

Simulation Evaluation of Wireless Web Performance in an IEEE 802.11b Classroom Area Network

Guangwei Bai Carey Williamson
Department of Computer Science
University of Calgary
Email: {bai,carey}@cpsc.ucalgary.ca

Abstract

This paper presents a simulation study of an IEEE 802.11b wireless LAN (WLAN) used as a classroom area network. The simulation is conducted using OPNET Modeler 9.1. The first part of the paper discusses parameterization and validation of the simulation model, based on empirical measurements in a wireless classroom environment. The second part of the paper presents a simulation study designed to estimate the number of clients that can be supported in the WLAN, as well as the user-perceived Web response time as a function of network load. The simulation results show that an IEEE 802.11b WLAN can easily support up to 100 clients with modest Web browsing activities. The results also show that protocol features such as persistent connections provide a significant performance advantage in a WLAN environment.

1. Introduction

Wireless access points are now commonplace on many university campuses [5, 6, 7, 17]. Technologies such as IEEE 802.11b wireless LANs (WLANs) have revolutionized the way people think about networks, by offering users freedom from the constraints of physical wires. Mobile users are interested in exploiting the full functionality of the technology at their fingertips, as wireless networks bring closer the “anything, anytime, anywhere” promise of mobile networking.

A natural step in the wireless Internet evolution is the convergence of technologies to form the “wireless Web”: the wireless classroom, the wireless campus, the wireless office, and the wireless home. Educators can embrace wireless Internet access to enhance the learning experience in the classroom for students with wireless laptops, through on-line access to lecture notes, demos, examples, quizzes, assignments, and supplementary reading material.

In this paper, we explore wireless Web performance in the context of classroom area networks. Our study is based in part on prior measurements from a small-scale wireless classroom experiment [2], wherein a wireless Web server was used in a “legacy classroom” environment to deliver selected course content to a graduate class with 13 students.

Our paper uses simulation to study a larger-scale classroom area network scenario. We use the OPNET Modeler 9.1 simulation environment, with its detailed models of IEEE 802.11b, TCP/IP, and HTTP. We parameterize the simulation model based on our previous classroom measurements, and validate the model against empirical measurements using simple Web workload models. We then build a model of browsing behaviour for a Web client, and use this model in a simulation study addressing the scalability of the classroom area network. Our experiments focus on the HTTP transaction rate and end-to-end throughput achievable in the wireless network environment, and the impacts of factors such as number of clients, Web object size, and persistent connections.

The simulation results in the paper show that an IEEE 802.11b WLAN can easily support up to 100 clients with modest Web browsing behaviour. Furthermore, HTTP protocol features such as persistent connections provide a significant performance advantage in a WLAN environment.

The remainder of this paper is organized as follows. Section 2 discusses background information on wireless Internet technologies, including IEEE 802.11b, TCP, and HTTP. Section 3 describes the simulation setup and methodology for our study. Section 4 presents the simulation results and analyses. Finally, Section 5 summarizes the paper and describes ongoing work.

2. Background and Related Work

2.1. The Web and Web Performance

Over the past ten years, the World Wide Web (WWW) has been the largest source of Internet traffic. The Web has made the Internet available to the masses, by providing location-independent, time-independent, and platform-independent access to information.

The Web relies primarily on three communication protocols: IP, TCP, and HTTP. The Internet Protocol (IP) is a connection-less network-layer protocol that provides global addressing and routing for datagram delivery on the Internet. The Transmission Control Protocol (TCP) is a connection-oriented transport-layer protocol that provides end-to-end data delivery across the Internet [15]. Among its many functions, TCP has flow control, congestion control, and error recovery mechanisms to provide reliable data transmission between sources and destinations. The robustness of TCP allows it to operate in many network environments. Finally, the Hyper-Text Transfer Protocol (HTTP) is a request-response application-layer protocol layered on top of TCP. HTTP is used to transfer Web documents between Web servers and Web clients. Currently, HTTP/1.0 [11] and HTTP/1.1 [12] are widely used on the Internet.

2.2. Wireless Internet and IEEE 802.11b WLANs

Wireless technologies are playing an increasingly prominent role in the global Internet infrastructure. One of the popular technologies in the wireless LAN market is the IEEE 802.11b standard. This popular “WiFi” (Wireless Fidelity) technology provides low-cost wireless Internet capability for end users, with up to 11 Mbps data transmission rate at the physical layer. The IEEE 802.11b standard defines the channel access protocol used at the MAC layer, namely Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). It also defines the frame formats used at the data link layer: 128-bit preamble, 16-bit Start-of-Frame delimiter, 48-bit PLCP (Physical Layer Convergence Protocol) header, followed by a 24-byte MAC-layer header and variable size payload, which can be used for carrying IP packets. Frames that are correctly received over the shared wireless channel are acknowledged (almost immediately) by the receiver. Unacknowledged frames are retransmitted by the sender after a short timeout (typically a few milliseconds), using the same MAC protocol.

2.3. Wireless Web Performance

The overall performance of the Web depends on the behaviours of Web clients, the Web server, and the network

in between. The primary challenge in the wireless Internet context is the characteristics of the wireless channel. Communication over wireless links often suffers from limited bandwidth, high error rates, and interference from other users on the shared channel. The obvious concern is that TCP and HTTP performance may degrade over wireless networks.

Our focus in this paper is on the performance of wireless Web access in a classroom area network. Our primary emphasis is on performance problems due to the wireless network bottleneck, and understanding how these problems affect user-perceived performance.

2.4. Related Work

There is a growing literature on wireless traffic measurement and Internet protocol performance over wireless networks [3, 5, 6, 13, 16, 17]. For example, Tang and Baker [16, 17] discuss wireless network measurements from two different environments: a metropolitan area network, and a local area network. More recently, Balachandran *et al.* [3] report on network performance and user behaviour for general Internet access by several hundred wireless LAN users during the ACM SIGCOMM conference in San Diego in 2001. They find that for this set of technology-literate users a wide range of Internet applications are used, user behaviours are diverse, and overall bandwidth demands are moderate. Kotz and Essein [7] characterize campus-wide wireless network usage at Dartmouth College, focusing on infrastructure mode using access points. Our own prior work focuses on the performance of standalone wireless Web servers in short-lived wireless ad hoc networks [2, 8].

3. Simulation Methodology

3.1. Simulation Environment

In this work, we use OPNET Modeler 9.1 for our network simulations. OPNET Modeler is a powerful communication system simulator developed by OPNET Technologies [9]. OPNET Modeler 9.1 assists with the testing and design of communications protocols and networks, by simulating network performance for wired and/or wireless environments.

The OPNET tool provides a hierarchical graphical user interface for the definition of network models. A network is constructed by graphically connecting network nodes via communications links. OPNET Modeler comes with an extensive model library, including application traffic models (e.g., HTTP, FTP, E-mail, Database), protocol models (e.g., TCP/IP, IEEE 802.11b, Ethernet), and a broad set of distributions for random variate generation (e.g., exponential, Er-

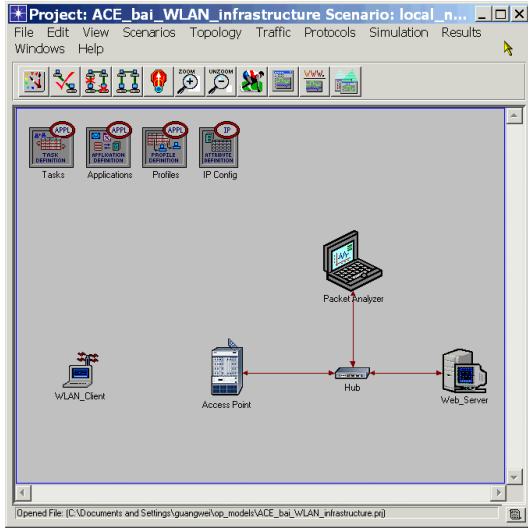


Figure 1. Single-client simulation scenario

lang, lognormal, Weibull, Pareto). There are also adequate facilities for simulation instrumentation, report generation, and statistical analysis of results.

3.2. Simulation Network Model

The IEEE 802.11 standard defines a set of wireless LAN protocols that deliver services similar to those found in wired Ethernet LAN environments. The IEEE 802.11 WLAN architecture is built around a Basic Service Set (BSS). A BSS is a set of stations that communicate with one another. When all the stations in the BSS can communicate directly with each other (without a connection to a wired network), the BSS is known as an *ad hoc* WLAN. When a BSS includes a wireless access point (AP) connected to a wired network, the BSS is called an *infrastructure* network. In this mode, all mobile stations in the WLAN communicate via the AP, providing access to stations on wired LANs and the world-wide Internet.

In our work, we use OPNET Modeler to model a simple infrastructure WLAN as shown in Figure 1. The network consists of a mobile client, a wireless Access Point (AP), an Ethernet-based Web server, an Ethernet Hub, and an OPNET ACE Packet Analyzer. The Web server is located on a 100 Mbps Ethernet LAN segment. The mobile client accesses content from the Web server via the AP, using the IEEE 802.11b protocol. (Additional clients are considered in later experiments.) The Ethernet hub allows the network monitor (ACE Packet Analyzer) to passively capture¹ and record all network activity to and from the Web server.

¹The current version of OPNET does not support ACE capturing traffic directly on the IEEE 802.11b WLAN.

The mobile client node represents a laptop with client-server applications running over TCP/IP. These applications make use of the WLAN connection, which can operate at either 1 Mbps, 2 Mbps, 5.5 Mbps, or 11 Mbps, as defined in IEEE 802.11b. We use only the 11 Mbps setting in our work. The Ethernet-based Web server node represents an HTTP server running over TCP/IP. The operational speed is determined by the wired link's data rate. In our configuration, the WLAN is the bottleneck. The ACE (Application Characterization Environment) node model is an add-on module to OPNET for capturing detailed packet traces, enabling the visualization and analysis of network traffic, and troubleshooting of network applications. This Packet Analyzer facilitates the real-time capture of all observed traffic in the simulation environment. In essence, it records all activity (i.e., frame transmissions, acknowledgements, collisions, and retransmissions). Decoding of the captured traces enables protocol analysis at the MAC, IP, TCP, and HTTP layers.

3.3. Simulation Design

A one-factor-at-a-time simulation design is used to study the impacts of many factors on wireless LAN performance and user-level Web performance. These factors include HTTP transaction rate, number of clients, Web object size, and HTTP/TCP protocol features. The simulation factors are summarized in Table 1. The values in bold font show the default levels used.

Table 1. Simulation factors and levels for wireless web simulation study

Factor	Levels
Number of Clients	1 , 2, 10, 20, 50, 100
Per-Client TCP Connection Request Rate (per second)	10 , 50, 80, 100, 200
HTTP Transfer Size (KB)	1 , 8, 32, 48, 64

3.4. Simulation Validation

Significant effort was spent on the parameterization and validation of the simulation model with our prior Web performance measurements in a WLAN environment [2, 8]. The goal was to make the OPNET simulation model match the empirical results closely.

Figures 2, 3, and 4 summarize the results from these validation tests. Figure 2 shows the structure of a typical HTTP transaction for a 1 KB Web object in OPNET, using HTTP/1.0. The first three packets constitute TCP's three-way handshake to open a connection reliably. The next few

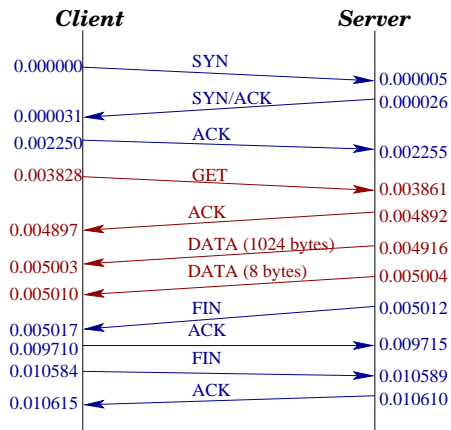


Figure 2. Example of HTTP/1.0 transaction

packets represent the exchange of the HTTP request (GET URL) and response (DATA). The final four packets represent TCP's closing FIN handshake. The entire HTTP transaction in Figure 2 takes about 10 milliseconds. The detailed timing structure was determined using the ACE Packet Analyzer.

The simulation results for the HTTP/1.0 transaction structure are quite consistent with observed empirical results, though some TCP implementations differ in the use of TCP DATA/ACK piggybacking and the use of TCP delayed acknowledgements. For example, the OPNET model uses separate DATA packets for the HTTP response header (8 bytes) and the Web object payload (1024 bytes), while most TCP implementations bundle these together into one packet. We obtained the closest match to the empirical traces when the OPNET simulation used the TCP Reno model, with a 1500-byte MTU at the IP layer, an initial congestion window (cwnd) size of 1 MSS, and an upper limit of 0.001 seconds for TCP delayed acknowledgements. Unless stated otherwise, these are the TCP/IP values used in all remaining simulation experiments.

Figure 3 shows the next set of validation results. These experiments assume a simplistic Web workload: a single Web client with a deterministic request arrival rate (i.e., successive requests are equally spaced in time, with a constant arrival rate) for a fixed-size 1 KB Web object, using HTTP/1.0 with non-persistent connections. The leftmost column of Figure 3 shows the empirical measurement results from this scenario for four different request arrival rates, ranging from 10 requests per second to 100 requests per second. The graphs show a time series representation of the TCP connection duration for each completed HTTP transfer. The second column of graphs shows the marginal distribution (also known as a frequency histogram, probability density function, or pdf) of the TCP connection duration. The last two columns present the simulation results

for the corresponding scenarios, with time series plots in the third column, and marginal distribution plots in the right-most column.

Figure 3 shows that there is close agreement between the simulation results and the empirical measurement results, at least at light (10 req/sec), medium (50 req/sec), and heavy (80 req/sec) load. The only anomaly that appears at these load levels is in the empirical measurement results in Figure 3(e) and (i). The spikes here are due to the X windows system running on the laptops, occasionally delaying a TCP connection handshake. There is no way to account for these operating system effects in the simulation model, so we simply ignore this (rare) event.

At higher loads (100 req/sec), the measurement results and the simulation results diverge. The empirical measurements in Figure 3(m) suffer from TCP packet losses. The losses occur at the (finite) link-layer queue of the client's wireless network interface, leading to TCP time-outs, retransmissions, and aborted connections. The simulation model appears to handle 100 requests per second just fine (Figure 3(p)), though the mean connection duration of 30 milliseconds indicates significant queueing delay somewhere in the system. More careful analysis shows that a large proportion of the TCP connections are reset (TCP RST). The reason for this is that OPNET Modeler 9.1 limits the maximum number of concurrent TCP connections between the same source and sink entities. If a new TCP connection is initiated too soon, an earlier one is reset.

At even higher request rates (e.g., 200 req/sec), the simulation model shows degraded performance because of excessive contention for the wireless channel. These results are similar to the empirical results for 100 req/sec, which are repeated in Figure 3(q), for easy visual comparison.

Figure 4 shows another set of validation results. These experiments assume a single Web client making 10 requests per second for a Web object, again using HTTP/1.0 with non-persistent connections. The size of the Web object is varied from one test to the next.

The leftmost column of Figure 3 shows the empirical measurement results from these tests, with Web object sizes ranging from 8 KB to 64 KB. The graphs show a time series representation of the TCP connection duration for each completed HTTP transfer. The second column of graphs shows the marginal distribution of the TCP connection duration. The last two columns present the OPNET simulation results for the same scenarios, with time series plots in the third column, and marginal distributions on the right.

There is qualitative agreement between the measurement results and simulation results in Figure 4, at least for 8 KB, 32 KB, and 48 KB transfers. Quantitatively, the simulation results report slightly higher TCP transfer times. For 64 KB transfers, two anomalies occur.

The first anomaly is in the empirical measurement re-

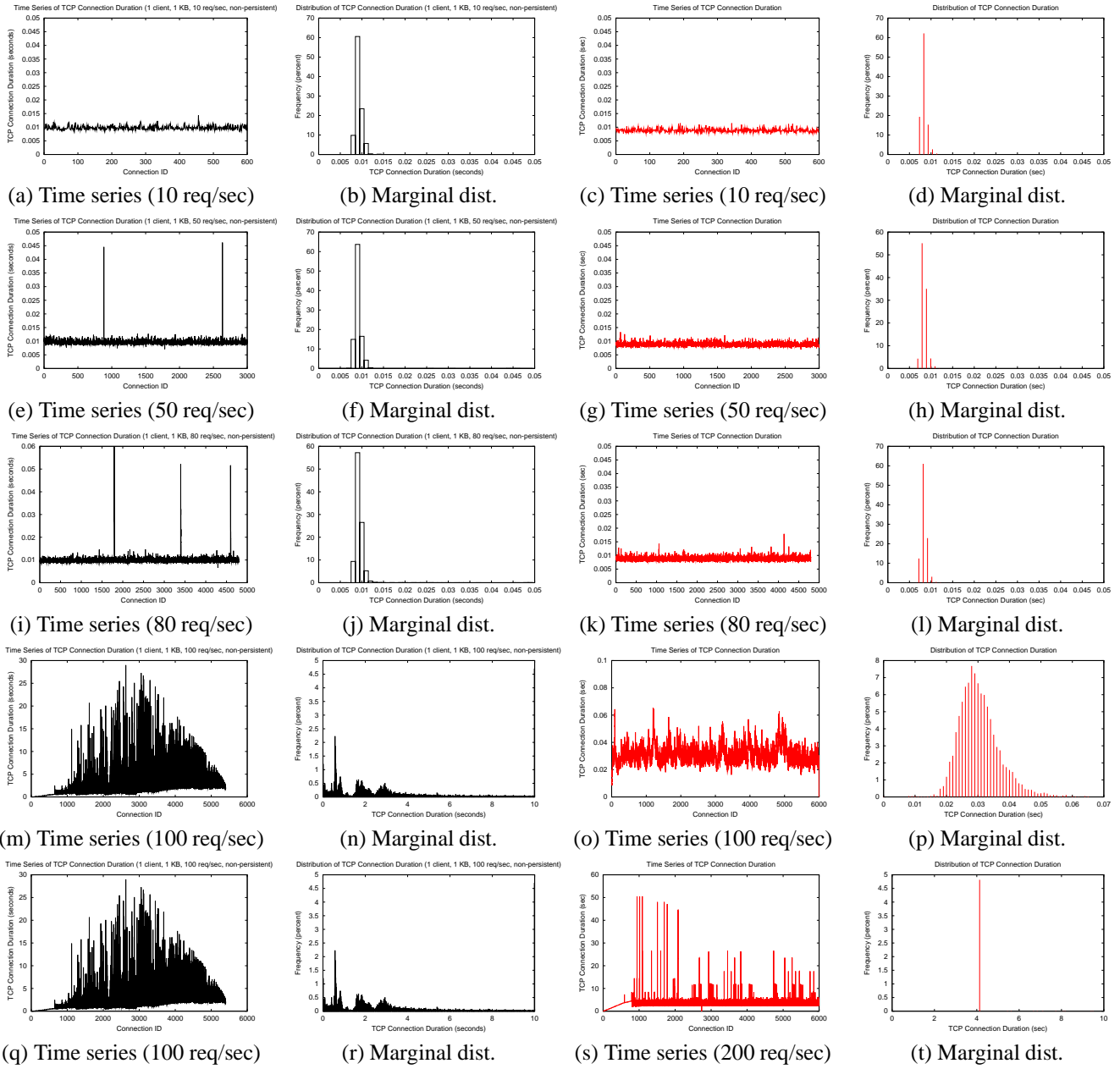


Figure 3. Simulation validation results for single-client scenario (deterministic arrivals, 1 KB transfers, HTTP/1.0, non-persistent). The leftmost two columns are empirical measurement results, while the rightmost two columns are OPNET simulation results.

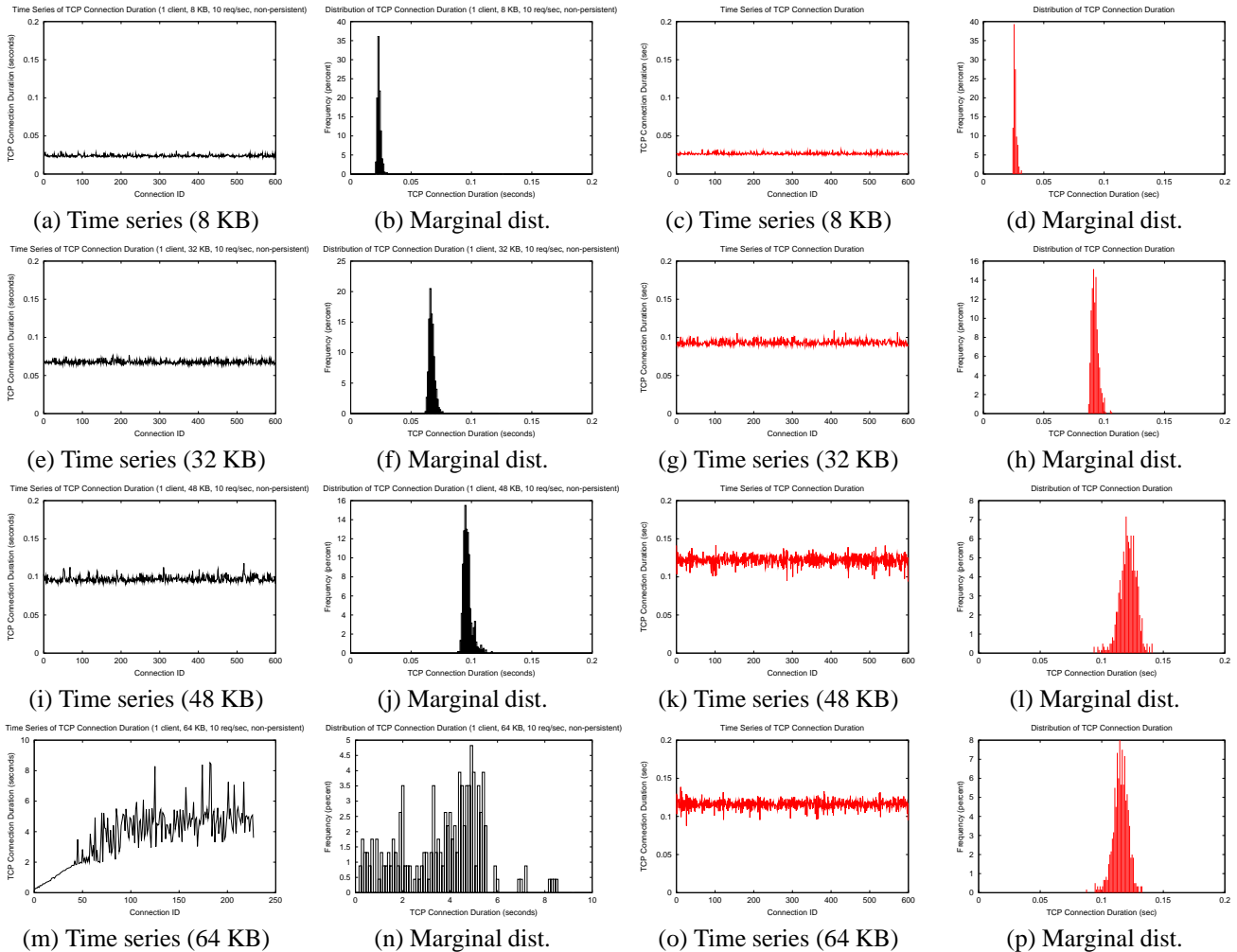


Figure 4. Simulation validation results for single-client scenario for different transfer sizes (deterministic arrivals, 10 req/sec, HTTP/1.0, non-persistent). The leftmost two columns are empirical measurement results, while the rightmost two columns are OPNET simulation results.

sults for 64 KB, where *network thrashing* occurs. Since it is not physically possible to complete a 64 KB transfer (45 TCP packets each carrying 1460 bytes of data) in less than 0.1 seconds (the time interval between successive requests) on an IEEE 802.11b WLAN, the server gradually builds up a huge backlog of waiting data, losing some packets, and inducing many TCP timeouts and retransmissions. Some HTTP requests abort because of the excessive delay.

The second anomaly is in the simulation results, where many TCP RST (reset) connections occur. Again, the reason for this is a limit enforced by OPNET Modeler 9.1 on concurrent TCP connections between the same source and sink entities. This simulation limitation produces anomalous behaviour that is quite different from that in the empirical measurements. Worse yet, the reported TCP connection durations in Figure 4(o) and (p) appear reasonable, but few of these connections actually transferred the requested amount of data. The same problem actually happens with 48 KB transfers in Figures 4(k) and (l).

In summary, Figures 3 and 4 show that the simulation model captures qualitative and quantitative characteristics of client-server Web transactions in the WLAN environment. However, the simulation model does not accurately predict the saturation points observed in the empirical measurements. We will be wary of these anomalies in the remainder of our simulation experiments with more general Web client workloads.

4. Simulation Results

This section presents selected results from our OPNET simulations of the network shown in Figure 1.

4.1. Experiment 1: Two Clients

The first experiment studies a two-client scenario, to see if there are fairness problems between two clients on a shared WLAN [2]. We consider a high load, with per-client request rates of 50 req/sec, for 1 KB Web objects using HTTP/1.0.

Figure 5 shows the results from the two-client scenario. In the high load case (50 req/sec from each client, for a total of 100 req/sec) in Figure 5, the two clients share the channel fairly, and experience similar user-level Web performance, and similar numbers of TCP resets. This result differs from empirical measurements with two clients, where the relative phasing between sources produces very different TCP packet loss behaviours for the two clients at the bottleneck queue, leading to distinct unfairness in terms of TCP and HTTP throughput [2].

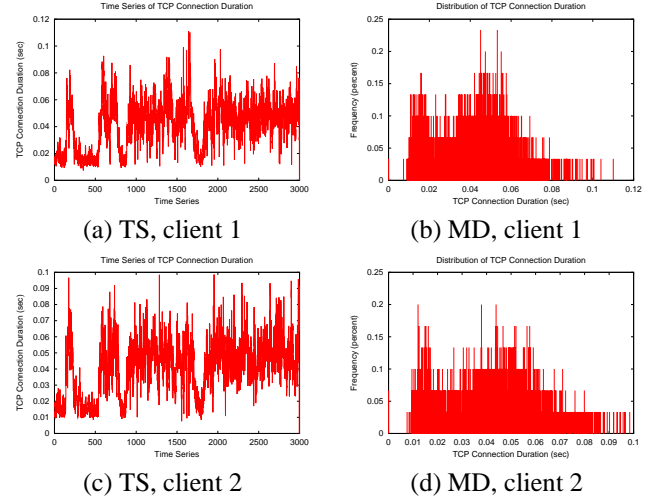


Figure 5. Simulation results for two clients (50 req/sec each)

4.2. Experiment 2: Persistent Connections

TCP connection handshaking adds a lot of overhead (and latency) to an HTTP transaction, when using non-persistent connections in HTTP/1.0. For example, only two of the packets in Figure 2 carry “useful” data; the others are control packets to establish, update, and release TCP connection state information. The overhead is particularly painful in a WLAN environment where each TCP packet must contend for access to the shared WLAN, using the MAC channel access protocol.

The overhead of TCP was one of the motivations for persistent-connection HTTP [10, 12, 14]. In a persistent connection, multiple HTTP transactions can be sent (sequentially, synchronously) on the same TCP connection, amortizing the overhead of the TCP SYN and FIN handshakes over multiple HTTP transfers.

The purpose of the next experiment is to demonstrate the performance advantages of persistent connections in a WLAN environment. In the OPNET simulation model, we changed HTTP/1.0 to HTTP/1.1, and set a 10-second persistent connection timeout. With these settings, we simulated different HTTP request rates for 1 KB Web objects.

An example of HTTP/1.1 usage appears in Figure 6. Here, three different HTTP transactions take place using the same (single) TCP connection. Each of the individual HTTP transactions takes about 3-4 milliseconds to complete, using only 4 network packets rather than 10. These transactions are about 3 times faster than with HTTP/1.0.

Similar results have been observed in empirical measurements of wireless Web traffic [2, 8], where persistent connections improve performance by a factor of 3.5 over

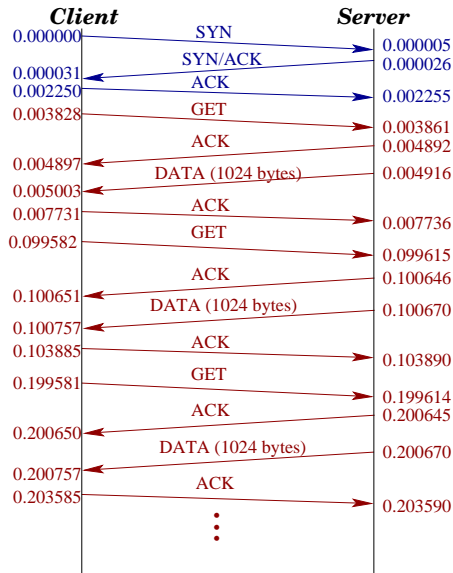


Figure 6. Example of HTTP/1.1 transactions

HTTP/1.0. The extra performance advantage in the empirical results comes from TCP DATA/ACK piggybacking, which reduces an HTTP/1.1 transaction to 2 network packets (one in each direction) in the common case. In practice, the effectiveness of persistent connections will depend on how many Web objects are retrieved over the same TCP connection before the persistent connection timeout expires.

Clearly, the advantages of persistent connections are significant in a WLAN environment. Fortunately, most Web browsers and Web servers today support persistent connections. All remaining simulation experiments use persistent connections, with a 10-second timeout.

4.3. Experiment 3: Web Browsing

The next experiment uses a simple model of Web client browsing behaviour. We model one simulated hour of Web browsing activity, as might occur during a lecture in a classroom area network. The Web client has a random “think time” between the generation of successive HTTP requests. We model the inter-arrival time between requests using an exponential distribution, with a mean of 10.0 seconds. We assume that each Web page retrieved has a very simple structure, consisting of a single Web object. The Web object sizes are randomly generated from a lognormal distribution, with a mean of 10 KB. Analysis of the output from a sample simulation run shows that the minimum object size generated is 841 bytes, the median is 7328 bytes, and the maximum is 188,863 bytes. We deem this range of sizes typical of Web workloads [1].

Figure 7 shows the simulation results for one client using this Web workload model. The client generates about

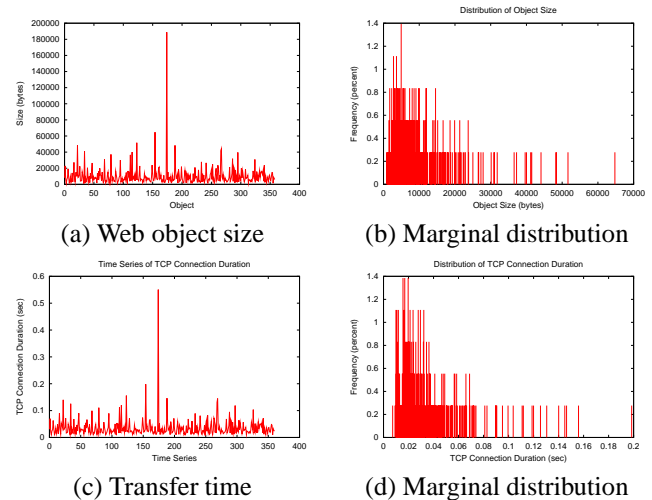


Figure 7. Simulation results for a single client web browsing

360 requests during the one hour period. The transfer time per request is clearly related to the object size retrieved. Ignoring protocol overheads, the average load generated by a single client is approximately 8 Kbits/sec (i.e., one 10 KB object every 10 seconds).

With this basic Web browsing model, we now consider multiple clients, as shown in Figure 8. We consider simulation models with 1, 2, 10, 20, 50, and 100 Web clients. Each client operates independently of other clients, downloading randomly sized Web objects from the Web server at random times.

Figure 9 shows an example of the browsing behaviour for four clients from a 10-client simulation model. Table 2 summarizes simulation results from this same model with three different HTTP and TCP protocol configurations. Clearly, persistent connections offer a performance improvement, while disabling TCP delayed acknowledgements also helps immensely. These protocol settings are used in all remaining experiments.

4.4. Experiment 4: Large Classroom Network

Our main interest is in the scalability of classroom area networks (i.e., how many clients can be supported in the WLAN, and how does user-perceived browsing performance degrade with network load).

Figure 10 shows the simulation results from these experiments. On each of the four graphs, the horizontal axis represents the number of simulated Web clients, ranging from 1 to 100 clients. Figure 10(a) shows the aggregate HTTP application-layer throughput for the Web clients, averaged over the simulated one hour period. This load is primarily a

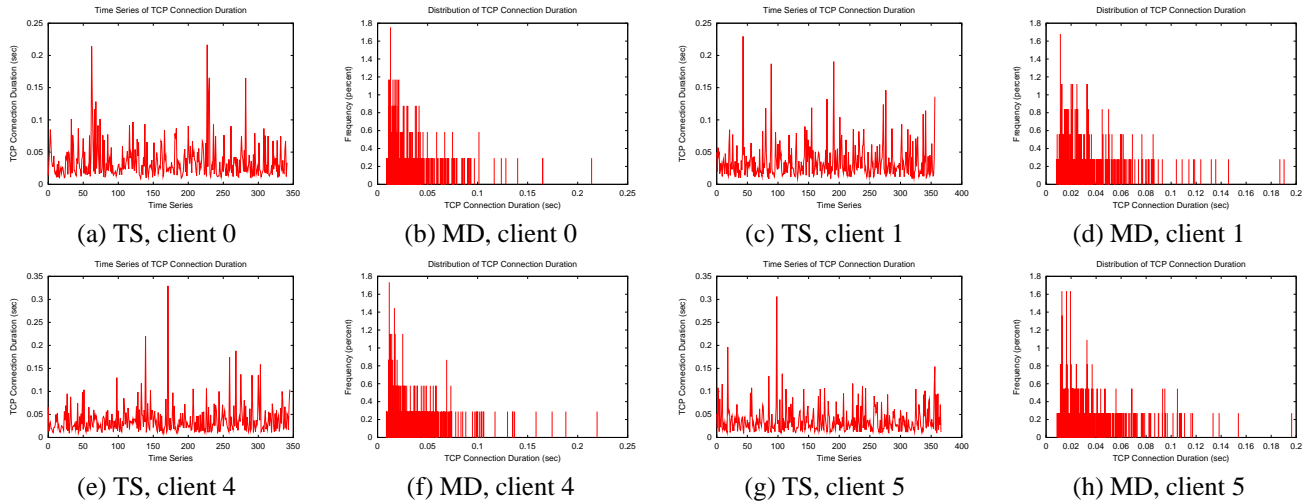


Figure 9. Selected simulation results for multiple clients web browsing

Table 2. Simulation results for different HTTP and TCP configurations (10 clients, random transfers)

Description of HTTP and TCP Protocol Configuration	HTTP Response Time (sec)		
	Minimum	Mean	Maximum
HTTP/1.0, Default Reno, Delayed ACK 0.200 sec	0.104	0.130	0.157
HTTP/1.1 , Default Reno, Delayed ACK 0.200 sec	0.069	0.100	0.138
HTTP/1.0, Default Reno, Delayed ACK 0.001 sec	0.020	0.031	0.045

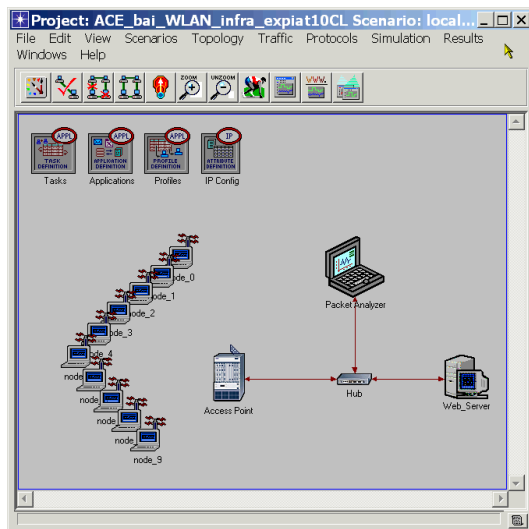


Figure 8. Multi-client simulation scenario

function of the number of clients, reaching almost 1.0 Mbps with 100 clients. HTTP/1.1 has a small advantage over HTTP/1.0; this advantage would increase significantly with more efficient TCP DATA/ACK packetization. Figure 10(b) shows the network-level throughput results. The network throughput is higher than the application-layer throughput because of protocol overhead (e.g., TCP, IP, headers, re-transmissions). Again, this load is a direct function of the number of clients. Figure 10(c) shows the mean HTTP transfer time, averaged across all transfers by all clients. Recall that the average Web object size is about 10 KB in these workloads. The mean HTTP transfer time increases slightly with the number of simulated clients, because of contention for use of the shared WLAN, and perhaps queuing delay at the server. Finally, Figure 10(d) plots the mean channel access delay for the shared WLAN channel. This graph shows an increase in channel access delay with the number of competing clients, as expected.

5. Summary and Conclusions

This paper presented a simulation study of an IEEE 802.11b wireless LAN in a classroom network scenario. The simulations, conducted using OPNET Modeler 9.1, were carefully parameterized and validated with empirical measurements from a WLAN environment.

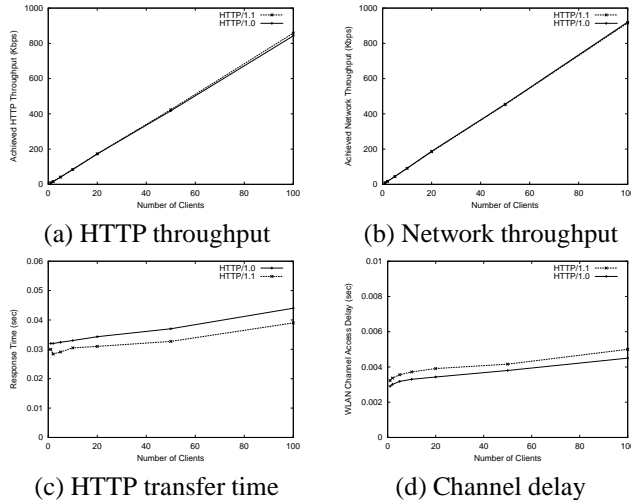


Figure 10. Simulation results for multiple clients

The simulation results show that an IEEE 802.11b WLAN can easily support up to 100 clients doing modest Web browsing. The results also show that persistent HTTP connections can provide a significant performance advantage in a WLAN environment.

Ongoing work focuses on extending our Web browsing model to represent more realistic Web workloads [4], and simulating larger network models.

Acknowledgements

Financial support for this research was provided by iCORE (Informatics Circle of Research Excellence) in the Province of Alberta, and by the Natural Sciences and Engineering Research Council of Canada, through NSERC Research Grant OGP0121969. The authors are grateful to Kehinde Oladosu for his assistance with the empirical measurements, and to the anonymous reviewers for their feedback on an earlier version of this paper.

References

[1] M. Arlitt and C. Williamson, "Internet Web Servers: Workload Characterization and Performance Implications", *IEEE/ACM Trans. Networking*, Vol. 5, No. 5, pp. 631-645, October 1997.

[2] G. Bai, K. Oladosu, and C. Williamson, "Portable Networks: Prototype and Performance", submitted for publication, 2003.

[3] A. Balachandran, G. Voelker, P. Bahl, and P. Rangan, "Characterizing User Behavior and Network Per-

formance in a Public Wireless LAN", *Proceedings of ACM SIGMETRICS*, Marina Del Rey, CA, pp. 195-205, June 2002.

- [4] P. Barford and M. Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluation", *Proceedings of ACM SIGMETRICS*, Madison, WI, pp. 151-160, June 1998.
- [5] B. Bennington and C. Bartel, "Wireless Andrew: Experience Building a High Speed, Campus-Wide Wireless Data Network", *Proceedings of ACM MOBICOM*, Budapest, Hungary, pp. 55-65, September 1997.
- [6] T. Hansen, P. Yalamanchili and H-W. Braun, "Wireless Measurement and Analysis on HPWREN", *Proceedings of Passive and Active Measurement Workshop*, Fort Collins, Co, pp. 222-229, March 2002.
- [7] D. Kotz and K. Essein, "Analysis of a Campus-Wide Wireless Network", *Proceedings of ACM MOBICOM*, Atlanta, GA, September 2002.
- [8] K. Oladosu, *Performance and Robustness Testing of Wireless Web Servers*, M.Sc. Thesis, University of Calgary, September 2003.
- [9] OPNET Technologies, www.opnet.com
- [10] V. Padmanabhan and J. Mogul, "Improving HTTP Latency", *Computer Networks and ISDN Systems*, Vol. 28, pp. 25-35, December 1995.
- [11] RFC 1945: "Hypertext Transfer Protocol – HTTP/1.0", www.ietf.org/rfc/rfc1945.txt
- [12] RFC 2616: "Hypertext Transfer Protocol – HTTP/1.1", www.ietf.org/rfc/rfc2616.txt
- [13] H. Singh and P. Singh, "Energy Consumption of TCP Reno, TCP NewReno, and SACK in Multihop Wireless Networks", *Proceedings of ACM SIGMETRICS*, Marina Del Rey, CA, pp. 206-216, June 2002.
- [14] S. Spero, "Analysis of HTTP Performance Problems", sunsite.unc.edu/mdma-release/http-prob.html
- [15] W.R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley, 1994.
- [16] D. Tang and M. Baker, "Analysis of a Metropolitan-Area Wireless Network", *Proceedings of ACM MOBICOM*, Seattle, WA, pp. 13-23, August 1999.
- [17] D. Tang and M. Baker, "Analysis of a Local-Area Wireless Network", *Proceedings of ACM MOBICOM*, Boston, MA, pp. 1-10, August 2000.