Application-Level Measurements of Performance on the vBNS[*]

Michele Clark   Kevin Jeffay
University of North Carolina at Chapel Hill
Department of Computer Science
Chapel Hill, NC 27599-3175 USA
{clark,jeffay}@cs.unc.edu

Abstract

We examine the performance of high-bandwidth multimedia applications over high-speed wide-area networks, such as the vBNS (Internet2). We simulate a tele-immersion application that sends 30 Mbps (2,700 datagrams per second) to various sites around the USA, and measure network- and application-level loss and throughput. We found that over the vBNS, performance is affected more by the packet-rate generated by an application than by its bit-rate. We also assess the amount of error recovery and buffering needed in times of congestion to present an effective multimedia stream to the user. Lastly, we compare our application-level measurements with data taken from a constellation of GPS-synchronized network probes (IPPM Surveyors). The comparison suggests that network-level probes will not provide information that a multimedia application can use to adapt its behavior to improve performance.

1. Introduction

Much work has been applied to the problem of maximizing the performance of multimedia applications on the public Internet. Reports of bursty throughput and often-crippling packet loss during multimedia transmissions have resulted in the call for quality-of-service (QoS) guarantees or differentiated services in the network. Meanwhile, multimedia applications are becoming more complex and demanding larger amounts of bandwidth in order to provide users with enriched services and more immersive, realistic interfaces.

For example, consider a tele-immersion system under development at the University of North Carolina [9]. This system attempts to provide a life-size 3-D tele-conference projected on three walls of a room. This application has the same latency and loss requirements as a traditional videoconferencing application, however, the tele-immersion application requires sustained bandwidth on the order of 20-40 Mbps. To support such high bandwidth applications, there is a push for the development of a new Internet in the United States with speeds 100 times faster than today's Internet. This comes in the form of the Internet2 and the *very-high-speed Backbone Network Service* (*vBNS*). But, with multimedia applications demanding an increasing amount of bandwidth, will these new high-speed services be enough? If these services are not enough, should the network provide QoS guarantees for multimedia applications, or should the application be made to adapt to congestion in the network?

In this paper, we focus on how applications can obtain knowledge about the performance of the network in order to adapt to changing conditions (*e.g.*, congestion). Applications can either rely on network-level measurements of latency and packet loss to estimate the degree of network congestion, or they can directly measure the performance of the network themselves. The difference is that network-level monitoring tools take more precise measurements but have no knowledge of application concerns such as packet inter-dependence. It is therefore unclear if one can use network-level measurements such as packet loss rates as indicators of a multimedia application's likely performance. For example, in this case ideally we would like to know the application's "goodput," which is the amount of data that the remote application received and can actually use (*e.g.*, the number of video frames received). Since only the application knows how media units are distributed among packets, is the application itself the best place to measure goodput?

We look at one high-speed backbone service, the *vBNS*, and report on its performance as seen by a tele-immersion application. The application-level measurements we obtain provide information that could be useful to multimedia application developers in creating applications that are resilient in the face of network pathologies. We found that the *vBNS*, is *access-constrained* [8], which means that the network's performance is more sensitive to the number of packets it must handle than to the number of bits. We also found that introducing relatively small amounts of error control into the data stream could at times result in large benefits in the overall performance of the application.

2. Background and Related Work

Researchers from both INRIA and Lucent have looked at the problem of packet loss on the public Internet and studied how it affects multimedia applications. Bolot *et al.* [3] reported on the occurrence of consecutively lost packets in low-bandwidth (~100 kbps) audio streams transmit-

ted over the public Internet and suggested the use of error control mechanisms such as Forward Error Correction (FEC) and Automatic Repeat Request (ARQ) to recover from these packet losses.

Boyce and Gaglianello [4] studied the effects of packet loss on MPEG streams of 384 kbps and 1 Mbps. They found that seemingly innocent packet loss rates could cause much higher frame loss rates. They also observed access constraints on the public Internet.

We look at application-level measurements taken over a backbone network that has over 10 times the bandwidth of the public Internet. Will the larger capacity of the *vBNS* be able to handle larger bandwidth applications?

The *vBNS*, sponsored by the National Science Foundation (NSF) and MCI Telecommunications (MCI World-Com), provides OC-12 (622 Mbps) IP-over-ATM backbone service to numerous universities and research labs in the United States [5]. Miller *et al*. [6] performed UDP throughput experiments to measure the performance of the *vBNS* backbone. These tests were performed between points on the backbone of the network over both non-routed, ATM-only, paths, and over paths routed via IP routers. Over the OC-3 links to the *vBNS* backbone, Miller reported a maximum UDP throughput of 135 Mbps for non-routed ATM traffic and 120 Mbps for IP routed traffic. UDP tests of the OC-12 ATM interfaces resulted in a throughput of 469 Mbps.

3. Measurements

We look at data captured by our own application and compare it to statistics collected by specialized monitoring machines located near some of our test sites.

3.1 Surveyor Measurements

Advanced Network & Services and the Common Solutions Group (CSG), as a part of the IP Performance Metrics (IPPM) program, have deployed "Surveyor" machines to various sites around the USA [1]. Each of these machines is equipped with a global positioning system (GPS) card, which receives time updates via satellite, effectively synchronizing all of the computers' clocks. These machines continually collect precise one-way delay and loss statistics between each pair of Surveyor machines [2].

There is a Surveyor located at UNC, as well as at one of our test sites, the University of Washington (UW). Another of our test sites, the University of Illinois-Chicago (UIC) does not have a Surveyor of its own, but there is a one at a nearby university, the University of Chicago.

Surveyors send out probe packets 3-5 times a second, which means that we will have a set of measures from a Surveyor at the same time that we run an experiment. As the Surveyors take network-level measurements, they have no notion of application-level data units such as frames or

of the inter-dependence of packets within frames. Our application-level measurements can reflect these dependencies (*e.g.*, in the form of frame loss rates rather than packet loss rates).

Network-level measurement tools, like the Surveyor, cannot be easily tuned to measure the performance that a specific application might receive. We found that the *vBNS* is access-constrained, such that the packet-rate is closely tied to the loss rate. Our test application has packet-rates that range from about 300 packets/second to almost 3,000 packets/second. If the network is access-constrained, an application will have very different packet loss rates depending upon its packet-rate. However, a Surveyor would report only one measure of packet loss for a certain path and time of day, and would not be able to provide application-specific loss rates.

3.2 Application-level Measurements

UNC is connected to the *vBNS* in Atlanta, GA. Traffic flows over 100 Mbps Ethernet, through a campus router, over the North Carolina GigaPOP at DS-3 (45 Mbps) rates to a direct connection to the *vBNS* in Atlanta [7]. Each of our test sites is also connected to the *vBNS* by at least a 100 Mbps Ethernet link. We ran our application and collected data at the University of Illinois-Chicago and the University of Washington.

We wrote an application that generates data at rates similar to a video stream produced by a high-bandwidth tele-immersion application. The data stream was partitioned into frames and sent in frame-sized bursts at a rate of 30 frames/s (fps). The sender was located at UNC, while the receiver was located at one of our remote test sites. The sender sent UDP packets, and the receiver recorded the header of each packet received, along with the time of receipt by the application. Post-processing of this data provided statistics on loss and delay-jitter.

Each set of tests consisted of sending a 5-minute, 3 Mbps stream (9 packets/frame, 270 packets/s) followed by a 5-minute, 30 Mbps stream (87 packets/frame, 2,610 packets/s) to a remote site. We ran a set of tests (3 Mbps followed by 30 Mbps) to each test site around 8 a.m., 10 a.m., 12 p.m., 2 p.m., and 4 p.m. EDT for several weeks. There was a 3-minute break between the 3 Mbps test and the 30 Mbps test, and a 15-minute break between tests to each site.

To measure the effect of a larger packet-rate versus the effect of a larger bit-rate on packet loss, we modified our video stream simulator to send frames of minimum-sized Ethernet packets at the same rate as the 30 Mbps stream. This resulted in a 500 kbps data stream (87 packets/frame, 2,610 packets/s). These tests were run in the same manner as above, but with the 500 kbps stream in place of the 3 Mbps stream. Tests were performed around 8 a.m., 10 a.m., 12 p.m., 2 p.m., and 4 p.m. EST.

4. Results

4.1 Average packet loss vs. average frame loss

Packet loss is an important factor to consider when determining the performance of a multimedia application. However, in an application that aggregates packets into frames, frame loss is also an important measure of application performance. Network-level measurements will only report packet loss rates. Can we directly translate that information into frame loss rates? We would expect that low packet loss rates would result in low frame loss rates, which in turn, would result in a successful experience for the user. Figure 1 shows how packet loss can impact frame loss. We count a frame as lost if one or more packets from that frame were not delivered. There are several incidents of low packet loss (less than 5%) that contribute to high frame loss (over 50%). If we can recover from these few packet losses, we could possibly make large gains in frame throughput (see Section 4.4).

4.2 Packet loss over time

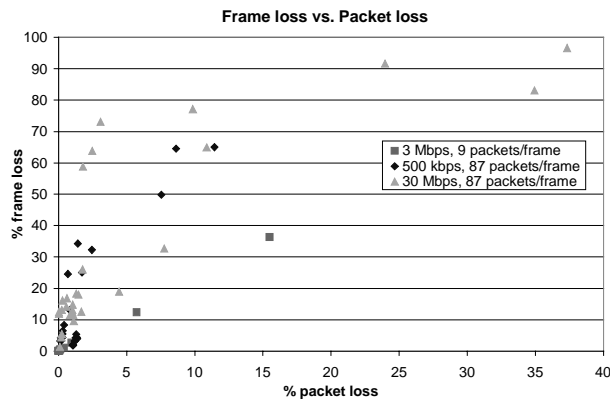Packet loss in a network can vary not only throughout the day, but also during the course of a 5-minute experiment. Router queues fill up and then are flushed, resulting in a bursty loss pattern. By recording the time that packets were delivered to the application, we were able to determine the percentage of packets lost during a specific interval of time. We plotted these numbers over 1-second intervals (Figures 2 and 3).

As shown in Figure 2, packet loss rates for the 3 Mbps streams are consistently lower than packet loss rates for the 30 Mbps streams. We believe that the difference lies in the increased packet-rate, rather than the increased bit-rate. To test this, we measured loss rates for a 30 Mbps stream and a 500 kbps stream, both of which send packets at a rate of 2,610 per second. Figure 3 shows that the loss rates for these two streams are comparable. We believe that this is because the network becomes access-constrained (*i.e.*, router queues fill with packets).

We can compare the data from Figure 3 to that collected by a Surveyor on the same route around 2 p.m. EST (Figure 4). Even though both the 500 kbps and 30 Mbps streams had loss rates averaging under 10%, the Surveyor reports only a few short bursts with losses of 20% and no losses for the remainder of the run. These graphs show data from the same link during the same time period, and yet their results are very different. The Surveyor data does

Figure 1. Effect of packet losses on frame losses. Each point represents the average loss over a 5-minute interval.

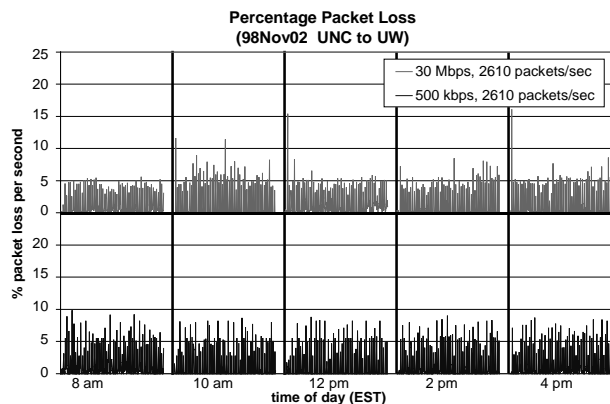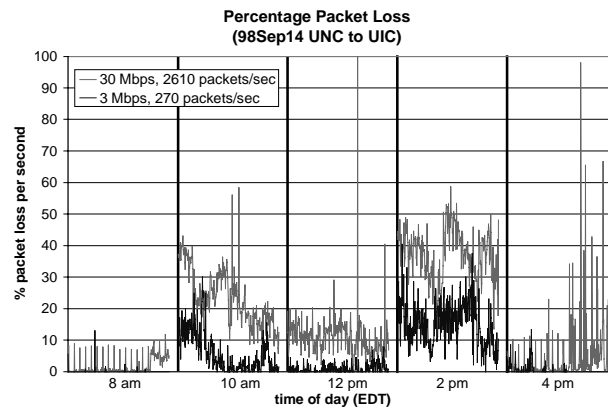**Figure 2.** Effect of packet-rate on packet loss. Each section shows the results from a 5-minute run.

Figure 3. Effect of bit-rate on packet loss. Each section shows the results from a 5-minute run.
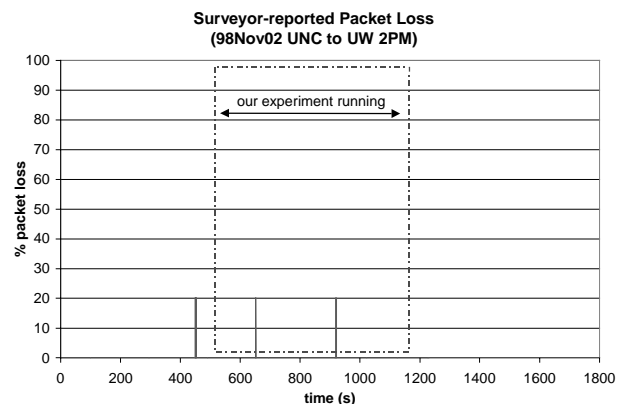

Figure 4. Network-level measurements of packet loss around 2 p.m. EST.

show that there was some packet loss, but it does not provide information about the overall bursty nature of loss that we see in the application-level measurements.

4.3 Packets lost per frame

In order to deliver a video frame suitable for playout, either all of the packets of a frame must be received, or the application must perform some type of error recovery. With large frames spanning many packets, error recovery is critical as the chances that a frame arrives with one or more missing packets increases. To give an indication of the amount of error recovery required for a stream, Figure 5 shows the CDF of frames with a certain number of packets missing per frame. The line at $x = 5$ on this graph shows how well recovering from the loss of 5 packets per frame would affect our overall frame throughput.

In Figure 5 during the 98Sep14 12 p.m. run, about half of the frames are missing more than 5 packets. Here error recovery of up to 5 packets marginally improves performance, but it does not make a large difference in frame throughput (Figure 6 – left side). When the majority of frames have packet losses of 5 or less, such as in the 98 Oct30 2 p.m. run, we expect that error recovery will effectively reconstruct the stream (Figure 6 – right side).

If there are fewer packets per frame, the chance of a frame missing a packet decreases. To reduce the packet rate, packets must grow. When packets exceed the minimum network MTU, the network will fragment packets resulting in an increased packet-rate. When designing a multimedia application, we must not only consider packet-rate (because of access-constraints), but also packet size and the network MTU.

4.4 Effect of error recovery on goodput

During periods of high loss, an application may need to aggressively perform error recovery in order to sustain a given level of quality. Figure 6 shows the actual frame goodput for different degrees of error recovery.
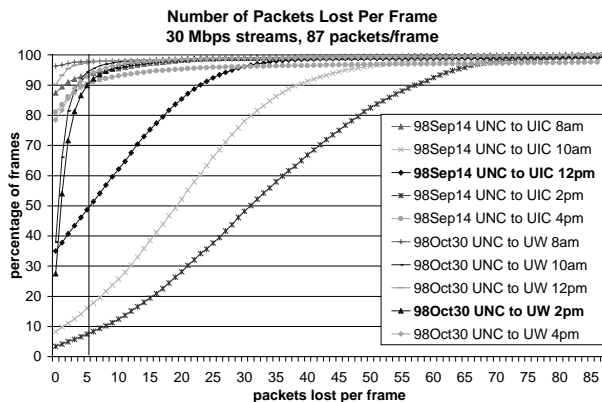
Error recovery performed on a (high loss) 98Sep14 transmission does not have a noticeable effect on frame throughput. Even correcting by as many as five frames only slightly increases the frame rate. At other times, error recovery has a dramatic effect. Just correcting for one packet in a 98Oct30 transmission almost doubles the frame goodput. Correcting for 5 packets brings the stream back to near its original 30 fps.

We can see that often during a run, only a small number of packets per frame are lost, so error recovery can greatly improve frame rates. At other times, congestion is so high that a large number of frames are missing many packets and will be immune to the effects of even the most aggressive error recovery schemes.

4.5 Delay jitter

We use send and receive timestamps to calculate a bound on the one-way delay and delay-jitter of packets and application-level frames. Delay-jitter gives us an idea of how jerky the media playout would be if these were real frames of video. Ideally, we would like these times to remain relatively constant throughout the run, but because of the burstiness of the traffic, these values will change.

Figure 7 shows our measurements of delay-jitter for a 30 Mbps stream. The spikes in the graph show that the range of jitter measured by the application is between 20-100 ms. In principle one can use these statistics to design a jitter-buffering scheme that would perform well in the presence of delay-jitter. By monitoring the average jitter over a run, one can dynamically estimate the amount of buffer space needed to minimize playout gaps.

Figure 8 shows the delay-jitter from a Surveyor on the same path. We would expect this to match up reasonably well with the delay-jitter statistics that our application measured, but the Surveyor shows a much lower range of delay-jitter, between 5-20 ms. Again, we see that even though the Surveyor is monitoring the same path as our
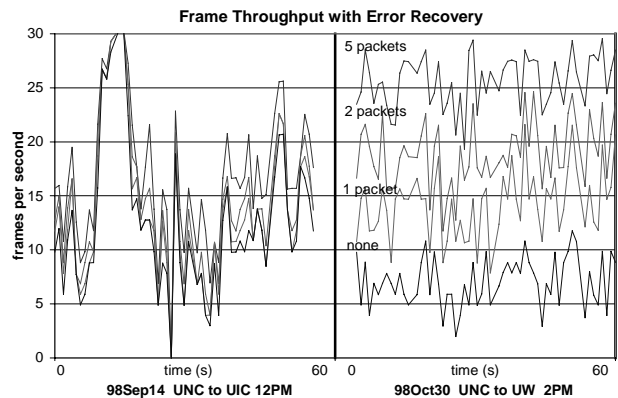
Figure 5. CDF of packets lost per frame. The line at *x=5* will help determine if error recovery of 5 packets will noticeably affect frame goodput.

Figure 6. Effect of error recovery on frame goodput. The left side corresponds to the stream in Figure 5 where 50% of the frames had 5 or more packets missing. The right side corresponds to the stream in Figure 5 where 90% of the frames are missing 5 packets or less.
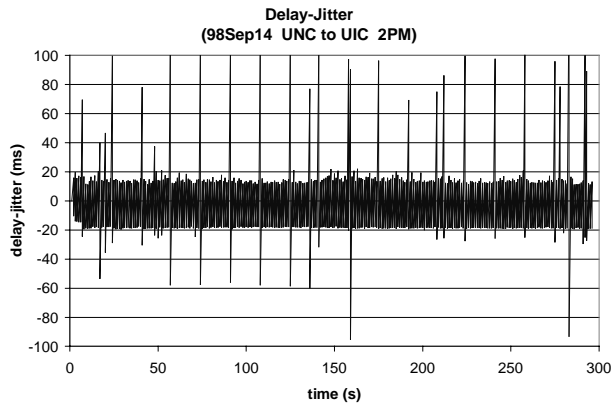
Figure 7. Delay-jitter as measured by our application.

Figure 8. Delay-jitter as measured by the network-level Surveyor.

application, we are presented with conflicting views of the network. This provides another illustration that network-level performance information is not likely to be useful to applications as they attempt to adapt their performance to levels of congestion the network. To be successful, applications must monitor their performance themselves.

5. Conclusions

We have studied an application's view of performance over the *vBNS*, a high-bandwidth backbone network service. Application-level measurements provide useful information on how to design multimedia applications in order to take advantage of the nature of high-speed services like the *vBNS*. To obtain these measures, we simulated a high-bandwidth tele-immersion application, sending data at rates of 500 kbps, 3 Mbps, and 30 Mbps to various test sites around the US. We used the application to measure loss and throughput statistics. By changing the bit-rate and maintaining a constant packet-rate, we saw that the network was access-constrained. Loss rates were as high for the 500 kbps stream as they were for the 30 Mbps stream. We can use these measurements to determine efficient packet sizes, packet-rates, and jitter-buffer sizes to maximize smooth playout of data at the receiver. We can also use these results in developing error control schemes that would best fit our applications.

These types of indications might not be possible to gather from network-level measurements alone. Network-level measures may give an estimate of packet loss, but goodput, which considers the inter-dependence of packets, would not be reported. Measurement devices, such as the Surveyor machines, give an idea of performance over a network, but might not accurately estimate how a particular application would perform. Since the network is sensitive to packet-rates, if an application has a higher packet-rate than the Surveyor probe packet-rate, actual loss could be much worse than observed by the Surveyor.
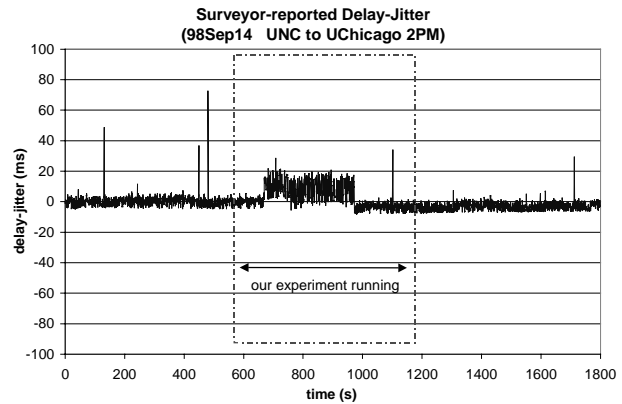
We have determined that it is possible to run a 30 Mbps application with reasonable latency over the *vBNS*, but not without times of significant packet loss. The performance of individual applications at any given time is impossible to predict, but with the types of measurements presented here, an application developer could estimate typical performance and, more importantly, design the application to handle periods of loss and extreme latency.

6. References

[1] Advanced Network & Services, *Surveyor Home Page*, *http://www.advanced.org/mm-support/support*, Oct. 1998.

[2] Advanced Network & Services, *About the Surveyor Project*, *http://www.advanced.org/surveyor/about. html*, Oct. 1998.

[3] Bolot, J-C., Crepin, H., & Vega-Garcia, A., *Analysis of Audio Packet Loss in the Internet*, Lecture Notes in Computer Science, Vol. 1018, 1995, pp. 154-165.

[4] Boyce, J. M. & Gaglianello, R. D., *Packet Loss Effects on MPEG Video Sent Over The Public Internet*, ACM Multimedia, Bristol, UK, 1998, pp. 181-190.

[5] MCI WorldCom, *MCI and NSF's very High Speed Backbone Network Service, http://www.vbns.net/*, October 1998.

[6] Miller, G. J., Thompson, K., and Wilder, R., *Performance Measurement on the vBNS*, In Proceedings of the Interop '98 Engineering Conference, Las Vegas, NV, May 1998.

[7] North Carolina Networking Initiative, *NCNI Architecture, http://www.ncgni.net/architechture.html*, November 1998.

[8] Talley, T. & Jeffay, K., *Two-Dimensional Scaling Techniques for Adaptive, Rate-based Transmission Control of Live Audio and Video Streams*, ACM Multimedia '94, San Francisco, CA, Oct. 1994, pp. 247-254

[9] Welch, G., *UNC Participation in the National Tele-Immersion Initiative, http://www.cs.unc.edu/~welch/teleimmersion.html*, October 1998.