

M.Tech. Project Report

A Study on the Performance of the HTTP Protocol over Wired-cum-Wireless LANs

A Project Report submitted in partial fulfillment of the requirements of the Degree of Master of Technology in Information Technology of Tezpur University

Himadri Barman

**Department of Information Technology
School of Science and Technology
Tezpur University
Tezpur 784028
Assam, India**

December 2002

A
Study on the Performance
of the
HTTP Protocol
over
Wired-cum-Wireless LANs

Department of Information Technology
School of Science and Technology
Tezpur University
Tezpur 784028

Forwarding Certificate

We hereby recommend that the project report prepared under our supervision by Himadri Barman entitled “**A Study on the Performance of the HTTP Protocol over Wired-cum-Wireless LANs**” be accepted in requirements for the Degree of Master of Technology in Information Technology of Tezpur University

Dhruba Kumar Bhattacharyya
(Reader and Head of the Department, IT)

Nityananda Sarma
(Lecturer, Department of IT)
Supervisor

Department of Information Technology
School of Science and Technology
Tezpur University
Tezpur 784028

Certificate of Approval

The foregoing project report, entitled “**A Study on the Performance of the HTTP Protocol over Wired-cum-Wireless LANs**”, is hereby approved as a credible study which has been carried out and presented in a satisfactory manner to warrant its acceptance as a prerequisite to the Degree for which it has been submitted. It is understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein. It has been approved only for the purpose for which it has been submitted.

Head of the Department, IT

Supervisor

External Examiner

**As part of the Committee on
Final Examination
For Evaluation of the Thesis**

Abstract

HTTP, the HyperText Transfer Protocol, today accounts for over 90% of all Internet traffic. This thesis is based on a comparative study on the performance of versions 1.0 and 1.1 of HTTP on a hybrid computer network model. The hybrid computer network model is a wired-cum-wireless model wherein a wireless LAN is considered as an infrastructural addition to a wired LAN. A few metrics are selected and defined for the performance analysis which will determine whether we can use the existing protocols, modify them or develop completely new protocols for using in a wired-cum-wireless network. The freely available open source software network simulator *ns-2.1* [6] is chosen as the tool for the performance evaluation. It will help in analyzing the various wireless networking hardware and topological configurations using the same setup and conclusions can be drawn accordingly. The empirical model of HTTP network traffic, proposed in [10] is selected as a basis for web browsing behaviour.

Contents

Abstract	v
List of Figures	viii
List of Tables	ix
Preface	x
Acknowledgements	xi
1. Report Introduction	1
1.1 Background	
1.2 Goal	
1.3 Contribution	
1.4 Document Outline	
2. Background	3
2.1 Wireless Networks	
2.2 Local Area Networks	
2.3 Wireless Local Area Networks	
2.3.1 Wireless LAN Configurations	
2.3.2 Benefits from Wireless LANs	
2.3.3 Wireless LAN – Technology Options	
2.4 The HyperText Transfer Protocol	
2.5 <i>ns</i> – Network Simulator	
2.5.1 Simulator Internals	
2.6 Nam – The Network Animator	
3 The Empirical Model of HTTP Network Traffic	10
3.1 Methodology	
3.2 Model	
3.3 Results	
3.4 Model Representation	
4 Performance Evaluation	13
4.1 Metrics	
5 Experimental Setup	15
5.1 The Model	

5.2 How the Web Pages are retrieved by the Clients from the HTTP Server	
5.3 Transfer Methodology	
5.4 Web Browsing Statistics	
5.5 Implementation	
5.5.1 Physical Implementation	
5.5.2 Simulation	
6 Conclusions	23
7 Future Work	24
Appendix A – Glossary	25
Bibliography	27

List of Figures

2.1 <i>nam</i> Interface	8
5.1 Model under Study	15
5.2 The WWW Page Transfer Process [4]	16
5.3 State Diagram for CSMA/CD	18
5.4 State Diagram for CSMA/CA	19

List of Tables

3.1 Summary of Traffic Traces	11
3.2 Quantities Modeled	11
3.3 Selected Measurement Results	12
5.1 Summary of HTTP Statistics	20

Preface

The work undertaken for the preparation of this report has been carried out at the Department of Information Technology under the School of Science and Technology at Tezpur University,

I am convinced that the work I have performed during the last 5 months have provided me with excellent insights in the field of networks in general and wireless networks in particular and also the various internet protocols. It further provided me with the opportunity to learn the basics of the simulator *ns*.

The accompanying CD-ROM contains materials related to the project besides a soft copy of the report. I hope to continue working on the future works mentioned later on this report. All related works will be made available at my home page at <http://himadri.koolhost.com>.

Himadri Barman
December 2002

Acknowledgements

During the course of this project, I received the guidance, help and encouragement from many. This acknowledgement is a small tribute to all of them. Thank you all!

At the very outset, I acknowledge the help and guidance of my supervisor, Mr. Nityananda Sarma for giving me the opportunity to work under him. It would be hard to comprehend the completion of this project without him by my side. His ability to analyze and understand the problems is amazing to say the least, and has enriched my understanding and thought process. Thanks to Dr. Dhruva Kumar Bhattacharyya, Head of the Department, Information Technology, Mr. Nitul Dutta, System Administrator, and Mr. Mohan Sewa, Office Assistant for making available to me all resources available at their disposal. I am equally thankful to my classmates and the other faculty members for their continued support and encouragement.

I am elated to put in record my deep appreciation for Mr. Sadhan Mahapatra, Lecturer, Department of Energy, Tezpur University for giving me valuable tips while writing my project report and his encouragement throughout the project period. I am grateful to Dr. Munindra Bora, Reader, Department of Mathematical Sciences, Tezpur University for making my stay possible at Tezpur University.

I will ever remain indebted to my parents for giving me the freedom to take my own decisions as also my brothers for their faith in me. Thanks to my friend Sabita for the support she extended, when I needed it the most. At last, a big thank you to all my friends – you are my most prized possessions.

- Himadri

Chapter 1

Report Introduction

1.1 Background

Wireless Local Area Networks (WLANs) are deployed almost everywhere today, mainly because of their enormous flexibility. It is expected that the usage of these networks will increase to cover large areas, such as university campuses, corporate office space, downtown stores, etc. With that, the number of users will increase and almost certainly there will be demands, the same as on wired networks, particularly Internet traffic. Though the wireless LAN market has seen a lot of growth worldwide recently, thanks to the wide support it's been receiving from the industry, it has not yet reached the position where it can completely replace the wired networks. The best option now is to integrate it with the existing wired networks and reap the benefits that come with it. Thus, there is enough scope for a study on wired-cum-wireless networks and various applications running over it. This study is an attempt in that direction.

1.2 Goal

The number of users as far as wired-cum-wireless network is concerned is on the rise and almost certainly there will be demands, the same as on wired networks, particularly Internet traffic. Therefore, the goal of the work undertaken as part of this report preparation is to undertake a comparative study on HTTP/1.0 and HTTP/1.1 which together handles more than 90% of all Internet traffic vis-à-vis the wired-cum-wireless networks and determine their suitability to the same.

1.3 Contribution

The main contribution of this work is its endeavour to use the existing results and studies, where necessary with suitable modifications to the current study, and avoid repetitive works.

1.4 Document Outline

The report is organized as follows:

Chapter 1: The Report Introduction gives a short introduction regarding the background, the goals, and the contribution of this report. You are going through it.

Chapter 2: This chapter discusses the background and techniques behind wireless networks and the simulation environment that is to be used for the evaluation of the performance of the HTTP protocol.

Chapter 3: This chapter outlines the empirical model of HTTP network traffic required for the simulation.

Chapter 4: This chapter dwells on the important aspect of performance evaluation.

Chapter 5: The most important chapter. It describes the model under study, the mechanics of web transfer in the model and the experimental setup.

Chapter 6: This chapter concludes the work.

Chapter 7: This chapter provides an outline on future works that may be undertaken along with some recommendations on where to take the work presented in the thesis one step further.

Appendix A: This appendix gives explanation of a few terms used in this thesis.

Chapter 2

Background

2.1 Wireless Networks

The two words *wireless* and *network* are in common usage today. Many companies either talk about product(s), have them or present futuristic concept product(s) that are to be used in a wireless networking environment. The concept of wireless networks incorporates several different network technologies, communication ranges and transmission bandwidths. They range from local coverage indoor networks (as IEEE 802.11) to large wide area coverage networks, as the third generation mobile telephony systems. One common thing about wireless networks is that they suffer from relatively low bandwidth and high bit error rates, compared to wired networks. The greatest advantage is that they are very flexible and easy to deploy.

2.2 Local Area Networks

A Local Area Network (LAN) is a data communication system, usually owned by a single organization that allows similar or dissimilar digital devices to talk to each other over a common transmission medium. According to the IEEE,

A local area network is distinguished from other types of data networks in that communication is usually confined to a moderate geographic area such as a single office building, a warehouse, or a campus and can depend on a physical communications channel of moderate-to-high data rate which has a constantly low error rate.

Thus we may regard a LAN as a resource sharing data communication network with the following characteristics:

- Short distances
- High speed
- Low cost
- Low error rate
- Ease of access
- High reliability/integrity

2.3 Wireless Local Area Networks

A wireless LAN (WLAN) or a Local Area Wireless Network (LAWN) is a flexible data communication system implemented as an extension to, or as an alternative for, a wired LAN and uses electromagnetic waves for transmitting and receiving data over the air, minimizing the need for wired connections. It is to be emphasized that WLANs frequently augment rather than replace LANs – often providing the final few metres of connectivity between a backbone network and the mobile user.

2.3.1 Wireless LAN Configurations

WLANs are basically of three different types of configurations:

1. **Independent WLANs:** The simplest WLAN configuration is an independent or peer-to-peer WLAN that connects a set of PCs with wireless adapters. Any time two or more wireless adapters are within range of each other, they can set up an independent network. These on-demand networks typically require no administration or pre-configuration. Access points can extend the range of independent WLANs by acting as a repeater, effectively doubling the distance between wireless PCs.
2. **Infrastructure WLANs:** Here, multiple access points link the WLAN to the wired network and allow users to efficiently share network resources. The access points not only provide communication with the wired network but also mediate wireless network traffic in the immediate neighbourhood.
3. **Microcells and Roaming:** Here, WLANs use cells, called microcells, similar to the cellular telephone system to extend the range of wireless connectivity. At any point in time, a mobile PC equipped with a WLAN adapter is associated with a single access point and its microcell, or area of coverage. Individual microcells overlap to allow continuous communication within wired network. They handle low-power signals and hand off users as they roam through a given geographic area.

2.3.2 Benefits from Wireless LANs

- **Mobility:** Provides users with access to real-time information anywhere within the domains of the network.
- **Installation speed and simplicity:** Easy and fast installation eliminating need to pull cables through walls and ceilings.
- **Installation flexibility:** Wireless technology allows the network to go where wires cannot.
- **Reduced cost of ownership:** While the initial investment required for wireless LAN hardware can be higher than the cost of wired LAN hardware, overall installation expenses and life-cycle costs can be significantly lower. Long-term cost benefits are greatest in dynamic environments requiring frequent moves, additions, and changes.
- **Scalability:** WLAN systems can be configured in a variety of topologies to meet the needs of specific applications and installations. Configurations are easily changed and range from peer-to-peer networks suitable for a small number of users to full infrastructure networks of thousands of users that allow roaming over a broad area.

2.3.3 Wireless LAN – Technology Options

1. **Spread Spectrum:** It is a wideband radio-frequency technique developed by the military for use in reliable, secure, mission-critical communication

systems. It is designed to trade off bandwidth efficiency for reliability, integrity and security. In other words, more bandwidth is consumed than in the case of narrowband transmission, but the trade off produces a signal that is, in effect, louder and thus easier to detect, provided the receiver knows the parameters of the spread-spectrum signal being broadcast. If a receiver is not tuned to the right frequency, a spread-spectrum signal looks like background noise.

2. **Narrowband Technology:** A narrowband radio system transmits and receives user information on a specific radio frequency. Narrowband radio keeps the radio signal frequency as narrow as possible just to pass the information. Undesirable cross talk between communication channels is avoided by carefully coordinating different users on different channel frequencies.
3. **Frequency-Hopping Spread Spectrum (FHSS) Technology:** It uses a narrowband carrier that changes frequency in a pattern known to both the transmitter and receiver. Properly synchronized, the net effect is to maintain a single logical channel. To an unintended receiver, FHSS seems to be short duration impulse noise.
4. **Direct-Sequence Spread Spectrum (DSSS) Technology:** It generates a redundant bit pattern for each bit to be transmitted. This bit pattern is called a chip. The longer the chip, the greater the probability that the original data can be recovered. Even if one or more bits in the chip are damaged during transmission, statistical techniques embedded in the radio can recover the original data without the need for retransmission. To an unintended receiver, DSSS appears as low-powered wideband noise and is rejected by most narrowband receivers.
5. **Orthogonal Frequency Division Multiplexing (OFDM):** The basic idea here is to make a broadband, high data transmission, as in DSSS. But in OFDM, the data is divided into several interleaved parallel bit streams where each stream modulates a separate sub carrier. By modulating the channel in this way, the channel frequency is divided into several independent sub channels. One of the great benefits with OFDM is that it uses the radio spectrum efficiently since the sub channels could be packed close together. One drawback is that due to the high frequency band (5GHz), the transmission range is rather small (30 to 150m only).
6. **Infrared (IR) Technology:** IR systems use very high frequencies, just below visible light in the electromagnetic spectrum to carry data. Like light, IR cannot penetrate opaque objects, it is either directed or diffuse technology. Inexpensive directed systems provide very limited range of a few feet and are occasionally used in specific WLAN applications. High performance directed IR is impractical for the mobile users and is therefore used only to implement fixed sub-networks. Diffuse or reflective IR WLAN systems do not require line-of-sight, but cells are limited to individual rooms.

2.4 The HyperText Transfer Protocol¹

The World Wide Web (WWW) is an architectural framework for accessing linked documents spread out over thousands of machines all over the mother of all networks, the Internet. From the users' point of view, the web consists of a vast, worldwide collection of documents, usually just called pages for short. Each page usually contains links (pointers) to other related pages, anywhere in the world. Users can follow a link (usually by clicking on it), which then takes them to the page pointed to. Pages that point to other pages are said to use hypertext. No wonder, the underlying protocol used to retrieve web pages is called the HyperText Transfer Protocol (HTTP).

The HTTP is an application-level ASCII protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless protocol which can be used for many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through extension of its request methods, error codes and headers. A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred.

HTTP has been in use by the World-Wide-Web (WWW) global informative initiative since 1990. The first version of HTTP, referred to as HHTTP/0.9, was a simple protocol for raw data transfer across the internet. HTTP/1.0, as defined by RFC 1945, improved the protocol by allowing messages to be in the format of MIME like messages, containing meta-information about the data transferred and modifiers on the request/response semantics and was the first to be used on a wide scale. But it had its inherent problems.

The problems with HTTP/1.0 are:

- ⊗ A separate connection is opened for each object on a page which means opening multiple connections for each page. This decreases throughput as a certain amount of time is wasted for setup and shutdown. Network load is also unnecessarily increased as packets used for setup and shut down don't carry application data.
- ⊗ Each connection consumes state in the HTTP server. HTTP uses TCP for its transport protocol and TCP specifies that a connection must remain in a waiting state for four minutes after it is closed. Thus, a server may be littered with many ports in this state. This undermines the server performance because the server takes time to determine which request a packet is destined for.
- ⊗ Most HTTP/1.0 connections are short, since most HTTP objects are small. The result is that TCP cannot utilize the full bandwidth of the network. TCP having a slow start mechanism, by the time the best throughput rate is achieved, the connection gets terminated. This causes the throughput experienced by the user to be unnecessarily low. A related problem is high internal fragmentation, which causes many small packets to be sent. Small packets are

¹ RFC 2616 refers it to as Hypertext Transfer Protocol, some others as Hyper Text Transfer Protocol

wasteful of network resources because packets have a fixed cost of certain network resources regardless of the size of the packet.

- ⊗ HTTP/1.0 is a strict request/response protocol which necessitates at least two network round trips for each retrieved object, which greatly decreases throughput on high latency networks like a wireless network.

HTTP/1.1 [7] was developed to address the above problems. It is backward compatible with HTTP/1.0 but connections are managed using a single persistent connection, which is left open for subsequent retrievals, thus solving the problems of HTTP/1.0 in a single stroke. Requests are also pipelined so that multiple requests may be issued before receiving any responses from the server. Having a single persistent, pipelined connection allows the HTTP/1.1 to reach a throughput approaching the bandwidth of the network while only sending the minimal number of packets.

2.5 *ns* – Network Simulator

ns – 2.1 is one of the most commonly used simulators today, in the networking research community, mostly because of its open source. The simulator is a discrete event simulator originally developed at LBL (Lawrence Berkeley Laboratory) at the University of Berkeley, within the VINT (Virtual InterNetwork Testbed) project. The main reason why this simulator has been frequently used is that it has substantial support, and accurate implementations, for TCP as well as other transport protocols over conventional wired networks. Because of its good TCP implementations, it has been often used to simulate and evaluate TCP performance.

Berkeley released the initial code that made wireless network simulations possible in *ns*. That code provided some support to model wireless LANs, but was fairly limited. As a result of the Monarch project at Carnegie Mellon University, the simulator was extended with support for *node mobility*, a realistic *physical layer*, *radio network interfaces* and an implementation of the *IEEE 802.11 DCF MAC protocol*. It was this contribution that made it possible to perform wireless simulations with *ns*.

2.5.1 Simulator Internals

ns is an object oriented simulator, written in C++, with an OTcl interpreter as front-end. *ns* uses two languages because the simulator needs to do two different kinds of things. On one hand, detailed simulations of protocols requires a system programming language which can efficiently manipulate bytes, packet headers, and implement algorithms that run over large data sets. For these tasks, run-time speed is important and turn-around time (run simulation, find bug, fix bug, recompile, re-run) is less important. On the other hand, a large part of network research involves slightly varying parameters or configurations, or quickly exploring a number of scenarios. In these cases, iteration time (change the model and re-run) is more important. Since, configuration runs once (at the beginning), run-time of this part of the task is less important.

The *ns* Manual advises to use OTcl

- for configuration, setup, and “onetime” stuff
- if you can do what you want by manipulating existing C++ objects

and use C++

- if you are doing anything that requires processing each packet of a flow
- if you have to change the behavior of an existing C++ class in any ways that weren't anticipated

2.6 Nam – The Network Animator

Nam [6] is a Tcl/Tk based animation tool that is used to visualize the *ns* simulations and real world packet trace data.

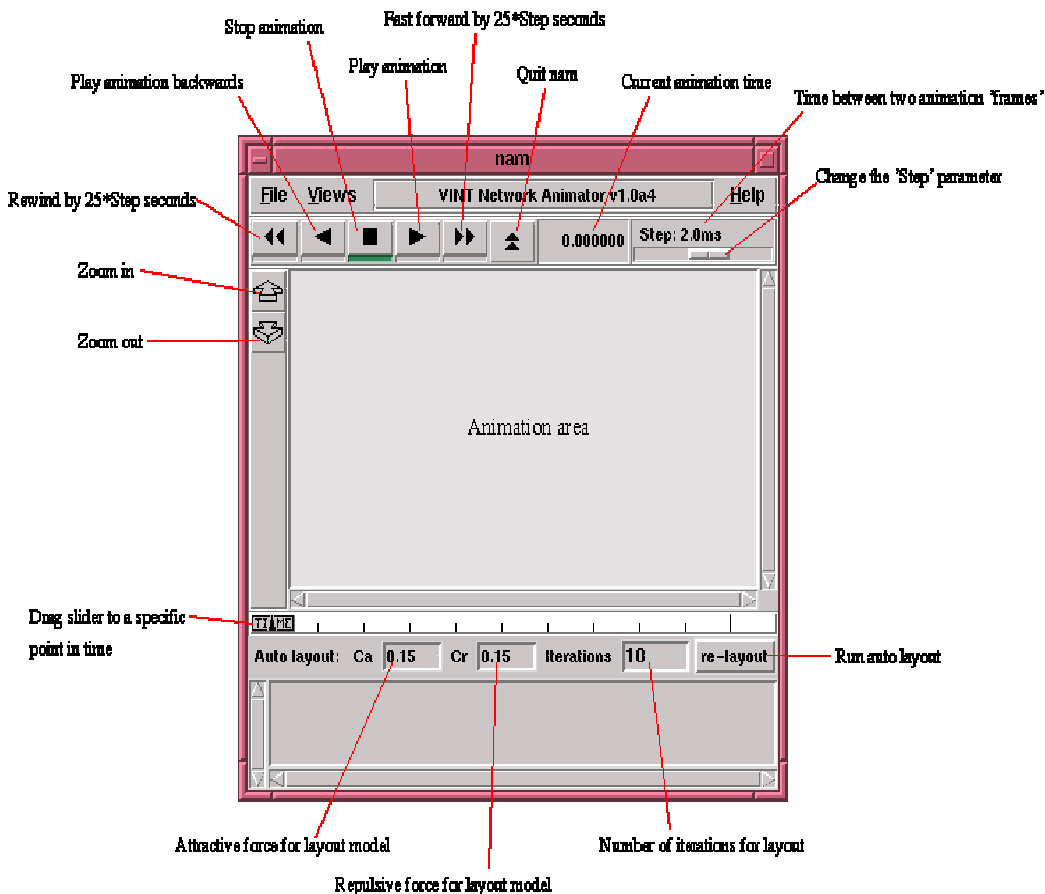


Fig. 2.1 nam Interface

The first step to use *nam* is to produce a “*nam trace*” file. This is usually created by the simulation script itself. The “*nam trace*” file contains topology information like nodes, links queues, node connectivity etc. as well as packet trace information.

The underlying design constraints for *nam* were that it is able to handle large amounts of trace data and that its animation primitives be adaptable so that it may be used in different types of network visualization. As a result, internally *nam* reads information from a file and keeps only a minimum amount of animation event information in memory. Its animation event has a fairly simple and consistent structure so that it can visualize many different situations.

Chapter 3

The Empirical Model of HTTP Network Traffic

Simulations have been a long-used tool for the evaluation of computer networks. In order for simulations to yield useful performance data, however, they require accurate models of the system under study and expected workload to be placed upon that system. In particular, workloads need to capture the various characteristics of network applications. Simulation studies of the HTTP environment require a model of the traffic patterns of the World Wide Web, in order to investigate the performance of this popular application. This chapter provides an outline of the methodology and the model as put forward in [10], with comments to support HTTP/1.1 and the wired-cum-wireless model under study.

3.1 Methodology

A packet trace-based approach was used for the model, principally because it aided in capturing the behavior of individual users and allowed the methodology to be used with the most popular currently deployed HTTP client. This approach did lead to the loss of higher level information such as the actual file accessed, but it was felt that such a characterization was not essential to a network workload model. Since most HTTP servers bind to the well known TCP port 80, by looking for all TCP packets to or from this well known port, the vast majority of HTTP traffic was believed to have been captured.

The freely-available *tcpdump* packet capture utility running on a DEC Alpha 3000/300 to record packet headers on a shared 10 Mbps Ethernet in the Computer Science Division at the University of California at Berkeley, during four periods in late 1995. This procedure saved the TCP and IP headers of each packet; a small number of payload bytes were also captured. These data were saved to disk for off-line processing.

The subnet examined was a stub network (no transit traffic), one of a dozen or so in use in the Computer Science Division. There were approximately a hundred hosts on this subnet; the majority of them being desktop UNIX workstations, each principally used by a single user. The user community consisted primarily of Computer Science graduate students. In the absence of statistics, *Netscape Navigator* was considered the most prevalent Web client used in the environment. There were also several WWW servers on the subnet, associated with various research groups.

The table below summarizes the traffic traces that were gathered for the study. The first three traces were collected as part of an effort to examine various types of network traffic (not just HTTP traffic) and the packet counts from these traces include

only those packets attributable to HTTP. The last traffic trace collected HTTP packets only. From these streams of packets, only those were extracted which comprised of HTTP connections originating from clients on the local network.

Start Time	End Time	Number of HTTP Packets
Tue Sep 19 16:12:33 1995	Thu Sep 21 07:53:22 1995	186068
Wed Oct 11 09:48:53 1995	Thu Oct 12 14:10:16 1995	458264
Wed Nov 1 11:22:47 1995	Thu Nov 2 10:53:12 1995	369671
Mon Nov 20 11:13:36 1995	Sun Nov 26 05:28:17 1995	676256

Table 3.1: Summary of Traffic Traces

Although complete figures for packet loss were not collected, a packet loss of approximately 6000 out of the 44,000,000 packets traced, were recorded during the November 1, 1995 trace (before filtering to isolate HTTP packets). These figures yielded a packet loss ratio of 0.014 %. Similar packet capture experiments using the same hardware and network produced loss figures consistent with the November 1, 1995 trace.

Comments: The above methodology suits the current study. So, the traffic traces already collected may be used for formulating a new model. But in a wireless network, significant packet loss is expected and this will need to be taken into account while modeling. It may also be noted that most of the networks outside educational or research institutions are not UNIX oriented, so it would be better to consider *Microsoft Internet Explorer* as the most prevalent Web client.

3.2 The Model

This section presents the various components of the model, which are summarized in the table below:

Quantity	Units	Description
Request Length	Bytes	HTTP request length
Reply Length	Bytes	HTTP reply length
Document Size	Files	Number of files per document
Think Time	Seconds	Interval between retrieval of two successive documents
Consecutive Document Retrievals	Pages	Number of consecutive documents retrieved from any given sever
Server Selection	Server	Relative popularity of any Web server, used to select each succeeding server accessed

Table 3.2: Quantities Modeled

Comments: The *Document Size* component of the original model needs to be done away with since in HTTP/1.1, persistent connection is used. It may be replaced with a

new component *Number of Requests Pipelined*, with the unit *Requests*. It will be used to describe the total number of requests pipelined by the HTTP client to be sent to the server. Also, no unit should be specified for the component *Server Selection*, which is felt to be unnecessary since this component specifies a relative quantity.

3.3 Results

The traffic traces and subsequent analysis, led to the derivation of various probability distributions for the different components of the model. The table below summarizes the more interesting facets of the measurements:

1. HTTP request sizes show a bimodal distribution.
2. HTTP reply sizes have a heavy – tailed distribution, and tend to be larger than request sizes.
3. A simple heuristic based on timing can be used to group individual files into documents.
4. The number of files per document tends to be small; 80% of documents required less than four file transfers.
5. HTTP requests to retrieve the first file of a multi – file Web document tend to be longer than subsequent requests.
6. Files retrieved as the first file of a multi – file Web document tend to be larger than subsequent files.
7. The number of consecutive documents retrieved from a given server tends to be small. 80% of visits to a server’s document space resulted in fewer than six documents being retrieved.

Table 3.3: Selected Measurement Results

Comments: There should be a few changes in the results observed with the proposed modifications and changing web content. Result 3 becomes redundant with pipelining. Result 4 should show an increase in the number of files per document as today’s web pages are very rich in content. This will have a cascading effect in the number of request messages being pipelined. Changes are also expected in Result 6 for similar reasons and in Result 7 for the simple fact that today’s Web sites offer specialized services.

3.4 Model Representation

CDF (Cumulative Distribution Function) representations were maintained for all distributions except a Zipf’s Law substitute to the *Server Selection* distribution. To use the model in a network simulator, it will need the simulation of activities of both the HTTP clients and the HTTP servers in the network.

Comments: In the top ten servers observed in the original study, only four servers were local, but in our model, all servers will be local. It will be interesting to see if it affects the difference in frequency counts of the top ten servers. If the difference is not striking, Zipf’s Law can well be considered a very bad substitute.

Chapter 4

Performance Evaluation

A source of ongoing frustration for users of the World Wide Web is the latency associated with data retrieval. Beyond a client's immediate connection to the Internet, there are essentially two possible sources of latency: the network and the server. Bottlenecks in the network exist due to congestion and/or limited bandwidth links along the path from the client to the server. Bottlenecks at the server occur when the server is heavily utilized either in the CPU or the disk system. This study concentrates on the latency associated with the hybrid network.

Version 1.1 of the HyperText Transfer Protocol (HTTP) was principally developed as a means for reducing both document transfer latency and network traffic. The rationale for the performance enhancements in HTTP/1.1 is based on the assumption that the network is the bottleneck in web transactions. This remains a valid assumption specifically for a wireless network which is inherently associated with high latency and low bandwidth.

[3] showed that on a Ricochet wireless network, which is based on IEEE 802.11, HTTP/1.1 doubles throughput over HTTP/1.0 and decreases the number of packets sent by 60 %. This is a significant improvement by any standard. But the study was performed on a relatively simple set up, consisting of a single client and a single server, a simple point to point network. So, this performance cannot be assumed to be true in case of a more complicated wireless network, more so in a network model like the one under study. This is because of the nature of the model where there is more than one client and at least two clients transmitting simultaneously and the resulting interference of their signals.

In addition, there is also the factor of transmitting over two different hops – client to gateway in a wireless environment and gateway to server in a wired environment. [2] shows that there is very little performance difference between HTTP/1.0 and HTTP/1.1 as far as network demands in a wired LAN is concerned. Since, the improvement in HTTP/1.1 over HTTP/1.0 come at a price, using it in the wired LAN may not be justified. If this is the case, we should consider some kind of connection management policy in the Gateway. This means the conversion of HTTP/1.1 messages to HTTP/1.0 messages when traffic is from the wireless domain to the wired domain and vice-versa.

The next section describes a few metrics which will help in determining the performance of HTTP over the model under study. This can be achieved by a detailed comparison of the results obtained as a part of this study with results obtained from similar studies.

4.1 Metrics

The metrics to be used for the performance evaluation are

- **Average Throughput (AT):** Measures the number of kilobytes (KB) per second delivered over the medium. This measure corresponds to the entire transfer in a single TCP connection.
AT = Total number of kilobytes received / Simulation time
Unit is KB/sec.
- **Packet Loss (PL):** Measures the number of packets discarded at the medium access layer.
PL = Number of packets lost / Total number of packets
Unit is %.
- **Average Access Delay (AAD):** Measures the time it takes to transmit a request message across to the HTTP server.
AAD = Sum of all access delays / Number of requests
Unit is ms (milliseconds).

Chapter 5

Experimental Setup

5.1 The Model

Simulation in simple terms is the act of performing experiments on a model of a given system. By a model, we mean a representation of the system under investigation. In our case, it is a wired-cum-wireless network. It has been modeled at the system level and involves analyzing the entire system from a performance standpoint.

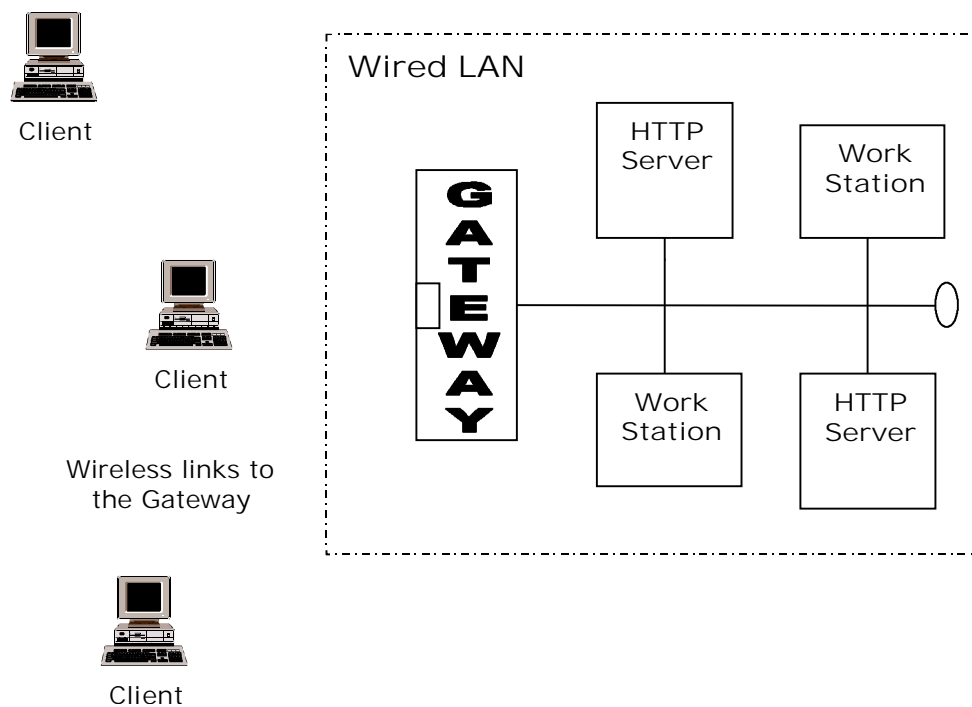


Fig. 5.1 Model under Study

The above figure shows the typical model to be used for our study. Here, the clients will be in a wireless LAN environment and the HTTP server(s) will be part of a wired LAN. The clients communicate with the desired HTTP server via a Gateway which is part of the wired LAN as well as the wireless LAN. The Gateway thus acts as the interface between the two dissimilar environments. The clients are in a fixed position at any given time. It is further assumed that the clients are independent of each other and can issue requests simultaneously. In fact, to obtain better results, it is made very

certain that at any given time, at least two clients are in the process of making requests. This model gives a very realistic view of an actual environment.

The clients do not use a cache because in spite of caching being widely employed, it is felt that it should not be employed while measuring performance as caching itself is a way of improving performance. Moreover, the study concerns itself with the performance of the underlying network and caching can undermine the results obtained. Also, packet loss, if any, will only be due to collisions and nothing else. This is an important assumption for the simulation

5.2 How the Web Pages are retrieved by the Clients from the HTTP Server

The Client passes a Universal Resource Locator (URL) message to a Domain Name System Server (DNSS) which resides in the Gateway node. The DNS replies with an Internet Protocol (IP) address of the server, which contains the web page the client desires. This transaction is usually done using the User Datagram Protocol (UDP). The client then issues a GET message to the server, with the name of the desired file. The server responds with the desired file using a REPLY message. This transaction is through TCP, the Transmission Control Protocol. Since, there are many inline objects in the requested file (e.g., inline GIF images), the client sends HTTP GET messages for each object on the ongoing TCP connection. (The above is the mechanics of secondary retrievals used by HTTP/1.1). Once the complete page along with the inline objects are received, the connection is closed.

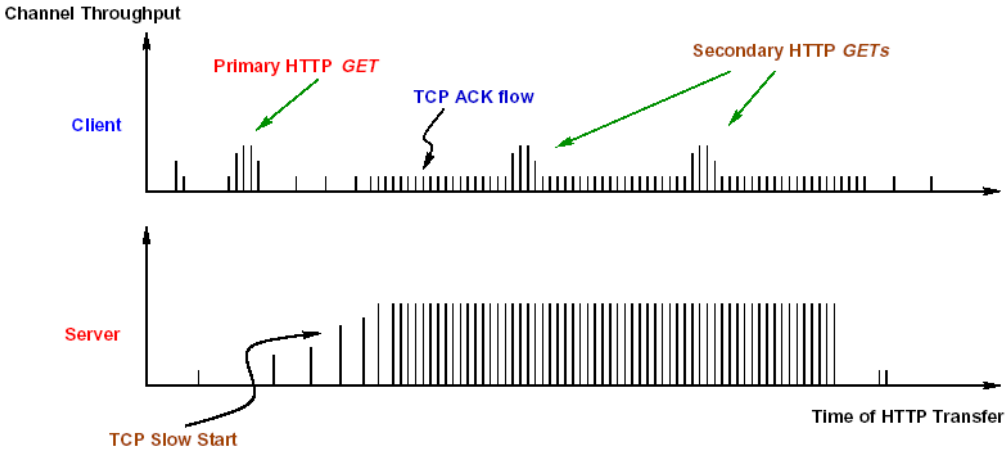


Fig. 5.2 The WWW Page Transfer Process [4]

The transfer process in HTTP/1.0 is a bit different, where separate connections are opened for each object in a web page. After each object retrieval, the corresponding connection is closed.

5.3 Transfer Methodology

Since, there are two distinct parts in the model under study, a wireless part and a wired part, the medium access strategy for each part will be different.

CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) is used for transmitting messages in the wireless channel. For the wireless part, one basic assumption is that, at any point of time at least two clients are generating requests. So, whenever a client has to send an HTTP request, it first listens to ensure that the channel is free. If the channel is clear, the request is made to the Gateway, Otherwise, it chooses a random “back off” factor which determines the amount of time, the client must wait until it can transmit its request. During periods, when the channel is clear, the clients decrement its “back off” counter. (When the channel is busy, the “back off” counter is not decremented). The client transmits its request after the “back off” counter reaches zero. Since, the probability that two clients will choose the same “back off” factor is minimal; collisions between transmitted packets are minimized. The actual transmission is again a combination of a few message exchanges. The client first sends out a short ready-to-send (RTS) packet containing information on the length of the packet. The Gateway, if it hears the RTS, responds with a short clear-to-send (CTS) packet. After this exchange, the actual request is transmitted. When the packet is received successfully, as determined by a Cyclic Redundancy Check (CRC), the Gateway transmits an acknowledgement (ACK) packet. The message exchanges eliminates the scenario wherein two distant clients which are not within each other’s radio range to sense their respective transmission status sends requests at the same time, resulting in the collision of packets. This is known as the hidden node problem.

The second part of the transmission will be on the wired part. For this CSMA/CD (Carrier Sense Multiple Access with Collision Detection) is used. Once the Gateway receives the request, after generating the required IP address, it senses the channel to transmit the request to the server. If the channel is free, the Gateway transmits. If due to some reason, another station of the LAN accesses the transmission media, there will be a collision. Transmission is then aborted and the detecting station sends jamming signals. The Gateway waits a random amount of time before retransmitting, thus reducing the likelihood of reoccurring collisions. If successive collisions continue to occur, the random time-out interval is doubled. When there are no more collisions, the transmission is completed. If more requests are pipelined, the procedure continues, else it is discontinued.

The reply part from the HTTP server is analogous with a few role reversals and is not elaborated here.

Figures 5.3 and 5.4 in the coming pages give the state diagrams for CSMA/CD and CSMA/CA respectively to reflect the state changes that the nodes in question (client or server) undergoes at various points of time in the transmission process.

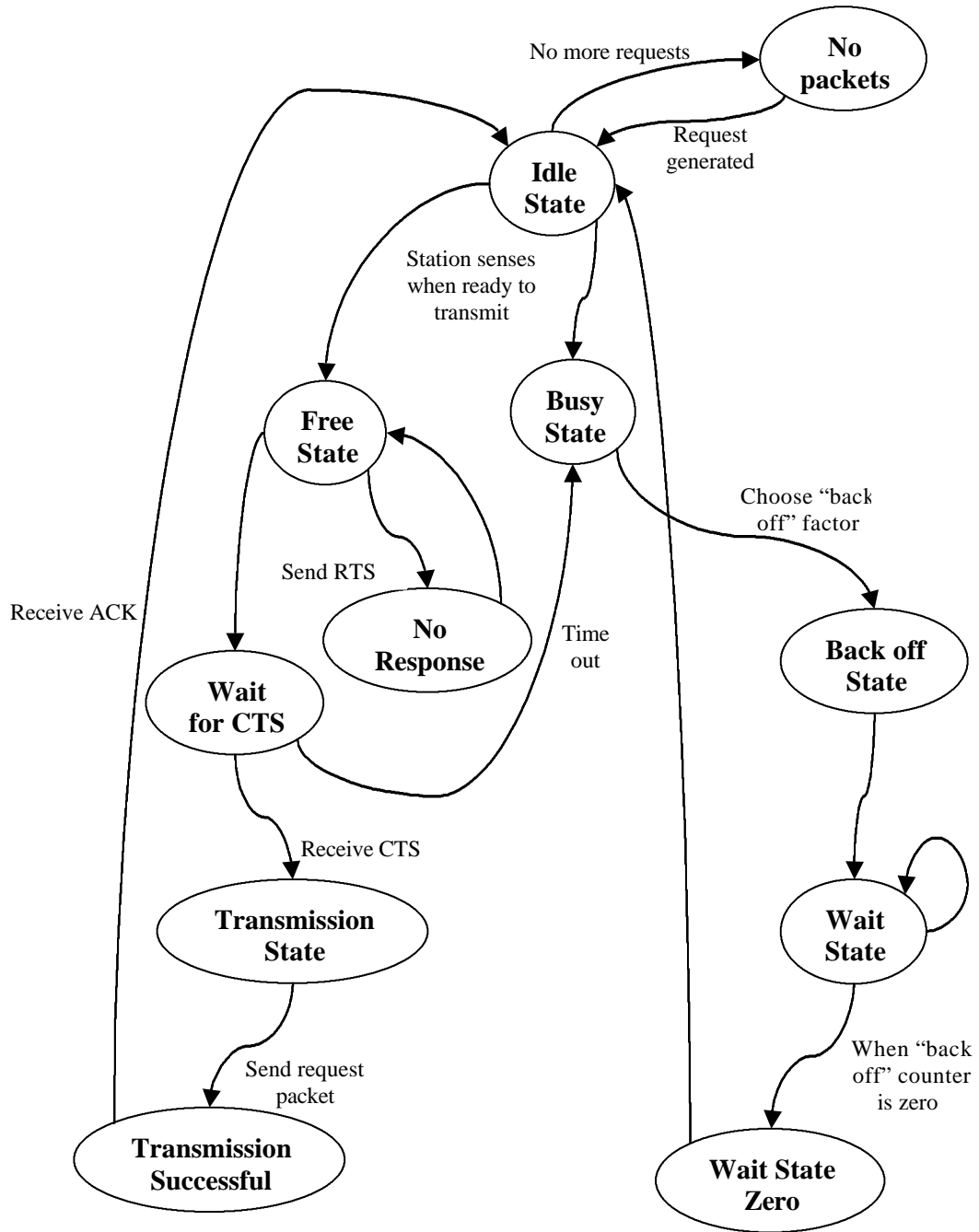


Fig. 5.3 State Diagram for CSMA/CD

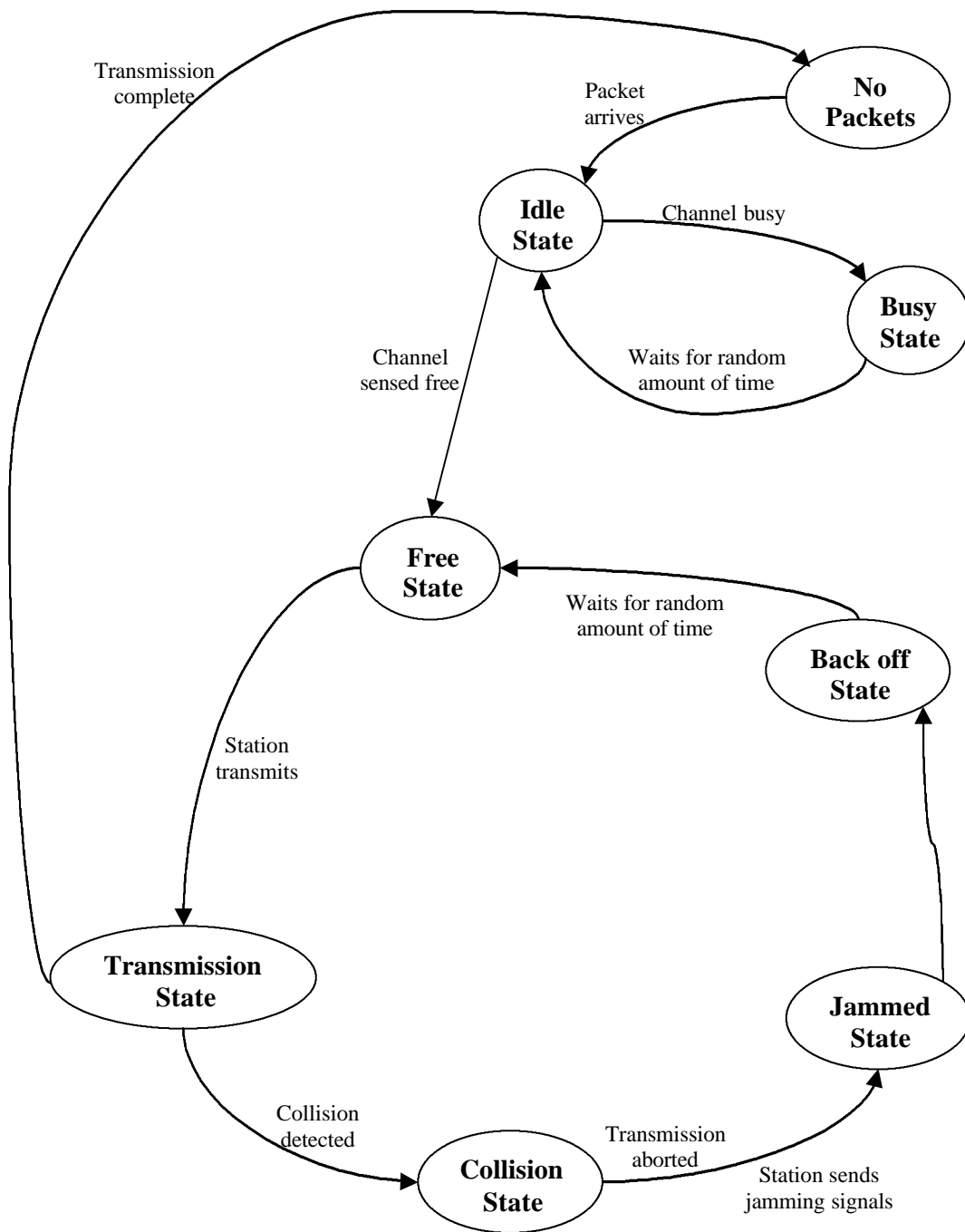


Fig. 5.4 State Diagram for CSMA/CA

5.4 Web Browsing Statistics

In order to drive a simulated HTTP session, the statistics of the Web transfer process must be known with a reasonable degree of accuracy. For the purposes of our present simulation, I have relied on the measurements taken in [10], which have already been outlined in Chapter 3 of this thesis. The statistics relevant to this simulation work are: client think time, HTTP GET length, HTTP REPLY length, inline object count, and consecutive file retrieval count. The average of all these statistics is shown below:

Parameter	Think Time	GET	REPLY	Objects	Consecutive Hits
Average	1313	398	17932	2.9	4.1
Units	Seconds	Bytes	Bytes	None	None

Table 5.1: Summary of HTTP Statistics

The CDFs described in [10] are already integrated into the network simulator *ns*. These allow the simulation of HTTP sessions in the model under study.

5.5 Implementation

There are two ways of implementing the model for the purpose of measuring the performance of HTTP over the said model. One is to go for a physical implementation and the other is to simulate the model. What follows is a brief overview of both. This study focuses on the simulation for reasons mentioned in the abstract.

5.5.1 Physical Implementation

In this section, we discuss the hardware and software that can be used to run the experiments. The IT Lab -2 of the Department of Information Technology is used to visualize the situation. The Gateway of the lab, *luit* represents the Gateway of our model. The web server of our University, *baan* and *manas* of IT Lab - 1 are the two HTTP servers. The entire network of the University represents the wired LAN. The gateway needs to be equipped with the necessary wireless cards as also the other clients. The HTTP servers are *linux* based and are loaded with *Apache* Web servers. The *linux* based clients are loaded with the *Netscape Navigator/Communicator* web client while *Microsoft Windows* clients are loaded with the proprietary *Internet Explorer*. The clients are configured so that they do not cache web pages.

5.5.2 Simulation

The simulation is to be performed on *ns*. The version of *ns* available is 2.1b8a. The basic functionalities for creating a wired as well as a wireless LAN exist. But creating the environment to our specific requirements is not simple with the available facilities. Using the existing facilities to create the hybrid network in *ns* gave unpredictable results in some test simulations. There is also facility for creation of HTTP Server/Client. But it also requires the creation of a Web Cache, which is not required in our study.

An outline of a simulation test script is given below:

```

set ns [new Simulator] ;# initialize the simulation

$ns node-config -addressType hierarchical
AddrParams set domain_num_ 2 ;# number of domains
lappend cluster_num 1 1 ;# number of clusters in each domain
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 4 4 ;# number of nodes in each cluster
AddrParams set nodes_num_ $eilastlevel ;# for each domainset trace

set trace [open out.tr w] ;# open a trace file
trace-all $trace
.
.
.
.
# setting some options for nodes in wired LAN
set opt_wired(bw) 10Mb
set opt_wired(delay) lms
set opt_wired(ll) LL
set opt_wired(ifq) Queue/DropTail
set opt_wired(mac) Mac/802_3
set opt_wired(chan) Channel
set opt_wired(tcp) TCP/Reno
.
.
.
# setting some options for wireless LAN
set opt_wireless(channel) Channel/WirelessChannel
set opt_wireless(prop) Propagation/TwoRayground
set opt_wireless(netif) Phy/WirelessPhy
set opt_wireless(mac) Mac/802_11
set opt_wireless(ifq) Queue/DropTail/PriQueue
set opt_wireless(ll) LL
set opt_wireless(ant) Antenna/OmniAntenna
.
.
# setting configurations for LL (wireless)
LL set delay_ 25us
LL set bandwidth_ 0 ;# not used
.
.
# setting for number of wireless nodes
set opt(nn) 3

# creating the wired LAN
set temp {0.0.0 0.1.0} ;# hierarchical addresses to be used
for {set i 0} {$i < $num_wired_nodes} {incr i} {
    set W($i) [$ns node [lindex $temp $i]]
    lappend wired_nodelist $W($i)
}
set wired_lan [$ns newLan $wired_nodelist $opt_wired(bw)
                $opt_wired(delay) \
                -llType $opt_wired(ll) \
                -ifqType $opt_wired(ifq) \
                -macType $opt_wired(mac) \
                -chanType $opt_wired(chan)]

```



```

.
.

#create Gateway node
set temp {1.0.0 1.0.1 1.0.2 1.0.3}    ;# hier address to be used for
                                   ;# wireless domain

set Gateway [ $ns node [lindex $temp 0]]
$Gateway random-motion 0             ;# disable random motion

#provide some co-ordinates (fixed) to base station node
$Gateway set X_ 1.0
$Gateway set Y_ 2.0
$Gateway set Z_ 0.0

# Note there has been a change of the earlier AddrParams
# function 'set-hieraddr' to 'addr2id'.

#configure for wirelessnodes
$ns node-config -wiredRouting OFF

# now create wireless nodes
for {set j 0} {$j < $opt(mn)} {incr j} {
    set wireless_node($j) [ $ns node [lindex $temp \
        [expr $j+1]] ]
    $wireless_node($j) base-station [AddrParams addr2id \
        [$Gateway node-addr]] ;# provide each wireless node with
                               ;# hier address of its base-station
}
.
.
.
.
.
$ns run

```

Chapter 6

Conclusions

The work behind this report was an attempt to systematically study and analyze the performance of HTTP over a wired-cum-wireless network. A wired-cum-wireless network is generally perceived as a network which is an internetworking of different wired networks by using a wireless part as a bridge. But, in this report, a wired-cum-wireless network is given a different meaning as represented by the model under study. This itself brings to the fore a fresh but relatively simple approach.

The model under study portrays a very realistic view of an actual environment and as such the study assumes additional significance. This stands out in comparison to other related works [2], [3].

The study also seeks to modify an important piece of work pertaining to modeling of the traffic patterns of the World Wide Web [10], though at a very theoretical level. This should provide new directions.

Reservations exist to the suitability of HTTP/1.1 as the only protocol of choice for wired-cum-wireless networks.

As many studies [12] indicate, one of the primary reasons for increased latency in wireless networks is the use of TCP. TCP stream – based sockets are desirable for reliable communication, but TCP is not optimized for performance on wireless networks. The extensive error checking of TCP can degrade performance over wireless links. A new or modified TCP protocol optimized for wireless networks is needed for reliable wireless communication. If that is the case, there will be a new problem as far as wired-cum-wireless networks are considered of dealing with two different transport protocols within the same setup. For example, in our case it will put extra load on the gateway as it will have to do the extra job of protocol conversion. Therefore, a new TCP like protocol is desirable for this kind of networks. This would require the accurate quantification of the MAC induced delay (MID) for HTTP downloads.

Chapter 7

Future Work

The existing facilities available in *ns-2.1* are not sufficient to accurately simulate our model. This strongly favours suitable additions and modifications to the facilities already available in *ns-2.1*. It will also be interesting to do an authentic evaluation of the mechanisms in an actual environment.

The modifications proposed for the model in [10] needs careful evaluation and experimentation. It would be an excellent idea to rework the whole thing, starting from collecting fresh packet traces. This is observed keeping in mind the fact that the nature of web content as well as the capabilities of Web clients and HTTP servers has undergone a sea change since the early days of the WWW.

Sufficient scope exists for extending and refining the model to make it more generalized. It can be taken up. Moreover, it can also be adapted to undertake other studies.

Appendix A

Glossary

access method: A technique used in telecommunications.

carrier sense: A device capable of detecting a constant frequency.

client: A computer that a user logs into to access LAN – attached resources and services.

collision: An event where two or more devices simultaneously perform a broadcast on the same medium. This term is used in Ethernet networks and also in networks where broadcast technology is implemented.

collision detection: The ability to detect simultaneous transmission attempt.

CSMA/CA: Part of the IEEE 802.11 standard, CSMA/CA is similar to CSMA/CD except that collisions cannot be detected in wireless environments as they can in wire-based environments. Before transmitting, workstations wait a predetermined amount of time to avoid collisions, and set up a point-to-point wireless circuit to the destination workstation.

CSMA/CD: It is the access methodology used by Ethernet media sharing LANs.

DNS: The Domain Name System has been created to provide the following key services: Uniquely identify all hosts connected to the Internet by name; Resolve, or translate, host names into IP addresses (and vice versa); Identify which services are offered by each host such as gateway or mail transfer, and to which networks these services are offered.

Ethernet: A data link level protocol. Although strictly speaking, Ethernet and 802.3 are conflicting standards, the term Ethernet is commonly used to refer to any IEEE 802.3 compliant network.

gateway: A hardware/software solution that allows for the translation between different physical signals and/or protocols.

IEEE: Acronym for Institute of Electrical and Electronics Engineers.

Internet: A collection of networks, routers, gateways, and other networking devices that use the TCP/IP protocol suite and function as a single, cooperative virtual network.

Internet Protocol (IP): A protocol used to route data from its source to its destination. A part of TCP/IP protocol.

IP address: The 32-bit address assigned to a host. It is a software address that on little-i internets is locally managed, but on the big-I Internet is dictated to the user (entity desiring access to the Internet). Actually an IP address consists of a network portion and a host portion. The partition makes routing efficient.

latency: The time it takes for a data packet to move across a network connection. While a packet is being sent, there is "latent" time, where the sending computer waits for a confirmation that the packet has been received.

medium: A physical carrier of electrons or photons. The medium may be hard, as in a type of cable, or soft, as in microwaves, for examples.

protocol: A set of rules that govern communication between hardware and/or software components.

RFC: Request for Comments. Proposed and accepted TCP/IP standards.

trace: A record of events captured and used to troubleshoot hardware and/or software or for analysis of the same.

Bibliography

- [1] Almquist Andreas, Anders Lindgren. *Quality of Service Schemes for IEEE 802.11 – A Simulation Study*. Master's Thesis, Luleå University of Technology, May 2001
- [2] Barford Paul, Mark Crovella. *A Performance Evaluation of Hyper Text Transfer Protocols*. Technical Report BU-TR-98-016, Boston University, Computer Science Department, Boston, October 1998.
- [3] Cheng Stephen, Kevin Lai, Mary Baker. *Analysis of HTTP/1.1 Performance on a Wireless Network*. Technical Report CSL-TR-99-778, Computer Systems Laboratory, Departments of Electrical Engineering and Computer Science, Stanford University, Stanford, February 1999
- [4] Connors Dennis P., Gregory J. Pottie. The Performance of HTTP Over Satellite Random Access Channels. In *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference*, San Diego, CA , January 2000.
URL: <http://citeseer.nj.nec.com/452942.html>
- [5] Das Sumitabha, *Unix Concepts & Applications, Second Edition*. Tata McGraw-Hill Publishing Company Limited, New Delhi. c1998.
- [6] Fall Kevin, Kannan Varadhan. *The ns Manual*. VINT Project, UC – Berkeley and LBNL, April 2002.
URL: <http://www.isi.edu/nsnam/ns/>
- [7] Fielding, et al. *Hypertext Transfer Protocol – HTTP/1.1*. RFC 2616, June, 1999.
- [8] Goldman James E., Philip T. Rowles. *Local Area Networks, A Business-Oriented Approach, Second Edition*. John Wiley & Sons, Inc. c2000.
- [9] Jhaveri Alay Pankaj. Networking Without Wires, *Science Reporter, Vol. 39 No. 6*. Pp 23 – 5. National Institute of Science Communication (CSIR), New Delhi – 110012.
- [10] Mah Bruce A. An Empirical Model of HTTP Network Traffic. In *Proceedings of IEEE INFOCOM '97* 1997
URL: <http://citeseer.nj.nec.com/369956.html>
- [11] Miller Michael A. *Data & Network Communications*. Delmar Thomson Learning. c2000.
- [12] Pentikousis K. *A survey of TCP in wired-cum-wireless environments*. State University of New York at Stony Brook, Department of Computer Science.

Bibliography

[13] Sadiku Mathew N.O., Ilyas Mohammad. *Simulation of Local Area Networks*. CRC Press Inc. c1995.

[13] Zeltserman Dave, Puoplo Gerard. *Building Network Management Tools with Tcl/Tk*. Prentice Hall PTR, NJ 07458. c1998.