The Effects of Mobility on Wireless Media Streaming Performance

Guangwei Bai Carey Williamson
Department of Computer Science
University of Calgary
2500 University Drive NW
Calgary, AB, Canada T2N 1N4
Phone: (403) 220-6780 Fax: (403) 284-4707
Email: {bai, carey}@cpsc.ucalgary.ca

ABSTRACT
This paper presents experimental measurements of MPEG-4 media streaming performance in an IEEE 802.11b WLAN environment, using a Darwin streaming server, several wireless clients, and a wireless network analyzer. We first study streaming performance in a simple scenario with no client mobility, and then demonstrate the impacts of user mobility on both network-level and user-level performance. Our results show that media streaming performance can degrade significantly in the presence of user mobility. Furthermore, the performance degradation affects all clients, not just those who are mobile.

KEY WORDS
Multimedia, Wireless Networks, Network Traffic Measurement, IEEE 802.11b WLANs

1 Introduction

The World Wide Web and multimedia streaming are two popular services on the Internet today. The Web has made the Internet available to the masses, through its TCP/IP protocol stack and the principle of layering. Users can access rich content, including digital audio and video media, using easy-to-use GUIs on their desktop or portable computing device. Multimedia streaming has seen increased demand on the Internet in recent years, and has drawn tremendous attention from both academia and industry.

Concurrent with these developments, wireless technologies have revolutionalized the way people think about networks, by offering users freedom from the constraints of physical wires. Wireless Internet access is widely available today, in laptop or handheld form, at relatively modest cost [6, 11]. Mobile users are interested in exploiting 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 these technologies to support wireless multimedia streaming [7, 8]. Typical streaming applications could include seminars, press conferences, news events, sports, and entertainment applications. Live multimedia streaming offers a remote audience an experience similar to being physically present at the event. Stored media streaming provides asynchronous access to events of interest, for on-demand entertainment or for archival purposes.

In this paper, we explore multimedia streaming performance in an IEEE 802.11b Wireless LAN (WLAN) environment. We focus on delivering multimedia (audio and video) clips from a streaming server on a wired network to mobile clients on a WLAN. We conduct streaming experiments both with and without client mobility. Experimental traffic measurements are collected using a wireless network analyzer, and used to characterize MPEG-4 media streaming performance in best-case and worst-case scenarios.

Our experimental results demonstrate two main observations. First, media streaming performance can degrade significantly in the presence of user mobility. Second, the performance degradation affects *all* clients in the WLAN, not just the clients who are mobile. These observations identify significant challenges for providing quality of service guarantees for multimedia streaming applications in WLAN environments.

The remainder of this paper is organized as follows. Section 2 discusses background information on IEEE 802.11b and multimedia streaming. Section 3 describes the experimental setup and methodology for our study. Section 4 presents the measurement results and analyses. Finally, Section 5 concludes the paper.

2 Background

2.1 Wireless Internet and IEEE 802.11b WLANs

Wireless technologies play a prominent role in today's global Internet infrastructure. One popular technology is the IEEE 802.11b WLAN standard known as "WiFi" (Wireless Fidelity). It provides low-cost wireless Internet access for mobile users, with physical layer data transmission rates of up to 11 Mbps.

The IEEE 802.11b standard defines two Medium Access Control (MAC) protocols, namely Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), and Request-To-Send/Clear-To-Send (RTS/CTS). It also defines the frame formats used at the data link layer. Frames that are correctly received over the shared wireless channel are acknowledged by the receiver. Unacknowledged frames are retransmitted by the sender after a short timeout (e.g., a few milliseconds), using the same MAC protocol.

Some vendor implementations of IEEE 802.11b dynamically switch between CSMA/CA and RTS/CTS based on the observed wireless channel conditions (e.g., excessive collisions). Some implementations also support dynamic rate selection, so that frames can be transmitted at either 1 Mbps, 2 Mbps, 5.5 Mbps, or 11 Mbps depending on the wireless channel quality.

2.2 Multimedia Streaming

Streaming technology delivers media over a network from a server to a client in real time. The media is not downloaded to a viewer's hard drive. Rather, the media is played as the client receives it (ignoring the buffering used at startup). If the client wishes to play the media again, the streaming process is repeated.

An end-to-end streaming system requires some content creation software, a streaming media server, and a client media player. Media clips can be created with production tools that convert audio, video, or animation to a format such as MPEG-4 for the server to stream. Streaming servers such as the Darwin Streaming Server (from Apple) or RealServer (from RealNetworks) can be used to deliver media clips to clients (e.g., running MP4Player, RealPlayer, or QuickTime).

The main networking protocols used for multimedia streaming are Real-Time Streaming Protocol (RTSP) [9], Real-Time Control Protocol (RTCP) [10], and Real-Time Protocol (RTP) [10]. These protocols define how to establish a connection and transmit the media from the server to the client. RTSP is a signalling protocol that is used to establish and manage a client/server streaming connection, including session initiation and media negotiation. RTSP is the highest level protocol. It handles the initial connection for the client to request a media file from the server. The server provides the information necessary for the client to render the media. The RTSP connection remains in place throughout the media streaming session, in case the client wishes to pause, stop, rewind, or replay the media stream, or even change the media stream selection. RTP and RTCP are the media transport protocols used together to transmit and control the actual media data. RTP is the standard protocol for real-time media transport over IP networks. It streams audio/video data and supports many codecs and media types, such as MPEG. RTCP is the adaptive feedback control protocol for RTP.

3 Experimental Methodology

3.1 Experimental Setup

The IEEE 802.11 standard allows two types of WLAN configurations. In *ad hoc mode*, all the stations in the WLAN can communicate directly with each other, without requiring a connection to a wired network. In *infrastructure mode*, the WLAN includes an Access Point (AP) connected to a wired network. In this mode, all mobile stations in the WLAN communicate via the AP, which provides access to services on wired LANs and the external Internet.

In our work, we use an infrastructure-based WLAN as shown in Figure 1. The media streaming server is on a 100 Mbps wired-Ethernet LAN. The WLAN consists of several wireless clients and an AP. In addition, we use a wireless network analyzer to monitor the wireless channel. Each laptop has a Cisco Aironet 350 Series Adapter for access to the IEEE 802.11b WLAN. The wireless cards operate in infrastructure mode.

The wireless clients run the MP4Player application, which is used to access and play media content from the streaming server. All content is accessed via the AP, using RTSP over TCP, RTP over UDP, RTCP over UDP, as well as the IEEE 802.11b MAC protocols.

The streaming server in our experiments is a Darwin Streaming Server [2]. The server sends streaming media to the clients using the RTP and RTSP protocols.

The AP in our experimental setup is a software-configurable AP on a laptop. This laptop has a Host AP driver for wireless LAN cards based on Intersil's Prism2/2.5/3 chipset. This HostAP feature enables 802.11b AP functionality. We run a specially instrumented Linux kernel on this laptop to record packet arrivals, packet departures, and packet queueing behaviour at the wireless network interface.

Network traffic measurements are collected using a wireless network protocol analyzer. The analyzer used is AiroPeek NX [1]. The analyzer receives all WLAN packets based on user-specified configuration parameters and stores

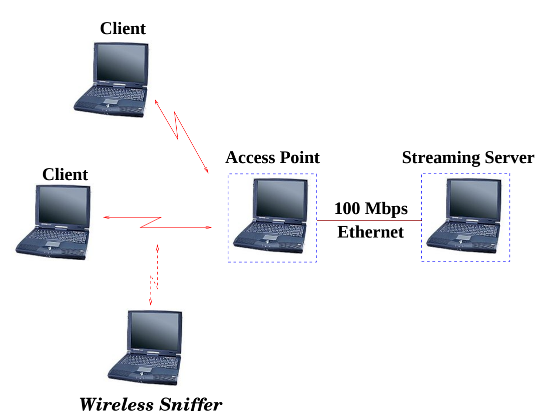


Figure 1. Experimental Setup for Streaming

these packets in memory. The software decodes the 802.11 protocols, providing information such as source address, destination address, data rate, protocol type, and payload size. We also use `tcpdump` [12] to record network packet events on the client and server machines.

3.2 Experimental Design

In our work, we stream a single MPEG-4 media clip (video and audio) over an IEEE 802.11b WLAN in our lab. The media clip is 100 seconds long, with 1000 kbps for the video and 128 kbps for the audio.

We vary both the number of clients in the streaming experiments, as well as the mobility characteristics of these clients. Measurements focus on both the network-level and user-level performance observed for wireless multimedia streaming.

4 Experimental Results

4.1 Experiment A: 1 Client, No Mobility

The first experiment uses only a single client, with no user mobility. The client, server, and analyzer laptops are all in the same office. The wireless channel is assumed to be excellent. This test indicates the best-case behaviour for wireless media streaming.

The network traces collected by `tcpdump` and the wireless network analyzer show that the media streaming session between the client and the server has three distinct phases. The three phases are:

1. Initialization.

    - The client requests a selected media file from the server, using RTSP over TCP.
    - The server returns information about the media format and available options to the client.
    - The client replies with `setup` to specify the protocol and ports used for transmission.
    - The server replies with the selected protocol and acknowledges the client's port numbers. In addition, the server indicates the port numbers for feedback sent by the client.
    - The client issues the `play` request.
    - The server responds with "OK", plus information for the client to synchronize with the upcoming RTCP and RTP transactions.

2. Media Transmission.

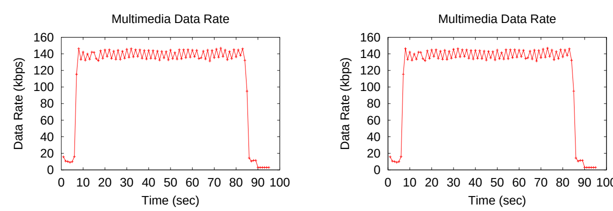
    - RTCP and RTP, running over UDP, work together. The RTCP info is sent to the odd numbered ports specified in the client's RTSP setup request (e.g., 1025 and 1027).
    - The RTP protocol packetizes and sends the media data to the even numbered ports (e.g., video packets to port 1024, and audio packets to port 1026).

3. Session Termination.

    - When the streaming is finished, the server initiates a four-way TCP handshake to close the RTSP/TCP connection.

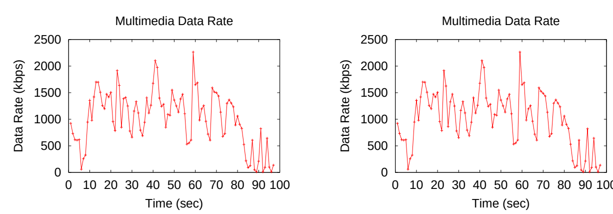
This three-phase structure is common in all of the experiments. In total, 28 RTSP/TCP packets were exchanged during the session setup and termination phases, among 11,236 TCP/UDP packets in the complete streaming session. The remainder of the paper focuses solely on the media data transmission phase of the wireless media streaming experiments.

Figure 2 and Figure 3 show the streaming data rate for audio and video content, respectively. These values are calculated from `tcpdump` traces collected at the client and server, with the data rate calculated using non-overlapping one-second intervals.



(a) Server (b) Client

Figure 2. Audio Traffic (1 Client, No Mobility)



(a) Server (b) Client

Figure 3. Video Traffic (1 Client, No Mobility)

Figure 2 shows that the audio bit rate is approximately constant at 128 kbps. The jagged nature of the plot arises from the 1-second sampling granularity used for the data rate calculation. Figure 2(b) shows that the client receives everything that the server sends. This result is as expected, since the wireless channel condition is excellent.

Figure 3 shows that MPEG generates a Variable Bit Rate (VBR) data stream. While the server sends MPEG video frames to the network at a constant frame rate of about 30 frames per second, the video content is bursty because the sizes of the compressed frames vary. MPEG uses a combination of *intra-frame coding* and *inter-frame coding* to remove spatial and temporal redundancies. This produces variable size I-frames, P-frames, and B-frames.

Figure 3 shows the fluctuation of video bit rate with time for this particular video sequence. The peak bit rate exceeds the mean bit rate by about a factor of 2. There is consistency between the sending rate at the server (Figure 3(a)) and the receiving rate at the client (Figure 3(b)). Again, this result is expected, because the wireless channel is excellent.

The queue behaviour at the MAC layer of the Access Point (AP) on a specially instrumented Linux kernel is illustrated in Figure 4. The WLAN supports this streaming application without any problem. The AP queue size never exceeds 20 packets.

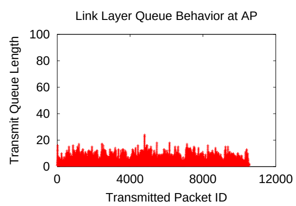
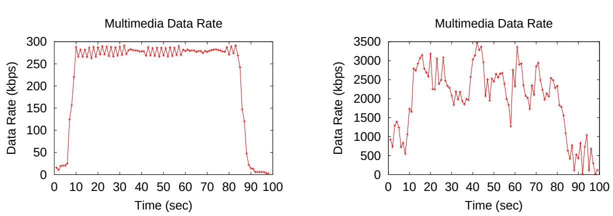


Figure 4. AP Queue (1 Client, No Mobility)

4.2 Experiment B: 2 Clients, No Mobility

Figure 5 shows the results for a two-client scenario, in which each client streams a media clip from the same server independently. Compared to the single-client scenario, the audio traffic (Figure 5(a)) is doubled. However, the video traffic (Figure 5(b)) is only about 50% higher. The reason is that the video data from multiple souces are statistically multiplexed on the network. The results show that the WLAN easily supports multimedia streaming for two clients. The AP queue size in Figure 6 never exceeds 30 packets.



(a) Audio (b) Video

Figure 5. Total Traffic (2 Clients, No Mobility)

4.3 Experiment C: 1 Client, With Mobility

The next experiment considers a one-client scenario with user mobility. The user with the client laptop physically

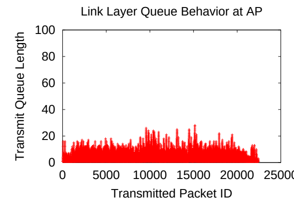


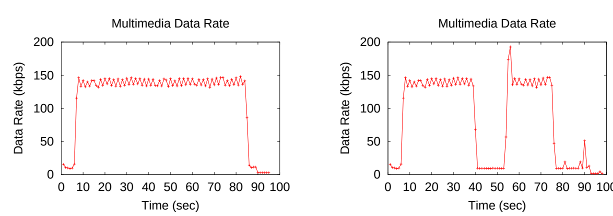
Figure 6. AP Queue (2 Clients, No Mobility)

walked in and out of the lab, and up and down the hall, during the media streaming. The streaming server and the AP remained in the lab.

In this scenario, the wireless channel conditions change with time. The channel quality depends upon many factors, including the distance between the client and the AP, and the obstacles between the client and AP. This experiment studies the impact of mobility (and thus the channel conditions) on the media streaming performance.

Figure 7 and Figure 8 illustrate the audio and video streaming rates at the network layer, as reported by `tcpdump` at the client and server. Obvious differences from the previous two experiments are observed: while the streaming server continues to transmit audio and video data at the proper media playback rates, the receiving client experiences occasional outages where little or no media content arrives.

Figure 7(b) and Figure 8(b) clearly show the degradation in media streaming performance for the mobile client, due to adverse wireless channel conditions caused by user mobility. The delivery problems also manifest themselves at the AP's queue (see Figure 9), where overflows and packet losses occur. The rest of this section studies this problem in more detail.

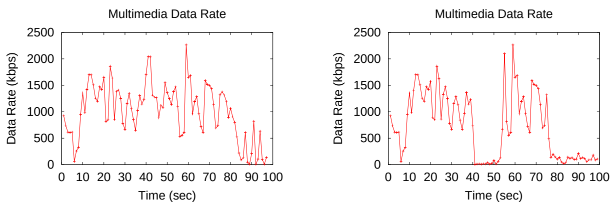


(a) Server (b) Client

Figure 7. Audio Traffic (1 Client, With Mobility)

Recall that we use a wireless network analyzer in our experiments to capture all activities from the wireless channel. A careful investigation of the captured trace shows that dynamic MAC-protocol selection and dynamic rate adaptation both contribute to the performance degradation.

In general, the basic CSMA/CA access mechanism is used most of the time to achieve low-latency frame transmissions. The CSMA/CA protocol can achieve about 500 frame transmissions per second on a typical WLAN, assuming good wireless channel conditions. However, after



(a) Server (b) Client

Figure 8. Video Traffic (1 Client, With Mobility)

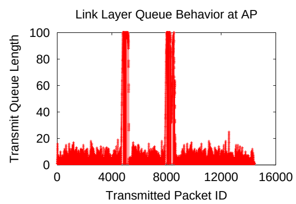


Figure 9. AP Queue (1 Client, With Mobility)

several failed (i.e., unacknowledged) frame transmission attempts, the protocol reverts to the RTS/CTS mechanism. If several attempts with RTS/CTS are also unsuccessful, then the 802.11b protocol discards the current frame, and proceeds to the next one in the queue for its possible transmission.

Similar behaviour is observed for dynamic rate adaptation. When the wireless channel condition deteriorates (e.g., excessive retransmissions), the AP adapts by changing from 11 Mbps to 5.5 Mbps, or from 5.5 Mbps to 2 Mbps, or from 2 Mbps to 1 Mbps for the physical-layer data transmission rate.

The additional latencies from MAC-layer retransmissions, RTS/CTS exchanges, and rate adaptation combine to reduce the effective “service rate” of the wireless network, leading to a backlog of frames waiting for transmission. Incoming frames continue to arrive from the media streaming server, which is unaware of the transient network outage for the client. When the AP queue fills, arriving frames are discarded. The end result is serious degradation in the media streaming quality for the mobile user, at least until the user moves again to a location with better wireless channel quality.

From the network traces, the streaming session can be divided into four phases, which are visually evident in Figure 7(b) and Figure 8(b). The phases are:

1. **Early (0-40 seconds):** The media transmission works well. Most data frames were transmitted at 11 Mbps, and many were received successfully. A few frames were transmitted at 5.5 Mbps. In this period, some retransmissions occurred, including some RTS/CTS exchanges, but all frames were eventually successful.

2. **Outage 1 (40-51 seconds):** The channel was very poor. The AP could not reach the client at all, and thus did not receive any acknowledgements from the client, or responses to RTS attempts. The physical-layer frame data rate was reduced to 1 Mbps.

3. **Recovery (51-77 seconds):** As the user returned toward the AP, the channel quality improved. Most data frames arrived at the receiver successfully. The data rate used for frame transmissions was gradually increased from 1 Mbps to 2 Mbps, then 5.5 Mbps, and finally 11 Mbps.

4. **Outage 2 (77-100 seconds):** The channel deteriorated again. In this period, the media transmission continued, but very inefficiently. Many RTS/CTS exchanges occured, with many RTS retransmissions needed. Many data frames were discarded by the AP because of excessive retransmissions, as well as queue overflow.

In even worse channel conditions, the client and server may disconnect completely. In this case, the media player (at the client) tries to establish a new connection to the server. In other cases, the player at the client freezes, indicating that the streaming session is beyond control.

The transitions in media streaming performance as a result of user mobility are sudden and drastic. Figure 9 shows that the MAC-layer transmit queue at the AP filled rapidly during the outages. With the default queue size setting of 100 in the Linux kernel, data packets from the upper layer overflow this queue, and many packet drops occur from this link-layer queue, even *before* the packets make it to the WLAN. User-level media streaming quality degrades drastically.

4.4 Experiment D: 2 Clients, With Mobility

Our final experiment considers two wireless clients, but only one of which is mobile. The foregoing results from Experiment C suggest that poor channel conditions for one client will degrade performance for both clients, since the wireless channel will be occupied by excessive retransmissions, RTS/CTS exchanges, and low-data-rate frame transmissions. The MAC-layer queue at the AP will fill with packets for the poorly-connected client, delaying the delivery of packets destined to the well-connected client.

The results from this experiment (not shown here, due to space limitations) confirm that this is indeed the case. The bottleneck is at the AP network interface, where packets wait at the link-layer queue for medium access on the wireless LAN. A specially instrumented Linux kernel shows this behaviour: the queue fills when the wireless channel quality is poor. There is no flow control or back-pressure mechanism to prevent the UDP layer from overflowing this queue.

In the worst case, many packets (including packets to the well-connected client) are dropped from the AP link-layer queue, *before* the packets make it onto the WLAN. Neither client receives sufficiently many packets for proper quality playback of the media stream. In other words, the mobility of one client can adversely affect the media streaming quality for other clients in the same WLAN.

5 Summary and Conclusions

This paper presented experimental measurements of MPEG-4 media streaming on an IEEE 802.11b WLAN. We focus on delivering multimedia (audio and video) content from a streaming server on a wired network to mobile clients on a wireless LAN. We conduct experiments both with and without client mobility, to characterize MPEG-4 media streaming performance in best-case and worst-case scenarios.

Our experimental results illustrate two main points. First, wireless media streaming performance can degrade significantly in the presence of user mobility. Inconsistent wireless channel quality and intermittent connectivity can lead to excessive retransmissions, dynamic rate adaptation, and RTS/CTS negotiations on the WLAN. These delays degrade the performance of the wireless streaming application. Second, the performance degradation affects *all* clients in the WLAN, not just the clients who are mobile. This problem occurs because of the shared queue at the AP.

These observations highlight the many challenges for providing quality of service guarantees for wireless multimedia streaming. One possible solution is to use per-flow queueing at the AP, or a buffer management scheme that provides fairness between flows. We will investigate these in future work.

Our ongoing work focuses on streaming experiments with multiple mobile clients (four or more) in the same WLAN environment. The clients access multiple MPEG-4 media clips of different bit rates. These experiments will further characterize wireless multimedia traffic, studying streaming performance and user mobility in more realistic scenarios. We believe that this work will provide more insight into the design of wireless multimedia protocols and applications.

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 (NSERC) of Canada.

The authors are grateful to Nayden Markatchev for technical assistance with the experimental setup. The authors also thank the anonymous reviewers for their constructive feedback on an earlier version of this paper.

References

[1] "Wild Packets' AiroPeek NX", *http://www.wi-fiplanet.com/reviews/article.php/1268971*

[2] Apple's Open Source Projects, "Darwin Streaming Server", *http://developer.apple.com/darwin/*

[3] G. Bai, K. Oladosu and C. Williamson, "Performance Issues for Wireless Web Servers", to appear *2004 International Workshop on Mobile and Wireless Ad Hoc Networking (MWAN)*, Las Vegas, Nevada, June 2004.

[4] G. Bai and C. Williamson, "Simulation Evaluation of Wireless Web Performance in an IEEE 802.11b Classroom Area Network", *Proceedings of IEEE 3rd International Workshop on Wireless Local Networks (WLN)*, Bonn/Königswinter, Germany, pp. 663-672, October 2003.

[5] IEEE 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 1999.

[6] D. Kotz and K. Essein, "Analysis of a Campus-Wide Wireless Network", *Proceedings of ACM MOBICOM*, Atlanta, GA, September 2002.

[7] T. Kuang and C. Williamson, "Hierarchical Analysis of RealMedia Streaming Traffic on an IEEE 802.11b Wireless LAN", *Computer Communications*, Vol. 27, pp. 538-548, 2004.

[8] A. Majumdar, D. Sachs, I. Kozintsev, K. Ramchandran and M. Yeung, "Multicast and Unicast Real-Time Video Streaming Over Wireless LANs", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No. 6, pp. 524-534, June 2002.

[9] RFC 2326, "Real Time Streaming Protocol", http://www.faqs.org/rfcs/rfc2326.html

[10] RFC 1889, "RTP, A Transport Protocol for Real-Time Applications", http://www.faqs.org/rfcs/rfc1889.html

[11] D. Tang and M. Baker, "Analysis of a Local-Area Wireless Network", *Proceedings of ACM MOBICOM*, Boston, MA, pp. 1-10, August 2000.

[12] `tcpdump`, http://www.tcpdump.org/