RealMedia Streaming Performance on an IEEE 802.11b Wireless LAN

Tianbo Kuang 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: {*kuang, carey*}*@cpsc.ucalgary.ca*

Abstract—

This paper presents experimental measurements of RealMedia audio/video streaming applications on an IEEE 802.11b wireless LAN. Empirical traffic measurements collected using a wireless network analyzer are used to characterize RealMedia streaming workloads, and to assess their impacts on wireless network performance. In addition, we study the relationship between the wireless channel error rate and the user-perceived quality of RealMedia streaming applications. We find that the RealMedia application provides relatively robust audio and video streaming quality in all but the poorest of wireless LAN conditions. Finally, competing TCP/IP Internet traffic is found to have relatively little impact on the quality of UDP-based RealMedia streaming sessions.

Keywords: Multimedia Streaming, Wireless LANs, IEEE 802.11b, Network Traffic Measurement, Workload Characterization

I. Introduction

Three of the most exciting and fastest-growing Internet technologies in recent years are the World Wide Web, multimedia streaming, and wireless networks. The Web has made the Internet available to the masses, through its TCP/IP protocol stack and the principle of layering: Web users do not need to know the details of the underlying communication protocols in order to use network applications. Multimedia streaming provides desktop access to real-time and on-demand audio and video applications for educational and entertainment purposes at home, office, and school. Wireless technologies have also revolutionalized the way people think about networks, by offering users freedom from the constraints of physical wires.

All three of these technologies are available today, in desktop or handheld form, at relatively modest cost. Mobile users are interested in exploiting the functionality of the technology at their fingertips, as wireless networks bring closer the "anything, anytime, anywhere" promise of mobile networking.

In this paper, we explore the convergence of these technologies by studying multimedia streaming applications for mobile Internet users in a wireless local area network (WLAN) environment. We focus on the RealMedia multimedia streaming application, delivering audio and video content from a wired-Internet RealMedia server to a single mobile client on an in-building IEEE 802.11b WLAN in the Department of Computer Science at the University of Calgary.

There are three objectives to our study. First, we seek to characterize the network traffic workload generated by the RealMedia streaming application, in order to study the impact on the WLAN. Such a characterization is useful in capacity planning studies for the design, deployment, and evolution of larger WLANs. Second, we seek to understand the relationship between wireless channel characteristics (i.e., channel error rate, packet loss, retransmission, delay) and the user-perceived quality of a video stream. We do this using a wireless "sniffer" to capture and analyze the WLAN traffic for a mobile user at a variety of physical locations in our WLAN environment. This portion of the study offers a subjective assessment of video quality as a function of channel error characteristics, with attendant explanations for the performance degradations based on network-level effects. Third, we attempt to understand the impacts of competing Internet TCP/IP packet traffic on the quality of a RealMedia streaming session. Again, we use WLAN traffic measurements to ascertain the behaviour of RealMedia streaming in the presence of bandwidth-hungry TCP traffic flows.

The remainder of this paper is organized as follows. Section II provides some technical background on IEEE 802.11b WLANs, and on the RealMedia streaming application used in our study. Section III describes our experimental setup for wireless LAN measurements of RealMedia traffic. Section IV describes our experimental measurement results, for each of the objectives identified above. Section V describes related work on measurements of wireless networks and multimedia streaming applications. Finally, Section VI summarizes the results and observations from our paper.

II. Background

A. IEEE 802.11b Wireless LAN Technology

The IEEE 802.11b WLAN standard [2] is a high speed extension (currently up to 11 Mbps) of the original 2 Mbps standard [3] in the 2.4 GHz band. The standard specifies the physical layer and Medium Access Control (MAC) layer protocols used.

The physical layer allows four different data rates to be used for packet transmissions: 1 Mbps, 2 Mbps, 5.5 Mbps, and 11 Mbps. The higher data rates are achieved using more sophisticated modulation schemes, while the lower data rates offer backward-compatibility with earlier 802.11 products. Data rate information is carried in the header of 802.11b frames (transmitted at 1 Mbps), so that the receiving station knows what clocking rate to use for the payload portion of the frame as it arrives.

The physical layer wireless channel is subject to loss, fading, and interference [6], [7]. These random noise fluctuations affect the signal quality for the receiver, and can result in corrupted packets, particularly when high data rates are used for transmission. For this reason, most 802.11b products dynamically adjust the data transmission rate (on a packet-by-packet basis) based on an estimate of the channel error rate.

The MAC layer regulates access to the shared (i.e., broadcast) channel in a wireless LAN. The typical MAC protocol used in 802.11b is Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA), also called Distributed Coordination Function (DCF). When a station wants to send packet, it first senses the channel. If the channel is idle for a certain period of time (called the Distributed Inter-Frame Space (DIFS)), it transmits the packet. Otherwise, it waits until the channel becomes idle for another DIFS plus some random time. If the channel is still busy, the station doubles the random waiting period and repeats the process.

The 802.11b standard also defines MAC-layer error control mechanisms. To combat the unreliability of the wireless channel, the standard requires a receiver to acknowledge each correctly received MAC frame. Frames that fail the Cyclic Redundancy Check (CRC) at the receiver are simply ignored. If no acknowledgment is received shortly after transmission, the sender resends the packet, repeating this process as necessary until an ACK is received or until it reaches the maximum retransmission threshold (e.g., 4), at which point the sender gives up, leaving the problem to higher layer protocols.

The 802.11b WLAN can be operated in two different modes. In *infrastructure mode*, all stations communicate via an Access Point (AP) connected to a wired network. In *ad hoc* mode, there is no access point; stations communicate with each other in a peer-to-peer fashion. In this study, we use infrastructure mode, with mobile clients requesting media content from a server on a wired network. We did not use the Point Coordination Function (PCF) of 802.11b for multimedia, since this feature is not supported on our WLAN.

B. *RealSystem Media Streaming Architecture*

RealNetworks offers RealSystem as the Internet solution for audio and video streaming. This system consists of RealServer (server), RealPlayer (client), RealProducer (codec), and a network [14]. The codec provides media content to the server in real-time or offline. The server is responsible for processing requests from clients (e.g. connection setup, play, pause, stop, teardown) and sending the media data to them in a controlled way (e.g., bandwidth adaptation, error control, intelligent packet dropping). The client receives and displays the audio and video content from the server, and provides explicit feedback to assist the server in traffic control. The network moves the media content between server and client, reliably or unreliably. The RealSystem architecture supports both real-time and on-demand streaming. We only study on-demand streaming in this paper.

The RealAudio and RealVideo contents are created in advance using RealProducer, and stored in RealMedia File Format (RMFF) files [4]. Each media packet in the encoded file has a stream number, timestamp, and size, plus a flag indicating whether it belongs to a key frame. The flag helps the server make decisions about which packets to transmit, retransmit, or skip when problems occur. The RMFF file header indicates how fast each stream should be delivered without degrading the quality. This rate is defined in the coding process and is called the target rate. As reported in an earlier study [8], RealServer sends both audio and video streams as Constant Bit Rate (CBR) streams.

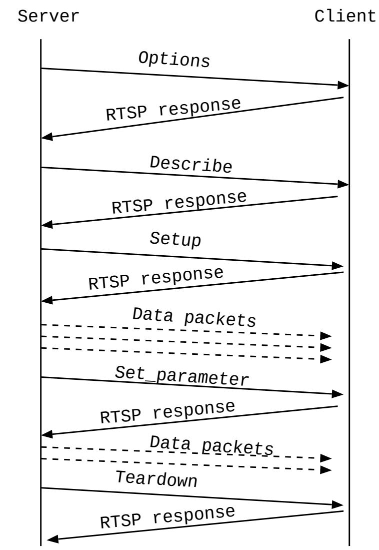


Fig. 1. Illustration of Real-Time Streaming Protocol (RTSP)

A streaming session is managed using the Real-Time Streaming Protocol (RTSP) [17]. This control connection is established between the RealServer and the RealPlayer, usually using TCP. A separate connection is set up for the actual media data. A backchannel is also set up for the client to request retransmssion of lost packets or report receiver statistics [16]. Besides the general start, stop, pause, and fast forward control functions, RealSystem uses RTSP to change the delivery parameters of an ongoing streaming session (e.g., using `set_parameter`). An example of a typical RTSP session is shown in Figure 1.

Before sending a media data packet, the RealServer encapsulates it using a *media packet protocol*. These protocols, such as the Real-time Transport Protocol (RTP) [18] or Real Data Transport (RDT) [15], facilitate the delivery and synchronization of real-time media data. The media packet is then carried by the transport-layer protocol. The default transport-layer protocol for audio/video streaming is UDP, though RealSystem supports TCP-based streaming as well (e.g., to traverse network firewalls).

To overcome network delay and delay variation, the RealPlayer buffers incoming data for a few seconds before it starts playing back the streams. If network conditions change during the playback, the RealSystem uses Adaptive Stream Management (ASM) [16] to manage the stream quality (e.g., to change the streaming rate, or to prioritize audio packets).

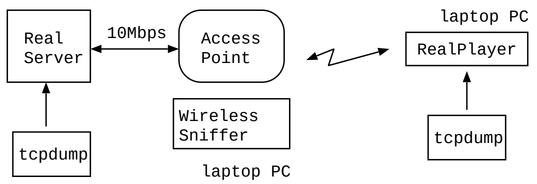


Fig. 2. Experimental Setup for RealMedia Streaming Measurements

TABLE I
CHARACTERISTICS OF EXAMPLE MULTIMEDIA STREAM

Item	Audio	Video
Duration (seconds)	68	68
Target Rate (kbps)	16.1	184
Total Packets	432	2850
Minimum Packet Size (bytes)	320	12
Maximum Packet Size (bytes)	320	685

TABLE II
QUALITATIVE CHARACTERIZATION OF WIRELESS CHANNEL CONDITIONS

State	Signal Strength
Excellent	$> 75\%$
Good	45%-75%
Fair	20%-45%
Poor	$< 20\%$

TABLE III
END-TO-END TCP THROUGHPUT IN DIFFERENT CHANNEL CONDITIONS

Trial	Throughput (Mbps)							
	1	2	3	4	5	6	7	8
Excellent	4.59	4.62	4.63	4.57	4.59	4.59	4.60	4.61
Good	3.41	3.58	3.76	3.41	3.83	3.96	4.03	4.07
Fair	2.24	2.36	2.48	2.30	1.85	1.92	2.26	1.70
Poor	0.46	0.03	0.01	0.05	0.02	0.05	0.05	0.08

III. EXPERIMENTAL METHODOLOGY

A. Experimental Environment

An experimental network was set up to study the performance of RealMedia streaming over a wireless LAN. The experimental environment is illustrated in Figure 2.

There are three machines and one access point (AP) in our experiments. The RealServer 8.0 software runs on a Linux machine with a 1.8 GHz Pentium 4 CPU. One laptop acts as the RealPlayer 8.0 client, and another laptop runs the Sniffer Pro 4.6 wireless network analyzer software. Both laptops have an 800 MHz Pentium III processor and a Cisco Aironet 350 network adapter. The AP is a Lucent RG-1000 residential gateway, connected to the server with a 10 Mbps Ethernet card. The AP uses infrastructure mode, and a maximum retransmission limit of 4 at the MAC layer.

Only one multimedia stream is considered in this paper. The streamed video clip is a segment of CBR-compressed rock concert. The characteristics of this clip are summarized in Table I.

B. Experimental Design

In our experiments, we study the streaming performance both with and without background TCP/IP traffic. When there is no competing traffic, we investigate the wireless channel characteristics and the impact on the streamed media quality. For ease of reference, we classify the wireless channel conditions into four qualitative categories: *Poor*, *Fair*, *Good*, and *Excellent*. These signal-strength categories are based on the Link Status Meter on the Cisco Aironet 350 devices, and on prior work by Eckhardt and Steenkiste [6] showing how error rate is related to signal strength. Table II summarizes these categories.

In the competing traffic experiments, background TCP traffic is generated between the streaming server and client. In order to isolate the effect of competing traffic from that of wireless channel error, these experiments used the Excellent channel condition.

For each experiment, three network traffic traces were captured: one trace at the server using `tcpdump`, one trace close to the AP using Sniffer Pro 4.6, and one trace at the client using `tcpdump`. The wireless sniffer traces were used to study the MAC-layer view of the channel, while the `tcpdump` traces were used to study the higher-layer view. The transport-layer protocol was UDP, while the streaming control protocol was RTSP with RDT.

IV. RESULTS

In this section, the experimental results are reported. We begin with a baseline test of WLAN performance. The streaming results without competing traffic are presented after that, followed by the results with competing TCP traffic.

A. Baseline Throughput Results

To establish a reference point for our measurements, we first determine the maximum end-to-end throughput achievable for bulk TCP data transfers in our experimental environment. For this purpose, we use `netperf` [12] to invoke 60-second TCP-STREAM tests between client and server, with an 84 KB receive socket buffer size. Each test was done 8 times.

Table III shows the measured throughput results under different channel conditions. Two observations are evident. First, the weaker the signal strength, the lower the throughput, and the greater the variability in the measured throughput. This is due to the complicated fading characteristics of the wireless channel. Second, we note that the maximum observed throughput of 4.6 Mbps for an Excellent channel is well below[1] the nominal rate of 11 Mbps, reflecting the overhead of the IEEE 802.11b protocol. This result indicates that the 10 Mbps Ethernet connection between the server and the AP is not a bottleneck in our experiments.

B. Streaming Performance without Competing Traffic

B.1 Subjective Assessment of Streaming Quality

The playback of the video and audio streams were very smooth for the Excellent and Good channel conditions. For the

[1] The measured throughput for another vendor's AP was 6.04 Mbps.

Fair channel, the playback of the video was jerky, indicating lost video frames, though the visual quality of displayed video frames was good. The sound quality was good too. Under the Poor conditions, the video playback was jerky, some individual pictures were blurry or truncated, and the audio quality deteriorated. In some cases, the attempt to set up the streaming connection failed. The following sections analyze the underlying causes of these anomalies.

B.2 Effect of Wireless Channel Characteristics

The error characteristics of a wireless channel affect higher-layer protocols. To study errors, we focus on the MAC-layer retransmission behaviour, since each MAC-layer retransmission indicates an error in either the data packet or its ACK.

Figure 3 presents our analysis of MAC-layer retransmissions in different channel conditions. The leftmost column of graphs presents a two-dimensional time series representation of MAC-layer retransmissions, with time on the horizontal axis, and MAC frame sequence number (modulo 100) on the vertical axis. Each '+' in these plots represents a retransmitted MAC-layer frame. The graphs in the rightmost column of Figure 3 show the data rate that the AP indicates in the MAC header of each transmitted frame, under different channel conditions. Comparing the two columns of graphs shows that the sending rate is (as expected) inversely related to the channel error rate. When the error rate is high, the sending rate is low, and vice versa.

The results in Figure 3 show bursty error conditions, since retransmissions are clustered both horizontally in the time domain, and vertically in the sequence number domain. In many cases, there are many consecutive MAC frames that require retransmission, particularly in the Fair and Poor channel conditions. In the Poor channel conditions, 67.5% of the media packets sent require at least one retransmission. While the retransmission rate is high, the MAC-layer retransmission strategy is able to recover most of the missing packets without exceeding the MAC-layer retransmission limit.

Although the 802.11b MAC layer is able to hide most of the wireless channel errors from higher layers, errors still affect the higher layers, in several ways. First, even if a packet reaches the client after MAC-layer retransmission, it takes longer, which affects the application's view of the available network bandwidth. Second, the physical layer's transmission rate is automatically adjusted, and this affects the application's view of the channel. Third, some errors are still left to the application layer to solve, which can trigger application-layer retransmission. The next two sections study the application-layer behaviour.

B.3 Application-Layer Streaming Rate

Figure 4 shows a structural overview of the RealMedia streaming behaviour. These graphs plot the media packet sequence number versus time, for the video (Figure 4(a)) and audio streams (Figure 4(b)).

In the Excellent channel condition, the RealServer starts with a constant 600 kbps video streaming rate. Once the RealPlayer client buffer fills (after 15 seconds or so), the RealServer switches to a 200 kbps video streaming rate (the target rate of this video stream), and maintains that rate for the duration of the session. The streaming structure is similar for both

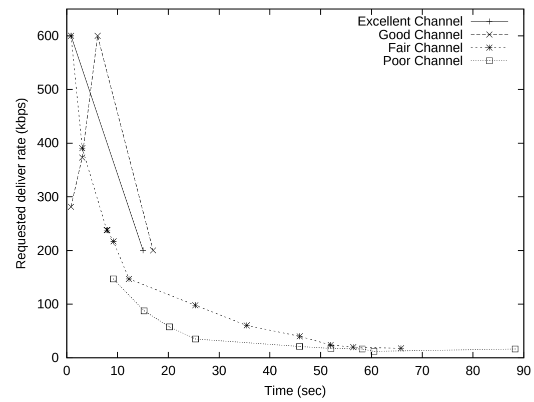


Fig. 5. Requested Streaming Media Delivery Rates in Different Channel Conditions

the video and audio streams, though the overall data rate of each is different (note the different scales on the vertical axes in Figures 4(a) and (b)).

The behaviour for the Good channel condition is qualitatively similar, though the errors experienced early in the session temporarily limit the streaming rate requested by the RealPlayer. Once these errors are handled, streaming rates of 600 kbps and 200 kbps are used throughout the session.

For the Fair and Poor channel conditions, the situation is quite different. In particular, the video streams have a lower streaming rate, well below the target rate. The high error rates cause the RealPlayer client to request a lower streaming rate. In many cases, the RealServer has to skip some media packets in order to meet the requested rate budget. This causes the jerky effect of video playback.

The audio streams, on the other hand, received their targeted rates for all channel conditions. The RealPlayer client requested a streaming rule that favours the audio stream. In all cases, the AP transmitted all 432 audio packets. The slopes of the plots in Figure 4(b) all match the target rate for the audio stream.

For reference purposes, Figure 5 shows the requested delivery rate by the RealPlayer client for the four channel conditions. It can be seen that for a Good or Excellent channel, the client was able to achieve the target rate of the streaming clip. However, for the Fair and Poor channels, the client was unable to achieve the targeted stream rate. The achieved rates for the Fair and Poor channels drop as low as 17.5 kbps and 12.1 kbps, respectively.

B.4 Application-Layer Retransmission

RealSystem uses application-layer retransmission to recover from network-layer transmission errors. In this section, we study the efficiency of this error recovery mechanism in the 802.11b WLAN environment.

Table IV shows the packet level statistics of the streaming experiments. The *packets sent* column shows the total number of video and audio packets transmitted by the server. The next column shows the total number of packets received by the clients. The other three columns present the MAC-layer packet loss, the number of NACKs sent by the client, and the number of successful application-layer retransmissions.

The results in Table IV show that the NACK-based retransmission is reasonably efficient in correcting errors. For example,

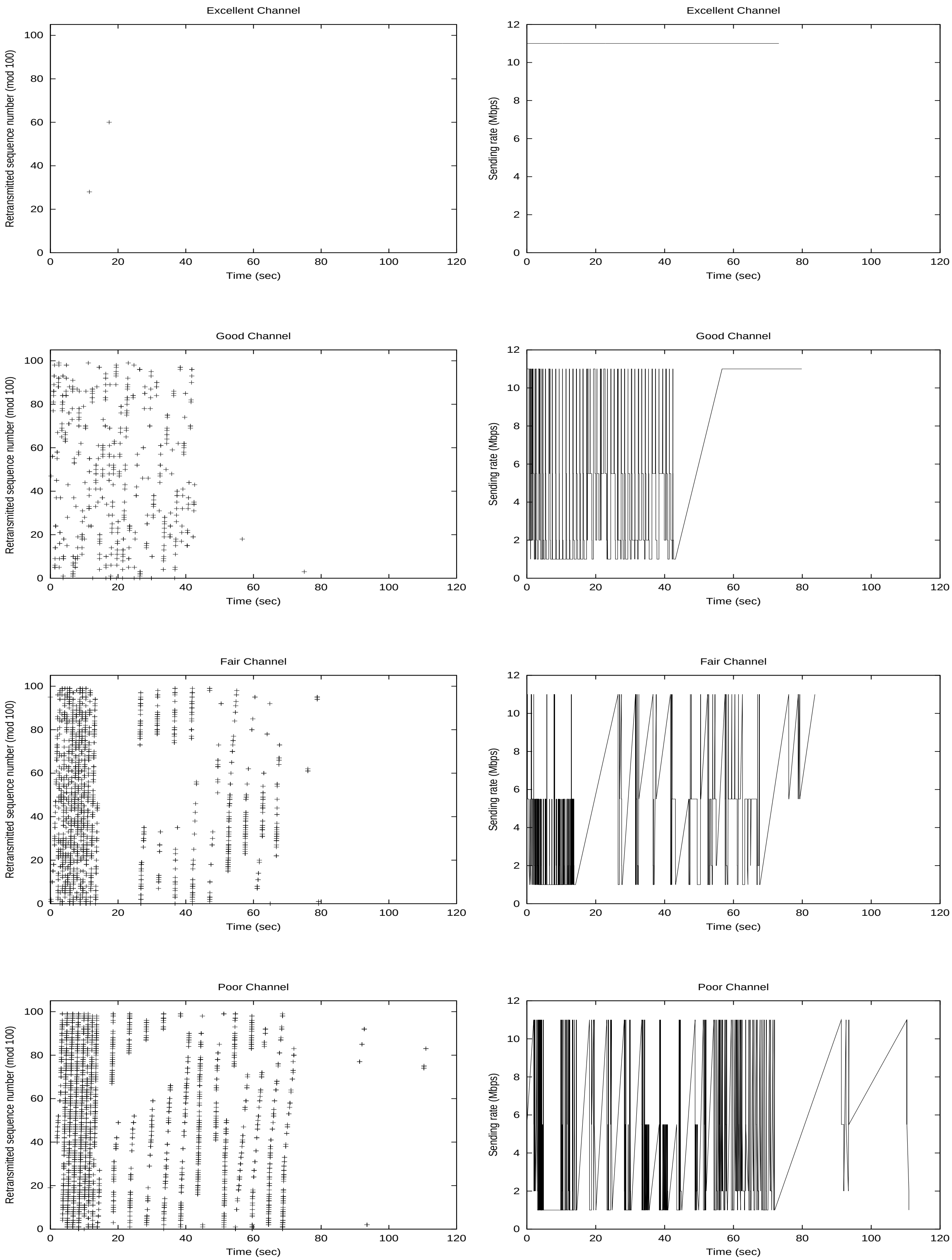
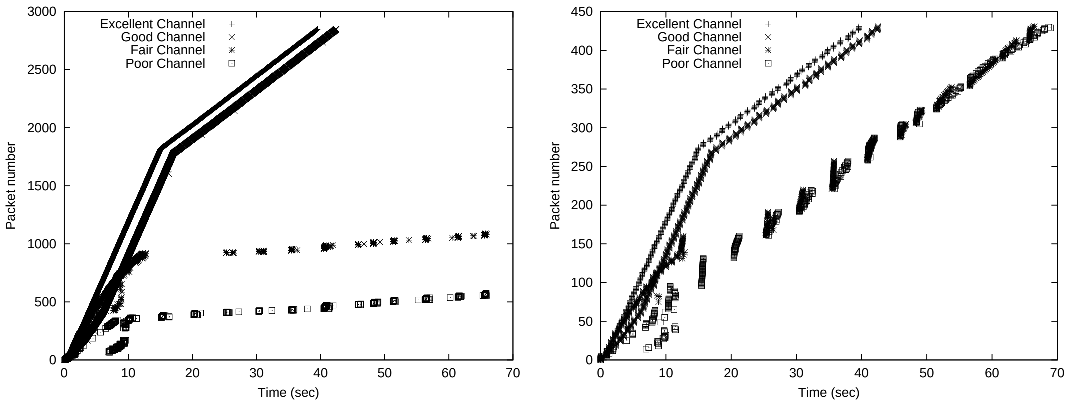


Fig. 3. Illustration of the Impacts of WLAN Channel Conditions on MAC-Layer Retransmissions (left) and Access Point Data Rate (right)



(a) Video streams

(b) Audio streams

Fig. 4. Structural Characteristics of RealMedia Streaming Network Traffic

with the Fair channel condition, 133 of the 148 retransmissions requested by the client were successful (89.9%). For the Poor channel condition, NACK-based retransmission is slightly less effective: 68 out of 96 retransmitted audio packets (70.8%) were successful, and 132 out of 192 video packets (68.8%).

Ignoring timing issues for the moment, the stream quality can be characterized by the "final" effective loss after MAC-layer and application-layer retransmission. The effective loss is zero for Excellent and Good channels. For the Fair channel, one audio packet was lost (0.2%) and 14 video packets (1.3%). This is why the individual picture quality is good. The effective loss for the Poor channel is 28.3% (163 packets) video packets, which explains the blurred video quality. The effective loss for audio packets is 6.9% (30 packets). This explains the occasional deterioration in audio quality.

In reality, every real-time streaming packet has a playback deadline. So even if a retransmitted packet is finally received by the client, it is useless if it misses its deadline. In our experiments (on a LAN), the only packets that miss the deadline are retransmitted packets. With Excellent and Good channels, no packets miss their deadlines (assuming a 4 second startup delay at the client). For the fair channel condition, there is only 1 video packet that misses its deadline. The Poor channel has 19 audio packets and 67 video packets that miss their deadlines. This increases the effective loss to 11.3% (49 packets) for audio and 40% (230 packets) for video in the Poor channel condition.

C. Streaming Performance with Competing Traffic

In this section we investigate the impact of competing TCP traffic on RealMedia stream quality, for the Excellent channel condition. A single client streams RealVideo and RealAudio from the server. At the same time, several TCP connections were created from the server to the client to do bulk file transfers.

Our experiments use a total of 10, 20, 30, 40, and 50 sources, one of which is the RealMedia stream of interest. Since the maximum achievable throughput of the WLAN is 4.6 Mbps, the bandwidth for each source in these scenarios should be 460 kbps, 230 kbps, 150 kbps, 115 kbps, and 92 kbps (assuming equally shared). Notice that for 10 and 20 sources, there is adequate bandwidth for the target rate of the media stream, while for the other scenarios there is not. We thus expect the server to reduce the streaming rate.

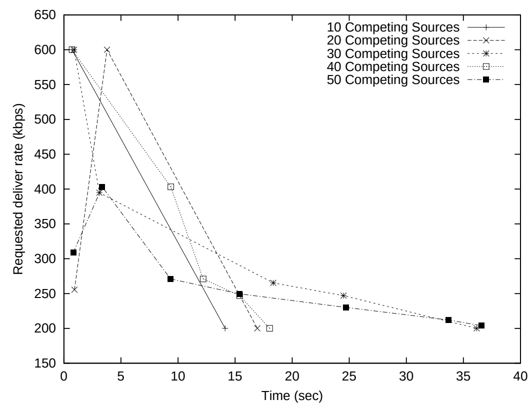


Fig. 6. Requested Media Streaming Rates in the Presence of Competing TCP Traffic

The results from this experiment appear in Table V. For 10 sources, the average throughput for each TCP connection shows that they share the bandwidth as expected. However, for more sources, the TCP streams get slightly less bandwidth than expected.

Figure 6 illustrates this fairness problem, by showing the media streaming rate requested by the RealPlayer client. When bandwidth is plentiful (e.g., 10 and 20 sources), the streaming takes place as usual at 600 kbps and 200 kbps. However, even when bandwidth is inadequate to support the target bit rate, the RealPlayer requests more than 200 kbps. This shows that the bandwidth adaptation algorithm at the RealPlayer is not TCP-friendly.

V. Related Work

To the best of our knowledge, there is no measurement study of wireless video streaming in the literature. Even for Internet video streaming, there are few empirical studies, though we mention several of them here. Mena *et al.* [11] studied the RealAudio traffic from a popular Internet audio server and

TABLE IV

APPLICATION-LAYER RETRANSMISSION STATISTICS FOR REALMEDIA STREAMING

Channel Condition	Stream Type	Packets Sent	Packets Received	MAC-Layer Loss	App-Layer NACK	Successful Retransmit
Excellent Channel	audio	432	432	0	0	0
	video	2850	2850	0	0	0
Good Channel	audio	432	432	0	0	0
	video	2850	2850	9	9	9
Fair Channel	audio	432	431	26	26	25
	video	1090	1076	122	122	108
Fair-poor Channel	audio	432	402	111	96	68
	video	575	412	295	192	132

TABLE V

REALMEDIA STREAMING PERFORMANCE WITH COMPETING TCP TRAFFIC

Total Num Sources	10	20	30	40	50
Avg TCP Thrpt (kbps)	461.1	221.6	143.1	105.6	87.3
Video Pkts Rcvd	2796	2782	2770	2802	2803
Audio Pkts Rcvd	426	424	425	428	432

found that RealAudio traffic shows non-TCP friendly behavior. Wang *et al.* [19] studied RealVideo traffic from several servers to different users. They found that users generally achieve good quality video with an average frame rate of 10 fps, though few achieve full motion video (24 to 30 fps). Loguinov *et al.* [10] conducted an emulated study of streaming low-bitrate MPEG-4 video to home users in more than 600 major U.S. cities. They reported the results in terms of the packet loss, round-trip delay, one-way delay jitter, packet reordering, and path asymmetry. Chesire *et al.* [5] analyzed the stream media workload generated between clients inside the University of Washington and servers outside.

VI. CONCLUSIONS

In this paper, we studied the performance of RealAudio and RealVideo streaming over an IEEE 802.11b wireless LAN under different channel error conditions. While the wireless channel has bursty error characteristics, the 802.11b MAC-layer retransmission mechanism is able to hide most physical-layer channel errors from higher-layer protocols. When needed, the application layer's NACK-based error control is effective in recovering missing packets. The subjective streaming quality is good for Excellent and Good channel conditions, while the Fair and Poor channel conditions produce jerky and blurred pictures. In the worst case, even the audio quality is bad.

In the presence of background traffic, RealPlayer does not compete fairly with TCP connections. When the bandwidth is scarce, the RealPlayer maintains its target rate, to the detriment of TCP flows.

Our future plans include a larger-scale study of wireless media streaming, and experimentation with dynamic packet fragmentation approaches to reduce the impacts of wireless channel errors on media streaming performance.

REFERENCES

[1] I. Aad and C. Castelluccia, “Differentiation Mechanisms for IEEE 802.11”, *Proceedings of IEEE INFOCOM*, Anchorage, AL, April 2001.
[2] ANSI/IEEE Standard 802.11b, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specificiations: Higher-Speed Physical Layer Extension in the 2.4 GHz band”, 1999.
[3] ANSI/IEEE Standard 802.11, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, 1999.
[4] R. Agarwal, J. Ayars, B. Hefta-Gaub, and D. Stammen, “Internet Draft draft-heftagaub-rmff-00.txt: RealMedia File Format”, March 1998.
[5] M. Chesire, A. Wolman, G. Voelker, and H. Levy, “Measurement and Analysis of a Streaming-Media Workload”, *Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS)*, San Francisco, CA, March 2001.
[6] D. Eckhardt and P. Steenkiste, “Measurement Analysis of the Error Characteristics of an In-Building Wireless Network”, *Proceedings of ACM SIGCOMM*, Stanford, CA, August 1996.
[7] G. Holland, N. Vaidya, and P. Bahl, “A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks”, *Proceedings of ACM/IEEE MOBICOM*, Rome, Italy, July 2001.
[8] T. Kuang and C. Williamson, “Measurement Study of RealMedia Traffic”, submitted for publication.
[9] A. Lindgren, A. Almquist, and O. Schelen, “Evaluation of Quality of Service Schemes for IEEE 802.11 Wireless LANs”, *Proceedings of IEEE Conference on Local Computer Networks (LCN)*, Tampa, FL, November 2001.
[10] D. Loguinov and H. Radha, “Measurement Study of Low-bitrate Internet Video Streaming”, *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, November 2001.
[11] A. Mena and H. Heidemann, “An Empirical Study of Real Audio Traffic”, *Proceedings of IEEEE INFOCOM*, Tel Aviv, Israel, March 2000.
[12] Netperf, http://www.netperf.org
[13] C. Papadopoulos and G. Parulkar, “Retransmission-Based Error Control for Continuous Media Applications”, *Proceedings of Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, 1996.
[14] RealNetworks, “RealSystem Production Guide”, http://service.real.com/help/library/index.html
[15] RealNetworks, “Using RTSP with Firewalls, Proxies, and other Intermediary Network Devices”, Version 2.0/rev. 2, http://service.real.com/firewall/fdev.html
[16] RealNetworks, “RealSystem G2 SDK”, http://service.real.com/help/library/index.html
[17] RFC 2326, “rfc2326: Real Time Streaming Protocol (RTSP)”, April 1998, http://www.ietf.org/rfc/rfc2326.txt
[18] RFC 1889, “rfc1889: A Transport Protocol for Real-Time Applications”, January 1996.
[19] Y. Wang, M. Claypool, and Z. Zuo, “An Empirical Study of RealVideo Performance Across the Internet”, *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, November 2001.