th} occurrence of loss in SSTP-1.3
Loss_counter	Index ($\in$ **N**) for the number of losses when a loss occurs in SSTP-1.3
MAC header	Header control bits in MAC frame
max[.]	Maximum function, which finds the maximum of the values in brackets
MG_z	Multiplexing Gain for *z* independent video streams combined for transmission, z $\in$ **N**.
N	Set of Natural numbers
$\mathbf{N}^*$	Set of Natural numbers - {0}
N_j	Number of packets in buffer at the j^{th} occurrence of loss; *j and* $N_j \in \mathbf{N}$
P, u, x, X	IP payload in bytes
Packet_Size(X)	Function (in BSE) that calculates the total frame size of a packet carrying *X* bytes
P_i	Packet number *i*; with $i \in \mathbf{N}$
P_L	Lost Packet
P_R	Peak Rate for video stream
PS_X	Size of the frame (in bytes) carrying a payload of *X* bytes; in BSE
pta	Peak-to-average ratio of the video stream
PX	Packet size in bytes, used in video experiments
r	Incremental difference between average service time values
R	Input of feedback control system in seconds

R_i	Response time (in seconds) of packet P_i is the time from when P_i arrives at the system till the time it departs from the system
R_{L-1}	Response time (in seconds) for the last packet (P_{L-1}) that departed just before loss
S	Average service time of test packet in packet pair queuing for the AP link model case
Sample_space	Statistical number of service time values available of the total number of transmitted packets (lost packets not counted) in SSTP-1.3
S_{avg}	Average service time value read from SSTP algorithm for a packet with payload X bytes
S_i	Service time of packet P_i is the time from when P_i enters the server until it departs from the system
s_i	Size of the i^{th} packet (in bits) sent in multi-packet (or packet-pair tests
S_{L-1}	Service time (in seconds) for the last packet (P_{L-1}) that departed just before loss
S_n	Service time of index n (n^{th} service time)
S_o	Original average service time of the 40B IP payload packet
SPC	Sniffer PC used to record, filter and analyze traffic in testbed
$t(0,0)$, t_0^0	Time of transmission (in seconds) of the first packet in the packet-pair test
$t(0,1)$, t_0^1	Time of transmission (in seconds) of the second packet in the packet-pair test
$t(d,0)$, t_d^0	Arrival time in seconds (at the destination) of the first packet in the packet-pair test

$t(d,1)$, t_d^1	Arrival time in seconds (at the destination) of the second packet in the packet-pair test
$T_{a,i}$	Time of arrival of packet P_i
T_{APL}	Throughput of AP in Mbps
$T_{d,i}$	Time of departure of packet P_i
T_i	Time of arrival of packet P_i
T'_i	Time of departure of packet P_i
Total_loss	The total number of lost packets in an experiment in SSTP-1.3
UBR(u)	Utilized Bit Rate (Mbps) of a link for a payload of *u* bytes
$UDC(\alpha, x)$	Uplink Downlink Contrast of AP, α, for IP payload x
$UST(\alpha, x)$	Uplink Service Time of AP, α, for IP payload x
V_z	Link bandwidth required for multiplexing z video streams
W_i	Waiting time of packet P_i is the time from when P_i arrives at the system until it enters the server
W_{L-1}	Waiting time (in seconds) for the last packet (P_{L-1}) that departed just before loss
WPCq	Wireless PC of index q in testbed; $q \in \mathbf{N^*}$

Chapter 1

Introduction

"The web of our life is of a mingled yarn, good and ill together."

William Shakespeare

This chapter briefly introduces concepts, which underlie the work and present the motivation. This work began with my observations to the good and the bad behavior of a commercial access point. I will focus on some problems of access points in wireless Local Area Networks (WLANs). This choice was due to my belief in the future of this technology, which can carry interesting services to better the users' quality of life. This is already apparent in educational premises, where it has already had high rate of deployment. This chapter consists of three sections: section 1.1 describes my motivation that has been supported by my advisors, section 1.2 presents my contributions clearly, and section 1.3 introduces the organization of the remainder of the thesis.

1.1. Motivation

The last century has witnessed the birth of a revolution in data communications so that we started this century with the media discussing digital communication technologies using the term "explosion". One could not avoid phrases like "Internet explosion" [84]. I was fully determined to increase my knowledge base to what I could see of importance this "explosion" will have, for instance, in maybe curing people more rapidly. In this context, I found that the protocols and the basic work behind this famous "explosion" had been mostly done in the late nineteen seventies and the early nineteen eighties. However, high-speed wireless communications was new, and I thought it would be significant for the future. I chose IEEE 802.11 wireless Local Area Network (WLAN) access points (APs) since they provide relatively high speeds for wireless communications and were becoming widely available. Nowadays many educational institutions as well as companies and even public spaces utilize this technology to give wireless access to users [77]. However, a scientific look at performance aspects led me to discover that there was an element in the quality of WLAN networks that most of the scientists in the field had neglected, and that affects all mobile nodes: the WLAN Access Point (AP). It is true that WLANs can be formed via ad hoc networking, however, statistics show that the majority of WLANs utilize a WLAN AP to connect multiple users into a Local Area Network, , which is in turn connected to the Internet [91]. Studies also show access points being deployed in public, private, educational, or business premises [65]. In addition, in ad hoc networking, most of the problems that exist, from a performance point of view, are related to the IEEE 802.11 medium access and how the protocol deals with collision avoidance. However, when APs are deployed, the access point itself introduces new characteristics to the network adding relatively significant delay that will affect throughput. Simply, enhancing performance of wireless LANs that utilize APs require examining the behavior of the AP itself, because all traffic to the end user passes through it. It directly affects quality parameters, most importantly delay.

I decided to study the AP as a system. I looked for answers by asking people in the field and searching for information on WLAN APs. After months of research, I was shocked that those deploying/selling APs could not give me satisfactory answers to my questions about the behavior of APs. Thus, I was intrigued to investigate APs by myself so that I could provide answers to my questions. I found that when I asked about the AP, some answers were hardware related and others were about administrative issues. I would classify the latter issues as operating system or software related. To the best of my knowledge, few have looked at the interaction of both, hardware and software from a performance point of view. So, I decided to try to model the AP as a system, especially since my advisors and I were not able to find published literature giving such a model for WLAN APs.

1.2. Contribution

I was able to model the delay attribute of IEEE 802.11b WLAN APs. The AP model is a queuing system with a single queue and a single server for the uplink (WLAN to Ethernet) and downlink (Ethernet to WLAN) directions. To verify this model, the Simple Service Time Producer (SSTP) algorithm was designed and developed. My advisor professor Rassul Ayani, helped me verify the first version of the algorithm, which I, now, call: SSTP-1.1 (presented and published as SSTP at MobiCom 2002, Atlanta, Georgia, USA) [32] and is published in [31]. The latest version, SSTP-1.3, is an enhanced version of SSTP-1.1, which deals with packet loss and has refined sample spaces for statistical and probabilistic results. Versions 2 (published in [27]) and 3 (published in [30]) of the algorithm were designed and developed by me. The implementation in MATLAB and later in C++ was also totally done by me. Another contribution is the Buffer Size Estimator Algorithm (BSE), which I designed and implemented. The buffering considered in WLAN APs is never directly visible to end users, and hence remains a black box. Using my queuing model, the BSE algorithm estimates the initial buffer size allocation in APs as well as identifies buffer adaptation schemes [27]. A key contribution of the work is my analytic solution for the average service time that a packet consumes while passing through a WLAN AP [30]. I analyzed the average service time and showed it to be an increasing linear function of payload, where the parameters of the service time formula are mechanically obtained from the results of the tests using the aforementioned algorithms. The information that the BSE algorithm provides to users together with the service time formula can help them understand the behavior of multimedia traffic over WLANs [29]. In addition, using the service time formula and the power of the packet-pair techniques for FIFO-queuing networks [47], I found a new model for the AP as a link with adaptive bandwidth [28]. The advantage of this link model is the simple analytic solution it provides for the throughput of a WLAN AP. The throughput of a WLAN AP is found to be a linear fractional transformation of payload. The throughput formula shows good correlation with measurements. From this transformation, my second advisor professor Gerald Q. Maguire Jr. suggested that I try to find a feedback control model for the throughput of the AP, and so I did. The idea of the feedback model (i.e. to derive it from the transformation) was professor Maguire's idea, however, my contribution was to make this idea a reality [28], and I succeeded in doing so.

I also introduce the notion of the Uplink Downlink Contrast and define two kinds of contrasts: Convergent and Divergent [33]. The Uplink Downlink Contrast is a new QoS metric to evaluate WLAN APs, especially suitable for real-time bi-directional traffic.

Moreover, I developed the testbed and the methodology for testing and preparations. I have designed the tests, and Mr. Daniel Forsgren[1] helped, in the test design for the video over WLAN APs.

The programs of the SSTP and the BSE algorithm were implemented firstly in MATLAB and then in C++ and were totally programmed by me.

Prior to this work, there had been no logical/mathematical model for WLAN access points, and the major contribution in this research work is introducing a mathematical model that can be used for the enhancement of the quality of services over WLANs.

1.3. Organization of Thesis

The rest of the thesis is organized as follows: chapter 2 gives background knowledge of related fields and research. Chapter 3 presents the queuing model of WLAN APs and average service time. Chapter 4 discusses buffering in WLAN APs. Chapter 5 introduces a new model for throughput of a WLAN AP. The model presented in chapter 5 is the data communications link model. Chapter 6 discusses video results in WLANs as compared to GPRS video results. Chapter 7 presents some conclusions. Chapter 8 examines open issues and presents ideas for future work.

I expect the reader to have a good mathematical background and basic knowledge of TCP/IP networking. References are presented alphabetically and numbered accordingly using Arabic Numerals. Numbers between *parentheses* relate to *equations*.

---

1 Mr. Daniel Forsgren is a Doctoral student and is also a co-author of [30], [31], and [32].

Chapter 2

Background and Related Work

"All men by nature desire knowledge."

Aristotle

To the best of our knowledge, this thesis is the first work directed to modeling the behavior of an IEEE 802.11 AP as a queuing system. However, there has been lots of research on modeling 'systems' in general and 'communication systems' in particular [70]. The rest of this chapter presents the necessary background for the chapters that follow. Brief introductions to systems, modeling, queuing theory, and quality of service are presented. Section 2.1 discusses the logical concept of systems. In section 2.2, basic modeling issues are presented. Section 2.3 concentrates on queuing definitions and theory. Quality of Service is discussed in section 2.4. The IEEE 802.11 standard is briefly described in section 2.5. Section 2.6 introduces modeling of communications networks.

2.1. Systems

The objective of this section is to present a definition for the concept of a system, which best suits computer networking. The word *system* was mentioned a few times earlier, and I often found it in scientific literature while looking at related work. While I could always grasp what the word meant in the text I was reading, I never looked more deeply into the definition of a system until I tried to define my *to-be-modeled* system. The simplest way to introduce a definition of a *system* is to look the word up in the dictionary. Hence, I used the Webster dictionary [90], where I found a few definitions, each of which was best suited for a family of objects as classified by Webster (see Appendix 4). For data processing objects (our interest), Webster defines such a ***system*** as: *a group of devices or artificial* *objects or an organization forming a network especially for distributing something or serving a common purpose.* In the scientific versions of this definition, a system is nothing but a collection of *connected objects* that interact (*communicate*) to accomplish some *task(s)*. This latter definition is used for the "system" of interest in coming chapters.

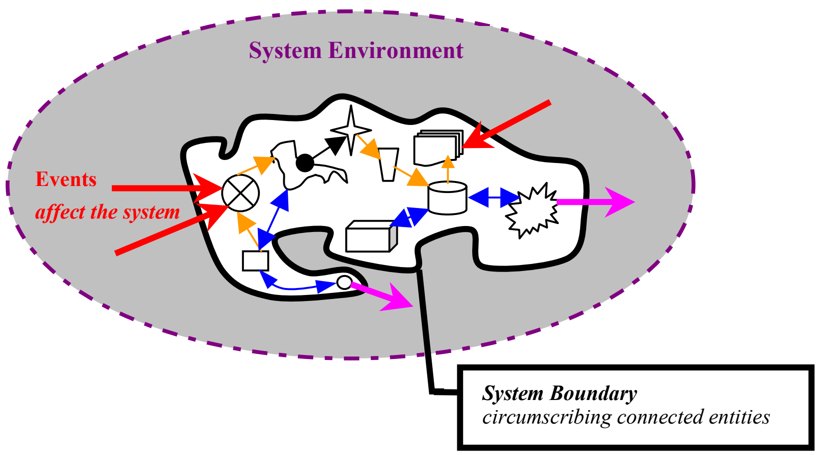

Figure 1. A system is a collection of interacting components, with a system boundary separating them from the system environment. Actions and events in the system environment affect the internal state of the system.

Nothing exists without a surrounding. When describing a system, it exists relative to its boundaries, which show the separation between the inside and the outside of the system. We have thus far focused on the system as defined by the inside of the system, i.e. the group of objects of interest. However, a system is often affected by actions happening outside its boundaries. According to [23], these external changes occur in the *system's environment*. Figure 1 shows a system with its surrounding environment. Changes that occur outside the system environment affect the system of interest only via events that cross the boundary. In order to understand how a system behaves, system constituents need to be defined. The following subsection discusses the components that make up a system.

2.1.1. System Components

All systems share a basic set of compoents, which must be defined in order to be able to understand the behavior of a system [4]. In fact, defining these components of a system and how these components connect is- by itself- the step which defines the system of interest. The rest of the work, related to how these connected components interact defines the behavior of the system.

The basic components of a system are: entity, attribute, activity, event, and state variables [40]. As a result of my literature review, studies, and research on systems, I present my definition for each component of a system:

Entity. A *system entity* is an *object* inside the system (inscribed by the system boundary as shown in Figure 1). By *object* I mean *something* or *someone* that is of interest to the model of the system. If one system entity, ceases to exist inside a system, then we have a new system that is different from the old system. For example, in a supermarket queuing, the customer is an *entity*.

Attribute. An attribute is a *trait/characteristic* of a system entity, hence it can also be referred to as an *entity attribute*. An attribute does not exist independently of an entity. An example of an attribute is the *amount of goods* the customer in queue wants to purchase.

Activity. An activity is a system component defined relative to time. It is an *amount of time* related to a state transition of some entity. In the supermarket example, an activity may be the *period of time* it takes a customer to purchase all the goods desired.

State. A state is a *mode* (*or condition*) of a part or the whole of a system. A system *can* be described by illustrating the interaction between its states (usually by using a state diagram). A state is described through *state variables* (*parameters*). The state variables are usually the interesting parameters, which- when collected- can describe the system. If the system is time dependent, then a snap-shot of the state variables at a moment of time describes the system at that moment. An example of a state is *the number of customers waiting* in a queue. Another example is defined by when a customer is being served.

Event. An event is an *action* that *may* affect the system by changing its state. If the event belongs to the environment of the system of interest, then it will affect the system. If the event does not belong to the system's environment, then it does not directly affect the system. When representing a system, it is enough to present the events within its environment. Events can be *classified* into two types: *external* events and *internal* events. External events are also known as *exogenous* [40] actions, which occur outside the system boundary, but within the system environment. Internal events are known as *endogenous* actions and occur inside the system. Both, external and internal events, affect a system by changing its state. Other terms that are synonymous to the term *event* are: *occurrence*, *action*, or *happening*. An example of an event is the *arrival of a customer to the queue*. Events can also be described through *event parameters*. An example of an event parameter is the *time of arrival* of a customer. In theory, an event is defined to be an instantaneous action. However, in realistic systems, this instantaneous property should be well handled when a system is being modeled. For example, if the arrival of a customer is an event, then this arrival consumes time by itself, which is the time period for the whole of the customer to arrive into the queue, assuming he enters with one leg first and the other leg follows. So, the customer arrives fully when both legs cross into the queue line. Consequently, we consider the whole process as instantaneous, and we only care about this customer when he is totally in the queue. The moments before the customer's complete entrance to the system are generally not of interest.

2.1.2. Classification of Systems

There are many classifications for systems, however, the most important for this thesis is the time-dependent classification since time is an important parameter in data communications systems. Consequently, I will start with time-related classification, and further the categorization of these systems [80] thereupon.

2.1.2.1. Time-Related Classification

From the temporal point of view, systems can either be *time-invariant* or *time-varying*. A **time-invariant** system is a system, whose event and state parameters do not change with time. Some electric circuits made of resistors and capacitors can be designed to be time-invariant. A **time-varying** system is a system, whose event and state parameters change with time. An example of a time-varying system is the supermarket queuing example, where state and event variables depend on the arrival time and the departure time of a customer.

Another time-related classification is that of *static* and *dynamic* systems. A **static** system is a system that is memory-less, i.e. its output at any instant of time depends at most on the input at the same time instant, but not on past or future samples of the input. An example of a static system is a data link, where the output of the link at one instant depends on what it gets as input at an earlier fixed instant. A **dynamic** system has memory, i.e. its output at a time instant depends on previous samples. An example of a dynamic system is a communication node, whose output scheduling at one instant depends on the average traffic it had received earlier; perhaps in order to adapt to varying traffic loads.

2.1.2.2. Discrete-or-Continuous Classification

A **discrete** system is that whose event and state variables do not change within a segment of a model. This segment could be a time period or any physical part of the model. For example, balls in an urn could be considered as a discrete system. A **continuous** system varies over the specified area. The area could also be a time span or a physical region. When the segment of interest is a time period, then we talk of a *discrete-time system* or a *continuous-time system*, which are the systems of major interest to this thesis. Data communication systems are often continuous-time systems since their inputs and outputs change continuously with time. It is worth mentioning that in real life, it is hard to have a wholly discrete or continuous system [40]. However, usually one type of behavior (discrete or continuous) dominates for most systems. Therefore we can assume the system to have the behavior of its dominant part.

2.1.2.3. Randomness-Based Classification

Classifying systems from a randomness point of view leads to two types: deterministic systems and stochastic systems. A system is said to be **deterministic** when a specific set of inputs always produces the same set of outputs. An example of a deterministic system is a machine where you input 2+3 and always get 5. On the other hand, a **stochastic** system introduces some degree of randomness to the output. A simple example of a stochastic system is the flipping of a coin, where the result is described with a degree of probability or chance.

2.1.2.4. Distribution-Based Classification

From a distribution point of view, there are two types of systems: *lumped* and *distributed*. In a **lumped** system, the model or part of the model is the same over the whole area. For example, a lumped model of a piece of metal assumes that the piece of metal has the same density over the entire piece, thus we do not care about variations. In a **distributed** system, if there were a number of different segments of the model, each segment should vary in its parameter-of-interest. For example, the density of a piece of metal could vary over the whole area of the piece.

2.1.2.5. Linearity-Based Classification

A system can be *linear* or *nonlinear*. A **linear** system satisfies the superposition principle, where the response of the system to a weighed set of inputs is the same as the sum of the correspondingly weighed individual responses of each of the inputs. A **nonlinear** system does not satisfy the superposition property.

2.1.2.6. Casualty-Based Classification

A system is said to be **casual** if its output depends on the current input and/or past inputs, but not future inputs. All real-time systems are examples of casual systems. A **non-casual** system is that whose output depends on future inputs. A non-casual system is harder to imagine than a casual one, especially if one thinks of time as the dependent variable. However, if one considers a parameter other than time to be a future parameter, such systems could be realized. For example, in image processing, the dependent variable might represent pixels to the left and to the right (i.e. *the future*) of the current position on the image, this constitutes a non-causal system [74].

2.1.2.7. Stability-Based Classification

A system is **stable** if a bounded input gives a bounded output. An example of a stable system, is a ball inside a box, where no matter how we shake the box, the ball will always be inside. If the output of a system diverges from a bounded input (i.e. grows infinitely), then the system is **unstable**. An example of an unstable system is a bottle on top of a wheel, where a small movement of the wheel will lead to the bottle falling.

2.2. Modeling

A model is a *representation of a system* aimed at studying the system [4, 40]. In modeling a system, it is necessary to carefully define the boundary between the system and its environment. The boundary depends on the desired parameters of study of the model, i.e. on the goal sought from the model. The simpler the model, the easier it is to use in practice. The main purpose of modeling a system is to be able to predict the behavior of the system under new environmental[2] conditions. In some cases, it is possible to experiment with the system itself, but in most cases we want to predict how the system would act under different events without having to impose new conditions on the real system, because it could be very costly. For example, if we want to improve queuing in a bank better, it is easier to model the system and try to analyze it under different arrival events than to try to change the actual queuing in the bank. Modeling for the sake of analysis can also lead to new system models that enables enhancement for better service. For instance, a traffic model summarizes the expected behavior of an application or a set of applications [73], which in turn can be used to enhance the performance of these applications.

---

[2] Environmental conditions refer to changes in system environment as shown in Figure 1.

It is important to note that a model, by definition, is a simplification of the system. Hence, one shouldn't expect the results of experimenting with the model to be exactly the same as reality. At the same time, the model should be detailed enough so that valid conclusions could be drawn about the real system [40]. Hence, it is very important to consider only the aspects that affect the goal of the model, i.e. the issue under study. Therefore, the same system may need different models depending on the different purposes. For example, if we aim at modeling delay in the supermarket system, we can use a delay-related model. In addition, we can use another model of the same supermarket, if we aim at investigating the inventory level of the system.

2.2.1. Types of Models

There are two types of models: *physical* and *logical* [40]. A **physical** model is a simplified realization of the real system. A **logical** model uses logical relations or mathematical equations to represent a system. The work in this thesis is focused on logical/mathematical modeling.

Classification of a model follows the system it refers to. Hence, the classification of systems that are discussed in section 2.1.2 applies to models. For example, if a system is time invariant, then the model of the system will be time invariant.

2.2.2. Model Components

As the model considers only those aspects of the system that are related to the specific goal (i.e. problem under study), a model is represented by some or all components as those discussed in section 2.1.1. In other words, a system must have the five components: entities, attributes, activities, states, and events, but a model of a system contains only a subset of these components that are related to the investigation.

2.3. Queuing Modeling

The intention of this section is to provide background knowledge about modeling of queuing systems [60]. One of the methods for analyzing systems (or special aspects of systems) is through modeling various aspects of the system. For systems that *store and* *forward* information, a good model for store-and-forward behavior is a queuing system (queues and servers) since the *storing* activity can be thought of as a *waiting* activity, and the *forwarding* activity can be modeled as a *serving* activity. This particular branch of modeling systems as queues and servers belongs to a more general and practical branch of mathematical study known as *queuing theory*[3], which has a large number of applications in the field of performance analysis [52].

In studying systems as queuing models, events and states may evolve randomly. Therefore, the mathematical fields of statistics and probability have to be used to quantify certain parameters so that the analysis of the system is mathematically tractable. In this thesis, I expect that the reader is generally familiar with statistics and probability. In particular, I assume that concepts of sample space, mean, variance, and standard deviation are familiar to the reader.

2.3.1. Queuing System Nomenclature

A queuing system is a group of entities to which another form of entities arrive according to an *arrival process* (*arrival event*) in order to receive service from the *service facility* (*server*) and then depart (*departure event*) upon completion of service by the server [60]. The service facility may consist of one or more servers, where each server can serve one arriving-entity at a time. The general term of the entity that the service facility consists of is *resource*, however, in this thesis, I will use the term *server*. If the server is busy, then an arriving entity joins a waiting queue of entities. The time when an entity receives service (enters the server) is dependent on the speed of the server (service rate), number of customers already waiting for service, and the queue management technique used. Figure 2 shows a schematic of a queuing system with one queue and one server. From a mathematical point of view, a queuing system can be broken down into three major components: input process, system structure, and output process [60].

---

[3] Queuing theory can be traced back in its origin to the Danish scientist and engineer, Agner Krarup Erlang (1878-1929), who discovered the need to understand the behavior of telephony networks and automatic dialing equipment during the early years of the previous century [12]. Thus, the interest in queuing theory was due to an interest in understanding and modeling communications networks at some time, and it is still very useful for the state-of-art communications.

2.3.2. Queuing Input Process

The input process to a queuing system has aspects: the size of the arrival population, arriving patterns, and behavior of the arriving entities [60].

2.3.2.1. Size of Arrival Population

The notion that is used in statistics and probability for the arrival population is the arrival *sample space*. This arrival sample space may be finite or infinite. An arrival sample space is finite in the sense that the arrival rate of entities is affected by the sample space. In other words, the total number of arriving entities is not too large compared to the number of entities that can be inside the system at a snapshot. An infinite sample space means that the number of arriving entities from external sources is large compared to entities that can be inside the system at a snapshot [60].

Whether the arrival sample space is finite or infinite, it has an impact on the queuing results. In some real communications systems, such as telephony networks, the arriving population is finite but relatively large, so they are treated as infinite for mathematical convenience.

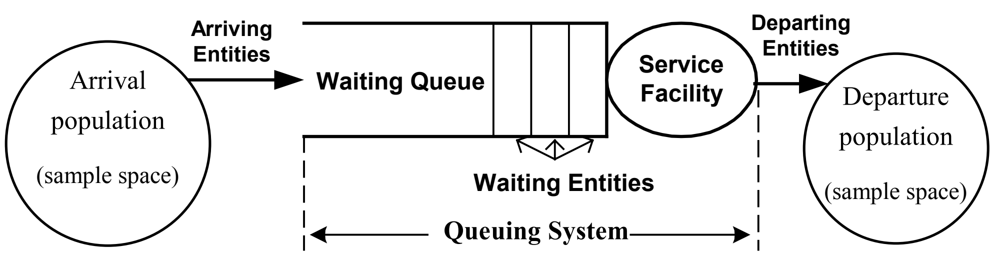

Figure 2. Queuing system with one queue and one server. *W* describes the queue, *S* the server, and *P* the waiting entity.

2.3.2.2. Arrival Patterns

Arriving-entities may arrive at the queuing system with a recognized (regular) pattern or in a random way. For a regular arrival pattern, we can describe the arrival process with one parameter: arrival rate [60]. For a random arrival process, one way is to try to fit a statistical distribution to the arriving pattern in order to generate several input sequences for the system. Some commonly used distributions in queuing theory are:

Ma: stands for *Memory-less* or *Markovian* process, which mathematically means a *Poisson* process;

De: stands for *Deterministic*, which mathematically means a fixed inter-arrival time;

E_k: stands for *Erlang* distribution of order *k;*

Ge*:* stands for *General* probability distribution;

GI: stands for *General and Independent (inter-arrival time)* distribution.

The distribution that is most commonly used in queuing theory is the Markovian (*Ma*) one, however, in data communication networks, we do not have Poisson arrivals, hence, we consider our systems in data communication to have a General (*Ge*) probability distribution.

2.3.2.3. Arriving-Entity Behavior

When entities arrive at a queuing system, they may behave differently, especially when the service facility is busy, or when the waiting queues are full (for finite waiting-queue systems). When an arriving entity leaves (or does not enter the waiting queue), because the service facility is busy and the queue is full, the entity is considered as a *lost* entity. Such a system is called a *blocking* system [60]. In some queuing systems, like telephony networks, the *probability of blocking* a call is one of the performance metrics used to evaluate the system. In data communication networks, other performance metrics are needed to suit traffic services that these networks offer.

2.3.3. *Queuing System Structure*

Two main parameters are important to understand the structure of a queuing system: the number of servers (with their relative layout) and the system capacity [60]. To further understand the behavior of a queuing system, one should gain knowledge of the arrival process (or behavior), queue management, and speed of the servers (service rate).

2.3.3.1. Service Facility Layout

The service facility may consist of one or more resources (servers). At the same time, servers can be arranged in different layouts. For instance, the service facility may consist of multiple servers in series, where a packet leaving one server enters a subsequent server and so on until it is served by the last server and departs. Another example of server layout is a parallel layout, where the service facility may consist of many servers in parallel and a single queue. In this parallel fashion, the arriving entity goes to the first server that is not busy.

2.3.3.2. System Capacity

System capacity is simply the maximum number of entities (from the arrival sample space) that a queuing system can have inside (including the waiting entities and the entities in the service facility). When there is no limit on the waiting queue size, then the system capacity is infinite and the blocking probability is zero. In our studying of data communication systems, we always have a finite limit on the buffer size for messages (or data packets) before they are served (and forwarded).

2.3.4. *Queuing Output Process*

Aspects that affect the departure (output) process are: the queuing discipline and service distribution [60].

2.3.4.1. Queuing Discipline

Queuing Discipline is defined as the way that waiting entities are selected to enter service. The most common queuing disciplines are:

- **FCFS**: First-Come-First-Served, sometimes also called FIFS (First-In-First-to-enter-Service);
- **FIFO**: First-In-First-Out, which is different from FIFS in that you can be served while another entity is in the service facility;
- **LCLS**: Last-Come-Last-Served, sometimes also noted as LILS (Last-In-Last-to-enter-Service);
- **LIFO**: Last-In-First-Out; (difference with LCLS is analogous to difference between FCFS and FIFO);
- **Priority**: a specific priority scheme for waiting entities is applied, where the waiting entities are classified into different groups with different assigned priorities. Entities with higher priorities are served first;
- **Process Sharing**: where the system capacity is divided amongst the waiting entities equally, i.e. when there are *n* waiting entities, the service facility devotes 1/*n* of its capacity to each entity;
- **Random**: where entities (to enter service) are chosen randomly.

2.3.4.2. Service time Distribution

When all waiting entities take the same time to be served, then the service pattern can be described by one parameter: service rate. However, in many systems, and especially in data communications, the service time for each entity is different. Hence, statistics and probability must be used. Types of service distribution are: M, D, E_k and G, as described in subsection 2.3.2.2.

2.3.4.3. Quality of Service

One of the most confusing topics in data networking today is Quality of Service (QoS) [63]. However, it seems that engineers and technicians use the term without pondering the exact definition, so that QoS has become a common word when talking about communications. In fact, QoS has different meanings for different people, i.e. it is to some extent subjective. However, a common level of understanding of QoS should be realized before starting any research that is dependent on this term. To do so, let us study the two words, *quality* and *service*, in communications. Quality has many meanings, but engineers generally use quality to describe the delivery of information in a reliable manner. Reliability here could also be subjective. On the other hand, the word service is with reference to the organization or the system that it belongs to. In data communications, it is usually used to describe what is offered to the end user, such as client-server applications [60]. Combining the two words in data communications could lead to ambiguity as well. However, to overcome this problem, QoS should be broken down to parameters. Hence, in this thesis we refer to the Free Online Dictionary of Computing [21], which defines QoS as *the performance properties of a network service, possibly including throughput, transit delay, and priority.* The QoS parameters as described in [62] by Ferguson and Huston are:

- **Latency or delay**: time for a packet to arrive at the destination or round trip;
- **Jitter**: variation in latency;
- **Bandwidth**: the amount of data that can be sent through a given communications circuit per second;
- **Reliability**: bit error rates, bit loss, packet loss; Note that terms are often mixed up, and the term QoS is sometimes used to refer to what we would rather call service guarantees or quality in general.

In addition, customer satisfaction plays a significant role in knowing the desired QoS, for instance through expectations, fulfillment, business models, charging models, service level agreements, and pricing.

If you have the possibility to guarantee a QoS level by controlling QoS parameters, then it is possible to offer service guarantees to applications or users. At the IP level this means that one wants to prioritize packets coming from different applications (identified by port numbers) or from different users. For example interactive services want low delay, while file transfer services want error free transmissions and real time applications require that the delay is less than an acceptable upper bound. Some Internet protocols (DiffServ) allow packets or streams to include QoS requirements.

The term QoS often relates to service guarantees, which are delivered to the end user through service providers. From the point of view of the network server, network management is essential to guarantee reliability, control bit errors, and delay [34]. Hence offering a guaranteed service means that QoS parameters of the network are consistent and predictable, which is a major engineering challenge in the world of packet based networks [63].

2.4. IEEE 802.11 Standard

There are seveal WLAN approaches such as HomeRF[4], ETSI's HIPELAN[5], IEEE 802.11, etc. WLAN standards cover the physical layer (PHY) and medium access control (MAC). However, in this thesis we only study access points that utilize the IEEE 802.11 WLAN approach, because it has been the most popular [71]. The IEEE 802.11 WLAN protocol has three major versions: IEEE 802.11b, IEEE 802.11a, and IEEE 802.11g. These three versions differ in their encoding methods, their internal logic, and the radios they use to broadcast at the required frequency for each version [71]. In this thesis work all the research was conducted with IEEE 802.11b WLAN APs. We chose this version, because currently it is the most popular of the three versions [71].

The basic differences, from a performance point of view, between these three versions is the operational speed or the transmission rate. IEEE 802.11b has a theoretical bit rate of 11Mbps, operates in the unlicensed 2.4GHz frequency band, and uses Complementary Code Keying (CCK) Direct-Sequence Spread Spectrum (DSSS) radio transmission technology. IEEE 802.11g uses more efficient encoding to increase the transmission rate to 54Mbps. IEEE does not expect to formally adopt an 802.11g standard until May 2003 [71]. There are now two sub-versions of IEEE 802.11g: one uses CCK and operates in the 2.4GHz range (i.e. it is compatible with IEEE 802.11b), and one uses Orthogonal

4 The HomeRF (The Home Radio Frequency) Working Group (HomeRF Resource Center, http://www.palowireless.com/homerf/about.asp) has developed a specification (Shared Wireless Access Protocol-SWAP) for a broad range of interoperable consumer devices. SWAP is an open industry specification that allows PCs, peripherals, cordless telephones and other consumer devices to share and communicate voice and data in and around the home without the complication and expense of running new wires. The SWAP specification provides low cost voice and data communications in the 2.4GHz ISM (Industrial, Scientific, and Medical) band.

5 HiperLAN is a set of WLAN communication standards primarily for European countries. There are two specifications: HiperLAN/1 and HiperLAN/2. Both have been adopted by the European Telecommunications Standards Institute (ETSI). The HiperLAN standards provide features and capabilities similar to those of the IEEE 802.11 WLAN standards. HiperLAN/1 provides communications at up to 20 Mbps in the 5-GHz range of the radio frequency (RF) spectrum. HiperLAN/2 operates at up to 54 Mbps in the same RF band. Neither has been widely adopted in the market place.

Frequency Division Multiplexing (OFDM) and operates in the 5GHz frequency range. The use of OFDM raises the transmission rate to 54Mbps.

Like IEEE 802.11g, the IEEE 802.11a uses OFDM, operates in the 5GHz range, and has a theoretical bit rate of 54Mbps, but does not have a sub-version that is compatible with IEEE 802.11b. For more information on the versions of the standard please refer to [35, 36, 37, 38].

2.4.1. General Description of the WLAN IEEE 802.11 Standard

The IEEE 802.11 Standard covers two networking layers for WLANs: the PHY (physical) and MAC layers. There exist major differences between WLAN networks and wired networks. For instance, in wired networks, the address of the source/destination is the equivalent of a physical location. However, in WLANs, the address is for a station (STA), which is a message destination but *not always* a fixed location [35].

The physical layers (PHYs) used in IEEE 802.11 also differ from those in wired media in the sense that IEEE 802.11 PHYs:

- utilize a medium that has no further boundaries outside of the reception and transmission range,
- have dynamic topologies, and lack full connectivity (so unlike wired networks, the assumption that each STA can listen to all other STAs is not always true),
- are unprotected from outside signals (as they are not in a shielded cable), hence communicate with less reliability than in wired PHYs,
- have time varying and asymmetric propagation properties, and

- have to handle *mobile* and *portable*[6] stations, hence propagation effects in this case can blur the distribution between portable and mobile stations. Moreover, battery power encourages lowering power consumption.

The IEEE 802.11 architecture is made of several components that interact to provide WLAN connectivity. The Basic Service Set (BSS) is the basic building block of an IEEE 802.11 LAN. The coverage area of the BSS is the Basic Service Area (BSA) [6]. A WLAN station (STA) is a member in the BSS if it is in its BSA [26]. There are two types of network architecture for IEEE 802.11: ad hoc network and infrastructure network. In ad hoc networks, the grouping of STAs into a BSS requires no infrastructure deployment leading to what is known as independent BSS or IBSS. Figure 3 shows an IBSS (WLAN ad hoc network).

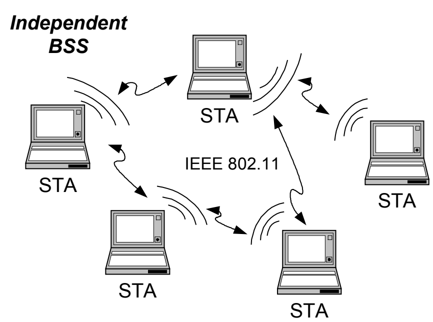

Figure 3. Ad hoc network of IEEE 802.11 WLAN STAs (IBSS).

Infrastructure networks utilize WLAN APs to connect a group of STAs to a wired infrastructure. Figure 4 shows an infrastructure network for an IEEE 802.11 AP, where the BSS encompasses STAs that are within the range of transmission and reception of the AP. The AP operates in a manner analogous to the operation of a base station in cellular telephony systems [26].

---

[6] A portable station can be moved from one physical location to another, but is only used while at a fixed location. On the other hand, a mobile station can access the network while in motion.

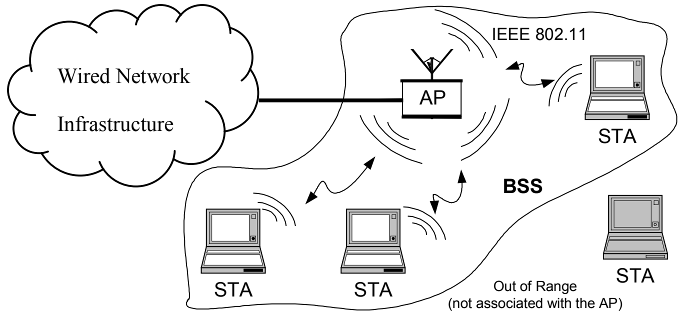

Figure 4. Infrastructure network of IEEE 802.11 utlizing a WLAN AP forming a BSS.

The *association* between an STA and a BSS is dynamic, i.e. an STA can be turned off, turned on, come within range, or go out of range [35]. To become a member of an infrastructure BSS, an STA must be *associated*.

2.4.2. IEEE 802.11 Physical Layer (PHY) Radio Technology Specifications

The IEEE 802.11 standard specifies three different PHY implementations: FHSS (Frequency Hoping Spread Spectrum), IR (Infrared), and DSSS (Direct Sequence Spread Spectrum). The FHSS operates on the 2.4 GHz ISM band (2.4000-2.4835GHz) [6]. The first channel has a central frequency of 2.402GHz, and all channels are 1MHz apart. The channel separation corresponds to 1Mbps of instantaneous bandwidth using two-level Gaussian Frequency Shift Keying (GFSK). The enhanced 2Mbps has 2 bits encoded at a time, by using four-level GFSK [6]. The IR implementation specifies a wavelength range from 850 to 950nm, and it is designed for indoor use only [6]. IR enables stations to receive line-of-site and reflected transmissions, with a bit rate of 1Mbps using 16-Pulse Position Modulation (PPM) and 2Mbps for the enhanced version using 4-PPM.

The IEEE 802.11b uses DSSS implementation; hence, I will concentrate on DSSS implementation. For more information on the other implementations and/or on DSSS itself, please refer to [35] and [37].

2.4.2.1. DSSS PHY

The DSSS PHY uses the 2.4GHz ISM band. Its 1Mbps basic rate is encoded using Differential Binary Phase Shift Keying (DBPSK). Its Enhanced 2Mbps rate uses Differential Quadrature Phase Shift Keying (DQPSK). Eleven sub-channels are utilized, each of which is 11MHz wide. Hence, the maximum channel capacity is 1Mbps if DBPSK is used [6], while IEEE 802.11b allows speeds of 5.5Mbps and 11Mbps [37].

2.4.2.2. DSSS Physical Layer Convergence Protocol (PLCP)

In DSSS, the PLCP (Physical Layer Convergence Protocol) provides a procedure where the MAC Protocol Data Unit (MPDU) can be converted *to and from* PLCP Protocol Data Unit (PPDU). Figure 5 shows the PLCP frame format, i.e. the PPDU. The receiver of the frame processes the PLCP Preamble and PLCP Header in order to aid in demodulation and delivery of the MPDU [35].

The PLCP Preamble and Header are transmitted using the 1Mbps DBSK modulation, and all transmitted bits are scrambled. A brief description of the fields of the PPDU follows, for more information please refer to [35]:

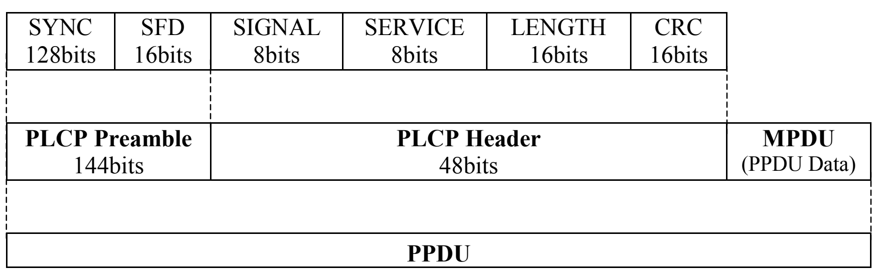

Figure 5. DSSS PLCP frame format (PPDU). The MPDU is the PPDU data field

- The SYNC filed consists of 128bits of scrambled 1 bits and is used for synchronization at the receiver;
- The SFD is a 16bit field, whose function is to indicate the start of PHY dependent parameters within the PLCP preamble;

- The SIGNAL field indicates the modulation that shall be used for transmission and reception of the MPDU;
- The 8bit Service field is reserved for future use;
- The 16bit length field is an unsigned 16-bit integer indicating the number of microseconds required to transmit the MPDU;
- The CRC field is a 16 bit frame check sequence, which protects the SIGNAL, SERVICE, and LENGTH fields.

2.4.3. MAC Sub-layer of IEEE 802.11

The MAC sub-layer in IEEE 802.11 deals with channel allocation procedures, protocol data unit (PDU), fragmentation, and re-assembly. The IEEE 802.11 supports three different types of frames: management, control, and data [6]. Management frames are used for association and disassociation with the AP, timing, synchronization, authentication, and de-authentication. Control frames are used for handshaking, positive acknowledgments (ACKs), and to end the Contention Free Period[7] (CFP). Data frames send data and can be combined with polling and ACKs during the CFP. Figure 6 shows the IEEE 802.11 MAC frame format.

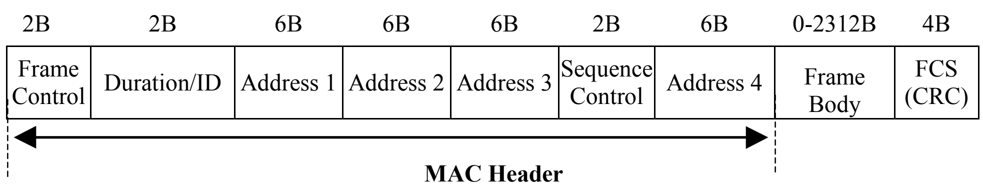

Figure 6. IEEE 802.11 MAC frame format (MPDU of the IEEE 802.11 DSSS PPDU). For more information on the Cyclic Redundancy Check (CRC) and the Frame Check Sequence, refer to section 6.2 in [89].

[7] CFP or Contention Free Period in IEEE 802.11 is the period of time when the medium usage is controlled by the AP, hence eliminating the need for STAs to contend for channel access [6].

The basic access method for IEEE 802.11 is the Distributed Coordination Function (DCF) that is based on CSMA/CA[8] [43]. CSMA/CD is not used, because an STA can not listen to the channel for collisions while transmitting. The STA senses the medium to check if it is idle so that it can transmit. If the medium is not idle, (i.e. busy is medium), then the STA waits for a period of time before checking again for the idle state, by entering a random back off procedure [35]. Figure 7 shows the time line for the DCF access method. DCF sits directly on top of the PHY and supports contention services [6].

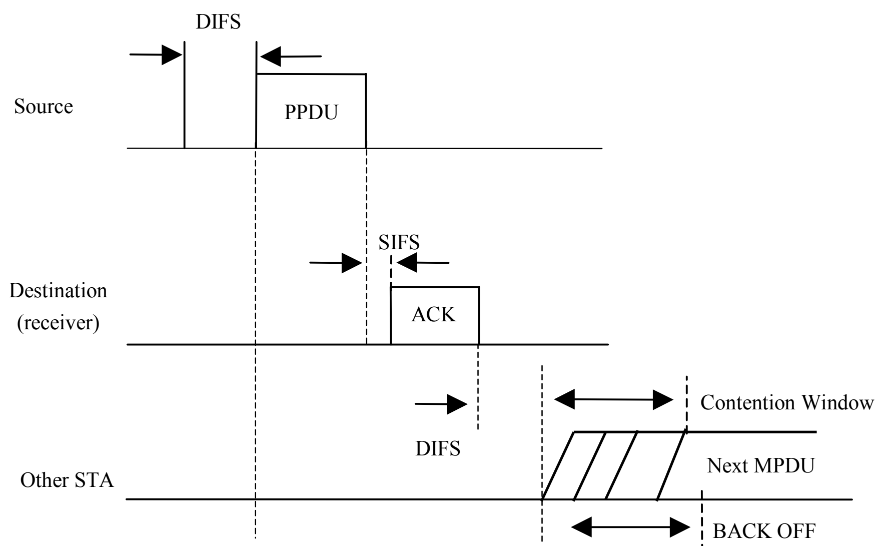

Figure 7. IEEE 802.11 DCF basic access method.

DCF is obligatory for all STAs; however, an optional access method exists as an extension to DCF, namely PCF (Point Coordination Function). PCF functions in conjunction with DCF were introduced so that it could be used for specific services like wireless multimedia applications. In ad hoc networks, only DCF can operate, however, in an infrastructure (i.e., with AP deployment) networks either DCF operates alone or coexists with PCF. In the work conducted in this thesis, only DCF was used; details on PCF can be found in [35].

[8] CSMA/CA stands for Carrier Sense Multiple Access / Collision Avoidance technique.

Priority access to the medium is controlled by the use of Interframe Spaces (IFSs). The fields used in order to complete the transmission of one MPDU (desired MAC frame) are:

- *DIFS*, standing for the DCF-IFS (Distributed Coordinated Function Interframe Space), which takes 50μsec per MAC frame;
- *PLCP Preamble and Header*, which is 272 bits per MAC frame (Figure 5);
- *SIFS*, which is the Short Interframe Space used for special acknowledgments, and it adds 10μsec per MAC frame. SIFS has the highest priority access;
- *ACK*, which is the link layer Acknowledgment indicates that the frame transmission was successful, and effectively adds 304 bits per MAC frame.

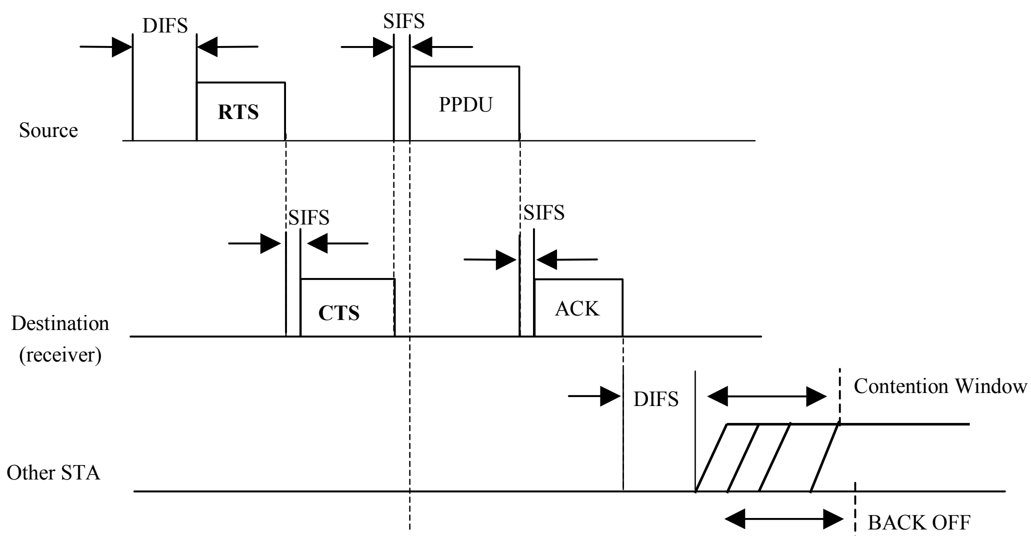

Figure 8. RTS/CTS access method for IEEE 802.11.

The DCF method described above depends on the ability of each STA to sense signals from other STAs within the BSS. However, this assumption is not always true. For instance, a station, *x*, could be out of reach of another station, *y*, while an AP can reach both. Hence, *x* and *y* could not detect each other's transmissions, thus leading to increased collisions. This problem is known as the *hidden node problem*. To solve the problem, a second method is described in the IEEE 802.11 standard, namely RTS/CTS[9]. RTS/CTS reserves the medium for a specified period of time before stations are allowed to transmit. This is done by having the STA transmit an RTS frame to the AP. All STAs that are able to listen to the RTS will have to wait for a period of time specified within the RTS frame to reserve the medium. When the AP receives the RTS frame from station, *x*, it sends a CTS reply to *x*, which other STAs can sense in order to read the reservation period sent by the AP in the CTS frame. For station, *y*, which can't listen to *x*, it will then not hear the RTS sent by *x*, but will hear the CTS sent by the AP, and hence learns the reservation period from the CTS and waits. Consequently collisions are avoided in a *hidden node* situation. Figure 8 shows the time line for the RTS/CTS access method. Other issues related to security and power management in the IEEE 802.11 are out of the scope of this thesis, interested readers should refer to [35] and [37].

2.5. Network Modeling Related Work

The major work related to my model is the research made by J-C. Bolot on end-to-end delay over the Internet [44]. Bolot used the measured round trip delays of small UDP packets sent at regular time intervals over the Internet, and he analyzed the packet delay and packet loss. The experiments were designed by varying the interval between the packets so that it is possible to study load behavior over different time scales. The main observation of Bolot was compression of sent packets and rapid fluctuation of queuing delays over small intervals. One more interesting result was that the loss in packets sent was random unless the traffic sent was utilizing a significant fraction of the available bandwidth. Bolot stretches his analysis to use his results for the design of control mechanisms for the Internet.

The similarities in [44] and my proposed work lie mainly in the model: the single server, single queue system. Moreover, in [44] the objective was to understand the packet delay

[9] RTS stands for Request to Send, and CTS stands for Clear To Send.

and loss behavior in the Internet. Similarly, my objective was to investigate delay and packet loss over the wireless LAN access point as a system. Understanding these performance parameters is essential for the proper design of algorithms of flow control. Moreover, this type of modeling helps choose parameters in both simulation and analytic studies. A key issue in both investigations is that they are essential for designing multimedia applications. For example, Bolot notes that the shape of the delay distribution is crucial for the proper sizing of playback buffers [44].

The most related work from the point of view of the wireless LAN access point is the investigation by Enrico Pelletta [19] on throughput of different brands of access points. The results in [19] are very valuable for comparison with our model for throughput of the wireless access point. Pelletta performed tests on different IEEE 802.11b access points for the downlink and uplink. His results have shown that the uplink and downlink throughput values are not identical for the access points investigated. Throughput results using our model will be compared with his results (sections 2.1 and 2.2 in [19]).

Another important related investigation is the work on scalable bridge architecture, by Thomas Rodeheffer et al. [76], where a new architecture for bridges was introduced: the SmartBridge. This new architecture combines the good features of IP routing and spanning tree bridging. The important part of this investigation for our work is the section related to the throughput of the bridge.

In [5], the authors deal with OSPF[10] measurements as Black-box measurements. My work does not deal with OSPF, but the treatment of the subject of study as a black box is interesting for the work on my model, which considers the access point as a black box. In my investigation I run external measurements (outside the black box) on time and delay, and analyze them to be able to derive internal parameters of the detailed model. Similarly, [5] presents black box external measurements for estimating delays for key internal tasks in OSPF. For example, the authors use the external measurements to estimate delays such as: processing Link State Advertisements (LSA), performing Shortest Path First calculations, updating the Forwarding Information Base, and flooding LSAs. The measurements were made on different Cisco routers.

Lai and Baker in [47] present a deterministic model of packet delay, which was very useful and inspiring for my work. The authors also derive packet pair property of FIFO queuing networks and a new technique for actively measuring link bandwidth. This work coincides partially with a portion of my thesis where I intend to use packet pair techniques for my FIFO queuing model of the wireless access point (see Section 5.2). This paper together with the studies on *packet streams sent over a path* and *the corresponding delays and losses* presented in [1, 8, 24, 42, 57, 66], constitute a basis for the understanding and development of the link model presented in Chapter 5.

In [3], throughput analysis of IEEE 802.11 wireless LANs was evaluated in relationship to link layer overhead. Sources of overhead were defined to be: gap time (which is the IFS), header fields for the PHY and the MAC layers, ACK frames, and the TCP. The authors of [3] measure the throughput of 2.4GHz products. After measurements and monitoring of actual exchange of frames, modeling was used. The close fit between the real measurement results and the modeling for currently available 2.4GHz product allows an accurate prediction for enhanced 2.4GHz versions and extends to further 5GHZ products. Throughput analysis of IEEE 802.11 LANs is crucial to understand my work.

In [25], the authors conduct a performance evaluation of the asynchronous data transfer protocols that are a part of the IEEE 802.11 standard [37] taking into account the decentralized nature of communications between stations, hidden stations, and the possibility of a node capture. The authors calculate system throughput for the purpose of evaluating the impact of spatial characteristics (like room architecture) on the performance of the system. The work is important for understanding the access methods in IEEE 802.11, which will affect the way I build and design the physical arrangement of nodes for experiments.

Stine presents in [39] an investigation that seeks design parameters for an IEEE 802.11 network, at both the physical and the protocol level. The author presents a design methodology and validates the performance of the network using simulation. The methodology is important for my work since I also present new methodology for testing as input to my analysis.

---

[10] OSPF or Open Shortest Path First is a widely known and used intra-domain routing protocol for IP based networks.

The IEEE 802.11 protocol is well presented and summarized in a chapter by B. P. Crow et al. [6] with particular emphasis on the medium access control sub-layer. Performance results are provided for packet-based data. The investigation in [6] shows that IEEE 802.11 networks may be able to carry traffic with time-bounded requirements using the point coordination function. This paper, together with, [2, 11, 22, 49, 67, 79, 82, 83, 85] form a good basis for understanding details of the link layer in IEEE 802.11, and for investigating the utilized bit rate of the WLAN medium as presented in Section 5.3.

In [45], the authors propose a transmission scheme to enhance the system capacity of wireless LANs. The scheme is the frame-based adaptive multi-rate transmission scheme. The throughput and delay were evaluated using simulations and the results show that they can be significantly improved compared with those of a single rate WLAN. This work helped me deepen my understanding of the delay process in IEEE 802.11.

Investigations in [10] concentrate on the performance of several models in IEEE 802.11. The model they used is called Message Retraining model, and it can be employed in situations where varying signal strength is expected to impact system performance. This method has a specific relevance where nodes in a given topology are unable to sense carriers from neighboring nodes. The model may also help in developing quality of service mechanisms for IEEE 802.11 wireless MAC protocols, which is the part that is interesting for my work.

In [56], protocols like IEEE 802.11 and GPRS are studied for vertical handoffs[11]. Simulation was used and the results were related to throughput and handoff delay in a vertical and horizontal handoff[12] in IEEE 802.11 and GPRS/EDGE networks. The results showed that the number of users affect the handoff parameters.

The explanation and work presented in [13] on multiaccess communications make a good basis for the understanding of delays in shared networks. In chapter 4 of [13], Bertsekas and Gallager present delay models for CSMA and other multiaccess communications schemes. These models helped shape my perception of the medium access and its delay attributes in WLANs.

---

[11] A *vertical handoff* is a handoff between stations that are using different wireless network technologies [54].

In [46], packet shapers are discussed, and some theorems are discussed for packet reshaping in cases of variable length packets. This helped my understanding of "packetization" and the theorems discussed were helpful for my investigation of variable packet effect on throughput and delay in wireless LAN access points.

In [59], the investigation of Multicast Inference of Network-internal Characteristics (MINC) developed and deployed methods to determine performance in the interior of a network from edge measurements. This idea is similar to my idea of calculating internal parameters of the desired system from external or edge measurements.

[12] A *horizontal handoff* is a handoff between stations that are using the same type of wireless network interface [54].

Chapter 3

Modeling WLAN Access Points as a Queuing System

"Everything should be made as simple as possible, but not simpler."

Albert Einstein

This chapter discusses the logical model of WLAN APs. Our main thrust in this chapter is to understand the behavior of the delay attribute in a WLAN AP for IEEE 802.11b. The derivation of the model is presented in detail as a queuing system. We use experimental tests that we designed for extracting the parameters of the AP model. Section 3.1 motivates modeling of WLAN APs and introduces the reader to general concepts in modeling. Section 3.2 presents the mathematical model proposed. Section 3.3 describes the experimental environment. Section 3.4 discusses the test design and algorithm used in calculating the model parameters . In section 3.5 we analyze the results to show the differences in performance of APs. Section 3.6 introduces the notion of Uplink-Downlink Contrast as a new performance parameter for WLAN APs. Section 3.7 concludes the chapter.

3.1. Introduction

Before modeling a system, its characteristics and parameters must be well defined. In fact, any group of elements can constitute a system as discussed in section 2.1, and thus any group of systems can be taken to be a system in itself. For systems that provide service, the service must be defined. Examples of services are processing, transmission, or management of traffic flow. The activity of interest is the storing and forwarding of packets in the WLAN AP. Therefore, the first idea to occur in one's mind is a queuing system. It could be a single queue and a single server or multiple queues and multiple servers. This chapter presents the queuing model of WLAN APs for IEEE 802.11b. Our model can be used to analyze and compare the performance of different WLAN access points. In this chapter, we focus on one main parameter: the delay introduced by a WLAN AP.

To provide suitable service, an understanding of the behavior of WLAN access points is essential. To understand how the performance of a system could be enhanced, the first step is to define the system of interest [4]. The advantages of our model are manifold; ranging from the ability to compare the performance of different APs, to the simple parameterization of the average time required to serve (forward) a packet passing through the AP. Moreover, our model is relatively simple, and having a simple model to represent a system is an advantage, especially for manufacturers of access points and for marketers, who can easily understand the difference service levels for different WLAN APs.

The key result presented is an analytic model for the average service time of WLAN APs in terms of payload size. Hence, the developer or the user of the access point can get a good estimate of the average time that a packet will need to be served by using a mathematical formula rather than running tedious measurements. Further analyses (Chapter 5) provides an analytic model for the throughput of a wireless LAN access point in terms of payload.

3.2. System of Interest

In our investigation, we try to understand the behavior of the delay attribute in WLAN APs. To create the theoretical model, some assumptions are made. The assumptions about the system should be carefully made since the end result, if used in industry, may influence practical engineering decisions as discussed in [92]. We isolate the AP and define the different events that occur. We consider the AP as a black box and define three events: *arrival, departure, and noise* (Figure 9). Since we are interested in the pure behavior of the system as a store and forward box, we study the system under noiseless conditions, hence we consider, for this investigation, that the *noise event* does not exist in the system environment[13]. This is similar to benchmarking of systems, where the system best-effort is studied (in a noiseless environment) to understand its maximum capabilities; for example,

[13] System environment is discussed in Section 2.1.

benchmarking a microprocessor with millions of NOP[14] opcodes in order to calculate its maximum MIPS[15]. The noise event shown in Figure 9 is just for illustration, i.e. it will not be considered in the parameterization of the performance of APs, because we look at the best-effort of an AP in order to compare it with the best effort of other APs.

When a packet enters the system, the parameter of interest concerning the arrival event is the time of arrival. Similarly, when a packet departs, we are interested in the departure time. We relate everything in the study from the point of view of the AP. Hence, when we define the *total delay* of a packet, which we refer to as the *response time*, to be the time difference between the departure time and the arrival time of the packet.

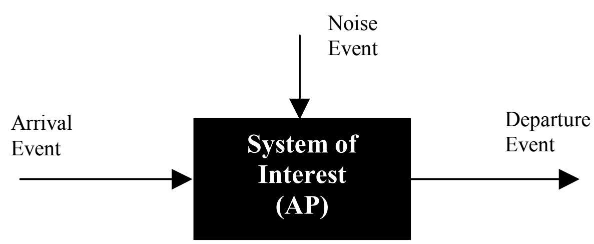

Figure 9. Logical model of the system of interest (AP) and events acting on it. We assume noise event effects to be nil, but is marked here to show all events that may act on the system.

The arrival and departure events are instantaneous events; however, as discussed in Section 2.1.1 the instantaneous property should be well defined when modeling a system. For our system (Figure 9), a packet is considered to have arrived at the system when all the bits of the packet are inside the system. Thus, the time span that the bits of the packet consume in order for the whole packet to arrive is neglected, and the arrival is considered instantaneous by taking the arrival time to be the instant of time when the entirety of the packet has entered the system. Similarly, the departure event is also considered an instantaneous event, and the time of departure is considered to be the instant of time just when the last bit of the packet has departed the system. In other words, the time spent for the bits of the packet to be transmitted is considered part of the delay of the system. By

---

[14] NOP is a microprocessor command, which does nothing but an opcode fetch, i.e. no operation.
[15] MIPS stands for Million Instructions Per Second, and it is the quantified unit used to measure performance of microprocessors.

this, we have defined two instantaneous events (arrival and departure) based on the exact instant of time when each event is triggered.

The logical packet (the entity that the system works on and hence delays) is the link layer frame, which carries an IP packet within its frame body. Figure 6 shows the MAC frame for IEEE 802.11. The description of the fields is defined in the IEEE 802.11 standard [35].

Inasmuch as time is an important factor in the study of our system, it is crucial to state whether the system is a discrete-event or a continuous-event system. The change in the number of packets (entities) inside the system is the factor that decides on the discreteness or continuity of the system [40]. In packet based data communications, there are often interframe spaces between packets transmitted over the same medium. More specifically, in WLANs, access to the transmission medium is controlled by the IFSs (see Section 2.4.3) in order to avoid collision. This time difference leads to packets being put on the medium at separate time instances, i.e. no two packets coexist at the same time unless there is a collision (which leads to loss of colliding packets). Thus, there is a time difference, which results in packets arriving at the system at separate points in time. Similarly, this leads to packets leaving the system at separate points in time. Since the number of packets inside the system changes when a packet arrives or when a packet departs, i.e. at separate instants of time, the system is a discrete-event system [40].

Since the wireless access point can forward traffic in two directions, we consider two cases: one where the traffic travels from the Ethernet side to the WLAN side, and another from WLAN to Ethernet. Modeling the WLAN STA to WLAN STA communication is also of interest, but not presented in this thesis since this does not traverse through the AP. Therefore, we define two traffic flows: *downlink traffic flow* (from Ethernet to WLAN) and *uplink traffic flow* (from WLAN to Ethernet). The system shown in Figure 9 considers any packet entering the AP, whether arriving from the Ethernet side or the WLAN side as arriving. Similarly, any packet leaving the AP, whether to the Ethernet medium or to the WLAN medium, is considered a departing packet. Hence the logical model of the system considers arrivals as packets entering the AP and departures as packets leaving the AP, regardless of their direction of flow.

After having defined the system and the events acting on it, we assume that the system can be modeled as a queuing system with a queue and a server or multiple queues and servers.

We ran experiments to measure the relationship between arrivals and departures. The arrival and departure times recorded from experiments have shown that the system can be modeled as a single server system with one queue. Figure 10 shows a detailed view of the AP system with internal entities. The arrival timestamps denoted by T_a indicate packets arriving to the system from either side: Ethernet or WLAN. Similarly, the departure timestamps denoted by T_d are for packets departing the system regardless of being on the uplink or the downlink.

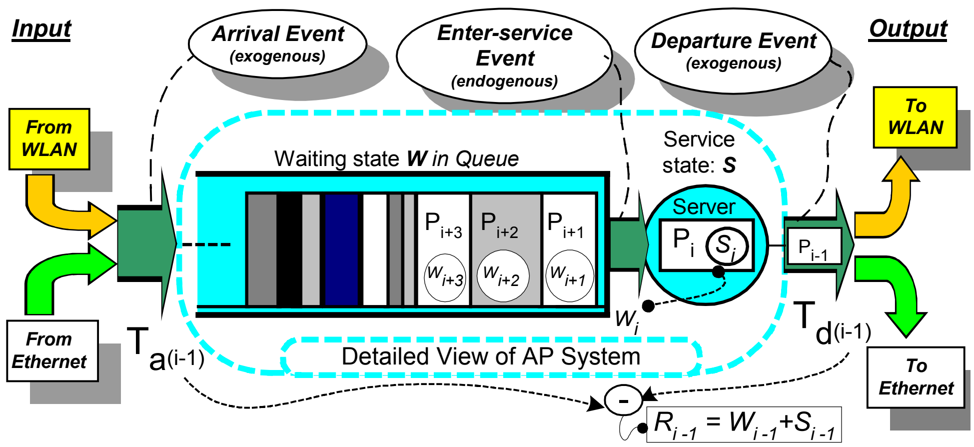

Figure 10. Detailed view of the model; T_a is the arrival time of the packet and T_d is the departure time of the packet. W_i and S_i are the waiting time and the service time of packet P_i , respectively.

Knowing that there are only one queue and one server, then, logically, we add one more event to the previously defined two events: *enter-service*. Figure 11 presents the event graph of the system with three events: *arrival*, *enter-service*, and *departure*. Within the different events there are state transitions. The system states are used to describe the state of the logical packet while being inside the system. We have three events, hence, there are two state transitions: *waiting* and *servic*e [60]. The waiting time and the service time of packet P_i are denoted by W_i *and* S_i, respectively, where i is a natural integer representing the logical *identification* of the packet with respect to its order of arrival. In other words, packet P_i arrives at the system before packet P_{i+1}. The waiting state ends with the state transition between the *arrival event* and the *enter-service event.* The service state ends with transition from the *enter-service event* and the *departure event*. Furthermore, the response of the system can be modeled as the total state transition from arrival to departure. The parameters of interest are the *waiting time* and the *service time*, respectively. We denote the *waiting time*, the *service time*, and *the response time* of a packet P_i as W_i, S_i, and R_i, respectively. Knowing the events, the system states, and the parameters of interest, we define the relationships between the event parameters and the state-parameters as follows:

1) The waiting time (W_i) of packet P_i is the time from when P_i arrives at the system until it enters the server;

2) The service time (S_i) of packet P_i is the time from when P_i enters the server until it departs from the system;

3) The response time (R_i) of packet P_i is the time from when P_i arrives at the system till the time it departs from the system.

The service time for our system includes the time required to check the headers, management time, and transmission time of the bits of the packet over the link of departure. Thus, the total time needed for the packet to leave the system since it arrived is the response time R_i, which can also be defined in terms of the waiting time and the service time as:

$$R_i = W_i + S_i. \tag{1}$$

Calculating R_i can be done simply because we can record the arrival and departure timestamps of every packet, P_i, that is entering or leaving the system. However, W_i and S_i are logical parameters, which can not be easily measured. Hence, we designed an algorithm to calculate the values of W_i and S_i for each packet, P_i, by using the recorded arrival and departure timestamps. We call the algorithm the SSTP (Simple Service Time

Producer). In Section 3.4.2, we present the third version of the algorithm (SSTP-1.3). The first and second versions were presented in [32] and [27], respectively.

In this respect, it is important to know to which classification the system under study belongs. With reference to the system classification discussed in Section 2.1.2, we can classify our AP system (Figure 9) to be a:

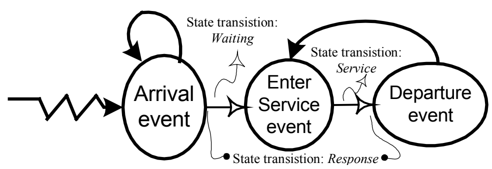

Figure 11. Event graph for the system of interest with the *enter-service* internal (endogenous) event.

1) *time-varying static system.* It is time-varying since its state and event parameters change with time, and it is static since its output at any instant does not depend on the past or future;
2) *continuous-time system* since its event parameters change with time. However, it is important to note that although our system is a continuous-time system, it is a discrete-event system, because events happen at **discrete** times. It is important to differentiate between the two: time and event classifications;
3) *stochastic system* since the system output is described by probabilities;
4) *lumped system* since there is no distribution of the model parts;
5) *linear system* since it satisfies the superposition principle;
6) *casual system* since its output does not depend on future inputs;
7) *stable system* since a bounded input gives a bounded output.

3.3. Testbed

The testbed (Figure 12) was designed to be able to test AP performance. We have two main entities besides the AP itself: PCs connected to the Ethernet side (denoted by EPCm, where m is the index number of the EPC) and PCs connected to the WLAN side (denoted by WPCq, where q is the index number of the WPC). Both, EPCs and WPCs act as traffic sources and sinks. In order to monitor the tests, we utilized a separate PC for traffic sniffing (SPC) as shown in Figure 12. We use the Linux-2.2.16 operating system [53] on all PCs. We use *tcpdump* [81] to passively record timestamps and other packet information. Moreover, when EPCs or WPCs act as traffic sources, they use MGEN-3.2 [58] for generating UDP streams. MGEN was only used for transmitting UDP packets as we used our own program modules to filter packets from *tcpdump* and analyze the results. Since accurate timestamping of arrivals and departures is essential for later analysis, we checked the clock drift in the monitoring PC (SPC) and the resolution of the *tcpdump* program on the SPC. We found that the resolution could give very accurate measurements from our tests. Moreover, we used one-second long experiments, hence the clock drift would be negligible and wouldn't correlate with subsequent measurements.

In some experiments, traffic flows from a single sender to a single sink. In other experiments, traffic flows from multiple senders to multiple sinks in both directions (downlink *and* uplink). EPCs and WPCs take turns in being senders and receivers. The dashed line in Figure 12 circumscribes the entities used in the *single-sender-to-single-sink* experiments: EPC1, WPC1, SPC, and the AP. We used an isolated environment, where there were no radio signals from other APs. To check for radio signals, we used the ORINOCO Client Manager to look at the power of signals from other access points [61].

A frame on the link is not a single entity, but rather serialized into a set of bits that are transmitted sequentially. The transmission time and retrieval time of a frame depend on the bit-signaling rate of the link. In our testbed, we use a 10Mbps Etherent [7] link and an 11Mbps IEEE 802.11b link. The retrieval of the departure timestamps by SPC can be done exactly the same way as that of WPCq since we use identical network interface cards (DELL Fast Ethernet 10/100 Base-TX, by 3Com, Model 3CCFE575CT-D [16]). However the sniffing speed for SPC for the bits of the frame transmitted on the Ethernet side may differ from that done by the AP. However, since these differences are very small on a

10Mbps Ethernet (relatively slower than the highest available speed nowadays; in Gbps), we assume that the differences are negligible, especially since we use short time spans for our experimental test runs, which we discuss below.

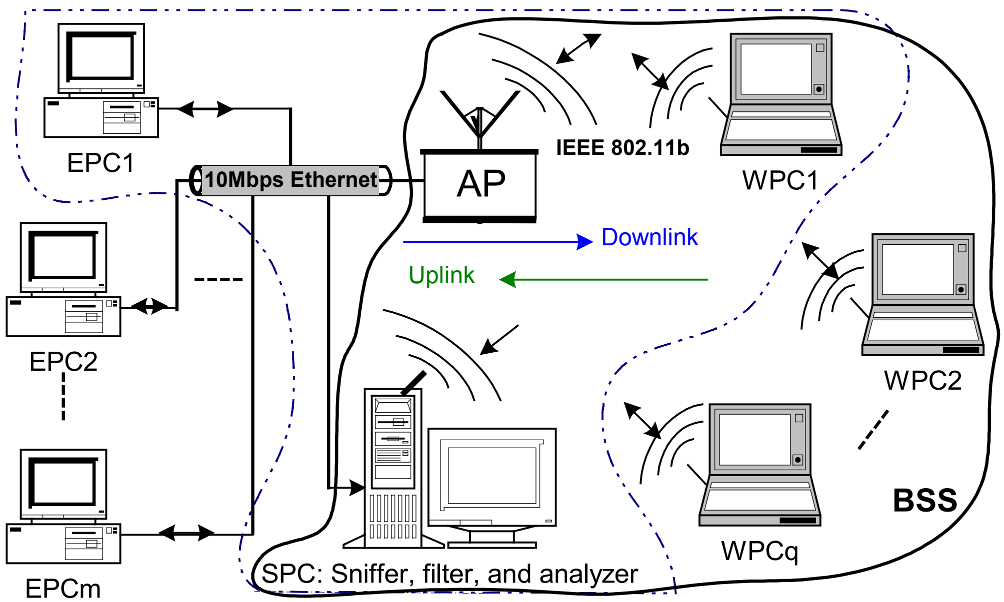

Figure 12. Testbed showing the *downlink* and *uplink* traffic flow directions. Downlink traffic consists of packets travelling from the Ethernet side (EPCs) to the IEEE802.11 side (to WPCs). Uplink traffic consists of packets travelling from the IEEE802.11 side (WPCs) to the Ethernet side (to EPCs). The wired link is a 10Mbps Ethernet. The WLAN link is IEEE 802.11b. SPC is a passive recorder used to dump traffic for consequent filtering and analysis. The dashed line encircles the set of equipment used for the AP tests to extract state/event parameters.

For accurate measurements, we ran experiments to test the *tcpdump* program itself and check whether the timestamps were suitable for measuring the arrival and departure times. Our measurements have shown that the *tcpdump* on Liunx-2.2.16 platform is a suitable tool to measure packet times since the inter-arrival time (space between two arriving packets) was large enough to be captured by the *tcpdump*. The experiments conducted with *tcpdump* recorded timestamps with resolutions lower than 32µs. The MGEN 3.2 module was used to send packets with inter-packet spacing larger than 100µs (depending on the experiment, thus could reach around 500µs). Hence, from the AP point of view, inter-arrival time periods are larger than 32μs. It is worth mentioning that the delay due to MGEN is a predictable delay and is much larger than the resolution used, so it can be detected and corrected.

Since we have one measurement device (SPC), synchronization of timestamps between sender and receiver is not necessary. So we stamp the packets by using one clock: that of SPC. However, another issue to consider in our experiments is the clock on SPC, which may drift in time. We measured the clock drift, and it was around 0.16ms drift per second. For tests of long time intervals, the PC clock drift can have a significant effect on the accuracy of the measurements. To overcome this problem, we use a one-second time-interval per test-run, and all PCs remain idle between tests. Hence the clock drifts will not affect the consequent experiments.

3.4. Measurement Methodology

Firstly, we sent packets in both directions, downlink and uplink, and we analyzed the departure events. Analysis of data from both directions showed that the system could be modeled as a single queue, single server, FIFO system. In the following section we describe the experimental test design used to extract parameters of the model.

3.4.1. Test Design

There are two main classes of tests: single-source-to-single-sink (SS) and multiple-sources-to-multiple-sinks (MM). The SS part is used to extract state parameters (waiting and service), and the MM tests were used to check the FIFO characteristic and queue management. Each part is made up of two subclasses of tests: downlink traffic tests, and uplink traffic tests. In the downlink test class, EPCs are the traffic sources, and WPCs are the traffic sinks. In the uplink test class, WPCs are the traffic generators, and EPCs act as receivers. In addition, tests were conducted when there was bi-directional traffic, i.e. EPCs and WPCs were both senders and sinks. In this thesis we present a portion of our results of this investigation focusing on delay and service time for unidirectional traffic. The two subclasses of tests are composed of identical sets. A set of tests is made up of clusters of experiments. In each cluster we fix all parameters and vary one traffic-related parameter (usually packet size) to investigate the effects. We call each experiment in a cluster an *experimental test run* (ETR). The ETR is the basic unit of tests. We describe an ETR for the clusters related to the results on unidirectional traffic. In each ETR, we send a stream of identical UDP datagrams from the sender(s) to the receiver(s). We use UDP because we designed our tests to have only forward traffic relative to the direction of flow [68], thus no traffic should come back in the opposite direction from the ETR data flow. For each ETR, we vary the size of the packets sent in a stream via increasing the payload of the UDP datagram by 32 bytes. The maximum number of bytes we used as UDP payload was 1472, because we are not interested in fragmentation, and sizes beyond the MTU may result in fragmentation [87, 88]. The headers and interframe spaces are included in the calculations of the utilized bit rate for the traffic streams to be sent, because there is a major difference between the link frames of the Ethernet and the IEEE802.11. This difference should be calculated in order to know how many data bits per second could be sent within the total number of bits transmitted in each ETR (i.e. considering overhead). The preamble on Ethernet is 8 bytes, and the Ethernet interframe space is 9.6μsec [87]. On the IEEE802.11, there are different scenarios [35]; however, our experiments use the DCF basic access method (Figure 7) described in Section 2.4.3. Our ETRs are controlled so that the WLAN overhead is described by four parameters per each MPDU per ETR: *DIFS* (50μs), *PLCP preamble and header* (272 bits), *SIFS* (10μs), and *ACK* (304 bits).

For statistical purposes, each ETR is repeated a number of times, which is up to the choice of the test designer. Three iterations of the same ETR is the minimal number accepted for statistical analysis. ETR clusters are built based on different utilized bit rates over the link. Each ETR involves passive traffic measurement using the *tcpdump* program on SPC to record traffic on both sides: Ethernet and WLAN. In addition, in each ETR a program filters the arrival time (T) and departure time (T’) for each packet (as shown in the three leftmost columns of Figure 14. For a lost packet, the departure time is considered as infinite and denoted with the value '-1'. We have two parts for each experiment: actual measurements and post-measurement. After clustering ETRs, we analyze the runs. At this point, it is crucial to note that the ETR measurements and filtering give information only about two parameters for each packet: the arrival time and the departure time. Consequently, we only have values for the exogenous parameters. However, we need information about endogenous parameters. To solve this problem, we designed the SSTP (Simple Service Time Producer) algorithm that looks at the ETR data and analyzes it to extract the values of the parameters we want. The algorithm is run on each ETR data set as discussed below.

3.4.2. Simple Service Time Producer

The Simple Service Time Producer (or SSTP) is an algorithm, i.e. it is a well-defined computational procedure that takes a set of values as input and produces a set of values as output [75].

```
1  Loss_counter = 0
2  Last_before_loss = 0
3  Total_loss = 0
4  j = 0
5  for i = 1 to n
6   do
7    if T'_i ≠ -1
8     then R_i = T'_i - T_i
9      if Loss_counter = 0
10         then if i = 1 or T_i ≥ T'_{i-1}
11                 then W_i = 0
12                      S_i = R_i
13              else if T_i < T'_{i-1}
14                 then W_i = T'_{i-1} - T_i
15                      S_i = T'_i - T'_{i-1}
16        if Loss_counter > 0
17        then  L_j = Loss_counter
18              T_j = T_{i-1}
19              j = j + 1
20              Loss_counter = 0
21               if i = 1 or {i > 1 and T_i ≥ T'_last_before_loss}
22                 then W_i = 0
23                      S_i = R_i
24              else if T_i < T'_last_before_loss
25                   then W_i = T'_last_before_loss - T_i
26                        S_i = T'_i - T'_last_before_loss
27   if T'_i = -1
28     R_i = W_i = S_i = -1
29     Total_loss = Total_loss + 1
30       if Loss_counter = 0
31         Last_before_loss = i - 1
32         Loss_counter = Loss_counter + 1
33       else if Loss_counter > 0
34         Loss_counter = Loss_counter + 1
35 Sample_space = n - Total_loss
```

Figure 13. Third version of Simple Service Time Producer (SSTP-1.3) used for calculating the response time (R_i, lines 8), the waiting time before entering service (W_i, lines 11, 14, 22, and 25), and the service time (S_i , lines 12, 15, 23, and 26) for each packet P_i in ETR data set.

The main goal of the SSTP algorithm (Figure 13) is to calculate the three state parameters discussed in Section 3.2 for each packet P_i: waiting time (W_i), Service time (S_i), and Response time (R_i). The algorithm scans the ETR data (three leftmost columns in Figure 14) and compares the time of departure of packet P_{i-1} (denoted by T'_{i-1}) with the time of arrival of packet P_i (denoted by T_i) for each packet in the ETR data set (see the cells marked by complete circles in Figure 14); where *i* is a natural integer that ranges from 1 to the *end of the ETR data se*t, n. In this thesis we present the third version of the algorithm (SSTP-1.3). The previous versions were presented in [27], [31], and [32].

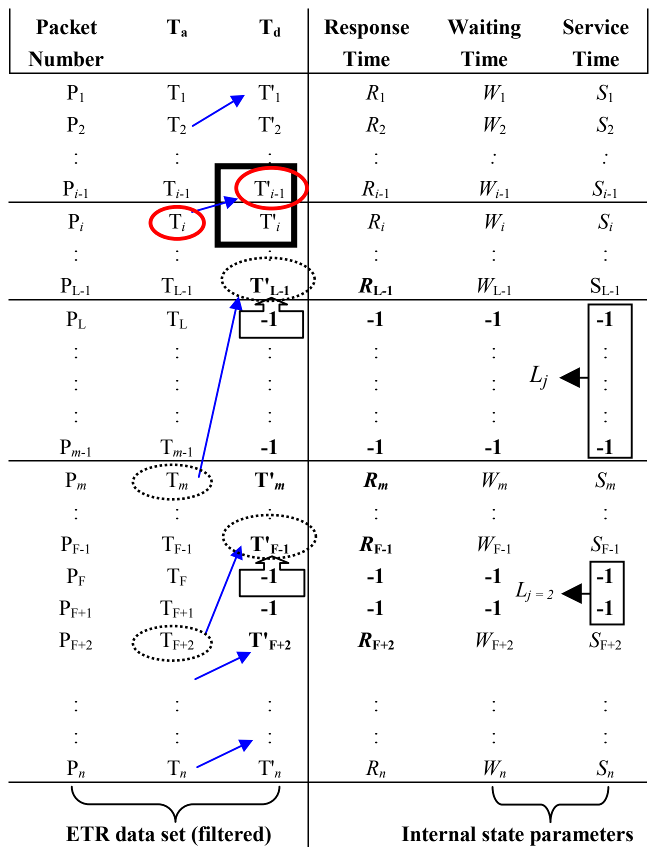

Figure 14. Analyzing ETR data using SSTP-1.3. T_a and T_d are the time of arrival and the time of departure of the packets, respectively. T_i and T'_i are the specific arrival and departure timestamps for packet P_i. The value '-1' is for lost packets. L_j is calculated by SSTP-1.3 to record the number of losses each time a loss occurs.

If the arrival time of a new packet (P_i) is later than the time when the previous packet (P_{i-1}) departed, then the waiting time for packet P_i is zero seconds, and its service time is simply the response time (R_i) since the service is assumed to process the arriving packet immediately ($W_i = 0$). However, if the time of arrival (T_i) of a new packet is earlier than the time of departure (T'_{i-1}) of the packet getting served, then the waiting time is the difference between the departure time (T'_{i-1}) of the packet in the service facility and the arrival time of the packet waiting in the queue (T_i). In this case, the service time is the difference between the time when a new packet (P_i) departs and the time when the previous packet (P_{i-1}) departed (see cells marked by the thick edges square in Figure 14).

When the SSTP-1.3 detects that a packet (P_L) has not departed, it considers the departure time to be infinite and records the value '-1' as the departure time in the output files. SSTP-1.3 counts the number of losses each time a loss occurs, and it also calculates the Sample Space of statistical sets.

The response time, waiting time, and service time for P_L will also be considered as infinite and denoted by the value '-1'. In cases of loss, the last packet that departed (P_{L-1}) is used by the algorithm for comparison with the arriving packet that departed just after loss (see cells marked by dotted circles in Figure 14). This comparison is used to calculate the waiting time and service time for the newly departing packet after loss occurs (see lines 21-26 in Figure 13). Using the algorithm, we calculate parameters from each of the ETRs. Our analysis of the different experiments gives the delay and the service time. A real SSTP output data example is provided in Appendix 3.

3.5. Results

The outcomes of the tests show that the assumption of single server, single FIFS queue holds true for downlink and uplink traffic tests. In bi-directional traffic tests, the first packet that enters the system will enter the server, but due to differences in transmission between Ethernet and WLAN, we leave the work on this matter for future work. For the WLAN APs we have tested, the delay on the uplink is always smaller than on the downlink. In this section we show results for two of the APs that we have studied. These two APs represent two classes of APs as discussed below, and the behavior of many other APs proved to be similar.

3.5.1. Response Time

The cumulative probability of the response time shows a piece-wise increasing function with one cutoff point. The cutoff point is always at the beginning of the plot (relatively small delays), showing a similar behavior for the first set of response times after which a sharp linear increase is observed. We noticed that as the packet size in a stream with specific utilized bandwidth increases, the cumulative probability of the response time also increases (compare the three different curves of Figure 15). Figure 15 shows different cumulative probability distributions for different packet size streams. As we increase the payload, the slope of the cumulative probability plot increases. In all experiments below, we utilized 2Mbps of the available bandwidth.

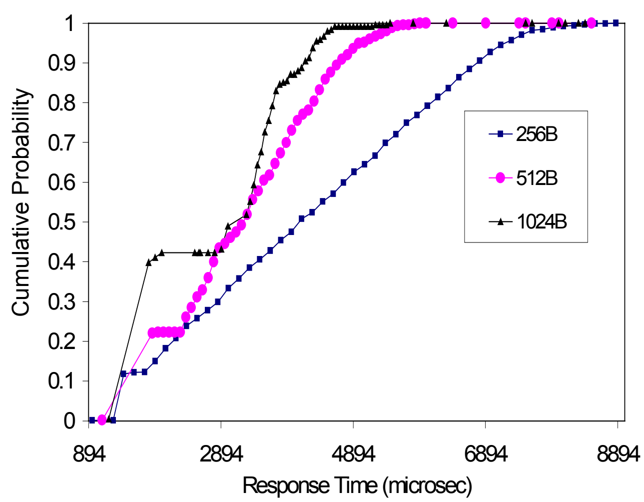

Figure 15. The cumulative probability of the response time increases in a piece-wise linear mode with one cutoff point, and it is larger for larger payloads utilizing the same bandwidth. The payload shown on the plot is UDP payload. The AP used is Lucent WavePoint II. The direction of the delay tests is Downlink (Ethernet to WLAN).

3.5.2. Directional Delay

Directional delay is delay that is related to the direction of data flow. In this section we look at the delay introduced due to service time. We define two directional service-time delays for the uplink direction and the downlink direction:

- *UST(α, x)* is the Uplink Service-Time of a packet carrying *x* bytes of IP payload, through AP "α";
- *DST(α, x)* is the Downlink Service-Time of a packet carrying *x* bytes of IP payload, through AP "α".

We found that an AP needs less service time for an uplink packet than a packet of the same size, but on the downlink. For example, for Lucent/ORINOCO WavePoint II [61] (AP1 in Table 1) the service-time for an uplink packet of 40 bytes of IP payload is an average of 152μs. However, if the same payload goes the opposite route (downlink), the average service time increases to an average of 894μs. The reason is that in our model we include the transmission time of the bits of each frame as part of the service time. The overhead for the wireless transmission of the frame is much larger than that for transmitting over Ethernet [35]. Table 2 shows the results of the average service-time values from the analysis of many ETR data sets for two Lucent/ORINOCO WLAN APs: WavePOINT-II and AP2000 [61]. Such analysis can be used to compare the performance of different APs.

Table 1. Comparison between two access points: AP1 is Lucent WavePOINT-II, and AP2 is Lucent AP2000. Uplink in both APs shows to have less service time than downlink. Comparing APs in both traffic directions proves AP1 to have lower service time than AP2.

IP Payload	AP1 Average Service Time (μs)		AP2 Average Service Time (μs)	
x (*bytes*)	*Downlink* *DST(**AP1**, x)*	*Uplink* *UST(**AP1**, x)*	*Downlink* *DST(**AP2**, x)*	*Uplink* *UST(**AP2**, x)*
40	894	152	999	171
72	918	190	1038	200
136	962	257	1140	274
264	1087	395	1374	419
520	1323	668	1820	733
1032	1750	1238	2673	1309
1480	2089	1705	3399	1876
Downlink and uplink average service times show that the server of AP1 is faster than that of AP2				

3.5.3. Service Time Formula

The key result of the work presented in this thesis is a formula for the average service time of a WLAN AP. Our experiments have shown that the average service time for a packet is a linear function of payload. The discrete-event nature of the system, which is discussed in Section 3.2, allows us to look at the *service-time* values, in relationship to payload, as *terms of a sequence*. Let us denote the general term of this sequence as S_n, where n is the experimental number of the packet (P_{n-1} arrives before P_n) and is directly related to the payload carried by the packet. Since our test design uses a 32 byte increment in the UDP payload, then each experiment will have a UDP payload that is divisible by 32. Therefore, n is a positive integer.

Table 2 shows the relationship between integer n and the IP payload. As discussed in Section 3.2, the definition of the states of the system gives a service-time that resembles the summation of three entities: the time required to check the frame headers, management time (consumed by AP to build address tables of associated hosts and run code management, which is a rare event in our 1s long experiments), and the time to transmit the bits of the packet.

Table 2. Comparison between integer n and the values of the IP payload. When the size of the UDP header (8B) is subtracted from the IP payload (x bytes), the result is divisible by n. k is a natural number, and X resembles the IP payload corresponding to packet k. Hence X is nothing but (32*k + 8) bytes. S_k is the average service-time of the packet whose sequence number is $n = k$ (i.e., carrying payload X).

Sequence number of packet in ETR (n)	IP Payload (x in bytes)	IP Payload (function of 32B)	S_n (μs)
1	40	32*1 + 8	S_1
2	72	32*2 + 8	S_2
3	104	32*3 + 8	S_3
: k :	: X :	: 32*k + 8 :	: S_k :
46	1480	32*46 + 8	S_{46}

So, the packet that enters or leaves the system is related to the frame headers it is encapsulated in. Since our test design for the ETRs uses only two PCs (see dashed line in Figure 12). Moreover, the header checks are constant for all packets since all headers have a constant size. Consequently, the transmission time of the frame plays a significant role in deciding the per-packet service time. Since transmission speeds (bps) of Ethernet and

IEEE 802.11 are constant, then the difference in transmission between one packet and another depends mainly on payload, as long as the headers are of identical sizes. In our test design, we use a 32B payload-increment, thus the difference in the average service-time values of two consecutive packets (back-to-back) is the time-difference to transmit 32 bytes with some little variations due to management time. However, management time is negligible compared to transmission time, because it is mostly code management in our 1s long experiments. Thus, denote this time difference by *r* and assume it is constant.

So, let $S_1, S_2, S_3, \ldots, S_{n-1}, S_n$ be the terms of the sequence of average-service-time values. From the previous analysis, the difference between any two consecutive terms of the sequence is the time constant, *r*. Hence,

$$S_k - S_{k-1} = r \quad \Leftrightarrow \quad S_k = S_{k-1} + r \; ; \quad k \in \mathbf{N}^* . \tag{2}$$

Equation (2) proves S_n to be the general term of an arithmetic progression with common difference *r*. What remains to be known for an arithmetic progression is its first term, which is discussed below. Let S_o (index *"o"* stands for *original*) be the first term of the sequence S_n. Consequently, we have the service time formula in (3).

$$S_n = S_o + (n-1)r \quad \mu s \tag{3}$$

where,

n = ($IP\_Payload_{[\text{in bytes}]}$ - $8\text{B}_{[UDP\ header]}$) /32B;

S_n = service time (μs) for a packet with IP payload of (32*n*+8)B = *general sequence term*;

S_o = original service-time (μs) for a packet with 40B IP payload = *sequence first-term* (S_1);

r = incremental difference in μs (calculated from linear regression of average service-time values of different payloads) = *sequence common difference*.

The proof for (3) is presented in Appendix 1. The value S_o in (3) can be calculated through numerical methods of averaging the service-time values calculated in the different ETR clusters of 40B IP payloads. From (3) we can see that the average service time grows linearly; hence, we can use linear regression of the average service-time values calculated by SSTP-1.3 to estimate the value of *r* [69]. In fact, *r* is directly proportional to the slope of the average service time curve. S_o and *r* are different for different WLAN APs. Using (3), the average service time for a downlink packet in a Lucent/ORINOCO WavePOINT-II AP [61] can be presented as:

$$S_n = 894 + 27(n-1) \ \ \mu s \, . \tag{4}$$

For a packet with IP payload 1032B, using (4) S_n will be around 1731µsec with an error of 1.08%. Note that this average service time is no longer a stochastic process, but is deterministic if the packet sizes in a flow are known. This is interesting since many applications produce well-known packet sizes. Studies could be made to improve application performance by knowing the best packet size to use over a path where an AP exists. The error for the values calculated by our equations and the experimental values range from zero to 3% of the maximum. Hence (3) has a maximum error of 3%. Thus, the manufacturer or the user of the AP has a good estimate of the service time per packet, a result that is valuable for modeling, simulation, and traffic shaping. It is important to mention that if the packet sizes in a flow are known, then the average service time is no longer a stochastic process, but is deterministic. This is very interesting since many applications produce known packet sizes. Therefore, application developers can run simulations of the model to study the delay that is imposed by the wireless AP for different flow patterns of the application. In addition, studies could be made to improve application performance by knowing the best packet size to use over a path, where a WLAN AP exists. Another note worth mentioning is that when using the results of the average service-time values like the ones presented in Table 1, the overhead time that is consumed to send pure data (that is of interest to the application) can be estimated. For example, to estimate the overhead on the uplink for an AP, α, we can subtract UST(α, 40B) from UST(α, 72B) to estimate the time spent serving 32 bytes of pure data. This is because the difference between two consecutive test packets (of 32B increment) is the difference to transmit 32B of pure data. To have a statistically acceptable result, we should average the ETR samples. However, this is already done in the calculations of the time constant, *r*. Since the first test-packet carries 40 bytes of IP payload (i.e., 32B of UDP payload-application data), then the time spent on overhead per packet can be estimated from the average service-time of the first test packet and *r* as: *UST*(α, 40B) - *r* in µs.

3.6. Uplink-Downlink Contrast

The differences between the uplink and downlink service times for the same AP are investigated to understand its performance as it would generally be used for bi-directional traffic. For our analysis, I introduce the notion of the *Uplink-Downlink Contrast* (UDC) of a WLAN AP. I define the UDC to be *the absolute value of the difference between the uplink and downlink service-time values in relation to packet size*. The UDC of a packet with a given payload of x bytes passing through an AP, α, is mathematically defined as:

$$UDC(\alpha, x) = | UST(\alpha, x) - DST(\alpha, x) | \tag{5}$$

where $UST(\alpha, x)$ and $DST(\alpha, x)$ are the directional service-time delays defined in Section 3.5.2.

Using (5) we can calculate the absolute value of the difference between the uplink and the downlink service-time of different APs for comparison.

Table 3 shows the Uplink-Downlink Contrast values for the APs given in Table 1.

Looking at the UDC values in

Table 3, we observe that as the payload increases, the UDC values of AP1 decrease, while the UDC values of AP2 increase. The reasons for these differences in behavior between APs are the subject of future investigation. In this particular case, the explanation is that AP2 spends more time checking the packet than AP1, this leads to a greater delay as the packet size increases. In fact, the increase in UDC for AP2 is mainly due to the relatively large increase in the downlink service-time as the payload increases (see Table 1). On the other hand, the increase in the uplink service time values for AP2 is not as relatively large as the increase on the downlink. One reason is that the Ethernet overhead time is more stable than the IEEE 802.11b overhead time for many APs due to the different checks (management including different standard implementation and code) that some access points make before transmission. In fact, there are different implementations of the IEEE 802.11b scenarios for checking headers and for radio transmissions for different WLAN access points. Hence, we look at these differences between APs from the point of view of their effects on performance, and we characterize a UDC as either convergent or divergent.

Table 3. Uplink-Downlink Contrast (UDC) of two Wireless LAN access point brands: AP1 is Lucent WavePoint-II, and AP2 is Lucent AP2000. UDC of AP1 decreases with increasing payload. UDC of AP2 increases with increasing payload.

IP Payload *X* (Bytes)	*UDC(**AP1**, x)* (μsec)	*UDC(**AP2**, **x**)* (μsec)
40	742	828
72	728	838
136	705	866
264	692	955
520	655	1087
1032	512	1364
1480	384	1523

3.6.1. Convergent UDC

Definition. *A WLAN access point, α, is said to have a convergent UDC if and only if:*

*UDC($\boldsymbol{\alpha}$, **x**) decreases as **x** increases, where x is the payload in bytes.* *(6)*

For a convergent UDC the downlink and uplink service-time values grow closer to each other as the payload increases. This is illustrated in Figure 16, where the uplink and downlink service-time curves for AP1 (Lucent WavePoint-II) are plotted.

Extrapolating the plots in Figure 16 (dashed lines) show that as the payload increases, two curves convergence to a point, where the downlink and uplink service-time values become identical. Beyond this point, the uplink service-time is larger than the downlink service-time. However, the point of convergence (around 2200B) is beyond the realistic limits of the standard maximum transmission unit (MTU) for Ethernet and IEEE 802.11. So, in reality this point of performance, where the uplink and downlink service-times are equal will not be reached. The linear behavior of the service time is not surprising since it has been shown in equation (3) that the average service time is a linear function of the payload. An interesting characteristic to look at is the rate of convergence of uplink and downlink service-times, which can be illustrated in the UDC-vs-payload plot, as shown in Figure 17. For a decreasing UDC curve, the UDC is convergent since this satisfies (6). In Figure 17, one can easily notice a decreasing UDC for AP1, which means that AP1 has a convergent UDC. The Convergent-UDC curve can be extended, and the point of intersection with the horizontal axis (payload) will be the point where the uplink and downlink service-times are identical for the given AP. In our analysis, we use the UDC plot as a characteristic curve of the wireless LAN access point. The dotted line is a theoretical extension, using the service time formula in (4) for AP1, to illustrate the intersection with the horizontal axis.

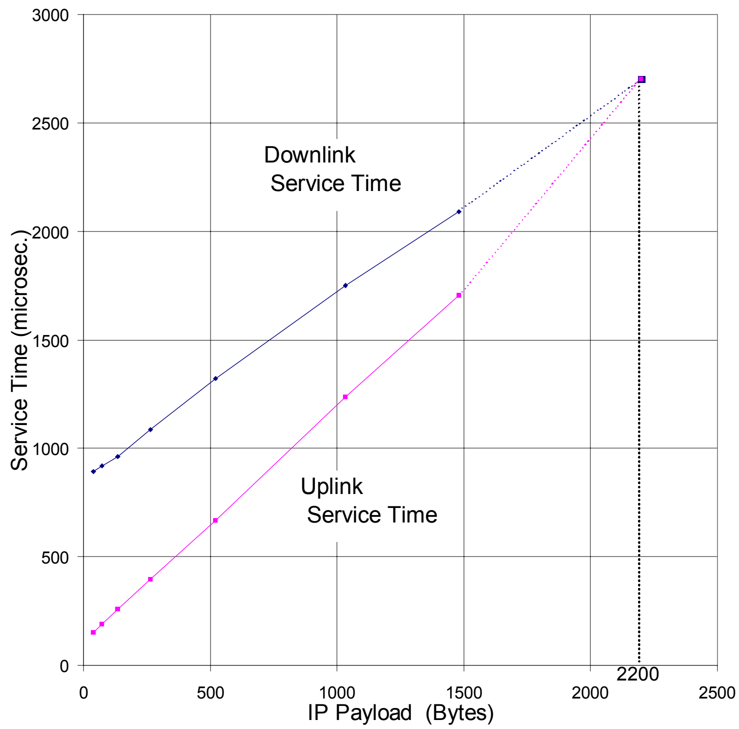

Figure 16. DST and UST of AP1 (Lucent WavePoint II). Stretched curves converge to a point, where the DST and UST are identical (corresponding to an IP payload value of around 2200 bytes). The solid lines are the measured values, while the dotted lines are theoretical extensions. The point of convergence lies beyond the realistic limit for the MTU used in Ethernet and IEEE 802.11b (1500 bytes).

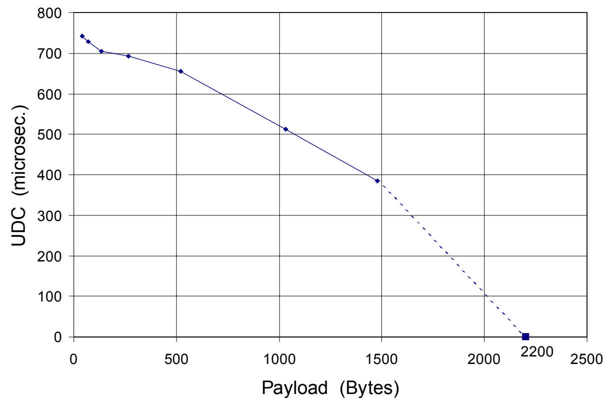

Figure 17. UDC plot for AP1 (Lucent WavePoint-II). The UDC is decreasing with increasing IP payload. AP1 has a convergent UDC. The solid line is the real curve. The dotted line is a theoretical extension.

3.6.2. Divergent UDC

The notion of a divergent UDC is the opposite of a convergent UDC.

Definition. *A WLAN access point, α, is said to have a divergent UDC if and only if:*

*UDC($\boldsymbol{\alpha}$, **x**) increases as **x** increases, where x is the payload in bytes.* *(7)*

When an AP has a divergent UDC, the uplink and downlink service-time values diverge from each other as the payload increases. This is illustrated in Figure 18, where the uplink and downlink service-time values for AP2 (Lucent AP2000) are plotted. The UDC graph for AP2 (Figure 19) shows that *UDC(AP2, x)* increases as payload *x* increases. Using (7), we conclude that AP2 has a divergent UDC, as is clear from Figure 19.

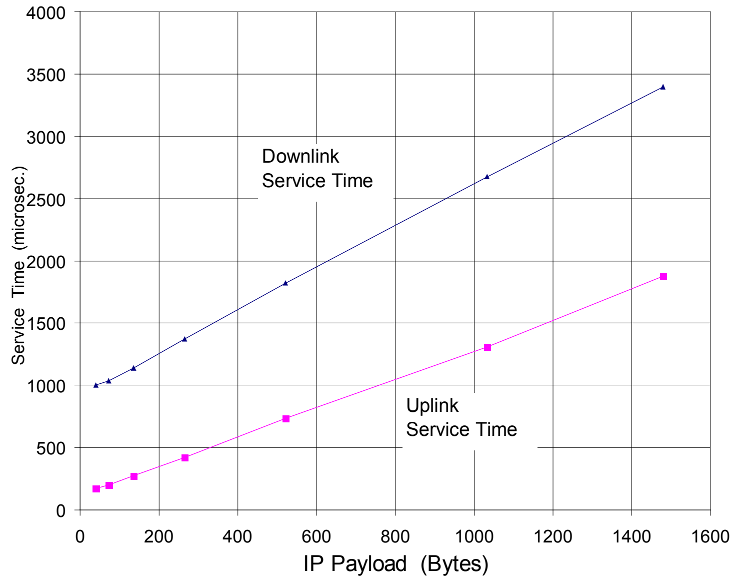

Figure 18. Downlink and Uplink Service-Times of AP2 (Lucent AP2000). The two plots diverge from each other as the payload increases.

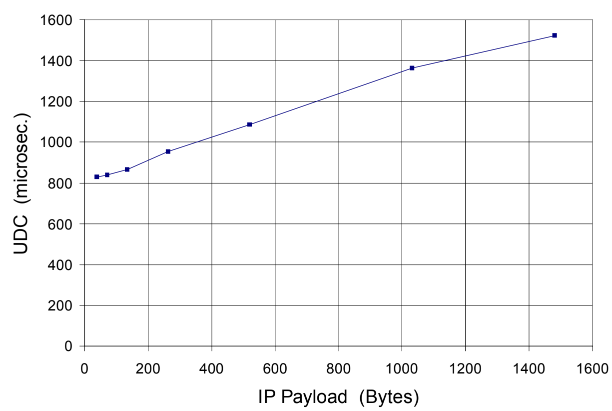

Figure 19. UDC plot for AP2 (Lucent AP2000). The UDC is increasing with increasing payload. AP2 has a divergent UDC.

3.6.3. The UDC as a QoS Parameter

The results on convergent and divergent UDC form a strong basis for studying the relationship between the brand of the WLAN AP and the QoS of specific applications over WLANs. For instance, an AP characterized by a convergent UDC is better suited for LANs where mobile users are interested in interactive real-time multimedia applications that produce *large* amounts of payload in both directions. The convergent UDC minimizes the differences in delay in both directions when the application produces large payloads in both directions (such as a two-way videoconference). So, if we know that the user is attached to a specific wireless LAN and we are using real-time applications with large packet sizes, the characteristics of Convergent UDC can be used for selecting an AP brand. Similarly, if the real-time application on a WLAN sends small packets, then a divergent UDC is preferable, because it will minimize the service time needed for the real-time communication packets at the cost of having larger delays for other applications that use larger packets. Video conferencing applications are a good example for such use of the UDC characteristic, because they transmit large or small packets depending on the video-codec[16] used. Moreover, most commercial IP video conferencing centers use specific multimedia applications. Hence, when supporting conferencing with wireless connections, or when one or more peers join a conference session through a wireless connection, the access point used will have a significant effect on the performance. In fact, the market is witnessing an increasing demand for IP video conferencing. Thus, the manufacturers of WLAN access points can also benefit from the UDC characteristic of APs to try to enhance the performance of APs for specific applications and deliver better services by introducing different products, which suit the various conferencing applications in the market. Currently, the UDC characteristic reveals two types (based on the three different AP brands we have tested): the convergent and the divergent UDC. Hence, just labeling the AP brands with convergent/divergent UDC characteristics will help the users to select suitable WLAN access points knowing the packet sizes of the applications used over the LAN.

---

[16] A codec is an electronic device that converts analog signals, such as video and voice signals, into digital form and compresses them to conserve bandwidth on a transmission path [20]. Codec is used for video conferencing systems. Video streams are referred to by the codec name they are recorded with.

3.7. Summary

This chapter presented a mathematical model for WLAN APs. Our experiments showed that our assumption of a single server, single FIFS queue is correct. The major observation is that the delay to serve a packet travelling from the wireless medium to the wired medium (on the uplink) is less than the delay to serve a packet, of the same payload, but travelling on the downlink. Using our model and analysis, we can compare performance of different brands of WLAN APs. A key result is an analytic model for the average service time of a packet in relationship to payload. The service time was analyzed and found to be a strictly increasing linear function of payload. In addition, we analyzed the absolute value of the difference between the uplink and downlink service-times for a given AP. We define the absolute value of the difference in time between the uplink and downlink to be the *Uplink-Downlink Contrast* (UDC). The results of this investigation show that as the payload increases, the UDC either decreases or increases depending on the brand of the access point. We introduced the notions of convergent and divergent UDC. A convergent UDC decreases with payload, while a divergent UDC increases with payload. Knowing the size of the packets sent by the application, the choice of a suitable wireless LAN access point can be based on the convergent/divergent UDC characteristic of the access point.

Chapter 4

Access Point Buffer Management

"When you reach the end of what you should know, you will be at the beginning of what you should sense."

Goubran Khalil Goubran

In this chapter we utilize the queuing model presented in section 3.2 to estimate parameters describing the buffering occurring in WLAN APs of IEEE 802.11b. We use experimentation to obtain these buffer-related parameters. This chapter focuses on estimation of initial buffer size. We designed and implemented an algorithm, which checks when losses occur and calculates the number of packets in the buffer just before loss. Section 4.1 introduces the dependency on previous results. Section 4.2 describes the experimental environment. Section 4.3 discusses the test methodology and the algorithm designed. Some results of applying the test methodology on APs are presented and discussed in section 4.4. Section 4.5 summarizes the chapter.

4.1. Introduction

This chapter builds on our previous results, which are presented in Chapter 3. We utilize a the single server, single queue, FIFS system model to estimate the parameters describing the buffering occurring in an AP. Based on a set of traffic measurements we are able to extract the parameters of the model. Buffer size is a very important parameter that has direct effects on the performance of WLAN APs, however, we are not aware of a careful analysis of buffer size in current APs. In order to estimate the buffer size, we construct a set of experiments that purposely try to cause packet loss due to the lack of buffer capacity. Based on these measurements we can estimate the size of the initially allocated buffer in bytes, because we are interested in the occurrence of the first loss.

Below, we present results on buffer size and adaptation of WLAN APs. The major observation is that the buffer adapts its size to the different loads. Adaptation occurs at cut-off points of losses by increasing the allocated buffer size. We designed and implemented the Buffer Size Estimator (BSE) algorithm that detects when a packet is lost and makes use of the SSTP algorithm for extracting some parameters for buffer size estimation. A key result is that different access points have different initially allocated buffer sizes, which leads to different initial losses and different adaptation thereupon. Therefore, the QoS, especially from the packet loss point of view, of an AP will not only be affected by the offered load, but also by the ability of the AP to adapt to the load, hence the dependence on buffer size. The buffer study on a particular access point can be used as a test to determine whether an AP is more suitable for certain applications or specific network loads than another AP.

4.2. Experimental Environment

The experimental environment is the same as the one shown in Figure 12. Hence, the testbed consists of traffic sources and sinks, an AP, and the traffic analyzer. The programs used are the same as the ones discussed in Section 3.3. However, for buffer size estimation, we designed and developed a new algorithm: the Buffer Size Estimator (BSE) [27]. The BSE algorithm depends on results calculated by of the SSTP-1.3 algorithm [30]. The BSE module is built using the C++ language. Since we have two traffic streams of uplink and downlink, then we have to run the buffer-related experiments in both directions. The same testbed is used, but the difference lies in the test design discussed below.

4.3. Experiments and Algorithms Used

In the following section we describe the experimental test design used to estimate the parameters related to buffering. These experiments and analyses focus on buffer allocation in WLAN APs.

4.3.1. Test Design

For buffer size estimation, we use the SS and the MM test classes as discussed in Section 3.4.1. The SS test class is used to calculate the initial buffer size. The MM test class is to check whether the same buffer allocation is used when multiple sources and multiple sinks are active. Additionally, there are two main types of tests: downlink traffic tests, and uplink traffic tests. Each of the aforementioned classes is composed of identical clusters of test runs, thus, describing one is sufficient. The only difference between the tests designed for buffer size estimation and the tests described in Section 3.4 is that we try to cause packet loss due to the lack of buffer capacity by purposely utilizing a high percentage of the available bandwidth with different packet sizes. Each experiment is repeated at least three times for statistical purposes. The results of filtering the recorded traffic (using *tcpdump*) of the tests are stored as BSE ETRs similar to the ETRs discussed in Section 3.4.1. Hence, the BSE ETR is the basic test unit for buffer size estimation.

After overloading the buffer and filtering the data in ETR data sets (first three columns in Figure 20), The SSTP algorithm is executed over the BSE-ETR data in order to give the SSTP output data file. The result is a file similar to the data shown in Figure 20. Then, the BSE algorithm is run over the SSTP output data file as discussed below.

4.3.2. BSE Algorithm

The BSE algorithm (Figure 21) is designed with the major aim of estimating the size of the allocated buffer. To do so, the BSE looks for packet loss in the SSTP output data file in order to relate *occurrences of loss* to *the number of test packets in the buffer*.

Two main parameters are needed from each BSE-ETR in order to estimate the buffer size: the *payload (X) in the test packets of the BSE-ETR* and *the average service-time* (S_{avg}) calculated by the SSTP from the BSE-ETR data set (line 1 of BSE in Figure 21). The payload is used to calculate the *size* (PS_X in bytes) of the test packet used (line 2 of BSE in Figure 21).

The BSE reads through the data of the SSTP output file looking for lost packets (marked by the value '-1' for departure time). When a packet is lost (see example of cell market by triangle in Figure 20), the BSE checks whether the loss has just happened (by checking the *BSE_loss_counter*) and whether there was enough information (*INFO*) to estimate the number of packets in queue. If there was enough information (i.e., at least one packet has departed before loss), then the BSE uses the *waiting time* of the last available packet in queue to estimate the number of packets in the buffer just before loss (see cell marked by a dotted circle in Figure 20). As the packets in our ETRs are identical, the waiting time (W_{L-1}) of packet P_{L-1} that last departed before loss is directly proportional to the number of packets in queue (N_j) at the j^{th} occurrence of loss as follows:

$$W_{L-1} = N_j * S_{avg}. \tag{8}$$

Hence j is a positive integer smaller than L.

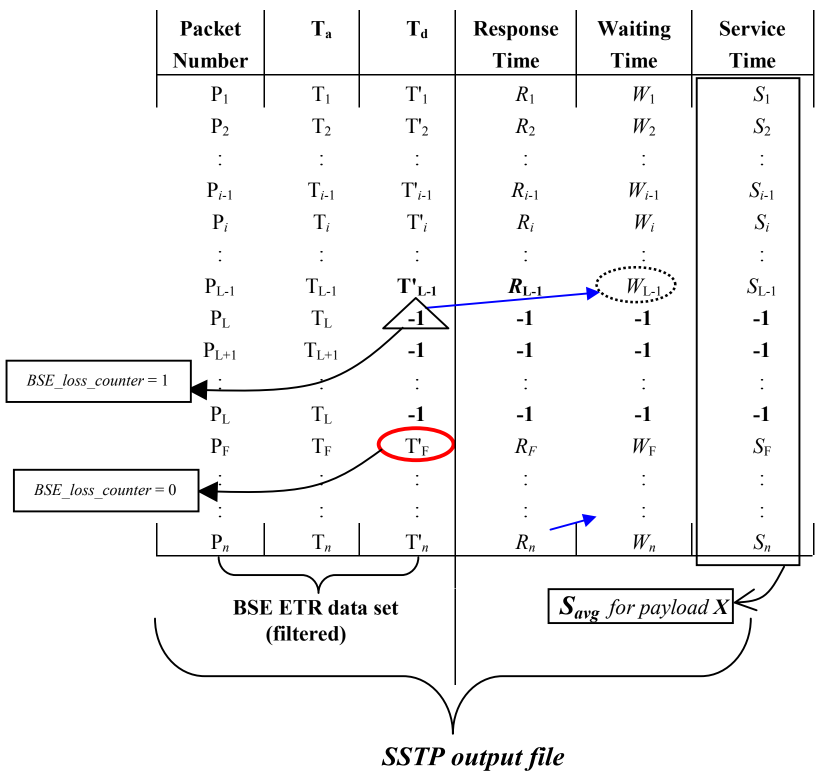

Figure 20. Output data file of the SSTP-1.3 to be used by the BSE algorithm. T_a and T_d are the time of arrival and the time of departure of the packet, respectively. The value '-1' is for lost packets. S_{avg} is the average service time calculated from the service time values (S_i). X is the value of the payload used in the BSE-ETR packets. L stands for 'Lost' and (L-1) is the index of the last packet in queue before loss.

In other words, the *number of packets* in queue just before loss is simply the *waiting time* of the packet that departed just before loss *divided by the average service-time* of the ETR packet (Figure 21, line 14). From (8) we conclude that the number of packets in the buffer will reach its first maximum (relative to the initially allocated buffer size) if packets after the j^{th} packet were lost. As the size of the test packets is known, we can now calculate the buffer size as the maximum number of packets in buffer multiplied by the packet size used in the ETR (Figure 21, line 15). For every data set, the BSE algorithm calculated the buffer size each time a loss is detected. These results were analyzed statistically [69] (to avoid measurement errors), and an estimate of the initial buffer size is computed.

```
1   Read S_avg and X from SSTP output data file
2   PS_X = Packet_Size(X)
3   BSE_loss_counter = 0
4   Buffer_test_success = 0
5   INFO = 0
6   j = 0
7   for i = 1 to n
8    do
9     if T'_i = -1
10      then if BSE_loss_counter = 0
11              then if INFO = 1
12                     then j = j + 1
13                          BSE_loss_counter = BSE_loss_counter + 1
14                          N_j = W_{i-1}/S_avg
15                          B_j = N_j*PS_X
16                          if j > 1
17                            then diff_{j-1} = B_j - B_{j-1}
18            else if BSE_loss_counter > 0
19                   then BSE_loss_counter = BSE_loss_counter + 1
20     else if T'_i ≠ -1
21            then if BSE_Loss_counter > 0
22                   then BSE_loss_counter = 0
23                 else if BSE_Loss_counter = 0
24                        then if i = n
25                               then Buffer_test_success = j
26                             else if i < n
27                                    then INFO = 1
```

Figure 21. BSE algorithm used for estimating the buffer size of the WLAN AP. S_{avg} is the average service-time calculated by SSTP for packet carrying a payload of *X bytes*. PS_X is the total packet size calculated by adding the headers of the packet to the value X (in bytes). N_j is the number of packets in buffer just before loss (i.e. the maximum number that the buffer allocated size could take at the j^{th} occurrence of loss). B_j is the estimated buffer size at the j^{th} occurrence of loss.

Moreover, the BSE algorithm can detect the difference between two consequently allocated buffer sizes in order to use this information for studying buffer adaptation to increasing loads (Figure 21, line 17 calculates $diff_{j\text{-}1} = B_j - B_{j\text{-}1}$). Hence users and AP manufacturers can determine the size of the buffer in an AP along their network path relative to the offered load.

4.4. Sample Results

The first set of results is the output data file of SSTP-1.3, this is needed for the BSE algorithm. In Table 4, we show the average service-time values (S_{avg}) on the downlink and the uplink for two different APs: APa is Lucent/ORINOCO WavePoint-II and APb is ORINOCO AP500 [61]. Though the results prove that APb has a faster server on the downlink than APa, this information is not enough to chose which of the two access points is more suitable for a certain application (whether delay sensitive or packet loss sensitive). That is because different buffer initial allocations also affect the decision on which AP is more suitable.

Table 4. Comparison between two access points: APa is Lucent/ORINOCO WavePOINT-II, and APb is ORINOCO AP500. Uplink in both APs shows to have less service time than downlink. Comparing APs in downlink traffic direction proves APb to have lower service time than APa, but on the uplink direction APa proves to have a faster server than APb.

Payload X (Bytes)	$S_{avg}(X)$ for APa (µsec)			$S_{avg}(X)$ for APb (µsec)	
	*DST(**APa**, X)*	*UST(**APa**, X)*		*DST(**APb**, X)*	*UST(**APb**, X)*
40	894	152		762	343
72	918	190		785	383
136	962	257		836	463
264	1087	395		953	618
520	1323	668		1110	915
1032	1750	1238		1599	1506
1480	2089	1705		1999	1996

In all experiments (on all APs studied), the BSE has shown that the buffer size is adaptive to offered load. As the load increases, the allocated buffer size increases. However, the first loss for a specific AP is always the same if the offered load was the same. Moreover, there was no packet loss detected on the uplink. All APs proved faster at serving the uplink traffic. This is because the uplink service-time is relatively small so packets will go out of the system before the buffer fills up on the uplink. However, on the downlink, the buffer filled up since packets had to wait longer in the queue while other packets were getting served. Table 5 shows the results on allocated buffer size (KB) when packet loss was encountered. The values presented are the results of many trials, and the calculated error is less than 2%. When there is no loss, we can not get any value for the buffer size, which means that the available buffer is suitable, as was the case on the uplink of all the APs that were investigated.

Table 5. Buffer size Comparison between the two access points in table 4: APa is Lucent/ORINOCO WavePOINT-II, and APb is ORINOCO AP500. Uplink in both APs shows to have no packet loss. Comparing APs shows APa to have a higher first buffer allocation than APb.

IP Payload (Bytes)	APa First Allocated Buffer Size (KB)		APb First Allocated Buffer Size (KB)	
	Downlink	Uplink	Downlink	Uplink
40	20	*No loss*	11	*No loss*
72	34	*No loss*	16	*No loss*
136	72	*No loss*	28	*No loss*
264	*No loss*	*No loss*	32	*No loss*
520	*No loss*	*No loss*	48	*No loss*
1032	*No loss*	*No loss*	77	*No loss*
1480	*No loss*	*No loss*	114	*No loss*

We observe from Table 5 that APa has a higher initial buffer sized allocation size than APb for the different loads utilizing the full bandwidth (which in our case was 10Mbps). Although APb has a lower downlink service-time than APa, if the application QoS is sensitive to packet loss, then APa proves to be more suitable than APb. However, if an application is sensitive to delay, then APb would be more suitable. All in all, the average service time and the allocated buffer size can be used in order to choose a suitable AP.

4.5. Summary

This chapter introduced the BSE algorithm for estimating the allocated buffer size in WLAN APs. We discussed our test methodology for estimating the parameters needed by the BSE, which depend on the output of the SSTP algorithm (discussed in Section 3.4). Our analysis of buffer size revealed that although an AP may have a faster server, the buffer allocation scheme used may affect the QoS from the packet-loss point of view. One observation is that the APs adapt their buffer to the different loads. Adaptation occurs at points of losses by increasing the allocated buffer size. These results will be shown in future work. A key result is that different APs have different initially allocated buffer sizes. The major result is that when using our model, test methodology, and algorithms, one can get a good estimate of the allocated buffer; hence, our model can be used to compare various APs and choose the most appropriate one.

Chapter 5

Access Point Link Model

"You see things, and you say: Why? But I dream things that never were, and I say: Why not?"

George Bernard Shaw

This chapter builds on our previous results in order to model WLAN APs of IEEE 802.11b as data communication links. We parameterize the bandwidth of the AP-link-model. The throughput of a data communication link is directly proportional to its bandwidth. Hence, knowing the bandwidth of the AP link model, we estimate the throughput of the AP. Section 5.1 sheds light on key effects due to the presence of an AP along an Internet path and its equivalence to as a link with variable bandwidth. Section 5.2 discusses a link model of a WLAN AP. In Section 5.3, we discuss the throughput of WLAN APs. A feedback control model for the throughput of WLAN APs is presented in section 5.4. Section 5.5 summarizes the chapter with concluding remarks.

5.1. Introduction

End-to-end performance on an Internet path depends on the performance of each node on the path. The largest portion of WLAN users are connected to the Internet backbone via a WLAN AP [65]. While many of these APs are connected to ADSL, cable modems, etc, an increasing number are connected via high speed network connections, which are capable of handling the full bandwidth of an IEEE 802.11b AP; thus the access bottleneck may move to the WLAN AP itself.

Inspired by the power of *packet-pair analysis in FIFO-queuing networks* [47, 72], this chapter builds on the results of Chapter 3 in order to model WLAN APs as data communication links with available bandwidth. As the rest of this chapter shows, when the AP is considered as a link, the bandwidth of the AP-link-model varies with varying loads. In fact, in the following sections, I show that the AP-link-model adapts the link speed of transmission to varying payloads. Hence, along an Internet path, the AP acts as if it were a data communications link with varying speeds (adaptive bandwidth). This does not mean that the AP will itself have different transmission speeds, but rather when packets pass through an AP, the whole process could be thought of and compared to the passage through a data link. Analyses in Section 5.2 show that the adaptive bandwidth is a strictly increasing linear fractional transformation of the packet size. Thus, the presence of an access point along a path will result in a link bandwidth that exhibits varying link speeds over a short interval of time (depending on the test packets themselves). Therefore, many end-to-end path bottleneck experiments will fail to provide applications with correct information about the condition of the path. This behavior of the AP leads to false service level expectations by applications, with especially negative effects on multimedia applications. The AP link-model helps us further understand the performance of WLAN APs, especially when looked at as separate nodes on a path. In this respect, APs will look and behave like links with varying speeds dependent on packet size. This performance does not mean that the transmission of the wireless link will change, but rather shows that while the packets are inside the AP, they can be thought of and considered as passing through a communication link, whose speed of transmission is dependent on packet size. This adaptive-bandwidth of the AP-link-model is used to calculate the throughput of the AP by utilizing the definition of link bandwidth of data communication links. An interesting model of the throughput of a WLAN AP is presented as a feedback control system in Section 5.4. Below we discuss the different logical and analytic steps required to parameterize the bandwidth of the AP-link-model and present the throughput of APs using a feedback control model.

5.2. Link Model

The packet-pair property [47] of FIFO-queuing networks (presented in Appendix 2) can predict an estimate of the difference in arrival times of two packets of the same size, sent from the same sender and received at the same destination. This property makes many assumptions that do not always hold true in most packet-based-networks [47]. However, in the test design and experiments on APs that were described in Section 3.4, the assumptions of the packet-pair property hold true. For this reason, we find in this property a very important basis for deeper analysis of the AP.

The applicability of the packet-pair property of our queuing model (presented in Section 3.2) inspired me to look at the WLAN AP as a *data communications link,* whose bandwidth is *b* bits per second (Figure 22). To complete the model, some assumptions are made. We consider the AP link model to be connected to two links: a 10Mbps Ethernet link on one side and an 11Mbps IEEE 802.11b link on the other side (Figure 22). Moreover, data can flow from the Ethernet side to the WLAN side (downlink) or from the WLAN side to the Ethernet side (uplink). A link is characterized by its speed of transmission, which is related to the link protocol, the number of nodes it can associate to, and the collision correction scheme used as well as the coding currently used.

Hence, we assume that the *speed of the link model* (*b* bits per second) could be smaller than, larger than, or equal to the link speeds of Ethernet or IEEE 802.11b. We also consider the link to be a *point-to-point link* with *no collisions*, because inside the AP the packets are queued without having to worry about a shared transmission medium. Consequently we consider that the link model of the AP has no preambles, no Interframe Spaces, and no link headers or trailers (we leave the link header and trailer to be taken care of by the Ethernet and WLAN media). Therefore, link frames of the AP model are the IP packets themselves, and these packets can queue back to back in the link without interframe spacing. Moreover, we assume that the propagation delay in the three links of Ethernet, IEEE 802.11b, and the *b* link of the AP to be negligible compared to transmission delay since the distances covered are too small to be considered in the calculations. Having two directions of flow (uplink and downlink), the AP link model will act with different transmission speeds for uplink and downlink. In this way, it is similar to many links where the speed of upload is different from the speed of download (e.g. ADSL links). Knowing the characteristics of the link model, we investigate the bandwidth of this link to parameterize its value.

5.2.1. Packet-Pair Property in FIFO-Queuing Networks and the AP Model

Utilizing packet pairs in FIFO-queuing networks makes use of a two-packet logical model [47], whose parameters are the difference in arrival times of two identical packets sent from the same source to the same destination. The arrival times in the packet-pair model are arrivals at the destination (see Figure A.2 in Appendix 2), hence, they resemble the departure time values of the two packets along the AP in our queuing model. Below we discuss the *assumptions of the packet-pair property* [47] to show how these assumptions are suitable for our model and analyses.

One of the assumptions of the packet-pair property is that the two test packets queue together at the bottleneck link (which is the AP link in this chapter) and not later. This assumption is true in the case of our experimental tests (discussed in Section 3.4) since we designed controlled tests, which use a set of back-to-back packets.

The second assumption of the packet-pair property states that the bottleneck node uses FIFO-queuing. In our previous work, we showed that the AP is modeled as a FIFO queue with one server for both the downlink and the uplink directions. Hence, the second assumption holds true for our model in both directions: downlink and uplink.

The third assumption considers that the transmission delay of the link under study is proportional to the packet size, and that the nodes are store-and-forward nodes. In this respect, our previous results (Section 3.5.3) presented the service time of the model as the time for checking headers, performing simple management, and transmitting the bits of the frame. The header-check time is constant. The management time is constant and is negligible compared to transmission time on both directions: downlink or uplink. Moreover, the transmission time depends on the number of bits in the packet. Therefore the service time is directly proportional to the transmission time, which- in turn- is directly proportional to the packet size (or payload), as (3) shows.

Finally, the packet-pair property does not consider per-link latencies, and in our link model we assume that link latencies are negligible compared to the transmission time on each of the three links: Ethernet, *b* link of the AP-link-model, and the IEEE 802.11b link.

5.2.2. AP as a Data Communications Link with Variable Bandwidth

Knowing that the assumptions of the packet-pair property discussed in Section 5.2.1 are true for our model, we consider the AP model as a link and investigate the bandwidth, *b*, of this link. Figure 22 shows the link model of the AP for a downlink flow example. The analysis is similar for both cases: uplink and downlink, with a difference reflected in the parameters of (3) and in the link header sizes.

The packet pair property states that [47]:

$$t(d,1) - t(d,0) = max\ \{[s_1/b], [t(0,1) - t(0,0)]\} \tag{9}$$

where,

- $t(d,0)$ and $t(d,1)$ are the arrival times (at the destination) of the first and second packets respectively;
- $t(0,0)$ and $t(0,1)$ are the times of transmission of the first and second packets respectively;
- s_1 is the size of the second packet in bits;
- b is the bandwidth of the bottleneck in bits per second.

The *d*-labeled timestamps resemble departures of the first and second packets from the AP-queuing-system respectively. Solving (9) for b is derived in [47] and presented in Appendix 2. The result of the solution is that the bandwidth of the bottleneck link is:

$$b = \frac{s_1}{t(d,1) - t(d,0)} \quad (bits \quad per \quad \sec.). \tag{10}$$

The denominator in (10) is the difference between the arrival time values of the packet-pair at the destination. These *packet-pair arrivals* are the departure time values calculated from the queuing model of the AP.

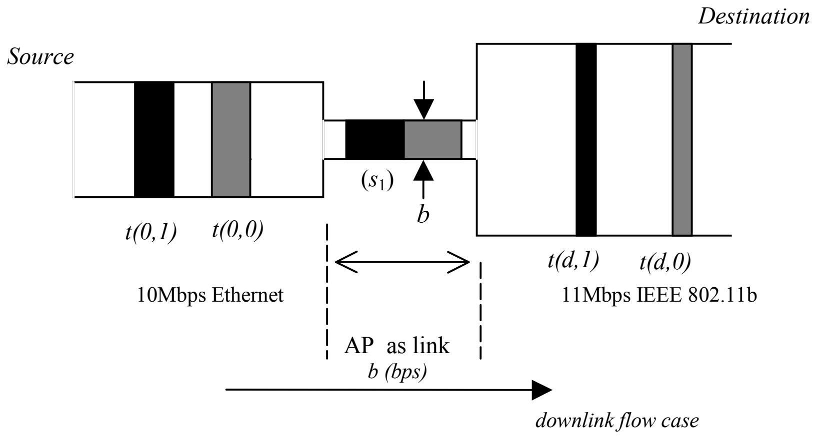

Figure 22. WLAN AP as link with bandwidth *b*. The bandwidth, *b*, represents the speed of the AP-link-model for the downlink flow case. The uplink flow case model is similar with a difference in the direction of flow of test packets. The black and gray rectangles represent data packets travelling from source to destination at different time shots. The gray packet left the source before the black packet, hence it reached the AP link model and the destination before the black packet. *t(0,0), t(0,1), t(d,0)*, and *t(d,1)* are the time values of the packets on the link. In the uplink flow case, *t(0,0)*, *t(0,1)*, and the *Source* are on the IEEE 802.11b side, and *t(d,0)*, *t(d,1)*, and the *Destination* are on the Ethernet side.

When packets in the queue are back-to-back, then the difference between the departures of the first packet and the second packet is the time that the server spent serving the second packet. The Interframe Space (IFS) is calculated as part of the average service-time. Since both test-packets are of equal sizes, substituting in (9) shows that the service time for any of the two packets is the same. Hence the denominator in (10) is the average service time of the test packet:

$$t(d,0) - t(d,1) = average\ service\ time\ of\ test\ packet. \tag{11}$$

Let us denote the average service time of any of the two test-packets as *S*. Equation (3) shows that *S* is a linear function of payload. Thus, we can express (3) in terms of payload (see ε.4 in Appendix 1) for this case to be:

$$S = \frac{r}{32}P + S_o - \frac{5}{4}r \qquad (12)$$

where,

- S is the service time of one of the two test packets in µs;
- P is the IP payload in bytes.

Substituting (12) in (11), we get:

$$t(d,1) - t(d,0) = \frac{r}{32}P + S_o - \frac{5}{4}r \ (\mu s). \qquad (13)$$

Substituting (13) in (10), we express b in Mbps:

$$b = \frac{8(P+h)}{\left(\frac{r}{32}\right)P + \left(S_o - \frac{5}{4}r\right)}\left(\frac{10^6}{1024 * 1024}\right) (Mbps) \qquad (14)$$

where,

- P is IP payload expressed in bytes;
- h is a constant representing the size of the IP header in bytes, i.e. 20B without the *Options* field in IPv4.

Equation (14) shows that the bandwidth of the link model is not constant but rather dependent on the packet size (or payload). Thus, the bandwidth of the link model adapts to different payloads. Consequently we call the bandwidth of the AP-link-model the adaptive bandwidth, b_a, which is simplified in (15) for IPv4 when the *Options* field is not utilized. We use (15), because in our experiments we utilized IPv4 without the *Options* field.

$$\boldsymbol{b_a} \cong 244.14 \frac{(\boldsymbol{P} + 20)}{(r\boldsymbol{P} + 32S_o - 40r)} \quad (Mbps). \tag{15}$$

To analyze the behavior of the adaptive bandwidth in terms of payload, I derive the partial derivative of b_a with respect to P:

$$\frac{\partial b_a}{\partial P} = 7812.48 \frac{S_o - 1.875r}{(rP + 32S_o - 40r)^2}. \tag{16}$$

Since r reflects the transmission time of 32 bytes plus some extra management time in the AP, then S_o is always greater than approximately twice the value of r. S_o will always be larger than the transmission time of 40B of IP payload plus link-layer-overhead, which is more than 2*r on both the downlink and the uplink. Hence,

$$S_o > 1.875r \Leftrightarrow S_o - 1.875r > 0. \tag{17}$$

Knowing (17), then (16) is always positive. From this analysis we conclude that b_a *is monotonically increasing* in terms of P. Consequently, the adaptive-bandwidth of the AP is an increasing linear fractional transformation of IP payload (a fraction of two linear functions of the variable of interest: IP payload). Hence, b_a is limited by the upper b_a bound (function of the largest IP payload) and the lower b_a bound (function of the smallest IP payload in the test design, i.e. 40 bytes) as shown in Figure 23.

Figure 23 shows two plots of two downlink adaptive-bandwidth equations derived for two different APs: AP1 is Lucent WavePOINT-II and AP2 is Lucent AP2000. Figure 24 shows plots for the same APs, but for the uplink adaptive bandwidth analytic solutions. The parameters, S_o and r, of the two APs, on the downlink and the uplink, were calculated as discussed in Section 3.4. Lucent WavePOINT-II shows a higher curve for the downlink and uplink adaptive bandwidths, hence better performance. Using our model and analysis different WLAN APs can be compared in terms of adaptive bandwidth.

This result of b_a can be interpreted as the operational speed of the AP-link-model increases to a maximum bound after which services suffer high delays and packet loss depending on the available buffer size. Figure 23 and Figure 24 show two bounds for the adaptive-bandwidth: the minimum b_a is for smallest IP payload (40B) and the maximum b_a for the largest IP payload (1480B).

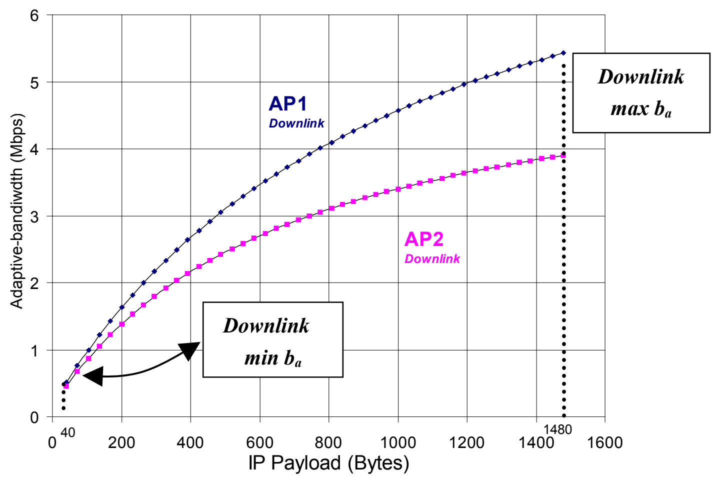

Figure 23. Downlink adaptive-bandwidth (Mbps), downlink b_a, of two WLAN APs. AP1 is Lucent WavePOINT-II and AP2 is Lucent AP2000. AP1 shows higher adaptive-bandwidth values for all payloads. The IP payload is in bytes.

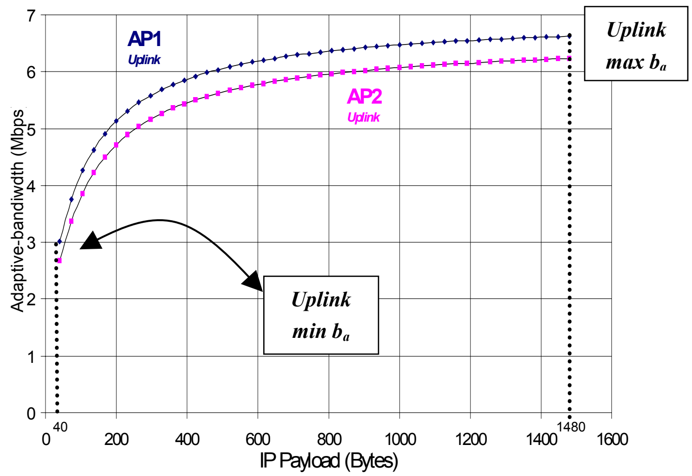

Figure 24. Uplink adaptive-bandwidth (Mbps), *uplink* b_a, of two WLAN APs. AP1 is Lucent WavePOINT-II and AP2 is Lucent AP2000. AP1 shows higher adaptive-bandwidth values for all payloads.

5.3. Throughput

Before defining the throughput of WLAN APs, it is important to draw attention to a very important observation about the throughput of WLANs of IEEE 802.11b in general. Hence, I will start by describing the difference between the speed commercially given for WLANs of IEEE 802.11b and the maximum throughput that could be expected from IEEE 802.11b.

5.3.1. IEEE 802.11b WLAN Maximum Throughput

In this thesis we are interested in the maximum throughput when the IEEE 802.11 DCF basic access method is utilized, as described in Section 2.4.3, because our ETRs are based on using DCF basic access method. I studied closely the transmission speeds in IEEE 802.11b and encountered a specification in the IEEE 802.11 standard that will always degrade the performance of IEEE 802.11b WLANs when speeds higher than 1Mbps are used for transmission.

The IEEE 802.11 standard specifies that the PLCP part of the PPDU (Figure 5) is always transmitted at 1Mbps [35]. Hence, when sending a data packet over an IEEE 802.11b WLAN with 11Mbps, the 144*bit PLCP Preamble* (18*B*) and the 48*bit PLCP Header* (6*B*) are transmitted at 1Mbps, i.e. one-eleventh of the desired 11Mbps speed. This definitely leads to lower performance than if all frame bits were transmitted at a speed of 11Mbps.

One of the problems I witnessed was users misunderstanding of the real transmission speed of the WLAN BSS they are associated with. In this sense, the problem is that when the user of an IEEE 802.11b WLAN decides to connect to the BSS at 11Mbps, he/she thinks the connection is real 11Mbps, as it is the case when the user connects to an Ethernet, where the speed is always 10Mbps (for 10 Base-T). However, the reality in an IEEE 802.11 medium is different from what is commonly understood by bandwidth when the chosen speed is higher than 1Mbps. The reality is that the available bandwidth is definitely less than 11Mbps when the 11Mbps bit rate is specified, because of the slower transmission of the PLCP part of the link frame. In this section, I address this issue in order to compare the expected throughput from a WLAN AP to actual throughput.

Let us consider some data to be sent over an IEEE 802.11b WLAN with 11Mbps, and let us try to calculate the real maximum bandwidth that could be utilized for an 11Mbps IEEE 802.11b WLAN. Consider $\tau(u)$ to be the time (in seconds) consumed to transmit one frame with IP payload of u bytes. Hence, all the parameters presented in Figure 7 are counted in the calculations of $\tau(u)$. Then, the number of frames that could be transmitted in one second is:

$$\eta_{(frames/s)} = \frac{1}{\tau(u)} \qquad (18)$$

where,

$\boldsymbol{\tau}(u) =$ *time to transmit*: **DIFS + PLCP + MAC header + FCS + IP header + IP payload (u) + SIFS + ACK + T$_{\text{backoff}}$.**

The time for backoff in our experiments will be the minimal used, which is 20µs for IEEE 802.11b. Sending only 24 bytes (48B + 6B) of PLCP overhead with 1Mbps speed, and assuming that the IP version used is IPv4 with no *Options*, makes $\tau(u)$ as follows:

$$\boldsymbol{\tau}(u) = \underset{\text{(DIFS)}}{50\mu s} + \underbrace{\underset{\text{(PLCP header and preamble)}}{\frac{192\text{bits}}{(1\text{Mbits/s})}} + \underset{\text{(MAC header and FCS)}}{\frac{34*8\text{bits}}{(11\text{Mbits/s})}} + \underset{\text{(IP packet)}}{\frac{(20+\boldsymbol{u})*8\text{bits}}{(11\text{Mbits/s})}}}_{\text{(PPDU)}} + \underset{\text{(SIFS)}}{10\mu s} + \underset{\text{(ACK)}}{\frac{304\text{bits}}{(11\text{Mbits/s})}} + \underset{(\textit{Backoff})}{20\mu s} \quad (19)$$

The real *Utilized Bit Rate* (*UBR*) is simply the number of frames per second (η) multiplied by the frame size used (depending on payload). Let the frame size with IP payload of *u* bytes be *f(u)*:

$$f(u)_{bits} = \underset{\text{(DIFS)}}{50*10^{-6}*11\text{Mbits}} + \underset{\text{(PLCP)}}{192\text{bits}} + \underset{\text{(MAC header plus FCS)}}{272\text{bits}} + \underset{\text{(IP packet)}}{(20+\boldsymbol{u})*8\text{bits}} + \underset{\text{(SIFS)}}{10*10^{-6}*11\text{Mbits}} + \underset{\text{(ACK)}}{304\text{bits}} + \underset{\text{(Backoff)}}{20*10^{-6}*11\text{Mbits}}. \quad (20)$$

Thus,

$$UBR(u)_{bps} = \frac{f(u)}{\tau(u)}. \quad (21)$$

With (21), we can calculate the maximum bandwidth that could be realized when using packets carrying IP payload of *u* bytes. For example, for a 40B IP payload stream, the *Utilized Bit Rate* that should be expected from an 11Mbps IEEE 802.11b WLAN is:

$$UBR(40\text{B})_{bps} = \frac{f(40\text{B})}{\tau(40\text{B})} \cong 5.8\text{Mbps}. \quad (22)$$

On the other hand, the utilized bit rate of a 10Mbps Ethernet is very near to 10Mbps. Hence there is a clear degradation in performance of IEEE 802.11 WLANs in terms of bandwidth as a QoS parameter. Using larger payloads can enhance this performance. The highest realized bit rate would then be for a stream with IP packets of the size of the link MTU (1500B for IEEE 802.11 WLAN).

Figure 25 shows the real utilized bandwidth in an 11Mbps IEEE 802.11b WLAN for different IP payloads. Note that in Ethernet, the preamble (8B) bits are sent at the same speed as data bits; hence, the utilized bandwidth of Ethernet is nearly 10Mbps due to the small 9.6μs interframe space. So, we expect the throughput of a stream to be less than 10Mbps, depending on the size of user data carried in the payload. I consider the IP payload to be the user data in my calculations of throughput. Hence the parameters used to calculate Ethernet throughput are presented in Table 6. The percentage utilization of the maximum utilized bit rate of WLANs for two APs is presented in Figure 26. Moreover, the real efficiency of WLANs in comparison with the ideal 11Mbps bit rate is calculated as a percentage in Figure 27. Hence, the degradation in performance in real WLANs can be seen with respect to the ideal 11Mbps speed.

Table 6. Ethernet frame components used to calculate throughput of Ethernet link [7].

Ethernet Frame	**Size (B)**	**Time (μs)**
Interframe Gap (Space)	Equivalent to 12.5	9.60
MAC Preamble	8	Consume 35.10
MAC header	14	
MAC trailer (CRC)	4	
IPv4 header, no *Options*	20	
IP payload (***user data***)	***u***	$(8u*10^6)/(10*1024*1024)$
Total	**58.5 + *u***	**44.70 + 0.76*u***

5.3.2. *Throughput of WLAN APs*

The throughput of a data communications link as defined by Stevens in [87] is directly proportional to the link bandwidth (Mbps). Hence, using the link model of the AP, we can conclude that the higher the link bandwidth curve for a given AP, the higher the throughput (bits per second) of the AP. The throughput of a data communications link with one source sending over the link is defined in as:

$$Throughput_{(bps)} = \frac{user\_received\_data}{time\_to\_transmit\_on\_link} \tag{23}$$

$$= \frac{user\_received\_data}{\left(all\_bits\_transmitted\_on\_link \big/ link\_speed\right)}$$

$$= \frac{user\_received\_data}{all\_bits\_transmitted\_on\_link}(link\_speed).$$

We consider the user-received data to be the IP payload, so the throughput of the link of the AP model, T_{APL} (Mbps), would be:

$$\underset{(Mbps)}{T_{APL}} = v.\boldsymbol{b_a} \tag{24}$$

where,

- $v = \dfrac{sum\_of\_all\_IP\_payloads\_received}{sum\_of\_all\_link\_frames\_transmitted}$;
- $\boldsymbol{b_a}$ *is the AP - link - model speed (adaptive bandwidth of the AP - link - model).*

In most links, the bandwidth has a constant value relative to the link protocol. Therefore, the throughput of the link is dependent on the ratio of the *received user data* to *total overhead and data*, when there is no collision. In my AP-link-model, there is no collision, but the bandwidth of the link (b_a) is a function of IP payload. Hence the throughput of the AP-link-model is dependent on IP payload. Examples of maximum throughput of two APs on the downlink are shown in Figure 25.

The analytic solution of the adaptive-bandwidth shows good correlation with the experimental results on the throughput of WLAN APs. Hence, if we consider the IP payload to be the user payload, then the analytic throughput values could be calculated from the adaptive bandwidth formula by using (24). Table 7 compares analytic and measured values of throughput. The measured throughput values are presented in [19]. As (24) shows, the throughput is directly proportional to the adaptive bandwidth. Since the investigation in [19] presents only throughput values for traffic streams, whose IP payloads were 1480B, we could only compare the measured throughput values with the analytic values for 1480B of IP payload (the maximum presented in Figure 23 and Figure 24).

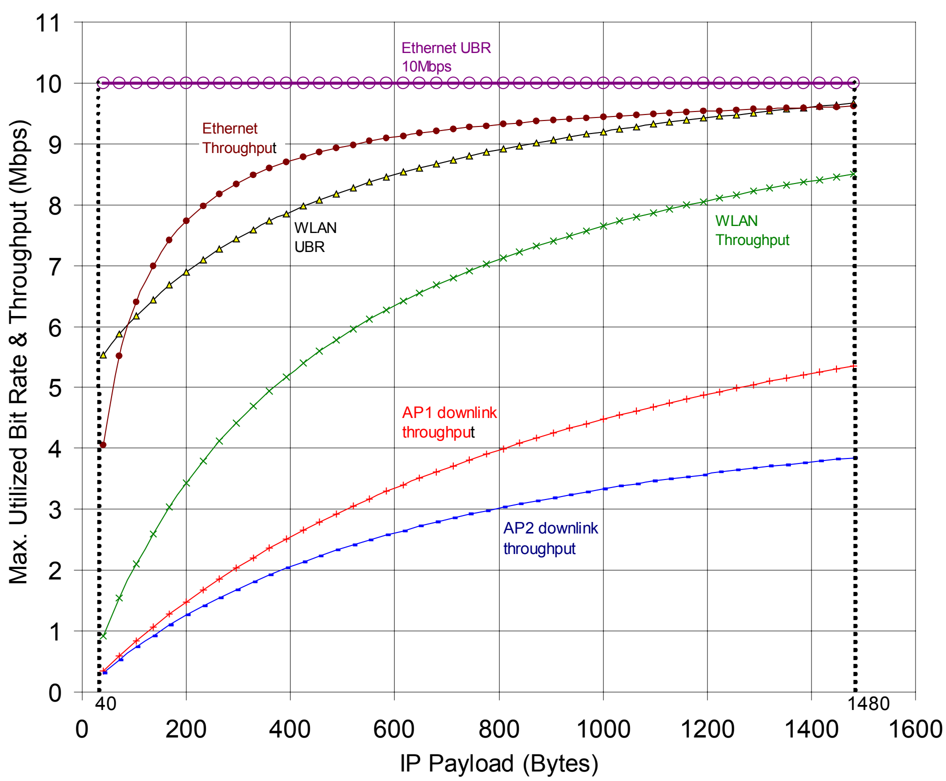

Figure 25. Maximum Utilized Bit Rate (UBR) and maximum throughput in Mbps. Ethernet maximum UBR is 10Mbps. IEEE 802.11b WLAN maximum UBR is a monotonically increasing function of payload and is always less than that of Ethernet. The throughput of 10Mbps Ethernet is a monotonically increasing function of payload. The throughput of the 11Mbps IEEE 802.11b WLAN link is a monotonically increasing function of payload, but always less than the throughput of 10Mbps Ethernet link. AP presence degrades performance. AP1 is Lucent WavePoint-II and AP2 is Lucent AP2000. Maximum throughput of AP2 is less than that of AP1 for all payloads. Both AP1 and AP2 have a monotonically increasing throughput in terms of payload.

The differences in the values of adaptive bandwidth and throughput in Table 7 are due to the value v in (24), which is not considered in the calculations of the adaptive bandwidth. Measurement errors due to clock drifts and time stamp resolutions in [19] lead to different standard deviations in the statistical results. Hence, looking at the standard deviation of the measured throughput of Table 7, the differences in the calculated and measured values could be explained. Consequently, good correlation between the measured values and the analytic values are observed.

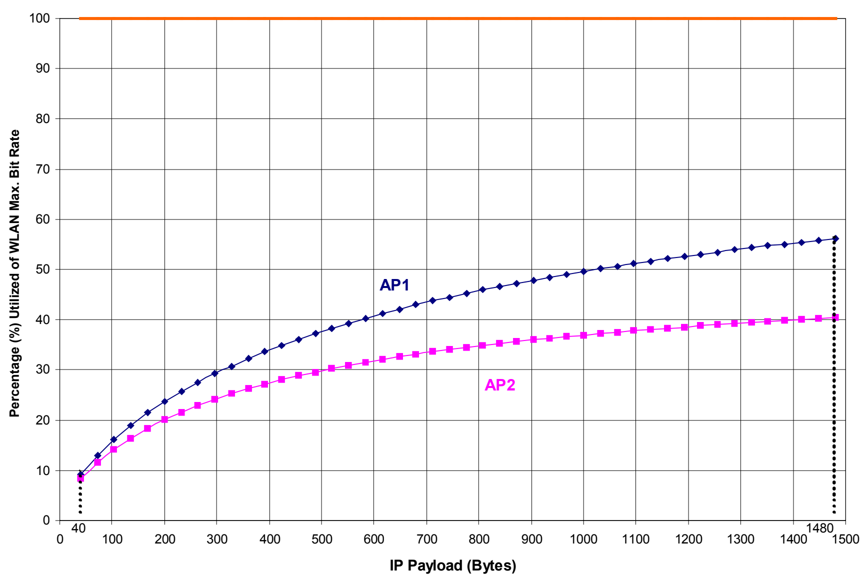

Figure 26. Percentage of the total WLAN bit rate used for two APs: AP1 is Lucent WavePoint-II and AP2 is Lucent AP2000. This graph can clearly show the percentage degradation in performance.

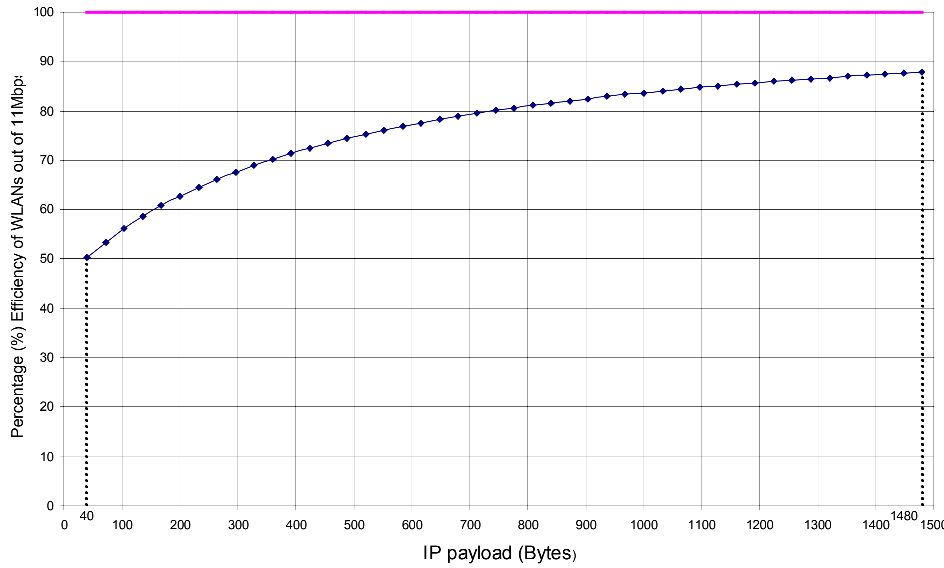

Figure 27. Efficiency of WLANs percentage as percentage of the ideal 11Mbps bit rate.

Table 7. Comparison between *measured throughput*, analytic throughput and *adaptive-bandwidth* for 1480B IP payload. Measured values and standard deviation values are from [19].

		AP1 Lucent WavePOINT-II Downlink	AP1 Lucent WavePOINT-II Uplink	AP2 Lucent AP2000 Downlink	AP2 Lucent AP2000 Uplink
Analytic	**Adaptive-Bandwidth (Mbps)**	5.41	6.62	3.89	6.23
	Throughput (Mbps)	5.35	6.54	3.85	5.93
Measured	**Maximum Throughput (Mbps)**	6.41	5.94	3.91	5.93
	Standard Deviation of Throughput (Mbps)	1.4 1	0.41	0.10	0.01

5.4. Feedback Control System for AP Throughput

The adaptive bandwidth formula inspires us to look at the throughput as a ratio of an output-function to and input-function of a feedback control system [48] as shown in Figure 28. Since there are two average service-time formulae for each AP (one for the downlink and one for the uplink), then there are two feedback control models for the throughput: uplink feedback model and downlink feedback model.

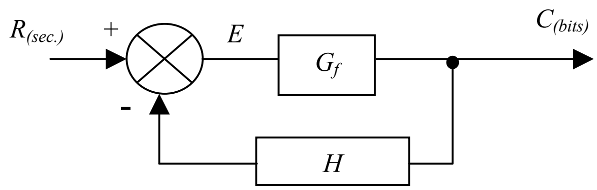

Figure 28. Feedback control model for throughput of the AP-link-model. R is the input function in seconds, and C is the output function in bits. G_f is the feed-forward transfer function, and H is the feedback transfer function. The ratio of C over R gives the throughput (in Mbps) of the AP-link-model. E is the error, and it is the difference between the input R and the product HC.

The functions G_f and H, as shown in (25) and (26), can be related to AP parameters and to the parameter v described in (24).

$$G_f = 244.14v\frac{P+h}{(32S_o - 40r)}. \tag{25}$$

$$H = \frac{rP}{244.14v(P+h)}. \tag{26}$$

Both uplink and downlink feedback models of throughput for the same AP can be represented as shown in Figure 28 with a difference in the values of S_o, r, and v that are the variables in G_f and H, respectively. Each direction of flow (uplink or downlink) has a different set of the three variables S_o, r, and v. Hence each of the two directions of flow has a different G_f and a different H.

Analyzing the system in Figure 28, we get the throughput (Mbps), which is the ratio of *C(bits)* to *R(sec.)* as shown in (27).

$$T_{APL} = \frac{C}{R}(bits\,/\,\mathrm{sec.}) = \frac{G_f}{1+G_f H} \quad (Mbps). \tag{27}$$

This model shows that the error (as described in feedback control being the value E shown in Figure 28), is due to the difference between the input, R, and the product HC. Hence, the *feedback transfer function*, H, plays a significant role in the value of the error, E. Consequently, analyses of the feedback system can be used to investigate the effects of the AP parameters, S_o and r, on the behavior of the AP. The feedback control model is beneficial for further analyses of the behavior of traffic passing through a WLAN AP. In this respect, one can separate the parameters that affect the feed-forward portion of throughput (G_f) from the parameters that decide the feedback portion (H) and its effects; a result that is useful for data traffic shaping, simulation and analysis. Furthermore, the equations of the feedback model can be reversibly studied for the effect of the design of the AP hardware and its correlation with protocol implementation on QoS.

5.5. Summary and Concluding Remarks

The main purpose of this chapter is to present a link model of the AP and a feedback control model used to parameterize the throughput of WLAN APs. The AP is looked upon as a data communications link. The link model of the AP builds on our previous results of the AP queuing-model and its service-time analytic solution. The packet-pair property of FIFO-queuing networks is shown to be very suitable for the link model analyses. We are able to parameterize the performance of the AP as if it were a link with adaptive-bandwidth, bounded by two limits: the lower and upper adaptive-bandwidth bounds. Performance of different AP brands could be compared using the adaptive-bandwidth characteristic. Adaptive-bandwidth of WLAN APs was analyzed and shown to be a strictly increasing linear fractional transformation of payload. The adaptive-bandwidth of the AP-link-model is used to calculate the throughput of the AP. Since the throughput of a data communications link is directly proportional to the speed of the link, then a major result is that when using our model and analysis, performance of different WLAN APs can be compared from a throughput point of view. An important observation about the real utilized bit rate in WLANs is presented in comparison to Ethernet utilized bit rate. I show that 11Mbps IEEE 802.11b WLANs have less utilization of the bit rate than the 10Mbps Ethernet. Moreover, when an AP is introduced, the throughput is degraded. The throughput of the AP is studied as the ratio of output to input of a feedback control system. Hence, some relationships between the AP parameters could be derived from the feedback loop model in order to enhance QoS relative to specific applications. Therefore, fruitful suggestions and implementations could be presented to manufacturers and designers of WLAN APs.

Chapter 6

Video Experiments

"Never mistake motion for action."

Ernest Hemingway

This chapter presents results of video experiments on WLAN APs and compares them to GPRS video experiments. The results are part of a larger research on multimedia traffic performance over WLANs and GPRS. The video investigation is a QoS study focusing on: bandwidth and packet loss. GPRS is an evolution of GSM providing a higher data rate. Moreover, expectations are that GPRS and WLAN will coexist. In our study, we investigate multimedia video traffic. The video codec used is H.261 with QCIF resolution. Section 6.1 is a brief overview of GPRS. Section 6.2 discusses multiplexing gain for GPRS. In this chapter we focus on minimum required bandwidth for acceptable QoS of QCIF H.261 video streams over the wireless and mobile medium, GPRS. Section 6.3 describes the experiments, presents results, and compares video performance of GPRS and WLAN APs through a set of metrics. Section 6.4 evaluates the results. Section 6.5 summarizes the chapter and points out some remarks.

6.1. Introduction

The wireless and mobile telecommunication world is experiencing a very critical transitional stage, where new QoS parameters are to be defined. Within a more general study on mobile systems evolution, we analyze multimedia traffic over GPRS and compare the results to those over WLANs. The motivation behind this study is due to the fact that many vendors have already started deploying WLAN and GPRS chipsets in their network interface cards and equipment. Hence, there could be many scenarios when both technologies are utilized by the same user with services shifting between WLAN links and GPRS connectivity. Therefore, differences in QoS between the two technologies will affect user satisfaction, which makes a motivating reason to study differences in performance especially when multimedia applications are considered.

GPRS represents an evolutionary step from the existing GSM system, where its purpose is to bring packet switched data services to the mobile system. This packet-based communication, being also wireless, makes comparison with WLANs interesting.

With GPRS, the user can always be connected to the network since charging is not based on the connection time. One of the other goals of GPRS is to try to provide higher speeds than traditional GSM systems. The maximum theoretical speed over GPRS is supposed to be around 115Kbps. This bandwidth is achieved with very good radio conditions, and when the network is fully developed. In practice, the starting GPRS speed would, to a large probability, be somewhere between 20Kbps and 56Kbps. An enhanced GPRS system called EDGE is supposed to bring the speed up to 384Kbps. This very evolutionary phase of mobile systems is believed to be only one step towards the *third mobile systems generation* (*3G*), which is expected to give speeds up to 2Mbps. The GSM system uses Time-Division Multiple Access (TDMA) with eight radio frequency time slots. A network operator can dedicate 0 to 8 of these time slots to GPRS. Each mobile terminal can send/receive in 1 to 8 time slots. It is believed that the first mobile terminals generation for GPRS will support 4 time slots downlink and 1 time slot uplink, which gives around 14Kbps uplink and 56Kbps downlink. With this great shift that GPRS will introduce to the wireless and mobile world, we are interested in investigating the quality of service that GPRS can offer to multimedia applications, mainly video quality. This is due to our belief that multimedia applications will be the killer future applications, especially when GPRS and WLAN are deployed on the same chipset. Our research in this area is long term; however, in this chapter we investigate few multimedia traffic parameters for one video standard. The format of the video streams we investigate over GPRS is H.261 with QCIF resolution. H.261 is chosen for its low bit rate [17], which suits packet based communication networks [14]. The H.261 video streams in the experiments are variable bit rate streams, which makes them more suitable for the medium [9]. Quarter-CIF (QCIF) has 176 pixels per line, and 144 lines [78]. QCIF is chosen, because it is mainly used for desktop videophone applications, i.e. the size will be suitable for a mobile unit. In addition, all codecs must be able to handle QCIF.

One of the interesting parameters for GPRS is the minimum bandwidth required for acceptable performance of H.261 video streams of QCIF resolution. We are also interested in self similarity since if we can define which type of videos show self similarity over GPRS, then GPRS vendors can learn more about how to deal with video over this medium [15][86]. In this respect, the Hurst parameter is calculated [41]. The Hurst parameter can be looked at as a self-similarity value; if near to 1, then this would be a sign of self-similarity. However, if it shows a value nearer to 0.5, then there is not much of self-similarity in the traffic.

6.2. Multiplexing Gain

In a study of multimedia over GPRS, it is very important to note that the standards with which the QoS is judged are subjective. Unfortunately, up till now, the judgements on acceptable QoS for multimedia streams are relative to the observer's personal standards [64]. Hence, we find it very important that, in our study, the minimum acceptable parameters investigated are defined by a representative number of people from different population backgrounds. Hence a common acceptable QoS is set to find the minimum bandwidth sought. To calculate the theoretical values for the minimum acceptable bandwidth, we use the *Multiplexing Gain* formula [51]:

$$MG_z = z\frac{P_R}{V_z} \tag{28}$$

where,

- P_R is the peak rate of the video stream;
- z is the number of independent streams combined for transmission, $z \in \mathbf{N}^*$;
- V_z is the link-bandwidth required to achieve desired QoS for the multiplexed stream of z sources (V_1 being the link bandwidth for a single source).

From (28), we conclude that:

$$MG_z = z\frac{P_R}{V_z} = \left(z\frac{V_1}{V_z}\right)\left(\frac{P_R}{V_1}\right) = \left(z\frac{V_1}{V_z}\right)MG_1 \tag{29}$$

where MG_1 is the multiplexing gain for one source.

$$\text{Equation (29)} \quad \Leftrightarrow \quad V_z = z\left(\frac{MG_1}{MG_z}\right)V_1. \tag{30}$$

The multiplexing gain is a parameter I use in order to achieve the minimum link-bandwidth for z streams. The multiplexing gain MG_z for z independent streams is given by,

$$\frac{1}{MG_z} = \frac{1}{pta} + \left(\frac{1}{MG_1} - \frac{1}{pta}\right)z^{\frac{1-2Hu}{2Hu}} \tag{31}$$

where,

- pta is the peak-to-average ratio;
- H_u is the Hurst parameter.

Many methods can be used to calculate the Hurst parameter (degree of self-similarity [50]), like time variance plot, R/S analysis [55], and periodogram method.

6.3. Experiments and Results

Figure 1 shows the testbed, which consists of two video senders, a GPRS emulator, a receiver, and a traffic measurement and analysis tool, NIKSUN NetVCR™ (*www.niksun.com*). Experiments 1-5 use one sender only, for they are dedicated to studying the bandwidth required for one video stream. On the other hand, experiment 6 concentrates on performance when multiple streams are sent over GPRS. For WLAN APs, we used the testbed shown in Figure 12, and we ran video experiments on the downlink and uplink for different video types. Table 8 shows the video streams used, where "Comm" is used in experiments 1 to 5. The packet time slots on the GPRS medium are set to 8 time slots throughout all the experiments since using less for video transmission will not lead to acceptable QoS. First, we look into whether there is any difference between the behavior of two entities: GPRS with no restricting limits, and WLAN AP. The associated results for the H.261 video stream are presented in Table 9, where one would conclude that when the GPRS is dedicated to one video stream, with no background users, it will most likely behave like Ethernet link. The same result is for a WLAN AP. However, when running the first experiment on 10 Mbps Ethernet and through and AP, we got no missing frames at the receiver end, while in running the experiment over GPRS, with 12dB, we had 2,670 video frames missing out of 6,306 video frames of the same stream (see Table 9). Figures 30-37 show the number of bytes (vertical axis) versus the packet size categories (horizontal axis).

Table 8. Video sequences used in the experiments. "Comm" is used in experiments 1 to 5.

Type of video	Length (mm:ss)	Total bytes	Total packets	Average bandwidth (bps)	Hurst param.
Commercial **"Comm"**	05:06	3,631,412	6,942	94,322	0.79
News	13:55	10,657,756	18,313	101,623	0.74
Talking head	12:51	11,682,038	13,682	120,901	0.61
Music	03:25	2,633,906	4,582	101,793	0.88
Total	44:14	33,402,294	56,205		

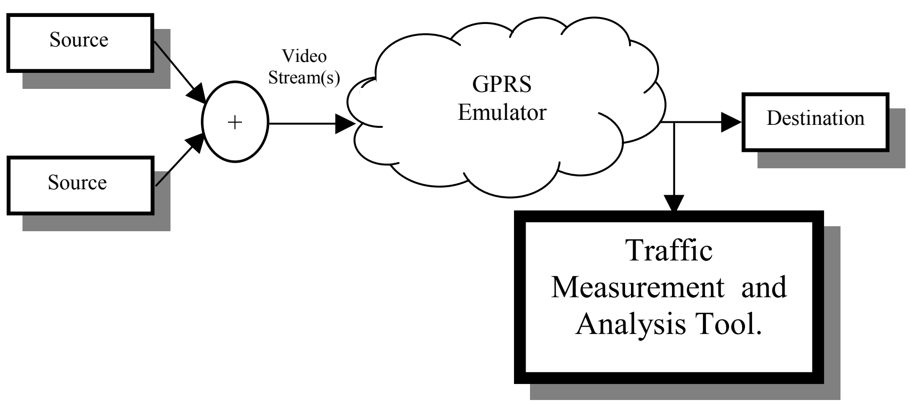

Figure 29. Testbed for bandwidth investigation of H.261 video quality over GPRS .

In Figures 30-37, the horizontal axes show packet size categories of size PX bytes, where labels 1, 2, 3, 4, 5, and 6 represent categories with 1B ≤ PX < 128B, 128B ≤ PX < 256B, 256B ≤ PX < 512B, 512B ≤ PX < 1024B, 1024B ≤ PX < 2048B, and PX = 2048B, respectively. In

Figure 29, the peak (bytes) is for the packets that are 512-1024 bytes in size, excluding 1024 byte packets. The second level peaks are for packets of 1024-2048 bytes (excluding 2048 byte packets) and 216-512 bytes (excluding 512 byte packets) respectively. Figures 31-37 can be read in a similar way for the associated experiments.

Table 9. Differences between GPRS (S/N=12 dB) with no restrictions and WLAN AP for "Comm". Same results on WLAN are for Ethernet as presented in [29].

	No limits on GPRS	**WLAN AP**
Total no. of bytes received	1.7284e+003	1.7289e+003
Median	3848	3788
Peak-to-Average ratio	5.3804	4.7394

In the presence of a WLAN AP, we ran experiments by sending the video streams and calculating performance metrics. The results on WLAN APs are shown in Table 10 and Table 11. We chose APa and APb (which are Lucent WavePoint-II and AP500), because we can compare the results on the video with the results on buffer size allocation presented in Section 4.4.

Looking at these results, we see that in the presence of a WLAN AP, some video packets are either transmitted with an erroneous checksum by the AP or are badly received at the wireless station. In both cases, these packets can not be used by the application at the receiver end. Moreover, we notice that for all videos except the "Comm", it took longer time on the downlink than on the uplink to be transmitted by the AP. This was due to the packet sizes used in the stream for "Comm". We notice few packets lost due to buffer filling, and these are presented in the tables as the IP packet loss. If we compare the data after transmission by the AP, we find that some video frames are missing compared to the total number of frames that each video consisted of. This is again due to the IP packet loss and the erroneous checksum (i.e. error in transmission by the AP). So the AP introduces some degradation to the video transmission, and comparing the AP video results to the original video data, the percentage of degradation can be calculated.

Table 10. Downlink video experiments on APa and APb. APa is Lucent WavePoint-II. APb is Lucent AP500.

Downlink Exps.	**APa**				**APb**			
	Video frames	*Packets with correct checksum*	*IP packet loss*	*Transmission Time (sec)*	*Video frames*	*Packets With correct checksum*	*IP packet loss*	*Transmission time*
Comm	6061	6916	0	374	6068	6926	2	306
News	14598	18261	4	838	14635	18312	3	838
Talk head	5736	13669	5	772	5732	13661	9	772
Music	4236	4545	0	205	4271	4581	1	205

Table 11. Uplink video experiments on APa and APb. APa is Lucent WavePoint-II. APb is Lucent AP500.

Uplink Exps.	**APa**				**APb**			
	Video frames	*Packets with correct checksum*	*IP packet loss*	*Transmission time (sec)*	*Video frames*	*Packets With correct checksum*	*IP packet loss*	*Transmission time*
Comm	6074	6932	2	306	6074	6932	1	307
News	14635	18310	3	838	14627	18304	1	837
Talk head	5740	13681	5	772	5740	13681	1	772
Musc	4271	4581	0	206	4271	4581	4	206

As for GPRS, we use the parameters in Table 12 and Table 14 to calculate difference with the values received by the application. The number of missing video frames is calculated. We believe that these numbers are very important to relate to acceptable QoS of the H.261 streams over GPRS. Fewer frames were missing in WLAN AP experiments than in GPRS experiments. These results are interesting to calculate the acceptable bandwidth for GPRS video transmission. The available bandwidth on WLANs is much higher than that on GPRS, hence studying the multiplexing gain and the minimum required bandwidth is interesting in the case of GPRS.

Table 12. Number of packets vs packet size for "Comm" video stream.

Packet Size Categories (Bytes)	**Count (Packets)**				
	WLAN	GPRS			
	Exp.1,fig.1	**Exp.2**,*fig.2* 12dB 0 BGU	**Exp. 3**,*fig.3* 15dB 0 BGU	**Exp. 4**,*fig.4* 12dB 20 BGU	**Exp.5**,*fig.5* 12dB 40 BGU
0 to 128	507	305	271	282	275
128 to 256	1089	651	594	585	623
256 to 512	2332	1332	1365	1302	1367
512 to 1024	2152	1312	1303	1232	1302
1024 to 2048	862	658	672	606	641
2048	0	14	19	12	21

Table 13. Statistics for "Comm" Video Stream in comparison with GPRS experiments

	"Comm" statistics	GPRS Statistics			
		Exp .2 12dB 0 BGU	Exp. 3 15dB 0 BGU	Exp. 4 12dB 20 BGU	Exp. 5 12dB 40 BGU
Total Number of Bytes	7262824	4639074	4674424	4366946	4614556
Average Rate (bps)	48419	77318	10387	72782	102546
Number of Packets	13884	8544	8448	8038	8458
Average Rate (pps)	12	18	23	17	23
Minimum Packet Size	67	69	68	67	69
Maximum Packet Size	1066	1066	1066	1066	1066
Mean Packet Size	523	542.96	553	543	546
Packet Size Variance	93519	101586	100581	99668	99399
Variance/Mean	178	187	182	183	182
Missing Video Frames	0	2670	2719	2811	2788

In the second experiment, a GPRS emulator is used. The S/N is 12dB i.e. worst case. However the main concern of this experiment is to measure the same parameters as in experiment 1 but over GPRS, hence we have no BackGround Users (BGU), i.e. the "Comm" video has all the available bandwidth. The results are presented in Figure 31. Collecting these parameters, we can also find a similarity in the QoS delivered as well as the shapes of the graphs in Figures 30 and 31. Around 41% of the frames are missing, but still the QoS is acceptable. The receiver end shows a rate ranging between 48Kbps and 62Kbps, which is a very good rate in our point of view for a video transmission with a QCIF resolution.

We also investigate the behavior and the used-bandwidth results for the 15dB S/N. We ran exactly the same experiment as in experiment 2, but with 15dB instead of 12dB. The result is shown in Figure 32. This experiment shows just a slight difference where the number of are large packets used is increased, i.e. the traffic concentration is more in the middle (512-1024B) than in experiment 2. The rate ranges between 48Kbps and 63Kbps i.e. acceptable.

In addition, we tried experimenting with forced background users over the GPRS network. The number of background users we have in experiment 3 is 20, with 12dB. The result can be seen in Figure 33. The behavior still shows a graph similar to the previous experiments. The rate at the receiver's end still shows a range between 40Kbps and 60Kbps.

In the fifth experiment we force 40 users with 12dB over GPRS. The behavior is similar to the previous experiments in terms of graphical shape (Figure 34), but the video quality drops down. In fact it is not acceptable at all. However, the rate at the receiver's end still shows a range between 48Kbps and 60Kbps.

In the sixth experiment our concentration is on the bandwidth when multiple streams are injected over GPRS, 12dB and 40 BGU. The results are predictable as shown in Figures 35, 36 and 37. Filtering the traffic of each stream alone is important to study the bandwidth from the multiplexing gain point of view. Figure 35 shows the result for the traffic of both H.261 video streams at the same time. Since both streams, when injected together, have their first level peaks at (512-1024B), as well as their second level peaks at the same points (Figures 36 and 37), then adding the two would lead to a graph with peaks at the same relative points (Figure 35). For the first stream, the rate at the receiver's end still shows a range between 21Kbps and 37Kbps. For the second stream, the rate at the receiver's end still shows a range between 30Kbps and 50Kbps. The rates show that the bandwidth is divided, and this is a normal behavior. Around 60% of the frames are lost in each video stream. The visual effects on the QoS can be observed, and the delay between the frames is not within the acceptable range when there are transitions in the video stream. We also run multiple streams over GPRS to investigate more on the multiplexing gain and suitable bandwidth for the set QoS. Results are shown in Table 16.

Referring to (30), and if we know MG_1 and V_1, then knowing V_z will be just a matter of knowing MG_z, which can be calculated using (31). For an acceptable QoS, we will use (28) to obtain a value of for $V_1 = P_R$, where P_R is investigated in the experiments to be around

70Kbps. This makes $MG_1 = 1$ for acceptable QoS. Hence (30) becomes $V_z = 70z / MG_z$ for acceptable QoS.

Table 14. *Number of packets vs packet size; 2 video streams; GPRS, 12dB, 40 BGU.*

Packet Size Categories (Bytes)	**Count (Packets)**		
	GPRS, 12 dB, 40 BGU		
	Total of Two video streams	First video stream **filtered**	Second video stream **filtered**
0 to 64	355	-	-
64 to 128	307	162	144
128 to 256	824	366	313
256 to 512	1433	747	699
512 to 1024	1452	730	728
1024 to 2048	624	304	305
2048	13	0	8

Table 15. *Two video streams over GPRS, 12dB, 40 BGU.*

	GPRS, 12dB, 40 BGU		
	Total of Two video streams	First video stream **filtered**	Second video stream **filtered**
Total number of Bytes	4949190	2456798	2401096
Average Rate (bps)	94270	16379	4573
Number of Packets	10016	4618	4394
Average Rate (pps)	24	4	10
Minimum Packet Size	60	72	68
Maximum packet size	1066	1066	1066
Mean Packet size	494	532	546
Packet Size Variance, B^2	10534	96162	97318
Variance/Mean	213	181	178
Missing video frames		4499	4399

Table 16. Multiplexing gain and min. bandwidth for a increasing number of video streams.

No. of Streams	Average (bits/interval)	Peak (bits)	Peak-to-Average	Hurst Param.	Multiplexing Gain	=>	Minimum Bandwidth (Kbps)
2	198,307	278,528	1.40	0.87	1.82	=>	76.9
3	297,458	455,384	1.53	0.79	1.83	=>	114.8
5	487,719	628,664	1.29	0.82	1.60	=>	218.8
10	970,794	1,246,264	1.28	0.83	1.50	=>	466.7
15	1,455,795	2,002,776	1.38	0.84	1.54	=>	681.8

6.4. Evaluation of Results

For WLAN APs, there is degradation in throughput due to packets neglected by the application because of erroneous checksums rather than IP packet loss in the AP itself. This is an interesting observation that inspires me to investigate more on the percentage of erroneous transmission by an AP brand. A study that could be interesting to UDP based applications. Moreover, The AP seems to introduce a degradation in quality but not as that of GPRS, basically due to the bandwidth difference. Hence it was interesting to look at the multiplexing gain of GPRS.

As we increase the number of users and limitations on the GPRS, our calculations lead to a final value that we would like to present. The minimum acceptable bandwidth for H.261 video streams QoS over GPRS is found - after many iterations and trials - to be around 70-80Kbps for one QCIF H.261 video stream. This number is not very satisfactory since the practical limit that GPRS can deliver now is around 50Kbps. However, work is going on to reach higher practical limits, and if 70Kbps is reached, then sending video streams with QCIF resolution will be possible for the defined QoS. When a bandwidth of less than 70Kbps over GPRS is reached, the video quality and the missing frames number are not acceptable. In this respect, and regarding the transmission of multiple streams to one receiver and two different application port numbers, the sharing of the bandwidth will surely happen. However, the bandwidth needed over GPRS for the H.261 video for z streams will be less than the sum of the peak rates of the two streams. In other words, multiplexing gain will occur and will be a value greater than 1,

$$V_z = z\frac{P_R}{MG_z} < zP_R = zV_1; \quad MG_z > 1, \quad z \in \mathbf{N}^* . \qquad (32)$$

The quality with multiple streams will always be less than for one video sent as shown in experiment 6.

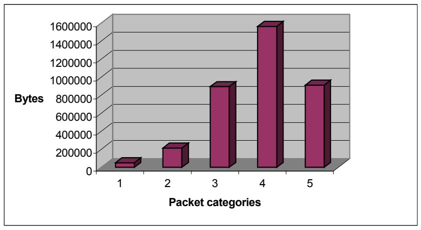

Figure 30. Exp. 1, "Comm" bytes vs packet categories over 10-BT.

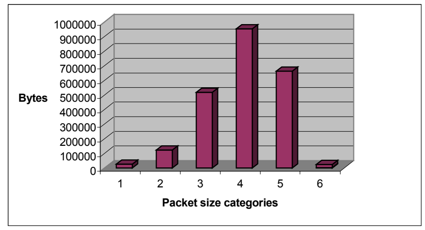

Figure 34. Exp. 5, "Comm" bytes vs packet size categories; GPRS, 12dB, 40 BGU.

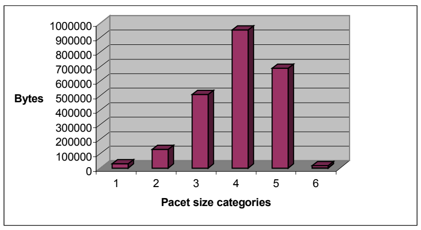

Figure 31. Exp. 2, "Comm" bytes vs packet size categories; GPRS, 12dB, 0 BGU.

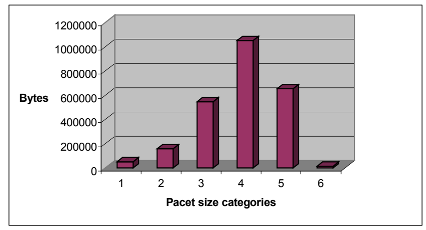

Figure 35. Exp. 6, two video streams traffic over GPRS, 12dB, 40 BGU.

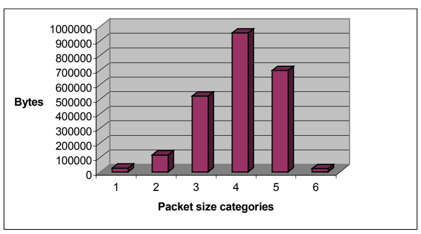

Figure 32. Exp. 3, "Comm" bytes vs packet size categories; GPRS, 15dB, 0 BGU.

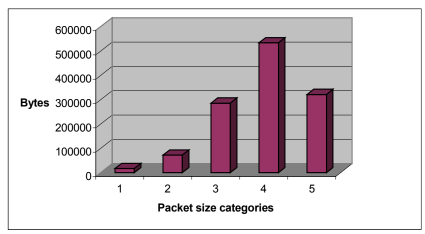

Figure 36. Exp. 6, first video stream over GPRS, 12dB, 40 BGU.

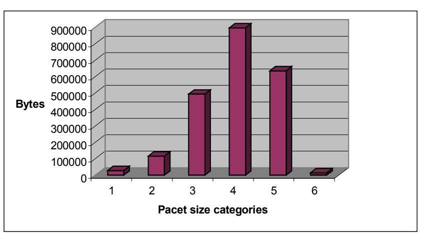

Figure 33. Exp. 4, "Comm" bytes vs packet size categories; GPRS, 12dB, 20 BGU.

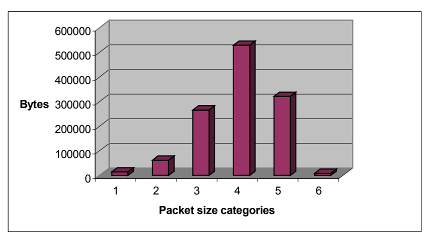

Figure 37. Exp. 6, second video stream over GPRS, 12dB, 40 BGU.

6.5. Summary

Motivated by the industry trying to deploy WLAN and GPRS chipsets on the same networking equipment, we investigate some multimedia traffic parameters over WLAN APs and GPRS. The video streams investigated are encoded in H.261 codec. QCIF resolution is chosen for investigation since it can be deployed on mobile units. The minimum bandwidth required for acceptable QoS of QCIF H.261 video is dependent on the peak rate of the video, the number of streams, and the medium multiplexing gain. For one video stream, the minimum bandwidth for GPRS is around 70Kbps, which is still not easy to achieve over real GPRS networks. However future GPRS generations are expected to supply this bandwidth and more. The encouraging part is that when two or more video streams are injected, they need less bandwidth than the sum of the peak rates of each. We hope that our study triggers more investigation in the filed of multimedia over GPRS from the traffic analysis point of view. We can see a clear degradation of the video quality on GPRS in comparison to a WLAN AP, and that is mainly due to the higher available bandwidth of WLANs. Though I was expecting less frame loss through a WLAN AP, my expectations were not totally true. Less IP packets were lost, but effectively, and from a throughput point of view, some frames were transmitted, but could not be used by the video receiver. Consequently, we lose quality, and since these frames are sent in IP packets, then we are also losing a percentage of the available bandwidth. This is an interesting issue, which motivates investigations in the *successful transmission of WLAN APs*, because it is one of the main elements of AP throughput together with *IP packet loss*.

Chapter 7

Conclusions

"If a man will begin with certainties, he shall end in doubts; but if he will be content to begin with doubts, he shall end in certainties."

Francis Bacon

We began by being uncertain how the WLAN AP worked, and then I made some assumptions and ended up with a queuing system model for the delay attribute and a link model for the throughput of APs. We used the link model to construct a feedback control system, whose ratio of output to input gives the throughput (Mbps) of a WLAN AP. In this chapter I present concluding remarks that summarize the work presented in this thesis.

In this thesis, we present two main models for the WLAN AP as a system of reference in order to understand the behavior of APs and evaluate their performance. We also introduce a feedback control model for the throughput of a WLAN AP. To the best of our knowledge, there has been no available model for the delay processing of WLAN APs running IEEE 802.11b, nor has there been an analytic solution for the throughput of a WLAN AP. This thesis tackled the problem of modeling and showed that the AP delay attribute could be modeled as a single server, single queue system with an analytic solution for the service time in terms of IP payload.

One of the key results of the work in this thesis is the parameterization of average service time in terms of IP payload and two AP characteristic-parameters. The AP parameters can be calculated through processing of some data that can be collected from specific tests. The tests were also designed for the purpose of extracting model parameters. The Simple Service Time Producer (SSTP) algorithm (implemented in C++) was designed in order to analyze the collected data. Because we have a statistical process of collecting data, the average service time of a packet has been analyzed and found to be an increasing linear function of IP payload. The corresponding average service time formula can help users and manufacturers to estimate (with errors lower than 3%) the time required to serve a packet passing through a WLAN AP by using an equation rather than running long experiments. This average service time solution is a result that can be inspiring and beneficial for simulation, traffic shaping and analysis in IP based networks where a WLAN AP exists on the path.

The average service time on the downlink is relatively higher than that on the uplink for all APs that run the IEEE 802.11b protocol for a packet with identical IP payload. This is due to the relatively higher overhead required for transmission over the wireless medium. The absolute difference between uplink and downlink average service time values for all payloads used was analyzed and the result was that a classification into two kinds of APs, of those tested. As payload increases, some APs had an increasing absolute difference between uplink and downlink, and others had a decreasing absolute difference with increasing payload. We introduced the notion of the Uplink-Downlink Contrast of an AP, α, and a packet with payload x in bytes, UDC(α, x), which is defined as the absolute difference between the uplink and downlink average service time values of the packet carrying a payload of x bytes. This notion made it simpler to characterize APs by introducing two characteristics to the UDC: convergent and divergent UDC. A convergent UDC is that where the absolute value of the difference between uplink and downlink average service time values decreases with increasing payload. A divergent UDC is the contrast where the absolute value of the difference between uplink and downlink average service time values increases with increasing payload. The UDC is very useful as a QoS parameter to decide on which AP to use, especially in the cases of real-time two-way traffic, like videoconferencing.

We also studied the buffer management and behavior and found out that the choice of a suitable AP is not only dependent on the service time, but it also depends on the application and how sensitive this application is to packet loss and delays. If the application was sensitive to packet loss, a larger buffer is better to use, otherwise if the application is sensitive to delay (like real audio), then smaller buffer sizes are better for the quality of service of the application. In order to estimate initial buffer sizes, the Buffer Size Estimator (BSE) algorithm was designed and implemented (in C++). Results show different initial and hence later buffer size allocations in different APs. The buffer adapts itself to payload as well. The more the payload, the more the buffer allocation is.

Throughput of a WLAN AP is also studied in a new way, where the AP is imagined as a link and modeled as such. We used the packet pair technique for FIFO queuing networks to solve the problem of the throughput analytically. The results prove that the throughput (Mbps) of a WLAN AP is a linear fractional transformation of IP payload; a result that is extremely important for evaluation of WLAN APs and WLAN networks. The results of the analytic solution are compared to real measurements of throughput and they show very good correlation.

Finally, being motivated by the plans to deploy different link layer technologies (namely WLAN and GPRS) on the same chipset by some vendors, some video experiments were run on WLANs and on GPRS and compared. The results show that the losses in application packets when an AP is deployed are not necessarily due to IP packet loss, but there is a percentage of packets that are erroneous (i.e. erroneous checksum). In GPRS, the IP losses were more, mainly due to less bandwidth, however the encouraging point in GPRS is that when more than one stream are on the link, the multiplexing gain shows relatively good values. This comparison study of video performance is an initial step we take in this direction since we see many vendors trying to have both technologies (GPRS and WLAN) on the same network card.

All in all, the results of the thesis can serve industry and end users to evaluate performance of WLAN APs and the corresponding WLAN networks by using our test designs, algorithms, analysis, and programs.

Chapter 8

Open Issues and Future Work

"A wise man will make more opportunities than he finds."

Francis Bacon

This chapter lists open issues in the presented research and thus suggests future work. Section 8.1 discusses research points that have been left without investigation. Section 8.2 looks at the open issues and suggests specific future tasks of research.

8.1. Open Issues in AP Performance

In this thesis, some issues remain open and have not been studied yet. For instance, noise in the WLAN medium had not been considered in its effect on delay. However, since we are trying to benchmark the AP for its maximum performance, then we can assume that with noise the performance will be worse than when evaluated without noise.

The case when multiple access points are present is also not investigated. Moreover, effects on background traffic, though it is not an issue to be considered in benchmarking, but could be interesting.

Throughput is studied for the AP without discussing the effect of multiple nodes on the wireless medium, and what that could add to the losses in quality of service. The thesis did not investigate the percentage of packets that are successfully transmitted by an AP (i.e. with a good checksum), and this is an important parameter for throughput analysis over WLANs.

8.2. Future Work Suggestions

The analyses and the logic used for modeling the delay attribute in WLAN APs can be applied in the same manner to model the delay attribute of other data communications nodes like switches, bridges, routers, etc. Hence the work in this thesis can and may be used as an example or basis for more future work on other data communications nodes. A major difference between modeling delay in WLAN APs and other communications nodes is that the WLAN AP links two entities connected via *different* link layer protocols. This difference leads to differences in headers and link access techniques. Another difference that exists for only special types of communications nodes (like routers) is the processing methodology, which includes more parameters for consideration in the case of a router. For instance, in WLAN APs, deciding a route is not very time consuming since the possibilities are simple and limited (a packet will go on the downlink or uplink or be duplicated on the WLAN side). However, in a router, the routing time is a significant parameter to consider when modeling the total delay attribute (response time of the system). In this thesis we have considered APs that utilize 10Mbps Ethernet on the wired side and IEEE 802.11b on the wireless side.

From the open issues and more reflection, I suggest some future work on the investigation of WLAN APs as listed:

- More detailed model for uplink, downlink, and bi-directional traffic taking in consideration more aspects of traffic.
- AP throughput model in relationship to nodes available on the IEEE 802.11 link.
- Stretching the work to bridges and routers.
- Investigating WLAN APs that utilize link layer combination of protocols other than the two studied in this thesis, namely 10Mbps Ethernet and IEEE 802.11b.
- Developing a simulator, which is a current topic of my research (work in progress).
- Traffic shaping can be used to allow emulation of a given access point, based on analyzing a set of measurements of a real access point.
- Running more multimedia experiments of future applications and compare them to the future 3G wireless networks from a performance point of view.
- Investigating whether a convergent-UDC AP will have a better downlink throughput than a divergent-UDC AP.
- Analyzing the uplink and downlink delays and the UDC of APs to determine the time required to serve the overhead bits on both directions (the downlink and the uplink) for a given AP.
- Test my programs against a Linux machine running packet forwarding code (such as the traffic shaping code).

- Research on using the service time analysis to estimate (calculate) the response time. For instance, a user may be interested to know what the average response time of a given AP with a given packet stream sequence is.
- Investigate the relationship between the AP brand and the percentage of packets sent with an erroneous Transport Layer Checksum, and its effects on the throughput formula of the AP, T_{APL}, presented in (24).

REFERENCES

[1] A. B. Downey, "Using patchar to estimate Internet link characteristics," ACM SIGCOMM Internet Measurement Workshop (ACM IMW'01), November 2001.

[2] A. Festag, J. Weinmiller, and A. Wolisz, "Interconnection of Wireless Cells- a Multicast-based Approach," 6th International Conference on Universal Personal Communication (ICUPC'97), October 1997.

[3] A. Kamerman and A. Aben, "Throughput performance of wireless LANs operating at 2.4 and 5 GHz," in Proceedings of the 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2000), London, pp. 190-195, September 2000.

[4] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, 3rd ed., McGraw-Hill, 1999.

[5] A. Shaikh and A. Greenberg, "Experience in Black-box OSPF Measurement," ACM SIGCOMM Internet Measurement Workshop (ACM IMW'01), November 2001.

[6] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE802.11 Wireless Local Area Networks," IEEE Communications Magazine, pp. 116-126, September 1997.

[7] C. E. Spurgeon, *Ethernet, The Definitive Guide*, O'Reilly & Associates, Inc., 2000.

[8] C. Estan and G. Varghese, "New Directions in Traffic Measurement and Accounting," ACM SIGCOMM Internet Measurement Workshop (ACM IMW'01), 2001.

[9] C. Huang, M. Devetsikiotis, I. Lambadaris, and A.R. Kaye, "Modelling and Simulation of Self-Similar Variable Bit Rate Compressed Video: A Unified Approach", in Proceedings of ACM SIGCOMM'95, Boston, pp. 114-125, 1995.

[10] C. Ware, J. Chicharo, and T. Wysocki, "Simulation of Capture Behavior in IEEE 802.11 Radio Modems," Journal of Telecommunications and Information Technology, March 2001

[11] C-S. Wu, G-K. Ma, and B-S. P. Lin, "Internet Access for Personal Mobile Equipment in a Wireless WAN Environment," IEEE Transactions on Consumer Electronics, vol. 43, no. 3, August 1997.

[12] D. Bear, "Principles of Telecommunication Traffic Engineering," IEE Telecommunications Series 2, 1980.

[13] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed., Prentice Hall, Inc., 1987.

[14] D. Loguinov and H. Radha, "Measurement Study of Low-bitrate Internet Video Streaming," ACM SIGCOMM Internet Measurement Workshop (ACM IMW'01), 2001.

[15] D. P. Heyman and T. V. Lakshman, "What Are the Implications of Long-Range Dependence for VBR-Video Traffic Engineering?", IEEE/ACM Transactions on Networking, vol. 4, no. 3, June 1996.

[16] Dell PCMCIA card, Fast Ethernet 10/100 Base-TX, by 3Com, Model 3CCFE575CT-D, *http://www.directron.com/3ccfe575ctd.html/.*

[17] Draft ITU-T Recommendation H.263 (1996): "Video Coding For Low Bit-rate Communication", 1996.

[18] E. H. Callaway, *Wireless Sensor Networks: Architectures and Protocols*, CRC Press, June, 2003.

[19] E. Pelletta, "Bridging Capacity of IEEE802.11b Access Points," Master thesis technical report, Royal Institute of Technology (KTH), Stockholm, Sweden, May 2002.

[20] Federal Standard 1037C, glossary of telecommunication terms, *http://glossary.its.bldrdoc.gov/fs-1037/dir-007/_1045.htm/.*

[21] Free Online Dictionary of Computing, http://wombat.d*oc.ic.ac.uk/foldoc/index.html/.*

[22] G. Bianchi , L. Fratta , and M. Oliveri , "Performance Evaluation and Enhancement of the CSMA/CA MAC Protocol for 802.11 Wireless LANs, " in Proceedings of the 7th International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC 1996), pp. 392-396, Taipei, Taiwan, October 1996.

[23] G. Gordon, *System simulation*, 2nd ed., Prentice-Hall, 1978.

[24] G. Iannaccone, M. May, and C. Diot, "Aggregate Traffic Performance with Active Queue Management and Drop from Tail," ACM SIGCOMM, Computer Communications Review, pp.4-13., vol. 31, no. 3, July 2001.

[25] H. Chhaya and S. Gupta, "Throughput and fairness properties of asynchronous data transfer methods in the IEEE 802.11 MAC protocol," in Proceedings of the 6th International Symposium on Personal, Indoor, and Mobile Radio Communications, (PIMRC 1995), vol.2, pp. 613-617, 1995.

[26] H-H. Liu and J-L. C. Wu, "A Scheme for supporting Voice over IEEE 802.11 Wireless Local Area Network," in Proceedings of the Natural Science Council ROC(A), vol. 25, no. 4, pp. 259-268, 2001.

[27] **I. Al Khatib** and R. Ayani, "Wireless LAN Access Points: Buffer Size Estimation," in Proceedings of the 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN'02), pp. 656-660, Stockholm, Sweden, 2002.

[28] **I. Al Khatib**, "Wireless LAN Access Points as Links with Adaptive Bandwidth: Throughput and Feedback Control," in Proceedings of the 10th International Conference on Telecommunications (ICT'2003), vol. 1, pp. 754-760, French Polynesia, February 23-March 1, 2003.

[29] **I. Al Khatib**, A. Franzen, and F. Moioli, "Minimum GPRS Bandwidth for Acceptable H.261 Video QoS", in Proceedings of IFIP Personal Wireless Communications (PWC'2000), pp. 147-156, Gdansk, Poland, September 14-15, 2000.

[30] **I. Al Khatib**, G. Q. Maguire Jr., R. Ayani, and D. Forsgren, "Wireless LAN Access Point Modeling as a Queuing System", in Proceedings of the Communications and Computer Networks 2002 Conference (CCN 2002), pp. 463-468, MIT, Cambridge, USA, November 4-6, 2002.

[31] **I. Al Khatib**, G. Q. Maguire Jr., R. Ayani, and D. Forsgren, "Wireless LAN Access Points as Queuing Systems: Performance Analysis and Service Time," to appear in ACM SigMobile Mobile Computing and Communications Review (MC^{2}R), last issue 2002.

[32] **I. Al Khatib**, G. Q. Maguire Jr., R. Ayani, and D. Forsgren, "Wireless LAN Access Points as Queuing Systems: Performance Analysis and Service Time," The Eighth ACM International Conference on Mobile Computing and Networking (ACM MOBICOM 2002 Conference, *sponsored by ACM SIGMOBILE*), extended abstract, Atlanta, Georgia, USA, September 23-28, 2002.

[33] **I. Al Khatib**, R. Ayani, and G. Q. Maguire Jr., "Wireless LAN Access Points Uplink and Downlink Delays: Packet Service-Time Comparison," in Proceedings of The 16th Nordic Teletraffic Seminar (NTS-16) Conference, pp. 253-264, Espoo, Finland, August 21-23, 2002.

[34] I. Kaj, "Stochastic Modeling in Broadband communications Systems," Lecture Notes, Department of Mathematics, Uppsala University, Sweden, 2001.

[35] IEEE802.11 standard, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1997.

[36] IEEE802.11a standard, (Supplement to IEEE Std. 802.11, 1999 Edition), Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 5GHz Band, 1999.

[37] IEEE802.11b standard, (Supplement to ANSI/IEEE Std. 802.11, 1999 Edition), Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4GHz Band, 1999.

[38] IEEE802.11d standard, (Amendment to IEEE Std. 802.11, 1999 Edition, IEEE Std. 802.11a-1999, IEEE Std. 802.11b-1999), Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Amendment 3: Specification for Operation in Additional Regulatory Domains, 2001.

[39] J. A. Stine, "Tactical Communications Using the IEEE 802.11 MAC Protocol," The IEEE Military Communications Conference (MILCOM'98), 1998.

[40] J. Banks, S. Carson, B. L. Nelson, and D. M. Nicol, *Discrete-Event System Simulation*, 3rd ed., Prentice-Hall, 2001.

[41] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-Range Dependence in Variable-Bit-Rate Video Traffic", IEEE Transactions on Communications, vol. 43, no. 2/3/4, pp. 1566-1579, 1995.

[42] J. Liu and M. Crovella, "Using Loss Pairs to Discover Network Properties," ACM SIGCOMM Internet Measurement Workshop (IMW'01), 2001.

[43] J. Zyren and A. Petrick, "Brief Tutorial on IEEE 802.11 Wireless LANs," PRISM, Intersil, Application Note, AN9829, February 1999.

[44] J-C. Bolot, "End-to-End Delay and Loss Behavior in the Internet," ACM SIGCOMM'93 Conference, September 1993.

[45] J-L. C. Wu, H-H. Liu, and Y-J. Lung, "An Adaptive Multirate IEEE 802.11 Wireless LAN," The 15th International Conference on Information Networking (ICOIN'01), Beppu City, Oita, Japan, 2001.

[46] J-Y. Le Boudec, "Some Properties of Variable Length Packet Shapers," EPFL-DSC Research Report DSC/2000/037/, 2000.

[47] K. Lai and M. Baker, "Measuring Link Bandwidths Using Deterministic Model of Packet Delay," ACM SIGCOMM, Computer Communication Review, vol. 30, no. 4, October 2000.

[48] K. Ogata, *Modern Control Engineering, 2nd Edition*, Prentice-Hall International, Inc., 1990.

[49] K. Pahlavan, T. H. Robert, and M. E. Chase, "Trends in Local Wireless Networks," IEEE Communications Magazine, March 1995.

[50] K. Park, G. Kim, and M. Crovella, "On the Effect of Traffic Self-similarity on Network Performance", the SPIE International Conference on Performance and Control of Network Systems, November 1997.

[51] K. R. Krishnan, A. L. Neidhardt, and A. Erramilli, "Scaling Analysis in Traffic Management of Self similar Processes", ITC 15, Elsevier Science B.V., 1997.

[52] L. Kleinrock, *Queuing Systems, Volume 1: Theory*, A Wiley-Interscience Publication, John Wiley and Sons, 1975.

[53] Linux-2.2.16 OS Release Notes, *http://www.linux.org.uk/VERSION/relnotes.2216.html/.*

[54] M. Stemm and R. H. Katz, "Vertical Handoffs in Wireless Overlay Networks," ACM Mobile Networks and Applications (MANET'01), August 2001.

[55] M. W. Garrett, W. Willinger, "Analysis, Modeling and Generation of Self-Similar VBR Video Traffic", Computer Communications Review, vol. 24, no. 4, pp. 269-280, January 1994.

[56] M. Ylianttila, M. Pande, J. Mäkelä, and P. Mähönen, "Optimization Scheme for Mobile Users Performing Vertical Handoffs between 802.11 and GPRS/EDGE networks," in Proceedings of GLOBECOM'01, November 2001.

[57] Mark Allman, "Measuring End-to-End Bulk Transfer Capacity," ACM SIGCOMM Internet Measurement Workshop (ACM IMW'01), November 2001.

[58] MGEN 3.2 tool set, *http://manimac.itd.nrl.navy.mil/MGEN/*, November 2000.

[59] MINC, Multicast-based Inference of Network-internal Characteristics, *http://www.research.att.com/projects/minc/*.

[60] N. C. Hock, *Queuing Modeling Fundamentals*, John Wiley and Son, 1998.

[61] ORINOCO Wireless for Lucent Access Points, *http://www.orinocowireless.com/*, January 2002.

[62] P. Ferguson and G. Huston , "Quality of Service in the Internet: Fact, Fiction, or Compromise?" INET'98 Conference, Geneva, Switzerland, July 1998.

[63] P. Ferguson and G. Huston, *Quality of Service, Delivering QoS on the Internet and in Corporate Networks*, Wiley Computer Publishing, John Wiley and Sons, Inc., 1998.

[64] P. Pruthi, D Ilie, and A. Popescu "Application Level Performance of Multimedia Services", SPIE International Symposium on Voice, Video, and Data Communications Boston, September 1999.

[65] PCC (Personal Computing and Communication) project, a Swedish research program financed by the Foundation for Strategic Research, *http://www.pcc.lth.se/.*

[66] R. L. Carter and M. E. Crovella, "Measuring Bottleneck Link Speed in Packet-Switched Networks," Technical report TR-96006, Department of Computer Science, Boston University, Online at: http://www.cs.bu.edu/faculty/crovella/papers.html, March 1996..

[67] R. O. Baldwin, "Improving the Real-Time Performance of a Wireless Local Area Network," Doctoral Dissertation, Faculty of the Virginia Polytechnic Institute and state University, June 23, 1999.

[68] R. Perlman, *Interconnections*, 2nd ed., Addison Wesley Longman, Inc., 2000.

[69] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers*, McGraw-Hill, 1998.

[70] S. Hyakin, *Communication Systems*, 2nd ed., John Wiley & Sons, 1983.

[71] S. J. Vaughan-Nichols, "Bull Market for IEEE 802.11 WLAN Chipsets," Computer Magazine, pp.17-19, November 2002.

[72] S. Keshav, "A Control Theoretic Approach to Flow Control," in ACM SIGCOMM Computer Communications Review, vol. 21, issue 4, pp.3-15, September 1991.

[73] S. Keshav, *An Engineering Approach to Computer Networking, ATM Networks, The Internet, and The Telephone Network,* AT&T, 1997.

[74] System Classifications and Properties, *http://cnx.rice.edu/modules/m10084/latest/*, January 2003.

[75] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, The Massachusetts Institute of Technology Press, 1990.

[76] T. L. Rodeheffer, C. A. Thekkath, and D. C. Anderson, "SmartBridge: A Scalable Bridge Architecture," ACM SIGCOMM 2000, Computer Communication Review, vol. 30, no. 4, October 2000.

[77] T. Neslson, "Wireless LAN Research Study, Made Available by the Wireless Etherent Compatibility Alliance, Wi-Fi™," Market Research Sponsored by Microsoft, October 2001. Online at:

http://www.weca.net/OpenSection/pdf/Wireless_LANResearch_ExecutiveSummary.pdf.

[78] T. Turletti, "H.261 Software Codec for Videoconferencing over the Internet", Research Report (Rapports de Recherche), Research Unit (Unité de recherche) no. 1834, INRIA-SOHIA Antipolis, 1993.

[79] T. Yoshimura, M. R. Jeong, H. Morikawa, and T. Aoyama, "Wireless Packet Scheduling for Adaptive Service over IEEE802.11," The First International Symposium on Wireless Personal Multimedia Communications (WMPC'98), pp. 314-319, Kanagawa, Japan, November 1998.

[80] Terms to Identify Systems, *http://pbl.cc.gatech.edu/bmed8101/97/.*

[81] The *tcpdump* network traffic recording program, *http://www.tcpdump.org/*.

[82] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's," ACM SIGCOMM'94, 1994.

[83] V. Hayes, "Tutorial on 802.11 to 802," Report, IEEE P802.11-96/49A.

[84] W. Badawy, M. Weeks, G. Zhang, M. Talley, and M. A. Bayoumi, "MRI Data Compression Using a 3-D Discrete Wavelet Transform, A low-Power System that Can Be Used to Compress MRI Data and for Other Medical Applications," IEEE Engineering in Medicine and Biology Magazine, vol. 21, no. 4, pp. 95-103, July/August 2002.

[85] W. Diepstraten, P. Belanger, "802.11 Tutorial: MAC Entity, MAC Basic Access Mechanism, Privacy and Access Control", Report, IEEE P802.11-96/49C, March 1996.

[86] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the Self-Similar Nature of Ethernet Traffic", IEEE/ACM Transactions on Networking, vol. 2, no. 1, February 1994.

[87] W. R. Stevens, *TCP/IP Illustrated, The protocols*, vol. 1, Addison-Wesley, 1998.

[88] W. R. Stevens, *TCP/IP Illustrated, Volume 3, TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols*, Addison Wesley Longman, Inc., 1996.

[89] W. Stallings, "Data and Computer Communications," 5th ed., Prentice-Hall, Inc., 1997.

[90] Webster Dictionary (Merriam-Webster Online), *http://www.webster.com/.*

[91] Wi-Fi Alliance, *http://www.wi-fi.org/.*

[92] Y. Joo, V. Ribeiro, A. Feldmann, A. C. Gilbert, and W. Willinger, "TCP/IP Traffic Dynamics and Network Performance: A Lesson in Workload Modeling, Flow control, and Trace-driven Simulations," ACM SIGCOMM, Computer Communications Review, pp.25-37, vol. 31, no. 2, April 2001.

Appendices

Appendix 1

Mathematical induction proof of average service time formula

We prove (3) as follows:

For n = 1, substitute in (3), then

$$S_1 = S_o + 0r = S_o \qquad (\varepsilon.1)$$

where S_o is the original Service time (for packet with 40B IP payload).

Equation (ε.1) means that the relation expressed in (3) holds true for $n = 1$ since S_o is the first term of the sequence (we start with 40B payload as first sequence element).

Knowing that (3) holds true for $n = 1$, let us assume that (3) holds true for a general sequence element *(k-1)* and try to generalize- by proof- that (3) holds for the sequence element *k*.

So, we assume that (3) holds true for $n = k - 1$, and we substitute the value of *n* in (ε.1) to get:

$$S_{k-1} = S_o + [(k-1)-1]r = S_o + (k-2)r. \qquad (\varepsilon.2)$$

In fact, the difference between two consecutive terms in the average service time sequence is estimated- by linear regression- to be the constant *r*. As (2) shows:

$$S_k - S_{k-1} = r \Leftrightarrow S_{k-1} = S_k - r. \qquad (\varepsilon.3)$$

Substitute the expression of S_{k-1} from (ε.2) in (ε.3), then

$$S_o + (k-2)r = S_k - r$$
$$\Leftrightarrow S_o + kr - 2r = S_k - r$$
$$\Leftrightarrow S_k = S_o + kr - 2r + r$$
$$\Leftrightarrow S_k = S_o + kr - r$$
$$\Leftrightarrow S_k = S_o + (k-1)r,$$

which shows that (3) holds true for the general sequence number $n = k$. Hence, (3) is correct by *mathematical induction.*

Knowing that *n* is related to payload, we can simply substitute the value of *n* in (3) to have an average service time model in terms of IP payload (P):

$$S_P = S_o + \left[\left(\frac{P-8}{32}\right) - 1\right] r$$

$$\Leftrightarrow S_P = \frac{r}{32} P + S_o - \frac{5}{4} r. \qquad (\varepsilon.4)$$

Equation (ε.4) holds for *P* that is a function of 32B as shown in

Table 2. Other cases will have an error larger than the error estimated for the service time model, however, due to linearity and due to the fact that linear regression was a suitable technique, the errors of such estimation are relatively large.

Appendix 2

Derivation of the packet pair property for FIFO queuing networks

The packet pair queuing model [47] uses a two-packet FIFO model (Figure A.2), which predicts the difference in arrival time values of two packets of the same size travelling from the same source to the same destination. I will state the packet pair property and then discuss the proof.

Packet Pair Property. *Let $b_{min(l)} \leq b_i$ ($\forall i$, $0 \leq i \leq 1$), then if we send two packets of the same size ($s_0 = s_1$) back to back ($t_0^0 \approx t_0^1$; t_0^1 is actually slightly larger than t_0^0), they will arrive with a difference in time equal to the size of the second packet divided by the smallest bandwidth on the path ($t_d^1 - t_d^0 = \dfrac{s_1}{b_{\min(d-1)}}$).*

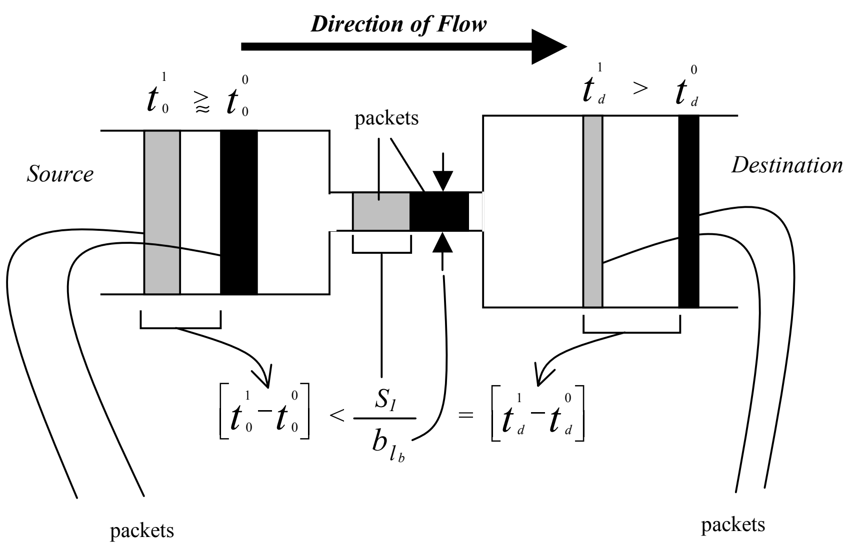

Figure A.2. Packet Pair Model for FIFO queuing networks.

The packet pair property, depends on the *multi-packet delay equations* [47]. The multi-packet model consists of a delay equation derived from two other equations: an arrival time equation and a queuing delay equation. The arrival time equations is:

$$t_l^k = t_0^k + \sum_{i=0}^{l-1}\left(\frac{s_k}{b_i} + d_i + q_i^k\right). \tag{ξ.1}$$

Equation (ξ.1) predicts that packet *k* arrives at link *l* at its transmission time (t_0^k) plus the sum over all the previous links, whose latencies are (d_i), transmission delays are $\left(\frac{s_k}{b_i}\right)$, and waiting time (in queue delays) are (q_i^k). The queuing delay due to other packets in the same flow is modeled as:

$$q_l^k = \max\left(0, t_{l+1}^{k-1} - d_l - t_l^k\right). \tag{ξ.2}$$

Equation (ξ.2) predicts that packet *k* is queued at the router just before link *l* from the time it arrives at that communication node (t_l^k) until it can begin transmitting. This is the time when the previous packet (*k*-1) arrives at the next communication node (t_{l+1}^{k-1}) minus the latency of link (d_l). An important assumption is that the first packet is never queued, i.e. $q_0^0 = q_1^0 = ... = q_{d-1}^0 = 0$. Moreover, queuing delay can not be negative, so the $\max(.)$ function makes the queuing equation zero. Combining (ξ.1) and (ξ.2), the multi-packet delay equation is:

$$t_l^k = t_0^k + \sum_{i=0}^{l-1}\left(\frac{s_k}{b_i} + d_i + \max\left(0, t_{i+1}^{k-1} - d_i - t_i^k\right)\right). \tag{ξ.3}$$

The packet pair property can then be derived from the multi-packet model equation, (ξ.3). To make the derivation clearer, two lemmas are defined: A.1 and A.2.

LEMMA A.1. Let $s_0 = s_1$, $t_0^0 = t_0^1$, $0 \le j \le n$. *Then* $t_j^1 - t_j^0 = \sum_{i=0}^{j-1} \max\left(0, t_{i+1}^0 - d_i - t_i^1\right)$.

Proof. From (ξ.3), we get:

$$t_j^1 - t_j^0 = \sum_{i=0}^{j-1}\left(\frac{s_1}{b_i} + d_i + \max\left(0, t_{i+1}^0 - d_i - t_i^1\right)\right) + t_0^1 - \left(\sum_{i=0}^{j-1}\left(\frac{s_0}{b_i} + d_i\right) + t_0^0\right).$$

Hence, $t_j^1 - t_j^0 = \sum_{i=0}^{j-1} \max\left(0, t_{i+1}^0 - d_i - t_i^1\right)$.

LEMMA A.2. Let $s_0 = s_1$, $t_0^0 = t_0^1$, $0 \le j \le n$. *Then*

$$t_{j+1}^0 - d_i - t_j^1 = \frac{s_0}{b_j} - \sum_{i=0}^{j-1} \max\left(0, t_{i+1}^0 - d_i - t_i^1\right).$$

Proof. From (ξ.3), we get:

$$\begin{aligned}
t_{j+1}^0 - d_j - t_j^1 &= \\
&= \sum_{i=0}^{j}\left(\frac{s_0}{b_i} + d_i\right) + t_0^0 - d_j - \left(\sum_{i=0}^{j-1}\left(\frac{s_1}{b_i} + d_i + \max\left(0, t_{i+1}^0 - d_i - t_i^1\right)\right) + t_0^1\right) \\
&= \sum_{i=0}^{j-1}\left(\frac{s_0}{b_i} + d_i\right) + \frac{s_0}{b_j} + d_j - d_j - \left(\sum_{i=0}^{j-1}\left(\frac{s_1}{b_i} + d_i\right) + \sum_{i=0}^{j-1} \max\left(0, t_{i+1}^0 - d_i - t_i^1\right)\right)
\end{aligned}$$

$$= \frac{s_0}{b_j} - \sum_{i=0}^{j-1} \max\left(0, t_{i+1}^0 - d_i - t_i^1\right).$$

Proof of the Packet Pair Property. Perform induction on d, which is the number of links. For $d = 1$, substitute in (ξ.3), then

$$\begin{aligned} t_d^1 - t_d^0 &= t_1^1 - t_1^0 \\ &= \sum_{i=0}^{0}\left(\frac{s_1}{b_i} + d_i + \max\left(0, t_{i+1}^0 - d_i - t_i^1\right)\right) + t_0^1 - \left(\sum_{i=0}^{0}\left(\frac{s_0}{b_i} + d_i\right) + t_0^0\right). \quad (\Psi.1) \end{aligned}$$

Use (ξ.3) again to simplify *(Ψ.1)*:

$$\begin{aligned} t_d^1 - t_d^0 &= \frac{s_1}{b_0} + d_0 + \max\left(0, t_1^0 - d_0 - t_0^1\right) + t_0^1 - \left(\frac{s_0}{b_0} + d_0 + t_0^0\right) \\ &= \max\left(0, t_1^0 - d_0 - t_0^1\right) \\ &= \max\left(0, \sum_{i}^{0}\left(\frac{s_0}{b_i} + d_i\right) + t_0^0 - d_0 - t_0^1\right) \\ &= \max\left(0, \frac{s_1}{b_0} + d_0 - d_0\right) \\ &= \frac{s_1}{b_0} \\ &= \frac{s_1}{b_{\min(d-1)}}. \end{aligned} \quad (\Psi.2)$$

Then, the case of $d = 1$ holds true. Now, by using Lemma A.1, then:

$$t_d^1 - t_d^0 = \sum_{i=0}^{j-1} \max\left(0, t_{i+1}^0 - d_i - t_i^1\right). \quad (\Psi.3)$$

Simplifying further and applying mathematical induction, then

$$t_d^1 - t_d^0 = \sum_{i=0}^{j-2} \max\left(0, t_{i+1}^0 - d_i - t_i^1\right) + \max\left(0, t_d^0 - d_{d-1} - t_{d-1}^1\right)$$

$$= \frac{s_0}{b_{\min(d-2)}} + \max\left(0, t_d^0 - d_{d-1} - t_{d-1}^1\right). \qquad (\Psi.4)$$

Applying Lemma A.2, then (Ψ.4) becomes:

$$t_d^1 - t_d^0 = \frac{s_0}{b_{\min(d-2)}} + \max\left(0, \frac{s_0}{b_{d-1}} - \sum_{i=0}^{d-2} \max\left(0, t_{i+1}^0 - d_i - t_i^1\right)\right). \qquad (\Psi.4)$$

Applying Lemma A.1 to (Ψ.4), then

$$t_d^1 - t_d^0 = \frac{s_0}{b_{\min(d-2)}} + \max\left(0, \frac{s_0}{b_{d-1}} - t_{d-1}^1 - t_{d-1}^0\right). \qquad (\Psi.5)$$

Using induction again and simplifying (Ψ.5), then

$$t_d^1 - t_d^0 = \frac{s_0}{b_{\min(d-2)}} + \max\left(0, \frac{s_0}{b_{d-1}} - \frac{s_0}{b_{\min(d-2)}}\right)$$

$$= \max\left(\frac{s_0}{b_{\min(d-2)}}, \frac{s_0}{b_{d-1}}\right). \qquad (\Psi.6)$$

Hence, there are two cases:

Case 1.

$$\frac{s_0}{b_{\min(d-2)}} \geq \frac{s_0}{b_{d-1}} \quad \Leftrightarrow \quad b_{d-1} \geq b_{\min(d-2)}. \tag{Ψ.7}$$

Knowing that $s_0 = s_1$, then from (Ψ.7) we get:

$$t_d^1 - t_d^0 = \frac{s_0}{b_{\min(d-2)}} = \frac{s_1}{b_{\min(d-1)}}. \tag{Ψ.8}$$

Case 2.

$$\frac{s_0}{b_{\min(d-2)}} < \frac{s_0}{b_{d-1}} \quad \Leftrightarrow \quad b_{d-1} < b_{\min(d-2)}. \tag{Ψ.9}$$

Knowing that $s_0 = s_1$, then from (Ψ.9) we get:

$$t_d^1 - t_d^0 = \frac{s_0}{b_{d-1}} = \frac{s_1}{b_{\min(d-1)}}. \tag{Ψ.10}$$

Appendix 3

An example of SSTP output data file

"Computers are useless. They can only give you answers."

Pablo Picasso

(Part of real data file for downlink flow of 40B IP payload packets utilizing 2Mbps)

The first column presents the packet number (identification). The second column presents the time or arrival at the AP system (in µs). The third column presents the departure time (in µs) from the AP system. The Fourth column is the Response time in µs. The fifth column is the waiting time (in µs), and the sixth column is the service time (in µs). The value '-1' means the packet was lost.

:	:	:	:	:	:
50739	1026345583541228	1026345583621838	80610	79484	1126
50740	1026345583541694	1026345583623088	81394	80144	1250
50741	1026345583542162	1026345583624525	82363	80926	1437
50742	1026345583542621	1026345583625707	83086	81904	1182
50743	1026345583543083	1026345583627173	84090	82624	1466
50744	1026345583543549	1026345583628395	84846	83624	1222
50745	1026345583544013	1026345583629927	85914	84382	1532
50746	1026345583544475	1026345583631119	86644	85452	1192
50747	1026345583544938	1026345583632305	87367	86181	1186
50748	1026345583545407	1026345583633757	88350	86898	1452
50749	1026345583545864	1026345583634982	89118	87893	1225
50750	1026345583546330	1026345583636162	89832	88652	1180
50751	1026345583546794	1026345583637721	90927	89368	1559
50752	1026345583547256	1026345583639246	91990	90465	1525
50753	1026345583547718	1026345583640688	92970	91528	1442
50754	1026345583548184	1026345583642150	93966	92504	1462
50755	1026345583548650	1026345583643296	94646	93500	1146
50756	1026345583549111	1026345583644786	95675	94185	1490
50757	1026345583549572	1026345583646014	96442	95214	1228
50758	1026345583550042	1026345583647198	97156	95972	1184
50759	1026345583550499	-1	-1	-1	-1
50760	1026345583550964	-1	-1	-1	-1
50761	1026345583551426	1026345583648696	97270	95772	1498
50762	1026345583551892	1026345583649825	97933	96804	1129
50763	1026345583552353	-1	-1	-1	-1
50764	1026345583552826	-1	-1	-1	-1
50765	1026345583553288	1026345583651167	97879	96537	1342
50766	1026345583553746	-1	-1	-1	-1
50767	1026345583554208	-1	-1	-1	-1
50768	1026345583554680	1026345583652406	97726	96487	1239
50769	1026345583555136	-1	-1	-1	-1
50770	1026345583555602	-1	-1	-1	-1
50771	1026345583556071	1026345583653610	97539	96335	1204
50772	1026345583556528	-1	-1	-1	-1
50773	1026345583556991	-1	-1	-1	-1
50774	1026345583557461	1026345583654802	97341	96149	1192
:	:	:	:	:	:

Appendix 4

Definition of the word: *system*

From the Webster dictionary [90].

Permission to use definition in this thesis assured on April 3, 2003, by: Anne Golob <*agolob@merriam-webster.com*>
Permissions Editor at Merriam Webster Inc.

Function: *noun.*

Etymology: Late Latin *systemat-, systema,* from Greek *systEmat-, systEma,* from *synistanai* to combine, from *syn-* + *histanai* to cause to stand -- more at STAND Date: 1603.

Meaning:
1 : a regularly interacting or interdependent group of items forming a unified whole <a number *system*>: as **a** (1) **:** a group of interacting bodies under the influence of related forces <a gravitational *system*> (2) **:** an assemblage of substances that is in or tends to equilibrium <a thermodynamic *system*> **b** (1) **:** a group of body organs that together perform one or more vital functions <the digestive *system*> (2) **:** the body considered as a functional unit **c :** a group of related natural objects or forces <a river *system*> **d :** a group of devices or artificial objects or an organization forming a network especially for distributing something or serving a common purpose <a telephone *system*> <a heating *system*> <a highway *system*> <a data processing *system*> **e :** a major division of rocks usually larger than a series and including all formed during a period or era **f :** a form of social, economic, or political organization or practice <the capitalist *system*>

2 : an organized set of doctrines, ideas, or principles usually intended to explain the arrangement or working of a systematic whole <the Newtonian *system* of mechanics>

3 a : an organized or established procedure <the touch *system* of typing> **b :** a manner of classifying, symbolizing, or schematizing <a taxonomic *system*> <the decimal *system*>

4 : harmonious arrangement or pattern **:** ORDER <bring *system* out of confusion -- Ellen Glasgow>

5 : an organized society or social situation regarded as stultifying **:** ESTABLISHMENT

Synonym: METHOD

Enclosed Publications

By Iyad Al Khatib

Publication 1

Iyad Al Khatib

Wireless LAN Access Points as Links with Adaptive Bandwidth: Throughput and Feedback Control

In Proceedings of the 10th International Conference on Telecommunications (ICT'2003), vol. 1, pp. 754-760, French Polynesia, February 23-March 1, 2003

IEEE Catalog Number: 03EX628
ISBN: 0-7803-7661-7
Library of Congress: 2002113141

Also available in IEEE Catalog Number: 03EX628C
ISBN: 0-7803-7662-5 (CD-ROM)
Paper reference: P754

Wireless LAN Access Points as Links with Adaptive Bandwidth: Throughput and Feedback Control

Iyad Al Khatib[⊥]
ikhatib@ieee.org

***Abstract*—The throughput of a Wireless LAN access point is an important parameter for QoS over wireless LANs. This paper builds on previous results in order to model wireless LAN access points as data communication links and parameterize the throughput of the access point. The previous work models wireless LAN access points of IEEE 802.11b as FIFS queuing systems, whose service time is a function of payload. This paper shows that a wireless LAN access point functions as a data communication link with variable bandwidth. It is possible to parameterize the performance of the access point as if it were a link with *adaptive-bandwidth*, bounded by two limits: the lower and upper *adaptive-bandwidth* bounds. Performance of different access point brands can be compared using the adaptive-bandwidth characteristic. Adaptive-bandwidth was analyzed and shown to be a strictly increasing linear fractional transformation of payload. The adaptive-bandwidth of the access-point link-model is used to calculate the throughput of the access point. The throughput of access points is studied in terms of a feedback control system, whose ratio of output to input gives the throughput in terms of access point parameters and offered load.**

Keywords: WLAN, access point, link model, bandwidth, throughput.

I. INTRODUCTION

Wireless LAN (WLAN) deployment is increasing as revealed by the market success of IEEE 802.11b products. Users are interested in network connectivity without giving up physical mobility [1]. This increase is directly coupled to an increasing number of WLAN access points, which connect WLAN users to the Internet backbone. Hence, today there is an increasing probability of having an access point (AP) as one of the nodes on an Internet path. While many of these APs are connected to ADSL, cable modems, etc, an increasing number are connected via high speed network connections, which are capable of handling the full bandwidth of an 802.11b AP; thus the bottleneck moves to the WLAN AP itself.

Inspired by the power of packet-pair analysis in FIFO-queuing networks [2][3], this paper builds on the previous results presented in [4] and [5]. By modeling WLAN APs as FIFS queuing systems, this paper shows that an AP functions as a communication link with variable bandwidth. The performance of the AP is parameterized as if it were a link with *adaptive-bandwidth*, bounded by two limits, the lower *adaptive-bandwidth* bound and the upper *adaptive-bandwidth* bound.

The adaptive-bandwidth of a WLAN AP is related to the IP payload in a packet. Analysis shows that the adaptive bandwidth is a strictly increasing linear fractional function of packet size. Thus, the presence of an AP along a path will result in a bottleneck link bandwidth that exhibits varying link speed over a short interval of time (depending on the test packets themselves and not necessarily due to other packets). Therefore, many end-to-end path bottleneck experiments will fail to provide applications with correct information about the condition of the path. This behavior of the AP leads to false service level expectations by applications, with especially negative effects on multimedia applications. The AP link-model helps further understand the performance of WLAN APs, especially when looked upon as separate nodes on a path. In this respect, APs will look and behave like links with varying speeds dependent on packet size. This performance does not mean that the transmission of the wireless link will change, but rather shows that while the packets are inside the AP node, they can be thought of and considered as passing through a communication link, whose speed of transmission is dependent on packet size. This adaptive-bandwidth of the AP-link-model is used to calculate the throughput of the access point by utilizing the definition of link bandwidth of a data-communication link. An interesting model of the throughput of a WLAN AP is presented as a feedback control system. Below we discuss the different logical and analytic steps required to finally present the throughput of WLAN APs through a feedback control model.

II. PREVIOUS WORK

This section describes the previous results upon which the AP-link-model builds. In the previous work with the co-authors of [4] and [5], we modeled the delay processing in the WLAN AP. To complete the theoretical model, a set of assumptions are made, where we look at the AP as a system of reference and define the different events that occur. We define two events: *arrival* and *departure*. When

[⊥] The author is a member of the IASTED (International Association of Science and Technology for Development), and he is currently a researcher and a doctoral student at the Department of Microelectronics and Information Technology, Royal Institute of Technology, Stockholm, Sweden. This work builds on the author's previous results of research that was funded by the Royal Institute of Technology. However, the model, design, investigation, and results presented in this paper are the explicit work of the author himself, as a research engineer: member of the Order of Lebanese Engineers and Architects (ID#: 20351), and member of the Swedish Union of Engineers (CIF).

ICT'03. 10th International Conference on Telecommunications 2003
0-7803-7661-7/03/$17.00©2003 IEEE.

a packet enters the AP or departs from the AP, the parameters of interest from the arrival and departure events are the time of arrival and time of departure, respectively. We define the total delay of a packet, which we refer to as the *response time*, to be the time difference between the departure time and the arrival time of a packet.

The logical packet (the entity that the system works on and hence delays) is the link layer frame, which carries an IP packet within its frame body. Figure 1 shows the MAC frame for IEEE 802.11b [6][7].

Time is an important factor in the study of our AP system; therefore, it is crucial to state whether the system is discrete or continuous. Since the number of packets inside the system changes when a packet arrives or when a packet departs, i.e. at separate points in time, then the system is a discrete-event system [8].

2B	2B	6B	6B	6B	2B	6B	0-2312B	4B
Frame Control	Duration/ID	Address 1	Address2	Address3	Sequence Control	Address4	Frame Body	FCS

MAC Header

Figure 1. MAC frame format for IEEE 802.11 [6].

Since the wireless access point can forward traffic in two directions, then we consider two cases: one where the traffic travels from the Ethernet side to the WLAN side, and another from WLAN to Ethernet. Modeling the WLAN to WLAN case is also of interest, but not presented in this paper. Therefore, we define two traffic flows: downlink traffic from Ethernet to WLAN and uplink traffic from WLAN to Ethernet. The AP system considers any packet entering the AP, whether coming from the Ethernet side or the WLAN side as an arriving packet. Similarly, any packet leaving the AP, whether it leaves to the Ethernet medium or to the WLAN medium, is considered a departing packet. Hence, the logical model of the system considers arrivals as packets entering the AP and departures as packets leaving the AP, regardless of direction of flow.

A. Queuing Model

After having defined the AP system and the events acting on the AP, we found that the processing could be modeled as a queuing system with a single server and a single FIFS queue [9].

It is worth mentioning that the service time in our model includes the time to check the frame headers, AP management time (spanning tree protocol and code management), and the transmission time of the whole frame (including interframe spaces and preambles).

B. Service Time Formula

Our experiments have shown that for a given AP, the average downlink service-time is larger than the average uplink service-time for packets with identical IP payload [4][5].

A key result of our previous work is an analytic solution for the service time of a packet in relation to payload. The average service time was analyzed and found to be a strictly increasing linear function of payload [5].

Since the system is a discrete event system, we looked at the average service time values as terms of a sequence. Hence, starting with a packet carrying 32 bytes of UDP payload and using the 32-byte payload-increment in our previous experiments, we were able, in [5], to parameterize the average service time, S_n, as:

$$S_n = S_o + (n-1)r \qquad (1)$$

where,

n = $(IP\_Payload_{[\text{in bytes}]} - 8B_{[\text{UDP header}]})$ /32B;

S_n = service time (μs) for packet with IP payload of ($32n$+8) bytes;

S_o = service time (μs) for packet with 40B IP payload;

r = incremental difference in μs (calculated from linear regression of average service times of different payloads).

The maximum IP payload used is 1480 bytes since values beyond the MTU may result in fragmentation [10]. S_o is calculated through numerical methods of averaging the service times calculated in the different experiments of 40 byte IP payloads, and r is directly proportional to the slope of the average service time curve. The curve is obtained through linear regression of the different average service-times [11]. The AP could be characterized by two service time formulae: one for the downlink and one for the uplink. Both, uplink and downlink formulae for the same AP are of the form presented in (1), however, the difference between the two formulae is in the values of S_o and r on the downlink and those of S_o and r on the uplink. Moreover, S_o and r, which we consider as the characteristic parameters of the AP, are different for different APs

The average service time formula provides a basis for the modeling and analysis of the AP as a data communication link.

III. LINK MODEL

The packet-pair property of FIFO-queuing networks can predict the difference in arrival times of two packets of the same size, sent from the same sender and received at the same destination. This property makes many assumptions that do not always hold true in data networks [3]. However, in our test design and experiments on the access point that were described in [5], the assumptions of the packet-pair property hold true. For this reason, we find in this property a very important basis for deeper analysis of the AP. Hence, we were inspired to look at the WLAN AP as a *data communication link,* whose bandwidth is b bits per second (figure 2). To complete the model, some assumptions are made. We consider the AP link model to be connected to two links: a 10Mbps Ethernet on one side and an 11Mbps

IEEE 802.11b link on the other side (figure 2). Moreover, data can flow from the Ethernet to WLAN side (downlink) or from the WLAN to Ethernet side (uplink). We also assume that the speed of the link model (*b* bits per second) could be smaller than, larger than, or equal to the link speeds of Ethernet or IEEE 802.11b. We also consider the link to be a one-to-one link with no collisions, because inside the AP the packets are queued without having to worry about a shared transmission medium. Consequently we consider that the link model of the AP has no preambles, no Interframe Spaces, and no link headers or trailers (we leave the link header and trailer time to be taken care of by the Ethernet and WLAN media). Therefore, link frames of the AP model are the IP packets themselves, and these packets can queue back to back in the link without interframe spacing. Moreover, we assume that the propagation delay in the three links of Ethernet, IEEE 802.11b, and the *b* link of the AP to be negligible compared to transmission delay since the distances covered are relatively too small to be considered in the calculations.

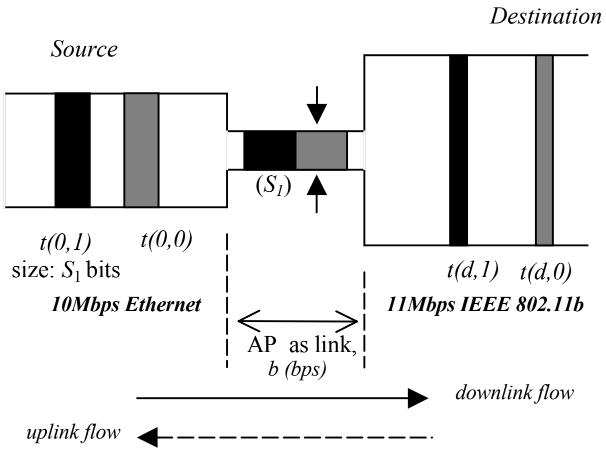

Figure 2. WLAN AP as link with bandwidth, *b (bps)*. The bandwidth, *b*, represents the bandwidth of the link model for the downlink flow case. The uplink flow case model is similar but with a difference in the direction of flow of test packets. In the uplink flow, *t(0,0)*, *t(0,1)*, and the '*Source*' are on the IEEE 802.11b side, while *t(d,0)*, *t(d,1)*, and the '*Destination*' are on the Ethernet side.

Having two directions of flow (uplink and downlink), the AP link model will act with different transmission speeds for uplink and downlink. In this way, it is similar to many commercial links, where the speed of upload is different from the speed of download (like ADSL links, for instance). Knowing the characteristics of the link model, we investigate the bandwidth of this link to parameterize its value. Figure 2 shows the bandwidth, *b* (bits per second), of the link model of the AP for the downlink flow case.

A. Packet Pair Queuing

Packet-pair in FIFO-queuing networks makes use of a two-packet logical model [3], whose parameters are the difference in arrival times of two identical packets sent from the same source to the same destination. The arrival times in the packet-pair model are arrivals at the destination, hence, they resemble the departure times of the two packets from the AP in our previous queuing model [5]. Below we discuss the assumptions of the packet-pair property to show how they suit our model and analysis.

One of the assumptions of the packet-pair property is that the two test packets queue together at the bottleneck link (which is the AP link in our model) and not later. This assumption is true in the case of our experimental tests since we designed controlled tests, which use a set of back-to-back packets.

The second assumption of the packet-pair property is that the bottleneck router/node uses FIFO-queuing. In our previous work, we showed that the AP is modeled as a FIFS queue with one server. Hence, the second assumption holds true for our model.

The third assumption assumes that the transmission delay of the link under study is proportional to the packet size, and that routers/nodes are store-and-forward. In this respect, our previous results showed that the service time of the model was nothing but the time for checking headers, performing simple management, and transmitting the bits of the frame. The header check-time is constant. The management time is constant and is negligible compared to transmission times. Moreover, the transmission time depends on the number of bits in the packet. Therefore the service time is directly proportional to the transmission time, which- in turn- is directly proportional to the packet size (or payload), as (1) shows.

Finally, the packet-pair property does not consider the per-link latency [3], and in our link model we assume that the link latencies are negligible compared to the transmission time on each of the three links: Ethernet, *b* link of AP-link-model, and IEEE 802.11b.

B. AP as a Link with Variable Bandwidth

Knowing that the assumptions of the packet-pair property hold true for our model, we consider the AP model as a link and investigate the bandwidth, *b*, of this link. Figure 2 shows the link model of the AP, for a downlink flow example. The analysis is similar in both cases: for uplink and for downlink with only a difference reflected in the parameters of (1), and in the link header sizes. The packet-pair property states that [3]:

$$t(d,1) - t(d,0) = \mathbf{max}\,\{[s_1/b],\ [t(0,1) - t(0,0)]\} \tag{2}$$

where,

- *t(d,0)* and *t(d,1)* are the arrival times (at the destination) of the first and second packets respectively;
- *t(0,0)* and *t(0,1)* are the times of transmission of the first and second packets respectively.
- s_1 is the size of the second packet in bits;
- *b* is the bandwidth of the bottleneck in bits per second.

The d-labeled timestamps resemble departures of the first and second packets from the AP-queuing-system respectively. Solving (2) for b is derived in [3]. The result of the solution is that the bandwidth of the bottleneck link is:

$$b = \frac{s_1}{t(d,1) - t(d,0)} \quad \text{(bits per second).} \qquad (3)$$

The denominator in (3) is the difference between the arrival times of the packet-pair at the destination. These *packet-pair arrivals* are the departure times calculated from the queuing model of the AP.

When packets in the queue are waiting back-to-back, then the difference between the departures of the first packet and the second packet in queue is the time that the server (of the queuing model) spent serving the second packet. The Interframe Space (IFS) is calculated within the average service-time. Since both test-packets are of equal sizes, substituting in (2) shows that the service time for any of the two packets is the same. Hence, the denominator in (3) is nothing but the average service time:

$$t(d,1) - t(d,0) = \textit{Average service time of test-packet.} \qquad (4)$$

Let us denote the service time of any of the two test-packets as S. Equation (1) shows that S is a linear function of payload. Thus, we can express (1) in terms of payload for this case to be:

$$S = \frac{r}{32}P + S_o - \frac{5}{4}r \quad (\mu\text{s}) \qquad (5)$$

where,

- S is the service time of one of the two test packets in µs;
- P is the IP payload in bytes.

Substituting (5) in (4), we get:

$$t(d,1) - t(d,0) = \frac{r}{32}P + S_o - \frac{5}{4}r \quad (\mu\text{s}). \qquad (6)$$

Substituting (6) in (3), we express b in Mbps:

$$b = \frac{8*(P+h)}{\left(\frac{r}{32}\right)P + \left(S_o - \frac{5}{4}r\right)}\left(\frac{10^6}{1024*1024}\right) \text{(Mbps)} \qquad (7)$$

where,

- P is IP payload expressed in bytes;
- h is a constant resembling the size of the IP header in bytes, i.e. 20B without the *Options* field in IPv4.

Equation (7) shows that the bandwidth of the link model is not constant but rather dependent on the packet size (or payload). Thus, the bandwidth of the link model adapts to different payloads. Consequently we call the bandwidth of the AP-link-model the adaptive bandwidth, b_a, which is simplified in (8) for IPv4.

$$\underset{\{IPv4\}}{b_a} \cong 244.14 \frac{(P+20)}{(r.P + 32.S_o - 40.r)} \quad \text{(Mbps).} \qquad (8)$$

To analyze the behavior of the adaptive bandwidth in terms of payload, the partial derivative of b_a with respect to IP payload, P, is derived in (9).

$$\frac{\partial b_a}{\partial P} = 7812.48 \frac{S_o - 1.875r}{(rP + 32S_o - 40r)^2} \qquad (9)$$

Since r resembles the transmission time of 32 bytes plus some extra management time in the AP, then S_o is always greater than twice r. S_o will always be larger than the transmission time of 40 bytes of IP payload plus link-layer-overhead, which count to more than $2r$ on both: the downlink and the uplink. Hence,

$$S_o > 1.875r \Leftrightarrow S_o - 1.875r > 0. \qquad (10)$$

Knowing (10), then (9) is always positive. From this analysis we conclude that b_a *is monotonically increasing* in terms of P. Consequently, the adaptive-bandwidth of the AP is an increasing linear fractional transformation of IP payload. Hence, b_a is limited by the upper b_a bound (as function of the largest IP payload) and the lower b_a bound (as function of the smallest IP payload in the test design, i.e. 40 bytes), as shown in figure 3.

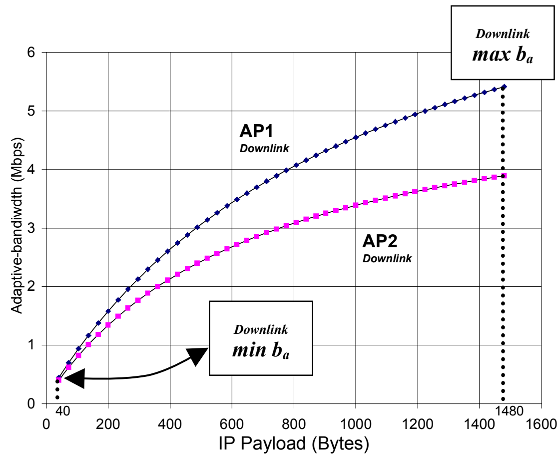

Figure 3. Downlink adaptive-bandwidth (Mbps), downlink b_a, of two WLAN APs. AP1 is Lucent WavePOINT-II and AP2 is Lucent AP2000. AP1 shows higher downlink adaptive-bandwidth values for all payloads. The IP payload is in bytes.

Figure 3 shows two plots of two downlink adaptive-bandwidth equations derived for two different APs: Lucent WavePOINT-II and Lucent AP2000 [12]. Figure 4 shows plots for the same APs but for the uplink adaptive-bandwidth analytic solutions. The parameters, S_o and r, of the two APs, on the downlink and the uplink, were calculated using the test design discussed in section II. Lucent WavePOINT-II shows a higher curve for the downlink and uplink adaptive bandwidths, hence better performance. Following our model and analysis, different wireless LAN access points can be compared in terms of performance.

This result of b_a can be interpreted as follows: the operational speed of the link model of the AP increases to a maximum bound after which services suffer high delays and packet loss depending on the available buffer size. Figures 3 and 4 show two bounds for the adaptive-bandwidth: the minimum b_a is for smallest IP payload (40B), and the maximum b_a is when the IP payload is maximum (1480B).

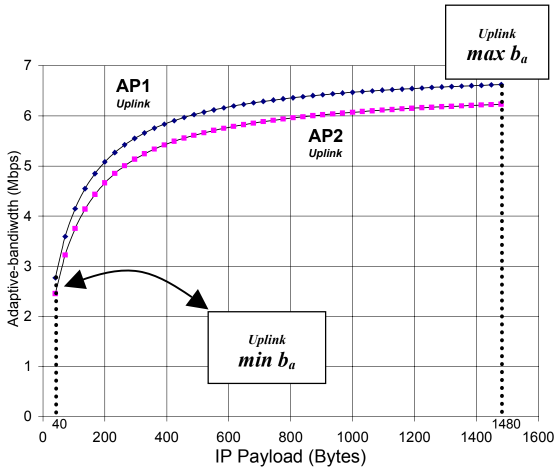

Figure 4. Uplink adaptive-bandwidth (Mbps), *uplink* b_a, of two WLAN APs. AP1 is Lucent WavePOINT-II and AP2 is Lucent AP2000. AP1 shows higher uplink adaptive-bandwidth values for all payloads.

The throughput of a link, as defined by Stevens [10], is directly proportional to the link bandwidth. Hence, using the link model of the AP, we can conclude that the higher the link bandwidth curve for a given AP, the higher the throughput (bits per second) of the AP. The throughput of a link with one source sending over the link is defined in [10] as:

$$Throughput_{(bps)} = \frac{User\ \ received\ \ data}{Time\ \ to\ \ send\ \ on\ \ link} = \frac{User\_data}{\left(All\_data\_sent \middle/ link\_speed \right)} = \frac{User\_data}{All\_data}\left(Link\_Speed\right). \tag{11}$$

So the throughput of the link of the AP model is:

$$T_{\substack{APL\\(Mbps)}} = v.\boldsymbol{b_a} \tag{12}$$

where,

$$v = \frac{Sum\_of\_all\_IP\_payloads\_received}{Sum\_of\_all\_link\_frames\_sent}.$$

In most links, the bandwidth has a constant value relative to the link protocol; hence, the throughput of the link is dependent on the ratio of the received user data to total overhead and data, when there is no collision. In the AP-link-model, there is no collision, but the bandwidth of the link is a function of payload. Hence the throughput of the link model is directly proportional to payload.

The analytic solution of the adaptive-bandwidth shows good correlation with the experimental results on the throughput of WLAN APs. Table 1 compares the analytic values of the adaptive bandwidth that are shown in figures 3 and 4, with the measured values of throughput that were investigated and presented in [13]. As (12) shows, the throughput is directly proportional to the adaptive bandwidth. Hence, the comparison was based on the proportionality of the two variables. Since the investigation in [13] presents only throughput values for traffic streams, whose IP payloads were 1480B, then we could only compare the measured throughput values with the analytic values of adaptive bandwidth for 1480B payload (the maximum presented in figures 3 and 4).

TABLE I. COMPARISON BETWEEN MEASURED THROUGHPUT AND CALCULATED ADAPTIVE BANDWIDTH FOR 1480B IP PAYLOAD. MEASURED VALUES AND STANDARD DEVIATION VALUES ARE FROM [13].

	AP1 Lucent WavePOINT-II		**AP2** Lucent AP2000	
	Downlink	*Uplink*	*Downlink*	*Uplink*
Analytic Adaptive Bandwidth (Mbps)	5.41	6.62	3.89	6.23
Measured Max. Throughput (Mbps)	6.41	5.94	3.91	5.93
Standard Deviation of Measured Throughput (Mbps)	*1.4 1*	*0.41*	*0.10*	*0.01*

The differences in the values of adaptive bandwidth and throughput in table 1 are due to the fact that we are comparing bandwidth with throughput to check correlation, and the value v in (12) is not considered in the calculations of the adaptive bandwidth. Moreover, measurement errors due to clock drifts and time stamp resolutions in [13] lead to different standard deviations in the statistical results. Hence, looking at the standard deviation of the measured throughput in the bottom row of table 1, the differences in the calculated and measured values could be explained. Consequently, good correlation between the measured values and the analytic values are observed.

C. Feedback Control System for AP Throughput

The adaptive bandwidth formula inspires us to look at the throughput as a ratio of an output-function to and input-function of a feedback control system as shown in figure 5. Since there are two average service time formulae for each access point: one for the uplink and one for the downlink, then there are two feedback control models for the throughput: uplink feedback model and downlink feedback model.

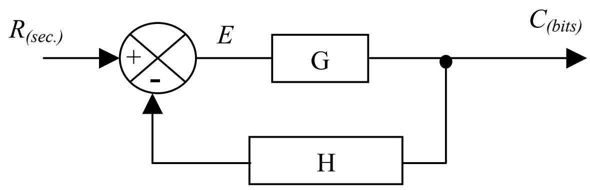

Figure 5. Feedback control model for throughput of the AP link model. *R* is the input fuction in seconds, and *C* is the output function in bits. *G* is the feed-forward transfer function and *H* is the feedback transfer function. The ratio of *C* over *R* gives the throughput (in Mbps) of the AP link model. *E* is the error, and it is the difference between the input *R* and the product *HC*.

The functions *G* and *H*, as shown in (13) and (14), can be related to AP parameters and to the parameter v described in (12).

$$G = 244.14(v)\frac{P+h}{(32S_o - 40r)}. \tag{13}$$

$$H = \frac{rP}{244.14(v)(P+h)}. \tag{14}$$

Both, uplink and downlink feedback models of throughput for the same AP can be represented as shown in figure 5 with a difference in the values of S_o, r, and v that are the variables in *G* and *H*, respectively. Each direction of flow (uplink or downlink) has a different set of the three variables: S_o, r, and v. Hence each of the two directions of flow has a different *G* and a different *H*.

Analyzing the system above, we get the throughput (Mbps), which is the ratio of *C(bits)* to *R(sec.)* as shown in (15).

$$T_{APL} = \frac{C}{R}(bits/\sec) = \frac{G}{1+GH} \text{ (Mbps)} \tag{15}$$

This model shows that the error (as described in feedback control being the value *E* shown in figure 5), is due to the difference between the input, *R*, and the product *HC*. Hence, the *feedback transfer function*, *H*, plays a significant role in the value of the error, *E*. Consequently, analysis of the feedback system can be used to investigate the effects of the AP parameters, S_o and r, on the behavior of the AP. The feedback control model is beneficial for further analysis of the behavior of traffic passing through a WLAN AP. In this respect, one can separate the parameters that affect the feed-forward portion of throughput, *G*, from the parameters that decide the feedback portion, *H*, and its effects; a result that is useful for data traffic shaping, simulation and analysis. Furthermore, the equations of the feedback model can be reversibly studied for the effect of the design of the AP hardware and its correlation with protocol implementation on QoS. Therefore, fruitful suggestions and implementation could be presented to manufacturers, designers, and implementers. Some relationships between the AP parameters could be derived from the feedback model in order to enhance QoS relative to specific applications. A future paper will discuss these results.

IV. Conclusion

The purpose of the paper was to present a feedback control model used to parameterize the throughput of a WLAN AP. The AP is looked upon as a link with adaptive-bandwidth bounded by two limits: the upper adaptive-bandwidth bound and the lower adaptive-bandwidth bound. The link model of the AP builds on previous results of the AP queuing-model and its service-time analytic solution. The packet-pair property of FIFO-queuing networks is shown to be very suitable for the link model analysis. The bandwidth of the link model of the AP was analyzed and found to be an increasing linear fractional transformation of payload. Since the throughput of a data communication link is directly proportional to the speed of the link, then a major result is that when using our model and analysis, performance of different wireless LAN APs can be compared in terms of throughput (as a QoS parameter). A key result is that the throughput of a WLAN AP can be modeled as the ratio of output to input of a feedback control system, whose feed-forward and feedback transfer functions are presented analytically.

Acknowledgment

I thank Daimler Sweden AB (*www.daimler.se*) for supplying me with different wireless LAN APs. Thanks to professor Gerald Q. Maguire Jr. at the Department of Microelectronics and Information Technology (IMIT), Royal Institute of Technology (KTH), Stockholm, Sweden, for the fruitful discussions. After I got the link model and the analytic solution of the throughput of the WLAN AP as a linear fractional transformation of payload, professor Maguire gave me the idea of trying to model this expression for a feedback control system. I would also like to thank Nicolas Baron at the Department of Technical Acoustics, Royal Institute of Technology (KTH), Stockholm, Sweden, for the valuable discussions on many mathematical concepts and theorems. Finally, I express thanks to Enrico

Pelletta at IMIT, KTH, Stockholm, Sweden, for supplying me with experimental results on AP throughput that he got from a long investigation on APs, which he had conducted.

REFERENCES

[1] PCC (Personal Computing and Communication) project, a Swedish research program financed by the Foundation for Strategic Research. http://www.pcc.lth.se/

[2] S. Keshav, "A Control-Theoretic Approach to Flow Control," in ACM SIGCOMM Computer Communications Review, vol 21, issue 4, pp. 3-15, September 1991.

[3] K. Lai and M. Baker, "Measuring Link Bandiwdth Using a Deterministic Model of Packet Delay," in Proceeingds of ACM SIGCOMM'00, vol. 30, no. 4, pp. 283-294, August 2000.

[4] I. Al Khatib, G. Q. Maguire Jr., R. Ayani, and D. Forsgren, "Wireless LAN Access Points as Queuing Systems: Performance Analysis and Service Time" to appear in SigMobile's Mobile Computing and Communications Review, last issue 2002.

[5] I. Al Khatib, G. Q. Maguire Jr., R. Ayani, and D. Forsgren, "Wireless LAN Access Point Modeling as a Queuing System," CCN 2002, the IASTED International Conference on Communications and Computer Networks, MIT, Cambridge, MA, USA, November 4-6, 2002.

[6] IEEE802.11 standard, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1997.

[7] IEEE802.11b standard, (Supplement to ANSI/IEEE Std. 802.11, 1999 Edition), Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band, 1999.

[8] J. Banks, S. Carson, B. L. Nelson, and D. M. Nicol, *Discrete-Event System Simulation*, 3rd ed., copyright Prentice-Hall, 2001.

[9] N. C. Hock, *Queuing Modeling Fundamentals*, John Wiley and Son, 1998.

[10] W. R. Stevens, *TCP/IP Illustrated, The protocols*, vol. 1, Addison-Wesley, 1998.

[11] C. Chapra and R. P. Canale, *Numerical Methods for Engineers,* McGraw-Hill, 1988.

[12] ORINOCO for Lucent wireless LAN access points. http://www.orinocowireless.com/

[13] E. Pelletta, "Bridging Capacity of IEEE 802.11b Access Points," Master thesis technical report, IMIT, Royal Institute of Technology, Stockholm, Sweden, May 2002.

Publications 2 and 3

Iyad Al Khatib, Gerald Q. Maguire Jr., Rassul Ayani, and Daniel Forsgren

Wireless LAN Access Points as Queuing Systems: Performance Analysis and Service Time

To appear in ACM SigMobile Mobile Computing and Communications Review (MC^{2}R), last issue 2002.

Also available at:

ACM MOBICOM'02
The Eighth ACM International Conference on Mobile Computing and Networking, Atlanta, Georgia, USA, September 23-28, 2002
http://www.acm.org/sigmobile/mobicom/2002/
Sponsored by ACM SIGMOBILE
ACM short paper (extended abstract) reference:
http://www.acm.org/sigmobile/mobicom/2002/posters/al-khatib.pdf

Wireless LAN Access Points as Queuing Systems: Performance Analysis and Service Time

Iyad Al Khatib *ikhatib@it.kth.se*
Gerald Q. Maguire Jr. *maguire@it.kth.se*
Rassul Ayani *rassul@it.kth.se*
Daniel Forsgren *forda@it.kth.se*
Department of Microelectronics & IT (IMIT), Royal Institute of Technology (KTH), Stockholm, Sweden

Since the approval of the IEEE 802.11b by the IEEE in 1999, the demand for WLAN equipment and networks has been growing quickly. We present a queuing model of wireless LAN (WLAN) access points (APs) for IEEE 802.11b. We use experimentation to obtain the characteristic parameters of our analytic model. The model can be used to compare the performance of different WLAN APs as well as the QoS of different applications in the presence of an AP. We focus on the delay introduced by an AP. The major observations are that the delay to serve a packet going from the WLAN medium to the wired medium (on the uplink) is less than the delay to serve a packet, with identical payload, but travelling from the wired medium to the WLAN medium (on the downlink). A key result is an analytic solution showing that the average service time of a packet is a strictly increasing function of payload.

I. Introduction

Today most WLANs use WLAN APs to connect multiple users to a wired backbone network [1]. To provide suitable service, an understanding of the behavior of WLAN APs is essential. The first step is to define the system of interest [2]. Based on our initial experiments, we model the WLAN AP as a queuing system. We are not aware of any study that has looked at the WLAN AP as a point of reference to be modeled as a queuing system. There are various advantages of our model ranging from the ability to compare the performance of different APs to the simple parameterization of service time. The key result is an analytic model for the average service time of the WLAN AP in terms of payload.

II. Logical model

In our investigation, we seek to model the delay processing in the WLAN AP. A set of assumptions was made. We isolate the AP and define two events: *arrival* and *departure* (figure 1). The parameters of interest are the arrival time (T_a) and the departure time (T_d).

Since the number of packets inside the system changes when a packet arrives or when a packet departs, i.e. at separate points in time, then the system is a discrete-event system [3]. The system (figure 1) considers any packet entering the AP, whether coming from the Ethernet side or the WLAN side, as an arriving packet. Similarly, any packet leaving the AP, whether it leaves to the Ethernet medium or to the WLAN medium, is considered a departing packet. The arrival and departure times recorded from experiments have shown that the system can be modeled as a single server system with one FIFS queue. We, then, add one more event, that of *entering the server* (figure 1), and we define the two system states: waiting and service [4]. The waiting time and the service time of packet P_i are denoted by W_i *and* S_i, respectively, where i is a positive integer representing the logical *identification* of the packet with respect to its time of arrival. We define the total delay of a packet P_i, or the *response time* (R_i), to be the time difference between the departure time and the arrival time of P_i, hence R_i is the sum of W_i and S_i.

The service time includes the time to check the headers of the link frame, code management time, protocol management time (spanning tree), and transmission time.

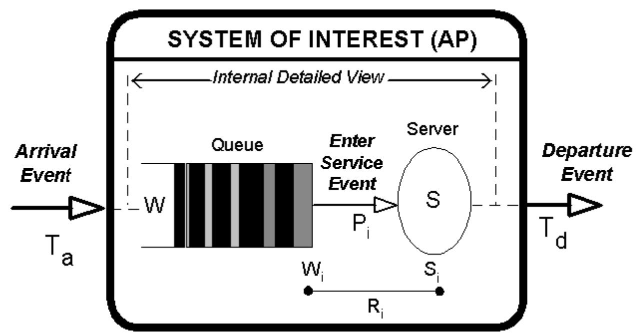

Figure 1: Logical model of the AP as a system of reference with two external events (arrival and departure) and one internal event (enter service). T_a and T_d are the arrival and departure times of packets entering or leaving the system, respectively. W and S are the *system-states* of *waiting* and *service*, respectively. W_i and S_i are the *waiting time* and the *service time* of packet P_i respectively, and $R_i = W_i + S_i$.

There is, however, a physical constraint that prevents direct measurement of W_i and S_i, because we can only easily measure the arrival and the departure times of packets. To solve this problem, we designed an algorithm that is described in section III.

III. Tests and SSTP algorithm

We send UDP packets to avoid any bits travelling backwards, and we increase the UDP payload by 32 bytes in each experiment. The maximum UDP payload we use is 1472 bytes, because sizes beyond the MTU may result in fragmentation [5]. We designed the SSTP (Simple Service Time Producer; figure 2) algorithm to calculate the values of internal-state parameters, W_i and S_i, from external-event parameters (T_a and T_d).

Packet #	T_a	T_d	Response time	Waiting time	Service time
P_1	T_1	T'_1	R_1	W_1	S_1
:	:	:	:	:	:
P_{i-1}	T_{i-1}	T'_{i-1}	R_{i-1}	W_{i-1}	S_{i-1}
P_i	T_i	T'_i	R_i	W_i	S_i
:	:	:	:	:	:
P_n	T_n	T'_n	R_n	W_n	S_n

Measured data set

```
1  for i = 1 to n
2    do Ri = T'i - Ti
3      if i = 1 or Ti ≥ T'i-1
4        then Wi = 0
5             Si = Ri
6      else if Ti < T'i-1
7        then Wi = T'i-1 - Ti
8             Si = T'i - T'i-1
```

Figure 2: SSTP algorithm calculates response time (R_i ,line 2), waiting time before entering service (W_i , lines 4 and 7), and service time (S_i , lines 5 and 8) for each packet P_i. T_i and T'_i are the measured arrival time (T_a) and departure time (T_d) of P_i, respectively.

IV. Results

IV.A. Directional delay

For the same AP, the uplink service time is less than the downlink service time for a packet with identical IP payload (figure 3). For all access points tested, on the same direction of packet flow (uplink or downlink), the average service time of a packet is monotonically increasing with payload, and that is mainly due to the increase in transmission time.

IP Payload (Bytes)	AP1 Average Service Time (µs)		AP2 Average Service Time (µs)	
	Downlink	Uplink	Downlink	Uplink
40	894	152	999	171
138	962	257	1140	274
264	1087	395	1374	419
520	1323	668	1820	733
1480	2089	1705	3399	1876

Downlink and uplink average service times show that server of AP1 is faster than that of AP2

Figure 3: Comparison of two Lucent APs. AP1 is WavePOINT-II. AP2 is AP-2000 [6]. Uplink has less service time than downlink.

IV.B. Service time analytic solution

The discrete-event system allows us to look at the service time values as terms of a sequence. Our analysis showed that the average service time, S_n, can be modeled as presented in equation (1). Since we use a 32B increment in the payload, then the UDP payload is always divisible by 32B, hence, n is a positive integer.

$$S_n = S_o + (n-1)r \tag{1}$$

where,

n = (IP_Payload$_{[\text{in bytes}]}$ - 8B$_{[\text{UDP header}]}$) /32B;

S_n = service time (µs) for packet with (32n+8)B of IP payload;

S_o = service time (µs) for packet with 40B IP payload;

r = incremental difference in µs (calculated from linear regression of average service times of different payloads).

S_o and r are different for different APs.

IV.C. Video-specific results

We use our analysis to compare QoS metrics of layered MPEG-1 and non-layered MPEG-1 videos of different types in the presence of an AP. The results of the video-test measurements show that for both videos, there are more delays on the downlink than on the uplink. Thus for a live videoconference, QoS will suffer from a relatively larger delay on the downlink. The type of AP used is very important, and the smaller the service time per packet size the better the performance of the video application, which may have varying sizes of UDP datagrams for the same video file or session. The results of the video analysis show good correlation with our AP model. Hence, we can use our results to enhance the performance of video applications over WLANs by sending more suitable payloads.

V. Conclusion

The main purpose was to present a logical model for wireless LAN APs as a single server, single queue, FIFS system. An interesting result is that uplink service time is relatively much smaller than downlink service time. Using our test design and analysis, one can get an analytic solution of the average service time, which is a linear function of payload. The model can be used to enhance performance of applications over WLANs, especially multimedia applications

References

[1] PCC (Personal Computing and Communication) project, a Swedish research program financed by the Foundation for Strategic Research, http://www.pcc.lth.se/

[2] M. Law, W.D. Kelton, *Simulation modeling and analysis*, 3rd ed., McGraw Hill, 1999.

[3] J. Banks, S. Carson, B. Nelson, D. Nicol, *Discrete-event system simulation*, 3rd ed., Prentice-Hall, 2001.

[4] J-C. Bolot, "End-to-end delay and loss behavior in the Internet", in proceeding of ACM SIGCOMM'93 (San Francisco, CA, USA), 1993, pp.289-298.

[5] W.R. Stevens, *TCP/IP Illustrated, The protocols*, vol. 1, Addison-Wesley, 1998.

[6] ORINOCO for Lucent wireless LAN Access Points, http://www.orinocowireless.com/

Publication 4

Iyad Al Khatib, Gerald Q. Maguire Jr., Rassul Ayani, and Daniel Forsgren

Wireless LAN Access Point Modeling as a Queuing System

in Proceedings of the Communications and Computer Networks 2002 Conference (CCN 2002), pp. 463-468, MIT, Cambridge, USA, November 4-6, 2002.

ISBN: 088986-329-6

WIRELESS LAN ACCESS POINT MODELING AS A QUEUING SYSTEM

Iyad Al Khatib, Gerald Q. Maguire Jr., Rassul Ayani, and Daniel Forsgren
Royal Institute of Technology (KTH)
Department of Microelectronics and Information Technology (IMIT)
Stockholm, Sweden
{ikhatib, maguire, rassul, forda}@it.kth.se

ABSTRACT

This paper presents a research study of wireless LAN access points for IEEE 802.11b, where we seek to model the access point as a queuing system. The model can be used to compare performance metrics of different wireless LAN access points and to investigate the QoS of specific applications in the presence of a wireless LAN access point. In this paper, we focus on two parameters: the delay introduced by a wireless LAN access point and the average service time required to serve a packet passing through an access point. A major result is an analytic solution for the average service time of a packet in relationship to payload.

KEY WORDS
Wireless LAN, access point, IEEE 802.11, modeling, queuing system, delay

1. INTRODUCTION

The increase in wireless LAN (WLAN) deployment is quite evident in private and public places, and users have shown great interest in getting connected without being tethered by a wire. Beyond this demand, a requirement for better performance is prevalent [1]. However, to enhance service, a fundamental step is to understand how the WLAN AP behaves. To realize how the performance of a system could be enhanced, the system of interest must be defined [2]. Based on our experiments, we model the WLAN AP as a queuing system. Many studies have been made on WLANs, however, most of the investigations are related to throughput and quality of service analysis of applications. We are not aware of any study that looks at the WLAN AP itself as a queuing system. The advantages of our model are manifold: ranging from having a test design and algorithm to compare performance of different types of WLAN APs to the simple parameterization of the average service time of a packet for a given AP.

Below, we present our results on delay and service time. The key result is the analytic model for the average service time of a WLAN AP in terms of payload. Hence, the developer or the user of the AP can estimate the average time that a packet will need to be served by using a mathematical formula rather than running tedious measurements.

2. MATHEMATICAL MODELING

We seek to model the processing in a wireless LAN access point (WLAN AP). A set of assumptions was made for this model. We logically isolate the WLAN AP and define the different events that occur. These events can be classified into two types: external and internal events. A set of parameters is associated with each event. In isolating the AP, we consider it as a black box and define two external events: arrival and departure (figure 1). Whether the packet arrives from the wired part or the wireless part, the model considers it as an arriving packet. Similarly, the model considers any packet leaving the system as a departing packet regardless of the medium it goes out on. We view all parameters in the investigation from the point of view of the WLAN AP itself. Hence, when a packet enters the system, we are interested in the time when the packet is in the system in its entirety, and we view this from the AP point of view as the arrival time of the packet. Similarly, when a packet leaves the system, we view the departure time as the time when the entire packet is out of the AP system. Consequently, we look at the relationship between arrivals and departures as the *response* of the system. Hence, the total processing delay of a packet, is defined as the *response tim*e, and is the time difference between the departure time and the arrival time of that packet. In packet based communications, there are often interframe spaces between packets transmitted over the same medium, i.e. packets are sent or received over the shared data communications media separated by some time difference to avoid collision. Due to this time difference, packets arrive at the system at separate points in time. Since the number of packets inside the system changes when a packet (in all its bits) leaves the system or when a packet enters the system, and since the packet is the entity that the system works on, then we are dealing with a discrete-event system [3]. We define traffic flows through the AP to be in two directions: from wired Ethernet to wireless LAN (downlink) and from wireless LAN to wired Ethernet (uplink). Modeling the WLAN to WLAN is out of the scope of this paper. We assume that the AP, as a system of interest, can be modeled as a queuing system with a certain number of queues and a number of servers. At this point, we ran experiments to check the number of servers and queues of the model. Analyses of experimentally recorded time stamps of arrival time and departure time values showed that the AP queuing system *can* be modeled as a single queue, single server system (figure 1). Knowing that there are only one queue and one server, then, logically, the system has one internal event: the enter-service event.

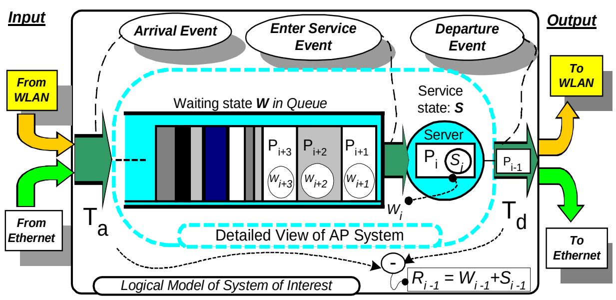

Figure 1. Logical model and detailed view of AP system. T_a is the arrival time and T_d is the departure time. W_i and S_i are the waiting time and the service time of packet P_i , respectively.

Within the different events there are state transitions. We have three events, hence, there are two state transitions. We refer to these two system states as: *waiting* and *servic*e [4]. The waiting state is the state transition between the *arrival event* and the *enter-service event.* The service state is between transition from the *enter-service event* and the *departure event*. Furthermore, the response of the system can be modeled as the total state transition from arrival to departure. The parameters of interest are the waiting time and the service time, respectively. We denote the waiting time, the service time, and the response time of a packet P_i as W_i, S_i, and R_i, respectively. Knowing the events, the system states, and the parameters of interest, we define the relationships between the event parameters and the state-parameters as follows:

1) The waiting time (W_i) of packet P_i is the time from when packet P_i arrives at the system until it enters the server.

2) The service time (S_i) of packet P_i is the time from when packet P_i enters the server until it departs from the system.

3) The response time (R_i) of packet P_i is the time from when packet P_i arrives at the system till the time it departs from the system.

Consequently, the response time can also be defined in terms of the waiting time and the service time as:

$$R_i = W_i + S_i. \quad (1)$$

Calculating R_i can be easily done since we can record the arrival and departure timestamps of every packet P_i that is entering or leaving the system. However, W_i and S_i are logical model parameters that can not be easily measured. Hence, we designed an algorithm to calculate the values of W_i and S_i for each packet P_i by using the recorded arrival and departure times. We call the algorithm the SSTP (Simple Service Time Producer). In section 4.1, we present the third version of the algorithm (SSTP-1.3). The first and second versions were presented in [5] and [6], respectively.

3. TESTBED

The testbed is designed to be able to test AP performance (figure 2). We have two main entities besides the AP itself: PCs connected to the Ethernet side (denoted by EPCm, where m is the index number of the EPC) and PCs connected to the WLAN side (denoted by WPCq, where q is the index number of the WPC). Both, EPCs and WPCs act as traffic sources and sinks. In order to monitor the tests, we prepared a different PC for traffic sniffing (SPC) as shown in figure 2. The operating system on all PCs of the testbed is Linux-2.2.16. To record timestamps and other packet information passively, we use tcpdump. Moreover, when EPCs or WPCs act as traffic sources, they use MGEN 3.2 [7] for generating UDP streams. MGEN was only used for transmitting UDP packets as we used our own program modules to filter packets from tcpdump and analyze the results. Since accurate timestamping of arrival and departure times is essential for later analysis, we checked the clock drift in the monitoring PC (SPC) and the resolution of the tcpdump program, which we used for measurements. We found that the resolution used could give very accurate measurements from our tests. Moreover, we used one-second based experiments, hence the clock drift would be negligible and wouldn't correlate with subsequent measurements.

To run experiments, where traffic flows from a single sender to a single sink, and to run others from multiple senders to multiple sinks on both directions (downlink and uplink), EPCs and WPCs take turns in being senders and receivers. The dashed line in figure 2 circumscribes the entities (EPC1, WPC1, SPC, and the AP) used in the *single-sender-to-single-sink* experiments.

We chose a surrounding environment, where there were no radio signals from other access points. To run the radio signal tests, we used ORINOCO Client Manager [8].

4. TEST DESIGN AND RESULTS

The first set of experiments we ran was designed to check whether modeling is applicable. These tests utilized multiple senders and multiple receivers on both sides (Ethernet and WLAN). We sent packets in both directions as described in section 2, and we analyzed the departure events. Analysis of data for both directions showed that the system could be modeled as a queuing system with a single FIFS queue and a single server. The case of bi-directional traffic was also studied, and it showed a FIFS queue management, however its results will be presented in a future paper. In the following section we describe the experimental test design that is used to extract results about the model.

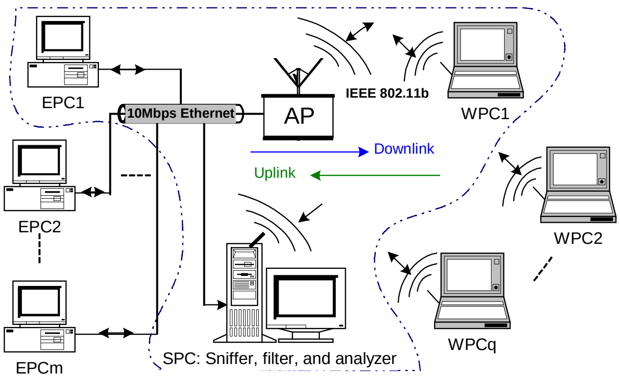

Figure 2. Testbed showing the downlink and uplink directions. Downlink traffic consists of packets travelling from the Ethernet side (EPCs) to the IEEE802.11 side (to WPCs). Uplink traffic are packets travelling from the IEEE802.11 side (WPCs) to the Ethernet side (to EPCs). The wired link is a 10Mbps Ethernet. The WLAN link is IEEE 802.11b. SPC is a passive "*sniffer*" to record traffic in order to use it for later filtering and analysis. The dashed line encircles the set of equipment used for the AP tests to extract model parameters.

4.1. AP Testing and SSTP-1.3

Our test consists of two main parts: single-source-to-single-sink (SS) and multiple-sources-to-multiple-sinks (MM). The SS part is used to extract parameters of the model, and the MM part of tests is used to check for the queue management and behavior. Each part is made up of two subclasses of tests: downlink traffic tests, and uplink traffic tests. In the downlink test class, EPCs are the traffic sources, and WPCs are the traffic sinks. In the uplink test class, WPCs are the traffic generators, and EPCs act as receivers. In addition, tests were conducted when there was bi-directional traffic, i.e. EPCs and WPCs were both senders and sinks. In this paper we present a portion of our results of this investigation focusing on delay and service time for directional traffic. The two subclasses of tests are composed of identical sets. A set of tests is made up of clusters of experiments. In each cluster we fix all parameters and vary one traffic related parameter (usually packet size) to investigate the effects. We call the experiment in a cluster an *experimental test run* (ETR). The ETR is the basic unit of tests. We describe an ETR for the clusters related to the results on directional traffic. In each ETR, we send a stream of identical UDP datagrams from the sender(s) to the receiver(s). We use UDP because we designed our tests to have only forward traffic relative to the direction of flow, thus no traffic should come back in the opposite direction of the ETR data flow. For each ETR, we vary the size of the packets to be sent in a stream via increasing the payload of the UDP datagram by 32 bytes. The maximum number of bytes we used as UDP payload was 1472, because we are not interested in fragmentation, and sizes beyond the MTU may result in fragmentation [9]. The headers and interframe spaces are thoroughly calculated before sending any traffic stream, because there is a major difference between the link frames of the Ethernet and the IEEE802.11. The preamble on Ethernet is 8 bytes, and the Ethernet interframe space is 9.6 μsec [9]. On the IEEE802.11, there are different scenarios [10]; however, our experiments are controlled so that the WLAN link layer interference is described by the four link layer parameters shown in figure 3 [11]:

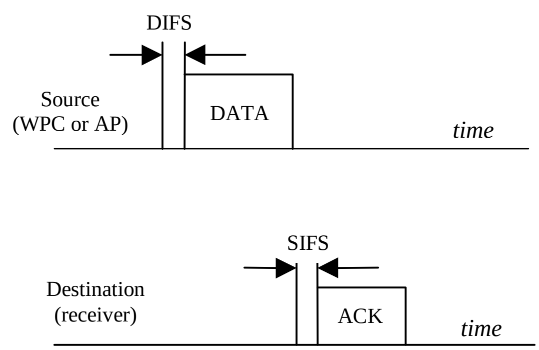

Figure 3. The time line showing the WLAN link frames present for each packet transmitted in the packet stream of our ETR. The DIFS is the Distributed Coordinated Function Interframe Space. The DATA is the link frame with preamble, PLCP, MAC overhead, and IP packet. The SIFS is the Short Interframe Space. The ACK is the link layer acknowledgment. Source and destination can be a WPC or the AP itself.

1) *DIFS*, which is the DCF-IFS (Distributed Coordinated Function Interframe Space). For our study, it is 50μsec per frame in each of our ETRs.

2) *Preamble and PLCP (Physical Layer Convergence Protocol) headers*, which is 272 bits per frame in each of our ETRs.

3) *SIFS*, which is the Short Interframe Space used for special acknowledgments, and it adds 10μsec per frame in each of our ETRs.

4) *ACK***,** which is the link layer Acknowledgment and it adds 304 bits per frame in each of our ETRs.

For statistical purposes, each ETR is repeated a number of time, which is up to the choice of the test designer. Three iterations of the same ETR is the minimal number accepted for statistical analysis. ETR clusters are built based on different utilized bit rates over the link. Each ETR involves passive traffic measurement using the tcpdump program on SPC to record traffic on both sides: Ethernet and WLAN. In addition, in each ETR a program extracts the arrival time (T) and departure time (T') for each packet. For a lost packet, the departure time is considered as infinite and denoted with the value '-1'. We have two parts for each experiment: actual measurements and post-measurement. After clustering ETRs, we analyze the runs. At this point, it is crucial to note that the ETR measurements and filtering give information only about two parameters for each packet: the arrival time and the departure time. Consequently, we only have values for the parameters of external events. However, we need information about internal-state parameters. To solve this problem, we designed a simple algorithm (SSTP) that looks at the ETR data and analyzes it to extract the values of the parameters we need for the model. The algorithm is run on each ETR data set. The result is a file similar to the data shown in figure 4.

Packet Number	T_a	T_d	Response Time	Waiting Time		Service Time
P_1	T_1	T'_1	R_1	W_1		S_1
P_2	T_2	T'_2	R_2	W_2		S_2
:	:	:	:	:		:
P_{i-1}	T_{i-1}	T'_{i-1}	R_{i-1}	W_{i-1}		S_{i-1}
P_i	T_i	T'_i	R_i	W_i		S_i
:	:	:	:	:		:
P_{L-1}	T_{L-1}	$\mathbf{T'_{L-1}}$	$\boldsymbol{R_{L-1}}$	W_{L-1}		S_{L-1}
P_L	T_L	**-1**	**-1**	**-1**		**-1**
:	:	:	:	:		:
:	:	:	:	:	L_j	:
:	:	:	:	:		:
P_{m-1}	T_{m-1}	**-1**	**-1**	**-1**		**-1**
P_m	T_m	$\mathbf{T'_m}$	$\boldsymbol{R_m}$	W_m		S_m
:	:	:	:	:		:
P_{F-1}	T_{F-1}	$\mathbf{T'_{F-1}}$	$\boldsymbol{R_{F-1}}$	W_{F-1}		S_{F-1}
P_F	T_F	**-1**	**-1**	**-1**	$L_{j=2}$	**-1**
P_{F+1}	T_{F+1}	**-1**	**-1**	**-1**		**-1**
P_{F+2}	T_{F+2}	$\mathbf{T'_{F+2}}$	$\boldsymbol{R_{F+2}}$	W_{F+2}		S_{F+2}
:	:	:	:	:		:
:	:	:	:	:		:
P_n	T_n	T'_n	R_n	W_n		S_n

Internal state parameters

Figure 4. Analyzing ETR data using SSTP-1.3. T_a and T_d are the time of arrival and the time of departure of the packet respectively. The value '-1' is for lost packets. L_j is calculated by SSTP-1.3 to record the number of losses each time a loss occurs to use in statistical analysis.

The algorithm scans the ETR data (the three leftmost columns in table 1) and compares the time of departure of packet P_{i-1} (denoted by T'_{i-1}) with the time of arrival of packet P_i (denoted by T_i) for each packet P_i where i ranges from 1 to the end of the data set n (see the cells marked by complete circles in figure 4). If the arrival time of a new packet P_i is larger than the time when the previous packet (P_{i-1}) departed, then the waiting time for packet P_i is zero seconds, and the service time is simply the response time.

However, if the time of arrival (T_i) of a new packet is smaller than the time of departure (T'_{i-1}) of the packet getting served, then the waiting time is the difference between the departure time of the packet in service (T'_{i-1}) and the arrival time of the packet waiting in the queue (T_i). In this case, the service time is the difference between the time when a new packet (P_i) departs and the time when the previous packet (P_{i-1}) departed (see cells marked by the thick edges square in figure 4). When the SSTP-1.3 detects that a packet (P_L) has not departed, it considers the departure time to be infinity and records the value '-1' as departure time in the output files. SSTP-1.3 counts the number of losses each time a loss occurs, and it also calculates the Sample space of the statistical set.

```
1   Loss_counter = 0
2   Last_before_loss = 0
3   Total_loss = 0
4   j = 0
5   for i = 1 to n
6    do
7     if T'i ≠ -1
8      then Ri = T'i - Ti
9       if Loss_counter = 0
10        then if I = 1 or Ti ≥ T'i-1
11              then Wi = 0
12                   Si = Ri
13             else if Ti < T'i-1
14              then Wi = T'i-1 - Ti
15                   Si = T'i - T'i-1
16        if Loss_counter > 0
17       then   Lj = Loss_counter
18              TLj = Ti-1
19              j = j + 1
20              Loss_counter = 0
21              if  i = 1 or {i > 1 and  Ti ≥ T'last_before_loss}
22              then Wi = 0
23                   Si = Ri
24             else if Ti < T'last_before_loss
25                then Wi = T'last_before_loss - Ti
26                     Si = T'i - T'last_before_loss
27    if T'i = -1
28      Ri = Wi = Si = -1
29      Total_loss = Total_loss + 1
30       if Loss_counter = 0
31         Last_before_loss = i - 1
32         Loss_counter = Loss_counter + 1
33       else if Loss_counter > 0
34         Loss_counter = Loss_counter + 1
35  Sample_space = n - Total_loss
```

Figure 5. Third version of Simple Service Time Producer (SSTP-1.3). used for calculating the response time (R_i , lines 8), the waiting time before entering service (W_i , lines 11, 14, 22, and 25), and the service time (S_i , lines 12, 15, 23, and 26) for each packet P_i in ETR data set.

The response time, waiting time, and service time for P_L will also be considered as infinite and denoted by the value '-1'. In cases of loss, the last packet that departed (P_{L-1}) is used by the algorithm for comparison with the arriving packet that departed just after loss (see cells marked by dotted circles in figure 4). This comparison is used to calculate the waiting time and service time for the newly departing packet after loss occurs (see lines 21-26 in figure 5). Using the algorithm, we calculate parameters from each of the ETRs. Our analysis of the different experiments gives the delay and the service time.

4.2. Results

The outcomes of the tests show that the assumption of single server, single FIFS queue holds true for downlink and uplink traffic tests. In bi-directional traffic tests, the first packet that enters the system will enter the server, but due to differences in transmission between Ethernet and WLAN, we discuss these results in our coming paper. For the WLAN APs we have tested, the delay on the uplink is always smaller than on the downlink (see section 4.2.1). Below, we discuss further results.

4.2.1 Directional Delay

We found out that for a given AP, the system needs less service time for an uplink packet than a packet of the same size but on the downlink (table 1). For example, analyzing a Lucent WavePOINT-II [8] AP shows that the average service time for an uplink packet of 32 byte UDP payload is 152µsec. While if an identical payload goes the opposite route, the average service time is 894µsec. This is because the service time definition includes the transmission time, and the overhead for the wireless transmission of the frame is much larger than that for transmission over Ethernet [10].

Table 1. Comparison between two access points: AP1 is Lucent WavePOINT-II, and AP2 is Lucent AP2000. Uplink in both APs has less service time than downlink. Comparing APs in both traffic directions proves AP1 to have lower service time than AP2

Payload (Bytes)	AP1 Average Service Time (µsec)		AP2 Average Service Time (µsec)	
	downlink	uplink	Downlink	uplink
32	894	152	999	171
64	918	190	1038	200
128	962	257	1140	274
256	1087	395	1374	419
512	1323	668	1820	733
1024	1750	1238	2673	1309
1472	2089	1705	3399	1876

Downlink and uplink average service times show that the server of AP1 is faster than that of AP2

The result is an increase in the service time of a packet. Table 2 shows the results of the average service times from the analysis of many ETR data sets for two Lucent WLAN APs: WavePOINT-II and AP2000 [8]. These results show that uplink service time is much smaller than downlink service time. Such an analysis can be used to compare the performance of different APs.

4.2.2 Service Time Formula

The key result of the work presented in this paper is a formula for the average service time of a WLAN AP. Our experiments have shown that the average service time for a packet is a linear function of payload. The discrete-event nature of the system allows us to look at the *service-time* values, in relationship to payload, as *terms of a sequence*. Let us denote the general term of this sequence as S_n, where n is the experimental number of the packet (P_{n-1} arrives before P_n) and is directly related to the payload. Since our test design uses a 32 byte increment in the UDP payload, then each experiment will have a UDP payload that is divisible by 32. Therefore, n is a positive integer. Our definition of the states of the system gives a service time that resembles the summation of three entities: the time required to check the frame headers, management time, and the time to transmit the bits of the packet [12]. So, the packet that enters or leaves the system is related to the frame headers it is encapsulated in. The management time is the time used by the AP to build its address tables of connected hosts plus other code management time. Since our test design uses two PCs only, we can consider the management time to be negligible compared to the service time and waiting time values. Moreover, the header checks are constant for all packets since all headers have a constant size. Consequently, the transmission time of the frame plays a significant role in deciding the per-packet service time. Since transmission speeds (bps) of Ethernet and IEEE 802.11 are constant, then the difference in transmission between one packet and another depends only on the payload, as long as the headers are of identical sizes. In our test design, we use a 32B payload-increment, thus the difference in the average service time of two consecutive packets is the time-difference to transmit 32 bytes with some little variations in time due to management time, which is negligible compared to transmission time. So, let S_1, S_2, S_3, …, S_{n-1}, S_n, be the terms of the sequence of average-service-time values. From the previous analysis, the difference between any two consecutive terms of the sequence is a time constant, which we denote by r. Hence,

$$S_k - S_{k-1} = r \quad \Leftrightarrow \quad S_k = S_{k-1} + r\ ; \quad k \in \mathbb{N}^*. \tag{2}$$

Thus, S_n is the general term of an arithmetic progression with common difference r and whose first term is S_o. Consequently, we have the service time formula in (3).

$$S_n = S_o + (n-1)r \tag{3}$$

where,

- n = (IP_Payload$_{\text{[in bytes]}}$ - 8B$_{\text{[UDP header]}}$) /32B
- S_n = service time (µs) for packet with IP payload of ($32n$+8)B
- S_o = service time (µs) for packet with 40B IP payload
- r = incremental difference in µs (calculated from linear regression of average service times of different payloads).

The value S_o in (3) can be calculated through numerical methods of averaging the service times calculated in the different ETRs of 40 byte IP payloads.

From (3) we can see that the average service time grows linearly; hence, we can use linear regression [13] of the average service times calculated by SSTP-1.3 to estimate the value of r. In fact, r is directly proportional to the slope of the average service time curve. S_o and r are different for different WLAN APs.

Using (3), the average service time for a downlink packet in a Lucent/ORINOCO WavePOINT-II AP [8] can be presented as:

$$S_n = 894 + (n\text{-}1)27 \quad \mu\text{s}. \qquad (4)$$

For a packet with IP payload 1032B, using (4) S_n will be around 1731 μsec with an error of 1.08%. Note that this average service time is no longer a stochastic process, but is deterministic if the packet sizes in a flow are known. This is interesting since many applications produce well-known packet sizes. Studies could be made to improve application performance by knowing the best packet size to use over a path where an AP exists.

The error for the values calculated by our equations and the experimental values range from zero to 3% of the maximum. Hence (3) has a maximum error of 3%. Thus, the manufacturer or the user of the access point has a good estimate of the service time per packet, a result that is valuable for modeling, simulation, and traffic shaping.

We are currently examining the QoS of multimedia streaming applications in the presence of WLAN APs. The codecs we use are layered MPEG-1, non-layered MPEG-1, and H.261 with QCIF resolution (corresponding to the resolution of many handheld devices). A future paper will compare these results.

5. CONCLUSION

The paper presented a mathematical model for WLAN APs. Our experiments showed that our assumption of a single server, single FIFS queue is correct. Our analysis of the delay showed that the time to serve a packet going from the wireless side to the Ethernet side is less than the time to serve a packet with identical payload but arriving from the Ethernet side and going to the wireless side. Using our model and analysis, we can compare performance of different brands of WLAN APs. The key result is that when using our model and our test design, one can get an analytic solution of the service time in terms of payload. The service time was analyzed and found to be a strictly increasing linear function of payload.

6. ACKNOWLEDGMENT

Special thanks to Daimler Sweden AB (www.daimler.se) for supporting us with different WLAN APs.

REFERENCES

[1] PCC (Personal Computing and Communication) project, a Swedish research program financed by the Foundation for Strategic Research.
http://www.pcc.lth.se/

[2] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, 2nd ed. (Singapore: McGraw-Hill, 1999).

[3] J. Banks, S. Carson, B. L. Nelson, and D. M. Nicol, *Discrete-Event System Simulation*, 3rd ed. (Upper Saddle River, NJ: Prentice-Hall, 2001).

[4] N. C. Hock, *Queuing Modeling Fundamentals* (West Sussex, England: John Wiley and Sons Ltd, 1998).

[5] I. Al Khatib, G. Q. Maguire Jr., R. Ayani, and D. Forsgren, Wireless LAN access points as queuing systems: performance analysis and service time, *MOBICOM'02 Conf.*, Atlanta, GA, USA, 2002, poster extended abstract.

[6] I. Al Khatib and R. Ayani, Wireless LAN access points: buffer size estimation, *Proc. MWCN'2002 4th IEEE Conf. on Mobile and Wireless Communications Networks*, Stockholm, Sweden, 2002, 656-660.

[7] MGEN 3.2 , traffic generator.
http://manimac.itd.nrl.navy.mil/MGEN/

[8] ORINOCO for Lucent WLAN APs.
http://www.orinocowireless.com/

[9] W. R. Stevens, *TCP/IP Illustrated*, *The protocols*, vol. 1 (One Jacob Way Reading, MA: Addison-Wesley, 1998).

[10] IEEE802.11 standard, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1997.

[11] IEEE802.11b standard, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band, 1999.

[12] J-C. Bolot, End-to-end delay and loss behavior in the Internet, *Proc. ACM SIGCOMM'93 Conf.*, San Francisco, CA, USA, 1992, 289-298.

[13] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers* (Singapore: McGraw-Hill, 1988).

Publication 5

Iyad Al Khatib and Rassul Ayani

Wireless LAN Access Points: Buffer Size Estimation

in Proceedings of the 4th IEEE Conference on Mobile and
Wireless Communications Networks (MWCN'02), pp. 656-660,
Stockholm, Sweden, 2002.

IEEE Catalog Number: 02EX614
ISBN: 0-7803-7605-6
Library of Congress: 2002108778

Also available in IEEE Catalog Number: 02EX614C
ISBN: 0-7803-7606-4
Paper reference: 110

Wireless LAN Access Points: Buffer Size Estimation

Iyad Al Khatib

Department of Microelectronics and Information Technology (IMIT), Royal Institute of Technology (KTH), Stockholm, Sweden
ikhatib@imit.kth.se

Rassul Ayani

Department of Microelectronics and Information Technology (IMIT), Royal Institute of Technology (KTH), Stockholm, Sweden
rassul@it.kth.se

***Abstract* - This paper utilizes a single server, single queue, FIFS system model to estimate the parameters describing the buffering occurring in wireless LAN access points of IEEE 802.11b. We use experimentation to obtain our model of the access point and buffer-related parameters. Using our test design, we are able to extract the parameters of an analytic equation giving the average service time of a packet as a function of packet IP payload. In this paper, we focus on buffer size estimation and adaptation. The major observation is that the buffer adapts its size to the different loads. Adaptation occurs at cut-off points of losses by increasing the buffer allocated size. We designed and implemented an algorithm, which checks when losses occur and calculates the number of packets in the buffer just before loss. A key result is that different access points have different initially allocated buffer sizes.**

Keywords: Wireless LAN, access point, IEEE 802.11b, performance, queuing system, buffer.

1. Introduction

Wireless LAN (WLAN) technology is spreading rapidly so that, today, users simply purchase a WLAN access point (AP), deploy it where they want, and connect themselves and perhaps their neighbors to the Internet. In addition, many schools and commercial sites have installed WLAN networks to support their employees, students or customers. This situation has created a change in the way computer users prefer to be connected. This increasing demand for WLANs is often coupled with a high load on the WLAN access points that connects the wireless network to the wired network. Although ad-hoc networking is an option for WLAN connections, today most WLANs use WLAN APs to connect multiple users to a wired backbone network [1]. In addition to these trends, there is also an increasing demand for better QoS over WLANs in general. To address these needs, a better understanding of WLAN APs is essential. This paper builds on our previous results, which modeled a WLAN access point as a queuing system [2]. The first step is to define the system of interest [3]. Based on our experiments and previous study, we model the wireless access point as a queuing system.

This paper utilizes a single server, single queue, FIFS system model to estimate the parameters describing the buffering occurring in an access point. Based on a set of traffic measurements we are able to extract the parameters of the model. Buffer size is thought to be a very important parameter that has direct effect on the performance of wireless access points, however, we are not aware of a careful analysis of buffer size in current WLAN access points. In order to determine buffer size we construct a set of experiments that purposely try to cause packet loss through the lack of buffer capacity. Based on these measurement we can estimate the size of the initially allocated buffer in bytes.

Below, we present results on buffer size and adaptation of the WLAN access point. The major observation is that the buffer adapts its size to the different loads. Adaptation occurs at cut-off points of losses by increasing the buffer allocated size. We designed and implemented the Buffer Size Estimator (BSE) algorithm that detects when a packet is lost and makes use of another algorithm (Simple Service Time Producer) for extracting some parameters needed for buffer calculation. A key result is that different access points have different initially allocated buffer sizes, which lead to different initial losses and different adaptation thereupon. Therefore the QoS, especially from the packet loss point of view, of an AP will not only be affected by the offered load, but also by the ability of the AP to adapt to the load, hence the initial buffer size. The buffer study on a particular access point can be used as a test to determine whether an AP is more suitable for certain applications or specific network loads than another AP.

2. AP Queuing System

In our investigation, we seek to estimate the buffer size of the model. In order to do so, we need to find the parameters of the queuing model of the AP. To complete the theoretical model, a set of assumptions have to be made. We isolate the wireless access point and define the different events that occur: arrival and departure (Fig. 1). When a packet enters the system the parameter of interest from the arrival event is the time of arrival. Similarly, when a packet departs, we are interested in the time of departure. We relate everything in the study from the point of view of the access point. Thus, we define the total delay of a packet, which we refer to as the *response tim*e, to be the time difference between the departure time and the arrival time of a packet. Time is an important factor in the study of our system; therefore, it is crucial to state whether the system is discrete or continuous.

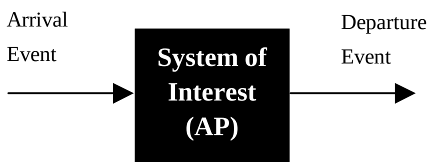

Fig. 1. System of interest isolated and events acting on it are marked; logical model. AP stands for Access Point.

Since the number of packets inside the system changes when a packet arrives or when a packet departs, i.e. at separate points in time, then we are dealing with a discrete system [4].

Since the wireless access point can forward traffic in two directions, then we will consider two cases: one where the traffic travels from the Ethernet side to the WLAN side, and another from WLAN to Ethernet. Modeling the WLAN to WLAN case is also of interest, but not presented in this paper. Therefore, we define two traffic flows: *downlink traffic* from Ethernet to WLAN and *uplink traffic* from WLAN to Ethernet.

After having defined the system and the events acting on it, we assume that the system can be modeled as a queuing system with a queue and a server or multiple queues and servers. The arrival and departure times recorded from experiments have shown that the system can be modeled as a single server system with one queue as shown in Fig. 2.

Hence we add one more event to the previously defined two: the event of entering the server (Fig. 3). Focusing on modeling the delay, we next define the two system states: *waiting* and *servic*e [5].

We conducted a series of experiments to measure: the *response time* of the system (as defined earlier), the waiting time in the queue, and the service time. However, there is a physical constraint that prevents direct measurement of the waiting time and service time variables, because we can only easily measure the arrival time and the departure time of a packet. To solve this problem, we designed tests that are described in section 4.

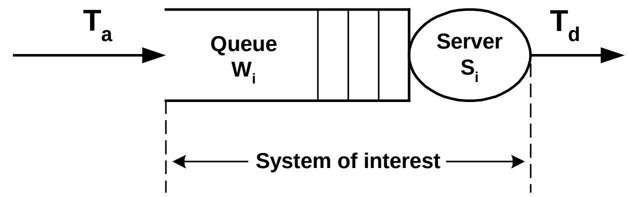

Fig. 2. Detailed view of the model; T_a is the arrival time of the packet and T_d is the departure time of the packet. W_i and S_i are the waiting time and the service time of packet P_i , respectively.

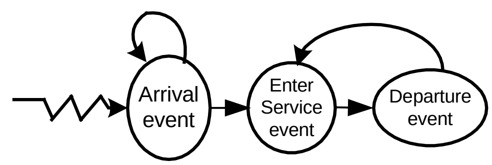

Fig. 3. Event graph for the system of interest with “enter service” event.

3. Experimental Environment

The experimental environment consists of four main entities: a traffic source (PC1/PC2), an access point, a receiver (PC2/PC1), and a traffic sniffer as shown in Fig. 4. The sender, receiver, and traffic sniffer are three PCs running Linux-2.2.16. The programs used were: tcpdump for reading network traffic and MGEN 3.2 [6] for generating UDP streams. We used one-second based experiments, hence the clock drift wouldn't correlate with subsequent measurements. MGEN was only used for sending UDP packets as we used our own program modules to filter packets from the tcpdump and analyze the results. These modules are built using the C++ language.

Since we have two traffic streams, the sender and receiver roles alternate between PC1 and PC2. In both cases, MGEN 3.2 was used to send traffic. Because these traffic streams are between two different media (IEEE802.11 and Ethernet), careful attention should be paid for the overheads of the link layers involved, namely Ethernet and IEEE 802.11b headers and trailers, as well as preambles and Interframe spaces.

The testbed was set up in an environment where no signals from other base-stations were present. We used the ORINOCO Client Manager [7] to check for radio signals.

4. Experiments and Results

In the following section we describe the experimental test design used to extract the parameters of the model and buffering. These experiments and analysis expose some of the important

QoS parameters, especially with regard to delay, packet loss and buffer size.

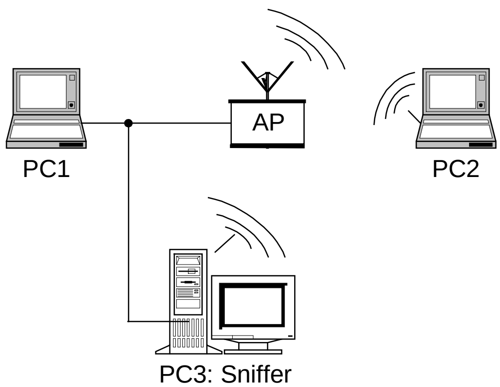

Fig. 4. The testbed, showing the downlink and uplink directions. Downlink traffic resembles packets travelling from the Ethernet side (PC1) to the IEEE802.11 side (to PC2). Uplink traffic resembles packets travelling from the IEEE802.11 side (PC2) to the Ethernet side (to PC1).

4.1. Test Design

There are two main classes of tests: downlink traffic tests, and uplink traffic tests. In the downlink tests class, PC1 acts as sender, and PC2 acts as receiver. In the uplink tests class, PC2 acts as sender, and PC1 acts as receiver. Each of the aforementioned classes is composed of identical sets, thus, describing one set is sufficient. The set of tests is made of clusters of experiments. We call the experiment in a cluster an *experimental test run* (ETR). Therefore, the ETR is the basic unit of the test design. In each ETR, we transmit a stream of UDP datagrams from the sender to the receiver and vary the packet size, by increasing the payload of the UDP datagram by 32 bytes. The maximum number of bytes we use as payload is 1472, because sizes beyond this may result in fragmentation [8]. It is very important to note that the headers and interframe spaces should be considered when transmitting, because there lies a major difference between the Ethernet side and the IEEE802.11 side. On the Ethernet side the preamble is 8 bytes, and the interframe space is 9.6 microsec [8]. While on the IEEE802.11 side, there are different scenarios, whose description [9] is beyond the scope of this paper. However, our experiments are controlled so that the link layer interference is described by only four link layer parameters [10]:

1) *DIFS*, which is the Distributed Coordinated Function Interframe Space, and it is 50 µsec per frame in our ETR.

2) *Preamble and PLCP (Physical Layer Convergence Protocol) headers*, which is 272 bits per frame in our ETR.

3) *SIFS*, which is the Short Interframe Space used for special acknowledgments, and it adds 10 µsec per frame in our ETR.

4) *ACK***,** which is the link layer Acknowledgment and it adds 304 bits per frame in our ETR.

Each ETR is repeated at least three times, and the ETR data are compared. The cluster of ETRs is built by utilizing different data rates of the available bandwidth. In each ETR, the tcpdump program is run on PC3 to record the traffic on both sides. In addition, in each ETR one of our modules is used to extract the arrival time (denoted as T) and departure time (denoted as T') for each packet. To solve the problem of having information about external event parameters only, we designed an algorithm that looks at the ETR data set and analyzes it to extract the values of the parameters we want. The result is a file similar to the data shown in Table 1. We call the algorithm SSTP, or the Simple Service Time Producer. Figure 5 shows the second version of the algorithm (SSTP-1.2), which is modified to encounter packet loss used in the buffer investigation. The service time for our system includes the time required to check the packet headers, management time, and the time spent transmitting the bits of the packet [11]. Thus,

$$R_i = W_i + S_i. \tag{1}$$

TABLE 1
DATA FILE AFTER ANALYZING ETR DATA USING SSTP-1.2. T_A AND T_D ARE THE TIME OF ARRIVAL AND THE TIME OF DEPARTURE OF THE PACKET RESPECTIVELY. THE VALUE '-1' IS FOR LOST PACKETS

Packet Number	T_a	T_d	Response Time	Waiting Time	Service Time
P_1	T_1	T'_1	R_1	W_1	S_1
P_2	T_2	T'_2	R_2	W_2	S_2
:	:	:	:	:	:
P_{i-1}	T_{i-1}	T'_{i-1}	R_{i-1}	W_{i-1}	S_{i-1}
P_i	T_i	T'_i	R_i	W_i	S_i
:	:	:	:	:	:
P_{L-1}	T_{L-1}	**T'_{L-1}**	***R_{L-1}***	W_{L-1}	S_{L-1}
P_L	T_L	**-1**	**-1**	**-1**	**-1**
:	:	:	:	:	:
:	:	:	:	:	:
P_n	T_n	T'_n	R_n	W_n	S_n

ETR data set

The SSTP-1.2 is needed to be run on the ETR data set before the buffer estimation algorithm since we need the values of the average service times. The SSTP-1.2 (Fig 5) looks at data in the three leftmost columns in Table 1 (the ETR data set), and compares the time of departure of packet P_{i-1} (T'_{i-1}) with the time of arrival of packet P_i (T_i) for each i ranging from 1 to the end of the data set n (see the cells marked by full circles in Table 1). If the arrival time of a new packet P_i is larger than the time when the previous packet departed, then the waiting time for packet P_i is zero seconds, and the service time is simply the response time. However, if the time of arrival of a new packet is smaller than the time of departure of the packet getting served then the waiting time is the difference between the departure time of the packet in server and the arrival time of the packet in queue. In this case, the service time is the difference between when the new packet departs and the time when the previous packet departed (see cells marked by the thick edges square in Table 1).

```
1   Loss_counter = 0
2   Last_before_loss = 0
3   for i = 1 to n
4    do
5     if  T'_i ≠ -1
6       then R_i = T'_i - T_i
7         if Loss_counter = 0
8           then if i = 1 or T_i ≥ T'_{i-1}
9                then W_i = 0
10                    S_i = R_i
11              else if T_i < T'_{i-1}
12                then W_i = T'_{i-1} - T_i
13                     S_i = T'_i - T'_{i-1}
14         if Loss_counter > 0
15         then  Loss_counter = 0
16              if  i = 1 or {i > 1 and  T_i ≥ T'_{last_before_loss}}
17                then W_i = 0
18                     S_i = R_i
19              else if T_i < T'_{last_before_loss}
20                  then W_i = T'_{last_before_loss} - T_i
21                       S_i = T'_i - T'_{last_before_loss}
22    if  T'_i = -1
23       R_i = W_i = S_i = -1
24         if Loss_counter = 0
25           Last_before_loss = i - 1
26           Loss_counter = Loss_counter + 1
27         else if Loss_counter > 0
28           Loss_counter = Loss_counter + 1
```

Fig. 5. Simple Service Time Producer version 2 (SSTP-1.2) modified from SSTP to suit the BSE algorithm. SSTP-1.2 is used for calculating the response time (R_i , lines 6), the waiting time before entering service (W_i , lines 9, 12, 17, and 20), and the service time (S_i , lines 10, 13, 18, and 21) for each packet P_i in ETR data set.

In case of packet loss (see the cells with value of -1, marked by a triangle), the SSTP-1.2 checks whether the loss has just happened. If there is no loss before this point, then the previous packet is considered as the last packet that departed before loss occurred. Then, the SSTP-1.2 stores the corresponding delay parameters (see dotted circles in Table 1), otherwise, the algorithm keeps reading lost packets and assigning the value -1 to the response time, waiting time, and service time.

The BSE algorithm uses the response time, and the average service time calculated by the SSTP-1.2. This information will be used to estimate the initial buffer size. To do so, we check for packet loss in the data sets, and when packet loss is detected, we use our BSE algorithm. This algorithm uses the waiting time and service time of the last available packet in queue to estimate the buffer size. So, if the L^{th} packet was lost, BSE checks if packet P_{L-1} has departed, and so on until a packet proves to have departed from the system before packet P_L. As the packets in our experimental test runs are identical, the waiting time, w, of the packet proving to have last departed before loss is directly related to the number of packets in queue multiplied by the service time of a packet:

$$w[P_j] = \text{Number of packets in queue} * S, \qquad (2)$$

where, S is the average service time of a packet used in the experiment;
P_j is the packet that last departed before loss, so j is a positive integer smaller than L.

Since the average service time is known through the averaging the values of the service times calculated by the SSTP-1.2 of the different ETRs. From Eq. (2) we can conclude that the number of packets enqueued will reach its first maximum (relative to the first allocated buffer size) if packets after the j^{th} packet were lost. Thus, substituting $w[P_j]$ and S (from the results of the SSTP algorithm) in Eq. (2) leads to an estimate of the maximum number of packets in the first allocated buffer. As the size of these packets is known, we can now calculate the buffer size as the maximum number of packets in buffer multiplied by the packet size used in the ETR. For every data set we run the BSE calculation each time a loss is detected. These results are analyzed statistically (to avoid measurement errors), and an estimate of the buffer first allocated size is computed [12]. Hence users and AP manufacturers can determine the size of the buffer in an AP along their network path relative to the offered load.

4.2. Results

The first set of results are those calculated by SSTP-1.2 and are needed for the BSE. In Table 2, we show the average service time values on the downlink and the uplink for two different APs to compare the server delay in both, and to use the average service time values in Eq. (2) for buffer estimation. The results prove that AP2 has a faster server on the downlink than AP1.

TABLE 2
COMPARISON BETWEEN TWO ACCESS POINTS: AP1 IS LUCENT WAVEPOINT-II, AND AP2 IS LUCENT AP500. UPLINK IN BOTH APS SHOWS TO HAVE LESS SERVICE TIME THAN DOWNLINK. COMPARING APS IN DOWNLINK TRAFFIC DIRECTION PROVES AP2 TO HAVE LOWER SERVICE TIME THAN AP1, BUT ON THE UPLINK DIRECTION AP1 PROVES TO HAVE A FASTER SERVER THAN AP2

Payload (Bytes)	AP1 Average Service Time (μsec)		AP2 Average Service Time (μsec)	
	downlink	Uplink	downlink	uplink
40	894	152	762	343
72	918	190	785	383
136	962	257	836	463
264	1087	395	953	618
520	1323	668	1110	915
1032	1750	1238	1599	1506
1480	2089	1705	1999	1996

In all experiments (on all access points studied), the BSE has shown that the buffer size is adaptive to offered load. As the load increases, the buffer allocated size increases. However, the first loss for a specific access point is always the same if the offered load was the same. Moreover, there was no packet loss detected on the uplink. Access points proved faster in serving on the uplink. This is due to the reason that the uplink service time is relatively small so packets will go out of the system before the buffer fills up. However, on the downlink, the buffer filled up since packets had to wait more in the queue while other packets were getting served. Table 3 shows the results of the buffer size when packet loss was encountered. The values presented are the results of many trials, and the error calculated is less than 2%. When there is no loss, we can not get any value of the buffer size, which means that the available buffer is suitable, and that was the case on the uplink of all the APs that were investigated.

TABLE 3
BUFFER SIZE COMPARISON BETWEEN TWO ACCESS POINTS: AP1 IS LUCENT WAVEPOINTII, AND AP2 IS LUCENT AP500. UPLINK IN BOTH APS SHOWS TO HAVE NO PACKET LOSS. COMPARING APS SHOWS AP1 TO HAVE A HIGHER FIRST BUFFER ALLOCATION THAN AP2

IP Payload (Bytes)	AP1 First Allocated Buffer Size (KBytes)		AP2 First Allocated Buffer size (KBytes)	
	downlink	Uplink	downlink	uplink
40	20	*No loss*	11	*No loss*
72	34	*No loss*	16	*No loss*
136	72	*No loss*	28	*No loss*
264	*No loss*	*No loss*	32	*No loss*
520	*No loss*	*No loss*	48	*No loss*
1032	*No loss*	*No loss*	77	*No loss*
1480	*No loss*	*No loss*	114	*No loss*

We observe from Table 3 that AP1 has a higher first allocation of buffer size than AP2 for the different loads utilizing the full bandwidth (which in our case was 10Mbps). Though AP2 has a lower downlink service time than AP1, if the application QoS is sensitive to packet loss, then AP1 proves to be more suitable than AP2. However, if an application is sensitive to delay, then AP2 would be more suitable. Currently we are working on refining this analysis as we aim to simulate WLAN APs using our model and parameters derived from actual APs for IEEE 802.11b and IEE 802.11a.

5. CONCLUSION

The main purpose of the paper was to estimate the initial buffer size that is allocated by WLAN APs. This paper discussed our test design for extracting the parameters of the model. Our analysis on the buffer size revealed that though an AP may have a faster server, the buffer allocation scheme used my affect the QoS from the packet loss point of view. The major result is that when using our model, test design, and algorithms, one can get a good estimate of performance parameters related to delay and packet loss, and the choice of a suitable AP can be made accordingly.

ACKNOWLEDGMENTS

We thank professor Gerald Q. Maguire Jr., at the Department of Microelectronics and IT (IMIT), Royal Institute of Technology (KTH), Stockholm, Sweden, for the fruitful discussions.

REFERENCES

[1] PCC (Personal Computing and Communication) project, a Swedish research program financed by the Foundation for Strategic Research. http://www.pcc.lth.se/
[2] I. Al Khatib, G. Q. Maguire Jr., R. Ayani, and D. Forsgren, "Wireless LAN Access Points as Queuing systems: Performance Analysis and Service Time," to appear as a SigMobile's Mobile Computing and Communications Review publication, 2002.
[3] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, 3rd ed., McGraw-Hill, 1999.
[4] J. Banks, S. Carson, B. L. Nelson, and D. M. Nicol, *Discrete-Event System Simulation*, 3rd ed., copyright Prentice-Hall, 2001.
[5] N. C. Hock, *Queuing Modeling Fundamentals*, John Wiley and Son, 1998.
[6] MGEN 3.2., traffic generator. http://manimac.itd.nrl.navy.mil/MGEN/.
[7] ORINOCO for Lucent wireless LAN access points. http://www.orinocowireless.com/.
[8] W. R. Stevens, *TCP/IP Illustrated, The protocols*, vol. 1, Addison- Wesley, 1998.
[9] IEEE802.11 standard, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1997.
[10] IEEE802.11b standard, (Supplement to ANSI/IEEE Std. 802.11, 1999 Edition), Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band, 1999.
[11] J-C. Bolot, "End-to-End Delay and Loss Behavior in the Internet ," Proceeding of ACM SIGCOMM'93, pp.289-298, 1992,
[12] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers,* McGraw-Hill, 1988.

Publication 6

Iyad Al Khatib, Rassul Ayani, and Gerald Q. Maguire Jr.

Wireless LAN Access Points Uplink and Downlink Delays: Packet Service-Time Comparison

in Proceedings of The 16th Nordic Teletraffic Seminar (NTS-16) Conference, pp. 253-264, Espoo, Finland, August 21-23, 2002.

ISSN: 1458-0322

Wireless LAN Access Points Uplink and Downlink Delays: Packet Service-Time Comparison

Iyad Al Khatib Rassul Ayani Gerald Q. Maguire Jr.
{ikhatib, rassul, maguire}@it.kth.se
Department of Microelectronics and Information Technology (IMIT)
Royal Institute of Technology (KTH), Stockholm, Sweden

Abstract

Wireless LAN access points are very important connecting nodes transferring traffic between two media in opposite directions. Hence the performance of the wireless LAN access point should be looked upon from two different reference points: uplink (from WLAN to Ethernet) and Downlink (from Ethernet to WLAN). This paper builds on our previous modeling of the wireless access point as a single server, FIFO, queuing system to analyze the service times in both directions. The previous analysis showed that the average service time is a function of payload. Measurements have revealed that the uplink service time is much smaller than the downlink service time for the same payload. In this paper, we investigate the absolute value of the difference between the uplink and downlink service-times. We refer to the absolute value of the difference in time between uplink and downlink as the UDC, or the "Uplink-Downlink Contrast". Results show that as the packet size increases, the UDC either decreases or increases monotonically depending on the brand of the access point. For a decreasing UDC, the absolute value of the difference between the uplink and downlink service-times decreases, hence the UDC is convergent. Similarly, the UDC is divergent if it increases with increasing packet size. These results can be used to select a WLAN access point given the size of packets transmitted by an application or multiple applications over a Local Area Network.

1. INTRODUCTION

Wireless Local Area Networks (WLANs) have garnered wide spread interest among users, and this interest is coupled with an increasing demand for better quality of service. Predictions are that more IEEE 802.11 wireless access points will be attached to LANs [1]. Analysis of the performance of wireless LAN access points is crucial since access points are the entities, which connect the mobile nodes to the wired backbone of a LAN. The access points we will consider connect two different media (Ethernet and IEEE 802.11 [2]). Since these *media access and control protocols* are quite different, it is important to look at the difference in behavior of the access point when traffic travels from the wired medium (Ethernet) to the wireless medium (WLAN) and when traffic travels from the wireless medium to the Ethernet backbone. We refer to these two directions for traffic flow through the access point as: *uplink* (from WLAN to Ethernet), and *downlink* (from Ethernet to WLAN).

An important QoS parameter for the analysis of a communication system is *delay* [3]. In this paper, the delay of a packet travelling through the wireless access point is investigated for both directions: the downlink and the uplink traffic-flows. Here we model the access point as a queuing system and analyze its uplink and downlink delays. The advantages of this model and the analysis presented in this paper range from the ability to compare performance of different access points, to the direct usage of the results by applications in order to enhance the perceived quality of service when an access point is deployed between two peers. Moreover, using our model and analysis helps in selecting a suitable access point given the sizes of packets transmitted by the application.

In this paper, we present results on the difference in service-time values between the uplink and the downlink for a WLAN access point. The major result is that the difference between the directional service-times varies from one type of access point to another. In fact, we have observed that the absolute value of the difference in service-times between the uplink and the downlink either increases or decreases with an increasing payload, depending on the brand of the access point. This is an interesting observation, because one would expect the absolute difference between the two directional service-times, for a given access point, to always decrease with an increase in payload, since the transmission-time increases. However, the results of our tests indicate that this is not true for all access point brands. The reasons behind such behavior are the subject of ongoing investigation. Finally, the observations led us to introduce a simple characterization of the access point based on the absolute value of the difference between the uplink and downlink service-times or the "Uplink-Downlink Contrast", which is presented below.

2. BACKGROUND

In this section we describe our previous work on modeling wireless LAN access points [4]. The model provides a basis for the analysis of directional delay over wireless access points for IEEE 802.11b.

Our previous investigation showed that wireless access points of IEEE 802.11b can be represented as a queuing system. We used experimentation to determine the relevant parameters of our model. In our experiments, we sent UDP packets through the wireless access points and analyzed the results. Each measurement incremented the UDP payload by 32 bytes. The minimum UDP payload we used was 32 bytes and the maximum was 1472 bytes, because values beyond this may lead to fragmentation [5]. We used UDP datagrarms, which enabled us to have more control over the tests, because UDP does not require any traffic flow in the reverse direction [5].

In the rest of this paper, we will use the acronym AP for access point. The time unit we use in our analysis of delay is microseconds (denoted as μsec).

2.1. Queuing Model

Our previous study of wireless APs has shown that a WLAN access point can be modeled as a single server, single queue, FIFO system, as shown in figure 1. The parameters of interest are time-related and are classified into two types: external and internal parameters. The external parameters are those, which can be calculated from events that take place outside the boundaries of the access point, for example, the arrival time (T_a) and the departure time (T_d). On the other hand, internal parameters are related to events that take place inside the access point [6]. For example, the service time (S_i) and the waiting time (W_i) of packet (P_i) are the two internal parameters that we seek in modeling the AP as a queuing system [7]. In this paper, we focus on delay.

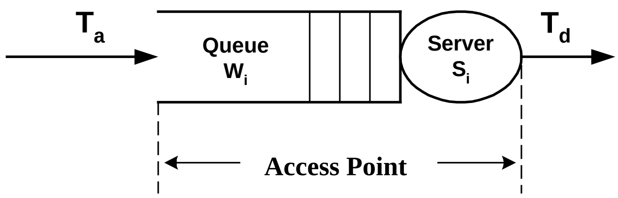

Figure 1. Queuing model of wireless access point of IEEE 802.11b. T_a is the arrival time of the packet, and T_d is the departure time of the packet. W_i and S_i are the waiting time and the service time of packet P_i , respectively.

We refer to the total delay for packet P_i as the *response time* (R_i) [6], and we define it in equation (1) as the sum of the waiting time of P_i in the queue and the service time of P_i:

$$R_i = W_i + S_i \,, \quad i \in \mathbb{N}. \tag{1}$$

It is worth mentioning that the service time in our model includes the time to check the headers, management time, and the transmission time of the frame.

2.2. Service-Time Analytic Solution

A key result of our previous work was an analytic solution for the service time of a packet in relation to the packet size. The average service time was analyzed and found to be a linear function of the payload. Since the number of packets in the system (AP) changes when a packet arrives or when a packet departs i.e., at different points in time, then the system is a discrete event system [8]. Hence, starting with a packet carrying 32 bytes of UDP payload and using the 32-byte payload-increment in our previous experiments, we were able to define the service time S_n [4] as:

$$S_n = S_o + (n-1).r \quad (2)$$

where, n = UDP_payload (in bytes)/32B = (IP_payload - 8)/32
S_n = service time (μsec) for a packet with IP payload of $(32.n + 8)$ bytes,
S_o = service time (μsec) for a packet with 32B UDP payload (i.e., 40B IP payload),
r = incremental difference in μsec.

S_o is calculated by averaging the service times calculated in the different experiments, and r is calculated using linear regression of service times of different packet sizes [9]. S_o and r are, in fact, the characteristic parameters of the access point i.e., they vary from one type of access point to another.

2.3. Directional Service-Time Delay

Our experiments have shown that for a given access point, α, the Downlink Service-Time, DST(α, x), is larger than the Uplink Service-Time, UST(α, x), for the same IP payload of x bytes. This result is illustrated in table 1, where we studied the service time for two AP brands: AP1 is Lucent WavePoint II, and AP2 is Lucent AP2000 [10]. One of the advantages of this study is that when using our model and analysis, one can compare the performance of different wireless LAN access points. However, in this paper, we focus on the comparison between the uplink and downlink average service time for the same access point.

TABLE 1
COMPARISON BETWEEN UPLINK AND DOWNLINK SERVICE TIME VALUES OF TWO ACCESS POINT BRANDS: AP1 IS LUCENT WAVEPOINT II, AND AP2 IS LUCENT AP2000. UPLINK IN BOTH APS SHOWS TO HAVE LESS SERVICE TIME THAN THE DOWNLINK.

IP Payload x (Bytes)	AP1 Average Service-Time (μsec)		AP2 Average Service-Time (μsec)	
	UST(*AP1, x*)	DST(*AP1 ,x*)	UST(*AP2, x*)	DST(*AP2 ,x*)
40	152	894	171	999
72	190	918	200	1038
136	257	962	274	1140
264	395	1087	419	1374
520	668	1323	733	1820
1032	1238	1750	1309	2673
1480	1705	2089	1876	3399

3. ANALYSIS OF DIRECTIONAL SERVICE-TIME VALUES

In this section, we investigate the absolute value of the difference between the uplink and downlink service times (as defined in our model) of wireless LAN access points. In section 3.1, we present the absolute value of the difference between uplink and downlink service-times as the "Uplink-Downlink Contrast" (UDC) of an access point. In sections 3.2 and 3.3 we define the convergent and the divergent characteristics of the UDC, respectively.

3.1. Uplink-Downlink Contrast

For our analysis, we introduce the notion of the *Uplink-Downlink Contrast* (UDC) of a wireless LAN access point, which is the absolute value of the difference between the uplink and downlink service-times in relation to packet size. The UDC of a packet with a given payload of *x* bytes is defined as:

$$\mathrm{UDC}(\boldsymbol{\alpha}, \boldsymbol{x}) = \left| \mathrm{UST}(\boldsymbol{\alpha}, \boldsymbol{x}) - \mathrm{DST}(\boldsymbol{\alpha}, \boldsymbol{x}) \right| \qquad (3)$$

where,

α is the brand of the access point,

$\mathrm{UST}(\boldsymbol{\alpha}, \boldsymbol{x})$ is the Uplink Service-Time of a packet with payload *x* bytes, for AP "α", and

$\mathrm{DST}(\boldsymbol{\alpha}, \boldsymbol{x})$ is the Downlink Service-Time of a packet with payload *x* bytes, for AP "α".

Using (3) we can calculate the absolute value of the difference between the uplink and downlink service-times of different access point brands. Table 2 shows the *Uplink-Downlink Contrast* values for the access points given in table 1.

TABLE 2
UPLINK-DOWNLINK CONTRAST (UDC) OF TWO WIRELESS LAN ACCESS POINT BRANDS: AP1 IS LUCENT WAVEPOINT II, AND AP2 IS LUCENT AP2000.
UDC OF AP1 *DECREASES* WITH INCREASING PAYLOAD.
UDC OF AP2 *INCREASES* WITH INCREASING PAYLOAD.

IP Payload *x* (Bytes)	UDC(*AP1*, *x*) (μsec)	UDC(*AP2*, *x*) (μsec)
40	742	828
72	728	838
136	705	866
264	692	955
520	655	1087
1032	512	1364
1480	384	1523

Looking at the UDC values in table 2, we observe that as the payload increases, the UDC values of AP1 decrease, while the UDC values of AP2 increase. The reasons for these differences in behavior between access points are the subject of future investigation. In this particular case, the radicalization is that AP2 spends more time checking the packet and deciding on the speed to be used for transmission than AP1, which leads to a higher cost in time as the packet size increases [11], [12]. In fact, the increase in UDC for AP2 is mainly due to the relatively large increase in the downlink service time as the payload increases (see the rightmost column in table 1). On the other hand, the increase in the uplink service time values for AP2 is relatively not as large as the increase on the downlink. One reason is that the Ethernet overhead time is more stable than the IEEE 802.11b overhead time for many access points due to the different checks that some access points make before transmission. In fact, there are different implementations of the IEEE 802.11b scenarios for checking headers and for radio transmissions for different WLAN access points [13], [14]. Hence, we look at these differences between APs from the point of view of their effects on performance, and we characterize a UDC as either convergent or divergent.

3.2. Convergent UDC

Definition: A WLAN access point, α, is said to have a *convergent UDC* if and only if:

$$\textit{as } \boldsymbol{x} \textit{ increases, } UDC(\boldsymbol{\alpha}, \boldsymbol{x}) \textit{ decreases, where } x \textit{ is the payload in bytes.} \tag{4}$$

For a convergent UDC the downlink and uplink service time values grow closer to each other as the payload increases. This is illustrated in figure 2, where the uplink and downlink service time values for AP1 (Lucent WavePoint II) are plotted.

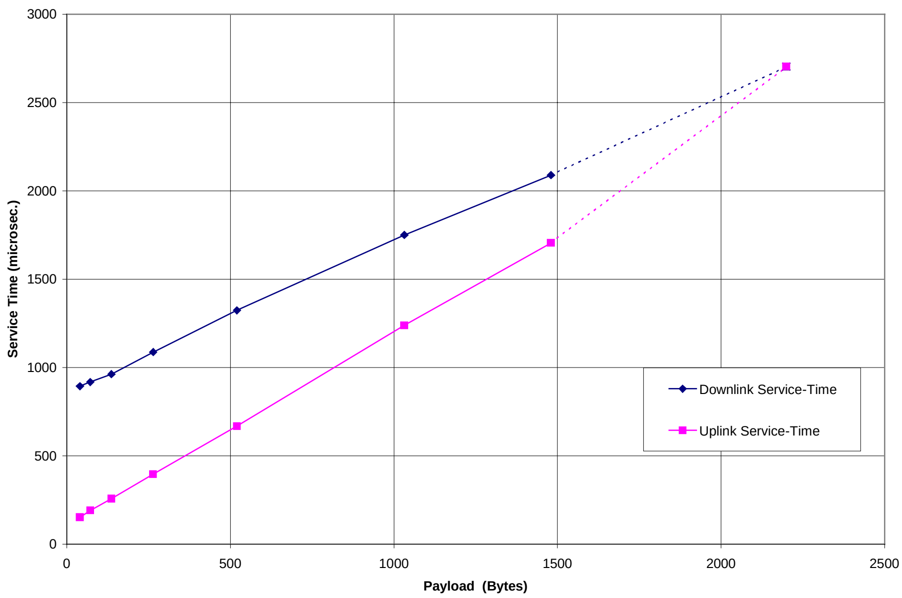

Figure 2. DST and UST of AP1 (Lucent WavePoint II). Stretched curves converge to a point, where the DST and UST are identical (corresponding to a payload value of around 2200 bytes). The solid lines are the measured values, while the dotted lines are theoretical extensions. The point of convergence lies beyond the realistic limit for the MTU used in Ethernet and IEEE 802.11b (1500 bytes).

The Uplink and Downlink Service time plots in figure 2 show that as the payload increases, there is a point of convergence, where the downlink service-time would be identical to the uplink service-time. Beyond this point, the uplink service-time is larger than the downlink service-time. However, the point of convergence (around 2200 bytes) is beyond the realistic limits of the standard maximum transmission unit (MTU) for Ethernet [5] and IEEE 802.11 [13]. So, in reality this point of performance where the uplink and downlink service-times are equal will not be reached. These extensions of the curves (dotted lines) illustrate the convergence of the service times of a wireless AP. The linear behavior of the service time is not surprising since it has been shown in equation (2) that the average service time is a linear function of the payload. An interesting characteristic to look at is the rate of convergence of uplink and downlink service times, which can be illustrated in the UDC-vs-payload plot, as shown in figure 3. For a decreasing UDC curve, the UDC is convergent since this satisfies definition (4). In figure 3, one can easily notice a decreasing UDC for AP1, which means that AP1 has a convergent UDC. The Convergent-UDC curve can be extended, and the point of intersection with the horizontal axis (Payload) will be the point where the uplink and downlink service-times are identical for the given AP. In our analysis, we use the UDC plot as a characteristic curve of the wireless LAN access point. The dotted line is a theoretical extension, using the service time formula in equation (2) for AP1, to illustrate the intersection with the horizontal axis.

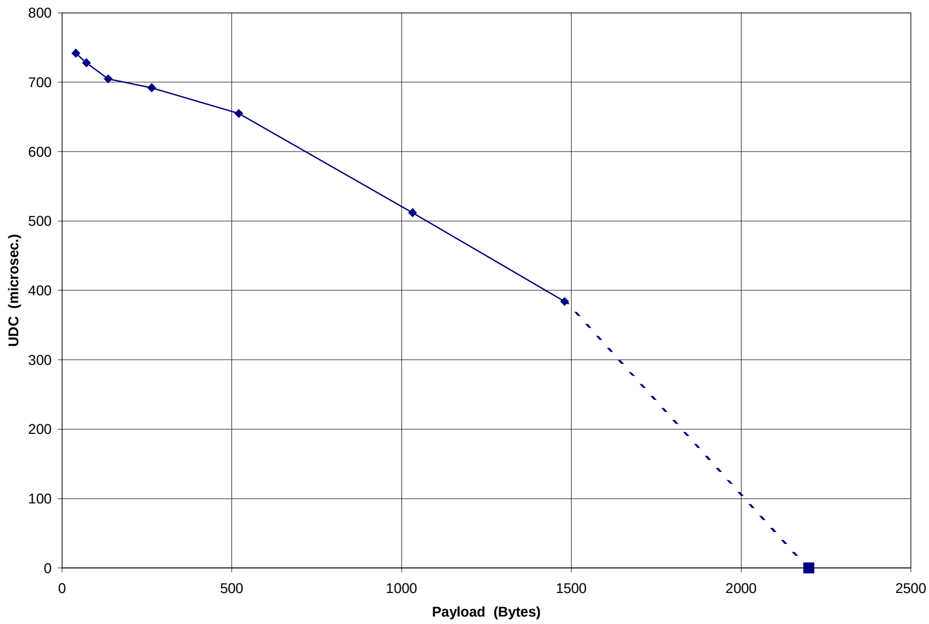

Figure 3. UDC plot for AP1 (Lucent WavePoint II). The UDC is decreasing with increasing payload. AP1 has a convergent UDC. The solid line is the real curve. The dotted line is a theoretical extension

3.3. Divergent UDC

The notion of a divergent UDC is the opposite of a convergent UDC.

Definition: A WLAN access point, α, is said to have a *divergent UDC* if and only if:

as $\mathbf{x}$ *increases,* $UDC(\boldsymbol{\alpha}, \mathbf{x})$ *increases, where x is the payload in bytes.* (5)

When an AP has a divergent UDC, the uplink and downlink service-time values diverge from each other as the payload increases. This is illustrated in figure 4, where the uplink and downlink service-time values for AP2 (Lucent AP2000) are plotted.

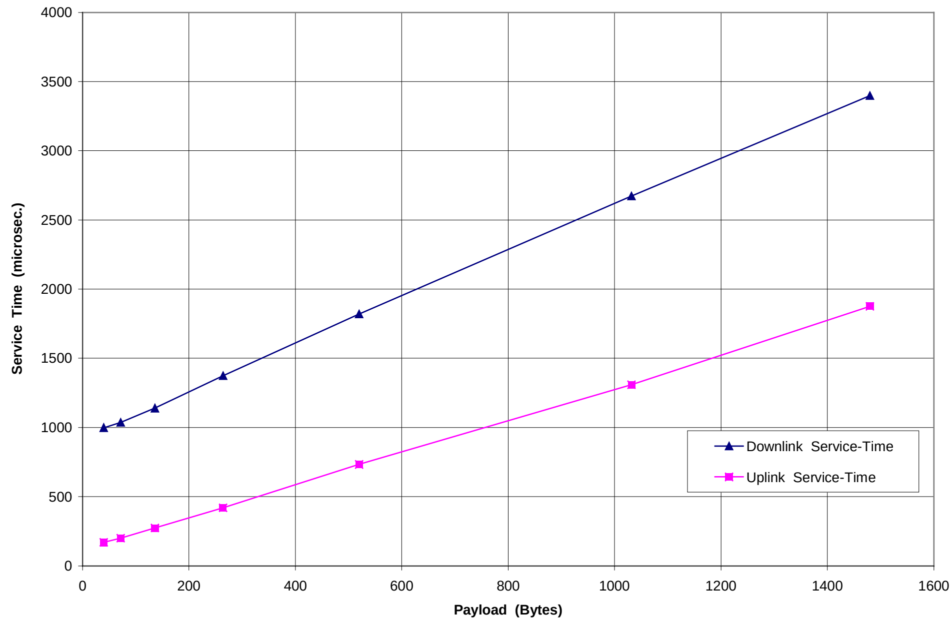

Figure 4. Downlink and Uplink Service-Times of AP2 (Lucent AP2000). The two plots diverge from each other as the payload increases.

The UDC graph for AP2 (figure 5) shows that UDC(*AP2*, *x*) increases as payload *x* increases. Using definition (5) we conclude that AP2 has a divergent UDC, as is clear from figure 5.

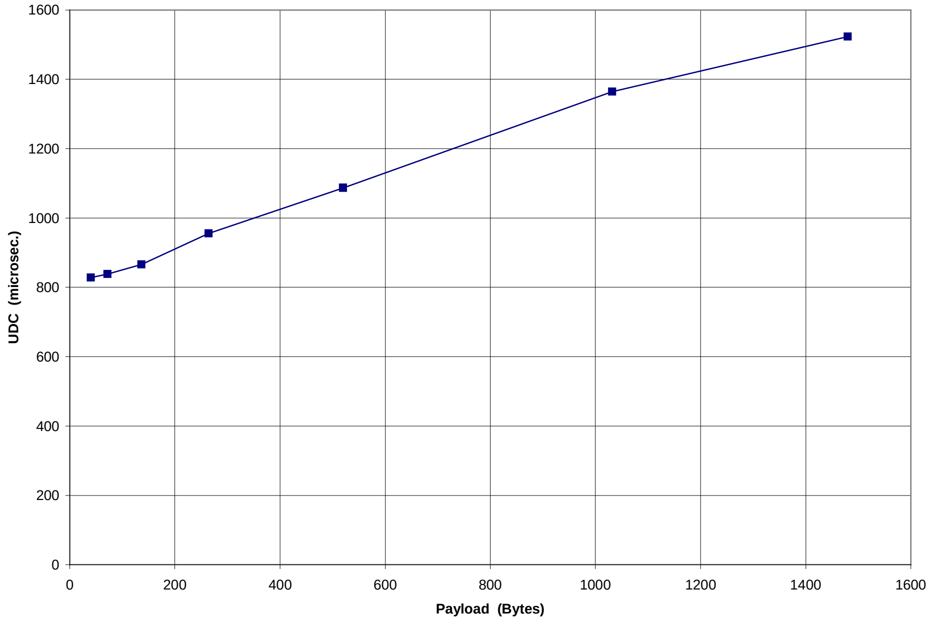

Figure 5. UDC plot for AP2 (Lucent AP2000). The UDC is increasing with increasing payload. AP2 has a divergent UDC.

4. UDC SIGNIFICANCE

The results on convergent and divergent UDC form a strong basis for studying the relationship between the brand of the WLAN access point and the QoS of specific applications over wireless LANs. For instance, an access point characterized by a convergent UDC is better suited for LANs where mobile users are interested in real-time multimedia applications that produce *large* amounts of payload. The convergent UDC minimizes the differences in delay in both directions when the application produces large payloads in both directions (such as a two-way videoconference [15]). So, if we know that the user is attached to a specific wireless LAN and we are using real-time applications with large packet sizes, the characteristics of Convergent UDC can be used for selecting an access point brand. Similarly, if the real-time application on a wireless LAN sends small packets, then a divergent UDC is preferable, because it will minimize the service time needed for the real-time communication packets at the cost of having larger delays for other applications that use larger packets. Video conferencing applications are a good example for such use of the UDC characteristic, because they transmit large or small packets depending on the video-codec used. Moreover, most commercial IP video conferencing centers use specific multimedia applications. Hence, when supporting conferencing with wireless connections, or when one or more peers join a conference session through a wireless connection, the access point used will have a significant effect on the performance. In fact, the market is witnessing an increasing demand for IP video conferencing, and we observe an increase in the number of IP videoconference centers. Thus, the manufacturers of wireless LAN access points can also benefit from the UDC characteristic of APs to try to enhance the performance of APs for specific applications and deliver better services by introducing different products, which suit the various conferencing applications in the market. Currently, the UDC characteristic reveals two types (based on the different AP brands we have tested): the convergent and the divergent UDC. Hence, just labeling the AP brands with convergent/divergent UDC characteristics will help the users to select suitable WLAN access points knowing the packet sizes of the applications used over the LAN.

Currently, we are investigating whether a convergent-UDC access point will have a better downlink throughput than a divergent-UDC access point. Moreover, we are analyzing the uplink and downlink delays and the UDC of access points to determine the time required to service the overhead bits on both directions: the downlink and the uplink, for a given access point.

5. CONCLUSION

We presented a performance analysis study of wireless LAN access points, for IEEE 802.11b. The results presented in this paper build on our previous work of modeling wireless LAN access points as single server, single queue, FIFO systems. We used the model and some of the previous results to analyze the absolute value of the difference between the uplink and downlink service times for a given AP. We define the absolute value of the difference in time between the uplink and downlink to be the *Uplink-Downlink Contrast* (UDC). The results of this investigation show that as the payload increases, the UDC either decreases or increases depending on the brand of the access point. We introduced the notions of convergent and divergent UDC. A convergent UDC decreases with payload, while a divergent UDC increases with payload. Knowing the size of the packets sent by the application, the choice of a suitable wireless LAN access point can be based on the convergent/divergent UDC characteristic of the access point.

ACKNOWLEDGMENTS

We express our thanks to Sermed Al-Abbasi at the *IT-University, Royal Institute of Technology (KTH), Stockholm, Sweden*, for the time he spent assigning different frequencies to the wireless access points near the testbed . We also would like to thank Dr. Luc Onana Alima from the *Distributed Computer Systems Group* at the *Department of Microelectronics and Information Technology (IMIT), Royal Institute of Technology (KTH), Stockholm, Sweden*, for the fruitful discussions and valuable remarks.

REFERENCES

[1] PCC (*Personal Computing and Communication*) project, a Swedish research program financed by the Foundation for Strategic Research, http://www.pcc.lth.se.

[2] T. Yoshimura, M. R. Jeong, H. Morikawa, and T. Aoyama, "Wireless Packet Scheduling for Adaptive Service over IEEE802.11," The First International Symposium on Wireless Personal Multimedia Communications (WMPC'98), pp. 314-319, Kanagawa, Japan, November 1998.

[3] K. Lai and M. Baker, "Measuring Link Bandwidths Using Deterministic Model of Packet Delay," ACM SIGCOMM, Computer Communication Review, Volume 30, Number 4, October 2000.

[4] I. Al Khatib, R. Ayani, and G. Q. Maguire Jr., "Wireless Access Points as Queuing Systems: Delay Probability and Service Time," submitted for publication.

[5] W. R. Stevens, *TCP/IP Illustrated, The protocols*, vol. 1, Addison-Wesley Publishing Company, 1998.

[6] A.M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, 3rd ed., McGraw-Hill, 1999.

[7] N. Chee Hock, *Queuing Modeling Fundamentals*, John Wiley and Son, 1998.

[8] J. Banks, S. Carson, B. L. Nelson, and D. M. Nicol, *Discrete-Event System Simulation*, 3rd ed., copyright Prentice-Hall, 2001.

[9] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers*, McGraw-Hill, 1988.

[10] ORINOCO, http://www.orinocowireless.com/.

[11] E. Pelletta, "Bridging Capacity of IEEE 802.11b Access Points," Master thesis, IT-University, Royal Institute of Technology (KTH), Stockholm, Sweden, 2002.

[12] J. A. Stine, "Tactical Communications Using the IEEE 802.11 MAC Protocol," MILCOM '98 Conference, unclassified paper 17.4, 1998.

[13] IEEE802.11 standard, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1997.

[14] IEEE802.11b standard, (Supplement to ANSI/IEEE Std. 802.11, 1999 Edition), Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band.

[15] A. Ganz, S. H. Park, and Z. Ganz, "Inline Network Encryption for Multimedia Wireless LANs," MILCOM '98 Conference, unclassified paper 17.1, 1998.

Publication 7

Iyad Al Khatib, Anders Franzen, and Fabio Moioli

Minimum GPRS Bandwidth for Acceptable H.261 Video QoS

in Proceedings of IFIP Personal Wireless Communications (PWC'2000), pp. 147-156, Gdansk, Poland, September 14-15, 2000.

ISBN: 0-7923-7941-1

Minimum GPRS Bandwidth for Acceptable H.261 Video QoS

Iyad Al Khatib, Anders Franzen, Fabio Moioli
Department of Teleinformatics, The Royal Institute of Technology
Ericsson Business Networks, and Ericsson Wireless LAN Systems
Sweden
iyad@it.kth.se, {Anders.franzen, Fabio.moioli}@era.ericsson.se

Abstract: As part of a larger research on multimedia traffic performance over GPRS, we present a QoS study focusing on one parameter: *bandwidth*. GPRS is an evolutionary phase and a critical step towards the third generation (3G) of mobile systems providing a data rate of 2Mbps. In our study on GPRS, we investigate *multimedia video traffic*. The video codec used is H.261 with QCIF resolution. Many parameters are under research; however, in this paper we focus on one parameter: *minimum required bandwidth for acceptable QoS of QCIF H.261 video streams over the wireless and mobile medium, GPRS.* Some other parameters of interest like the hurst parameter and multiplexing gain are tackled.

1. INTRODUCTION

The wireless and mobile telecommunication world is experiencing a very critical transitional stage, where new QoS parameters are to be defined. Within a more general study on mobile systems evolution, we analyse multimedia traffic over GPRS, a new phase of mobile communication media. GPRS represents an evolutionary step from the existing GSM system, where its purpose is to bring packet switched data services to the mobile system. With GPRS, the user can always be connected to the network since charging is not based on the connection time. The final billing scheme is not totally defined yet, but the main point is that the user should not care about connection time.

One of the other goals of GPRS is to try to provide higher speeds than traditional GSM systems. The maximum theoretical speed over GPRS is supposed to be around 115Kbps. This bandwidth is achieved with very good radio conditions, and when the network is fully developed. In practice, the starting GPRS speed would, to a large probability, be somewhere between 20Kbps and 56Kbps. An enhanced GPRS system called EDGE is supposed to bring the speed up to 384Kbps. This very evolutionary phase of mobile systems is believed to be only one step towards the third mobile systems generation (3G), which is expected to give speeds up to 2Mbps.

The GSM system uses Time-Division Multiple Access (TDMA) with eight radio frequency time slots. A network operator can dedicate 0 to 8 of these time slots to GPRS. Each mobile terminal can send/receive in 1 to 8 time slots. It is believed that the first mobile terminals generation for GPRS will support 4 time slots downlink and 1 time slot uplink, which gives around 14Kbps uplink and 56Kbps downlink.

With this great shift that GPRS will introduce to the wireless and mobile world, we are interested in investigating the quality of service that GPRS can offer to multimedia applications, mainly video quality. Our research in this area is long term; however, in this paper we investigate few multimedia traffic parameters for one video standard. The format of the video streams we investigate over GPRS is H.261 with QCIF resolution. H.261 is chosen for its low bit rate [10]. The H.261 video streams in the experiments are variable bit rate streams, which makes them more suitable for the medium [4]. Quarter-CIF (QCIF) has 176 pixels per line, and 144 lines [9]. QCIF is chosen, because it is mainly used for desktop videophone applications i.e. the size will be suitable for a mobile unit. In addition, all codecs must be able to handle QCIF.

The parameter we focus on throughout the experiments is the minimum GPRS bandwidth required for acceptable QoS of H.261 video streams of QCIF resolution. We are also interested in self similarity since if we can define which type of videos show self similarity over GPRS, then GPRS vendors can learn more about how to deal with video over this medium [3, 6]. In this respect, the Hurst parameter is calculated. The Hurst parameter can be looked at as a self-similarity value; if near to 1, then this would be a sign of self-similarity. However, if it shows a value nearer to 0.5, then there is not much of self-similarity in the traffic.

In a study of multimedia over GPRS, it is very important to note that the standards with which the QoS is judged are subjective. Unfortunately, up till now, the judgements on acceptable QoS for multimedia streams are relative to the observer's personal standards [8]. Hence, we find it very important that, in our study, the minimum acceptable parameters investigated are defined by a representative number of people from different population backgrounds. Hence a common acceptable QoS is set to find the minimum bandwidth sought.

To calculate the theoretical values for the minimum acceptable bandwidth, we use the Multiplexing Gain formula [5]:

$$G_n = nR_p/C_n \qquad \ldots (1)$$

where R_p is peak rate for the video stream; n is the number of independent streams combined for transmission; and C_n is the link-bandwidth required for the desired QoS for the multiplexed stream of n sources (C_1 being the link bandwidth for a single source).

$$(1) \Rightarrow G_n = nR_p/C_n = [nC_1/C_n][R_p/C_1] = [nC_1/C_n]G_1 \qquad \ldots (2)$$

where G_1 is the multiplexing gain for one source.

$$(2) \Leftrightarrow C_n = n[G_1/G_n]C_1 \qquad \ldots (3)$$

Here we think of the multiplexing gain as a parameter to use in order to achieve the minimum link-bandwidth for n streams where $n \in \boldsymbol{N}^*$, the set of natural numbers - $\{0\}$. The multiplexing gain G_n for n number of independent streams is given by,

$$\frac{1}{G_n} = \frac{1}{b} + \left(\frac{1}{G_1} - \frac{1}{b}\right) n^{\frac{1-2H}{2H}} \qquad \ldots (4)$$

where b is the peak-to-average and H is the Hurst parameter. Many methods can be used to calculate the Hurst parameter, like time variance plot, R/S analysis [2], and periodogram method.

2. EXPERIMENTS AND RESULTS

Figure 1 shows the testbed, which consists of two video senders, a GPRS emulator, a receiver, and a traffic measurement and analysis tool, NIKSUN NetVCR^{TM}. All the experiments, except the last one, use one sender only, for they are dedicated to studying the bandwidth required for one video stream. On the other hand, the last experiment concentrates on the performance when multiple streams are sent over GPRS. Table 1 shows the video streams, where "Comm" is the stream used in experiments 1 to 5. The packet time slots on the GPRS medium are set to 8 time slots throughout all the experiments since using less for video transmission will not lead to acceptable QoS. First, we look into whether there is any difference between the behavior of two media: GPRS with no restricting limits, and 10-BT. The associated results for the H.261 video stream are presented in table 2, where one would conclude that when the GPRS is dedicated to one video stream, with no background users, it will most likely behave like Ethernet. However, when running the first experiment on 10 Mbps Ethernet, we got no missing frames at the receiver end, while in running the experiment over GPRS, with 12dB, we had 2,670 video frames missing out of 6,306 video frames of the same stream (see table 4, Exp. 1 and 2). Figures 1-8 show the number of bytes (vertical axis) versus the packet size categories (horizontal axis).

Table 1. Video sequences used in the experiments. "Comm" is used in experiments 1 to 5.

Type of video	**Length (mm:ss)**	**Total bytes**	**Total packets**	**Average bandwidth (bps)**	**Hurst param.**
Music 1	05:18	1,769,912	7,941	44,387	0.81
Music 2	03:39	3,027,270	4,745	109,584	0.78
Music 3	03:25	2,633,906	4,582	101,793	0.88
News	13:55	10,657,756	18,313	101,623	0.74
Talking head	12:51	11,682,038	13,682	120,901	0.61
Commercial **"Comm"**	05:06	3,631,412	6,942	94,322	0.79
Total	*44:14*	*33,402,294*	*56,205*		

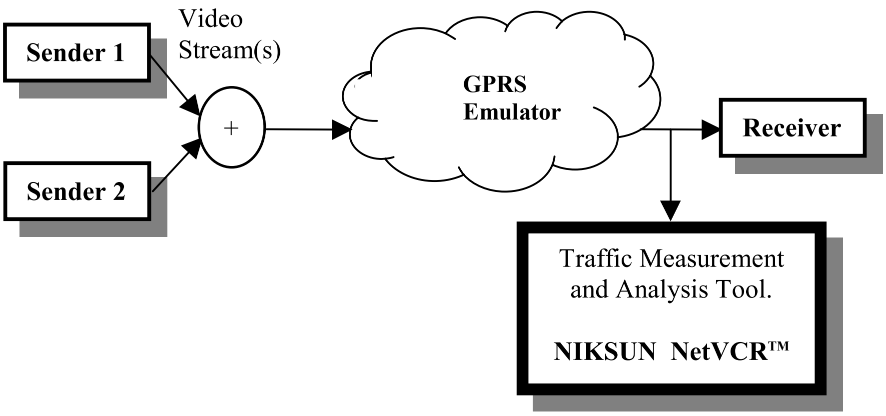

Figure 1. Testbed of bandwidth investigation for H.261 video quality over GPRS.

Hence in figure 1, the peak (bytes) is for the packets that are 512-1024 bytes in size, excluding 1024 byte packets. The second level peaks are for packets of 1024-2048 bytes (excluding 2048 byte packets) and 216-512 bytes (excluding 512 byte packets) respectively. Figures 2 through 8 can be read in a similar way for the associated experiments.

Table 2. Differences between GPRS with no restrictions and 10BT for "Comm". S/N=12 dB.

	No limits on GPRS	**10 Mbps Ethernet**
Total no. of bytes received	1.7284e+003	1.7289e+003
Median	3848	3788
Peak-to-Average ratio	5.3804	4.7394

We use the parameters in tables 3 and 4 to do some calculations for comparisons with the values received by the application. In this respect, we would like to note that the video is observed in real time and then the number of missing video frames is calculated. We believe that these numbers are very important to relate to acceptable QoS of the H.261 streams over GPRS. The fact that no frame is missing in experiment 1 can also be seen while watching the video stream live. Experiment 1 is also used as a comparison basis for acceptable QoS.

Table 3. Number of packets vs packet size for "Comm" video stream.

Packet Size Categories (Bytes)	**Count (Packets)**				
	10 BT	*GPRS*			
	Exp.1,fig.1	**Exp.2**,*fig.2* 12dB 0 BGU	**Exp. 3**,*fig.3* 15dB 0 BGU	**Exp. 4**,*fig.4* 12dB 20 BGU	**Exp.5**,*fig.5* 12dB 40 BGU
0 to 128	507	305	271	282	275
128 to 256	1089	651	594	585	623
256 to 512	2332	1332	1365	1302	1367
512 to 1024	2152	1312	1303	1232	1302
1024 to 2048	862	658	672	606	641
2048	0	14	19	12	21

Table 4. Statistics for "Comm" Video Stream over 10 Base-T and GPRS.

	10 BT	GPRS			
	Exp. 1	**Exp .2** 12dB 0 BGU	**Exp. 3** 15dB 0 BGU	**Exp. 4** 12dB 20 BGU	**Exp. 5** 12dB 40 BGU
Total Number of Bytes	7262824	4639074	4674424	4366946	4614556
Average Rate (bps)	48418.83	77317.90	103876.09	72782.43	102545.69
Number of Packets	13884	8544	8448	8038	8458
Average Rate (pps)	11.57	17.80	23.47	16.75	23.49
Minimum Packet Size	67	69	68	67	69
Maximum Packet Size	1066	1066	1066	1066	1066
Mean Packet Size	523.11	542.96	553.32	543.29	545.58
Packet Size Variance, B^2	93519.15	101585.97	100581.10	99668.34	99399.37
Variance/Mean	178.78	187.10	181.78	183.45	182.19
Missing Video Frames	0	2670	2719	2811	2788

In the second experiment GPRS is used. The S/N is 12dB i.e. worst case. However the main concern of this experiment is to measure the same parameters as in experiment 1 but over GPRS, hence we have no BackGround Users (BGU) i.e. the "Comm" video has all the bandwidth. The results are presented in figure 2. Collecting these parameters, we can also find a great similarity in the QoS delivered as well as the shapes of the graphs in figures 1 and 2. Around 41% of the frames are missing, but still the QoS is acceptable. The receiver end shows a rate ranging between 48Kbps and 62Kbps, which is a very good rate in our point of view for a video transmission with a QCIF resolution.

We also investigate the behavior and the used-bandwidth results for the 15dB S/N. We run exactly the same experiment as in experiment 2, but with 15dB instead of 12dB. The result is shown in figure 3. This experiment shows just a slight difference where there are more of large packets i.e. the traffic concentration is more in the middle (512-1024B) than in experiment 2. The rate ranges between 48Kbps and 63Kbps i.e. acceptable.

To get more practical results, we force background users over the GPRS network. The number of background users we have in experiment 3 counts to 20, with 12dB. The result can be seen in figure 4. The behavior still shows a graph similar to the previous experiments. The rate at the receiver's end still shows a range between 40Kbps and 60Kbps.

In the fifth experiment we force 40 users with 12dB over GPRS. The behavior is similar to the previous experiments in terms of graphical shape (figure 5), but the video quality drops down. In fact it is not acceptable at all. However, the rate at the receiver's end still shows a range between 48Kbps and 60Kbps.

In the sixth experiment our concentration is on the bandwidth when multiple streams are injected over GPRS, 12dB and 40 BGU. The results are predictable as shown in figures 6, 7 and 8. Filtering the traffic of each stream alone is important to study the bandwidth from the multiplexing gain point of view. Figure 6 shows the result for the traffic of both H.261 video streams at the same time. Since both streams, when injected together, have their first level peaks at (512-1024B), as well as their second level peaks at the same points (figures 7 and 8), then adding the two would lead to a graph with peaks at the same relative points (figure 6). For the first stream, the rate at the receiver's end still shows a range between 21Kbps and 37Kbps. For the second stream, the rate at the receiver's end still shows a range between 30Kbps and 50Kbps. The rates show that the bandwidth is divided, and this is a normal behavior. Around 60% of the frames are lost in each video stream. The visual effects on the QoS can be observed, and the delay between the frames is not within the acceptable range when there are transitions in the video stream. We also run multiple streams over GPRS to investigate more on the multiplexing gain and suitable bandwidth for the set QoS. Results are shown in table 7.

Referring to equation 3, if we know G_1 and C_1, then knowing C_n will be just a matter of knowing G_n, which can be calculated using equation 4. For an acceptable QoS, we will use equation 1 to get to a $C_1 = R_p$, where R_p is investigated in the experiments to be around 70Kbps. This makes $G_1 = 1$ for acceptable QoS. Hence (3) becomes: $C_n = n(1/G_n)(70) = 70n(1/G_n)$.

Table 5. Number of packets vs packet size; 2 video streams; GPRS, 12dB, 40 BGU.

Packet Size Categories (Bytes)	**Count (Packets)**		
	GPRS, 12 dB, 40 BGU		
	Total of Two video streams	First video stream **filtered**	Second video stream **filtered**
0 to 64	355	-	-
64 to 128	307	162	144
128 to 256	824	366	313
256 to 512	1433	747	699
512 to 1024	1452	730	728
1024 to 2048	624	304	305
2048	13	0	8

Table 6. Two video streams over GPRS, 12dB, 40 BGU.

	GPRS, 12dB, 40 BGU		
	Total of Two video streams	First video stream **filtered**	Second video stream **filtered**
Total number of Bytes	4949190	2456798	2401096
Average Rate (bps)	94270.27	16378.65	45735.16
Number of Packets	10016	4618	4394
Average Rate (pps)	23.85	3.85	10.46
Minimum Packet Size	60	72	68
Maximum packet size	1066	1066	1066
Mean Packet size	494.13	532.00	546.45
Packet Size Variance, B^2	10533.95	96162.32	97318.30
Variance/Mean	213.17	180.75	178.09
Missing video frames		4499	4399

Table 7. Multiplexing gain and min. bandwidth for a increasing number of video streams.

No. of Streams	Average (bits/interval)	Peak (bits)	Peak-to-Average	Hurst Param.	Multiplexing Gain	=>	**Minimum Bandwidth**
2	198,307	278,528	1.40	0.87	1.82	=>	76.9Kbps
3	297,458	455,384	1.53	0.79	1.83	=>	114.8Kbps
5	487,719	628,664	1.29	0.82	1.60	=>	218.8Kbps
10	970,794	1,246,264	1.28	0.83	1.50	=>	466.7Kbps
15	1,455,795	2,002,776	1.38	0.84	1.54	=>	681.8Kbps

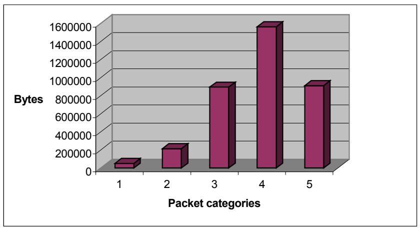

Figure 1. Exp. 1, "Comm" bytes vs packet categories over 10-BT.

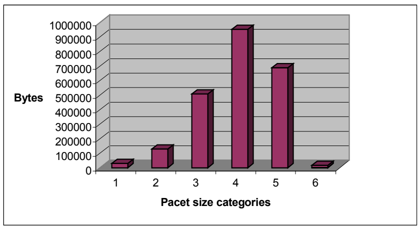

Figure 2. Exp. 2, "Comm" bytes vs packet size categories; GPRS, 12dB, 0 BGU.

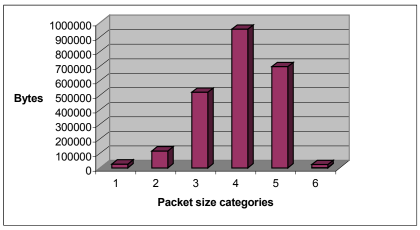

Figure 3. Exp. 3, "Comm" bytes vs packet size categories; GPRS, 15dB, 0 BGU.

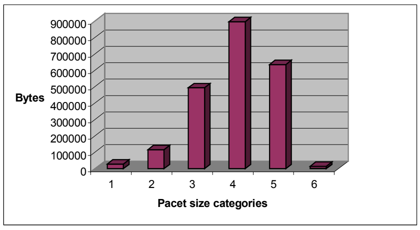

Figure 4. Exp. 4, "Comm" bytes vs packet size categories; GPRS, 12dB, 20 BGU.

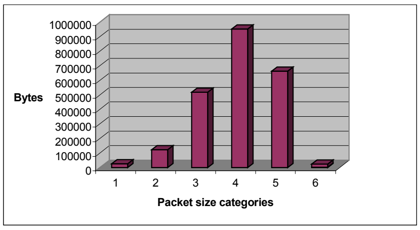

Figure 5. Exp. 5, "Comm" bytes vs packet size categories; GPRS, 12dB, 40 BGU.

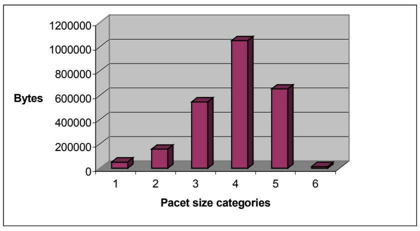

Figure 6. Exp. 6, two video streams traffic over GPRS, 12dB, 40 BGU.

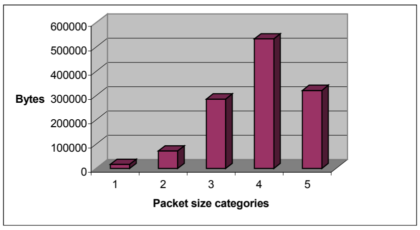

Figure 7. Exp. 6, first video stream over GPRS, 12dB, 40 BGU.

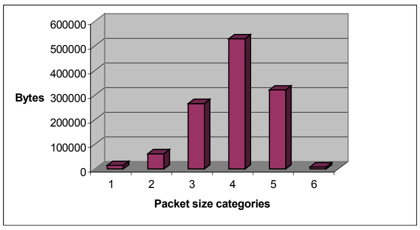

Figure 8. Exp. 6, second video stream over GPRS, 12dB, 40 BGU.

Figures 1-8. Horizontal axes show packet size categories of size X bytes, where labels 1, 2, 3, 4, 5, and 6 represent categories with $1 \le X < 128$ bytes, $128 \le X < 256$ bytes, $256 \le X < 512$ bytes, $512 \le X < 1024$ bytes, $1024 \le X < 2048$ bytes, and $X = 2048$ bytes respectively.

3. EVALUATION OF RESULTS

As we increase the number of users and limitations on the GPRS, our calculations lead to a final value that we would like to present. The minimum acceptable bandwidth for H.261 video streams QoS over GPRS is found - after many iterations and trials - to be around 70-80Kbps for one QCIF H.261 video stream. This number is not very satisfactory since the practical limit that GPRS can deliver now is around 50Kbps. However, work is going on to reach higher practical limits, and if 70Kbps is reached, then sending video streams with QCIF resolution will be possible for the defined QoS. When a bandwidth of less than 70Kbps over GPRS is reached, the video quality and the missing frames number are not acceptable. In this respect, and regarding the transmission of multiple streams to one receiver to two different application port numbers, the sharing of the bandwidth will surely happen. However, the results in table 7 clearly show that the bandwidth needed over GPRS for the H.261 video for (*n*) streams will be less than the sum of the peak rates of the two streams. In other words, multiplexing gain will occur and will be a value greater than 1;

$$C_n = nR_p/G_n < nR_p = nC_1 \; ; \qquad G_n > 1, n \in \boldsymbol{N}^*.$$

The quality with multiple streams will always be less than for one video sent as shown in experiment 6.

One parameter that seems promising for more research is the Hurst parameter shown in figure 9 with a *Log variance* vs *Log lag* plot. Since the self similarity is an interesting parameter to look at when all the presented data is available [7], we look at the Hurst parameter for two streams. The two H.261 video streams over GPRS show a Hurst parameter of around 0.97, with 12dB, 40 BGU. This means that the self-similarity is highly probable to occur [1]. This still needs more study to be conducted, but it is a very interesting start.

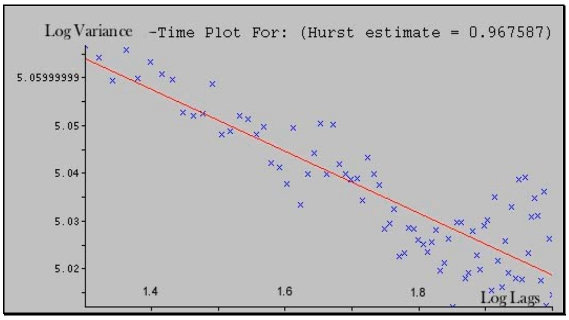

Figure 9. Hurst parameter plot for 2 video streams over GPRS, 12dB and 40 BGU.

4. CONCLUSION

We investigate some multimedia traffic parameters over GPRS, the third generation of mobile systems. The video streams investigated are encoded in H.261 codec. QCIF resolution is chosen for investigation since it can be deployed on mobile units. The minimum bandwidth required for acceptable QoS of QCIF H.261 video is dependent on the peak rate of the video, the number of streams, and how much the medium can have of multiplexing gain. For one video stream, the minimum bandwidth is around 70Kbps, which is still not easy to achieve over GPRS. However future GPRS generations will be able to supply this bandwidth and more. The encouraging part is that when two or more video streams are injected, they need less bandwidth than the sum of the peak rates of each. We hope that our study triggers more investigation in the filed of multimedia over GPRS from the traffic analysis point of view. The Hurst parameter is also presented briefly.

REFERENCES

[1] Jan Beran, Robert Sherman, Murad S. Taqqu and Walter Willinger, "Long-Range Dependence in Variable-Bit-Rate Video Traffic", IEEE Transactions on Communications 43, No. 2/3/4, pp. 1566-1579, 1995.

[2] M. W. Garrett, W. Willinger, "Analysis, Modeling and Generation of Self-Similar VBR Video Traffic", Computer Communications review vol. 24, no. 4, pp. 269-280, 1997.

[3] Daniel P. Heyman and T. V. Lakshman, "What Are the Implications of Long-Range Dependence for VBR-Video Traffic Engineering?", IEEE/ACM Transactions on Networking, vol. 4, no. 3, June 1996.

[4] C. Huang, M. Devetsikiotis, I. Lambadaris and A.R. Kaye, "Modelling and Simulation of Self-Similar Variable Bit Rate Compressed Video: A Unified Approach", Proc. of the ACM Sigcomm'95, Boston, pp. 114-125, 1995.

[5] K. R. Krishnan, A.L. Neidhardt, and A. Erramilli, "Scaling Analysis in Traffic Management of Self similar Processes", ITC 15, Elsevier Science B.V., 1997.

[6] W. Leland, M. Taqqu, W. Willinger and D. Wilson, "On the Self-Similar Nature of Ethernet Traffic", IEEE/ACM Transactions on Networking vol. 2, no. 1, February 1994.

[7] Kihong Park, Gitae Kim Mark Crovella, "On the Effect of Traffic Self-similarity on Network Performance", Proceedings of the 1997 SPIE International Conference on Performance and Control of Network Systems.

[8] P. Pruthi, D Ilie, A. Popescu "Application Level Performance of Multimedia Services", SPIE International Symposium on Voice, Video, and Data Communications Boston, Sep 19-22 1999.

[9] Thierry Turletti, "H.261 Software Codec for Videoconferencing over the Internet", Rapports de Recherche, Unité de recherche no. 1834, INRIA-SOHIA Antipolis, 1993.

[10] Draft ITU-T Recommendation H.263 (1996): "Video Coding For Low Bitrate Communication".