# Measurements of the Congestion Responsiveness of Windows Streaming Media

James Nichols, Mark Claypool, Robert Kinicki, and Mingzhe Li
Computer Science Department at Worcester Polytechnic Institute
Worcester, MA, 01609, USA
{jnick|claypool|rek|lmz}@cs.wpi.edu

## ABSTRACT

While previous research has shown that streaming media can respond to network congestion, it is not known to what extent commercial products are responsive. This research characterizes the bitrate response of Windows Streaming Media (WSM) to changes in network capacity and loss rate. A streaming media test bed was built to systematically vary network and content encoding characteristics and measure WSM congestion responsiveness. The results demonstrate that WSM improve response to congestion when content is encoded into several bitrates by the content provider.

**Categories and Subject Descriptors:** C.2.m [Computer-Communication Networks]: Miscellaneous
**General Terms:** Performance, Design.
**Keywords:** Streaming Media, Measurement.

## 1. INTRODUCTION

Unlike traditional network applications such as file transfer or Web browsing, streaming media often uses UDP to meet its bitrate and timing requirements. Without application-level congestion control, streaming media flows over UDP may not be TCP-friendly[1] and thereby cause retransmissions, queuing delays, and timeouts for concurrent TCP flows. In the worst case, unresponsive UDP flows can consume all the available network capacity with packets that are subsequently dropped at a congested router [4].

The response to congestion of commercial streaming media players has a large impact on network performance. Of the three dominant commercial streaming media products (Microsoft Windows Streaming Media, RealNetworks RealSystems, and Apple QuickTime), RealNetworks has the most content stored on the Web, with Microsoft Windows media having a close second [7]. Since our earlier work has explored RealNetworks video [2], and there are indications that Microsoft Windows media will soon become the dominant streaming solution, this work focuses on Windows Streaming Media (WSM).

[1]A flow is *TCP-Friendly* if its bitrate does not exceed the maximum bitrate from a conformant TCP connection under equivalent network conditions.


*NOSSDAV'04,* June 16–18, 2004, Kinsale, County Cork, Ireland.


Past work on streaming video has included passive measurement studies [10, 3], performed through log analysis, and active measurement studies using custom tools [2, 6]. However, without being able to systematically control server-side streaming parameters and measure their impact on network performance, these studies provide little insight into the interaction between streaming client and server.

WSM supports *Intelligent Streaming* [1], an application level means of reducing the streaming bitrate. Intelligent Streaming lowers the bitrate of the stream in mid-playout during network congestion and then switch back to a higher bitrate when congestion abates. Research on RealNetworks [2], which also supports a bitrate adjustment technique called *SureStream*, suggests using such an approach in response to congestion effectively reduces the impact of streaming media on other flows. However, to the best of our knowledge, there has been no published systematic study of the responsiveness of WSM to congestion.

Encoded content must contain multiple bitrates to make effective use of the features that Intelligent Streaming or related technologies provide [1, 2]. Multiple bitrate encoding places many different encoded video streams into one streaming object while allowing the client to request a single object for streaming. Upon connection, the server determines which encoded bitrate contained in the object to send based on client and network performance. With single bitrate encoding or with multiple bitrate encoding where the lowest bitrate stream is selected, upon congestion, the server is left to "thin" the stream by first decreasing the video frame rate, and then, if needed, by sending only audio.

The goal of this investigation is to experimentally measure the behavior of WSM in the presence of congestion under various network conditions and content encoding parameters with particular attention to the effects of Intelligent Streaming and content encoding on network performance. This study shows that the content provider can make judicious decisions about encoding rates and number of encoding levels to improve the responsiveness of WSM to network congestion. Network practitioners should find this information useful as they strive to accommodate the increasing amounts of streaming traffic. Moreover, the level of detail provided with respect to the behavior of video streams will facilitate building more sophisticated models of streaming traffic than the simple CBR (constant bitrate) distributions used by many researchers.

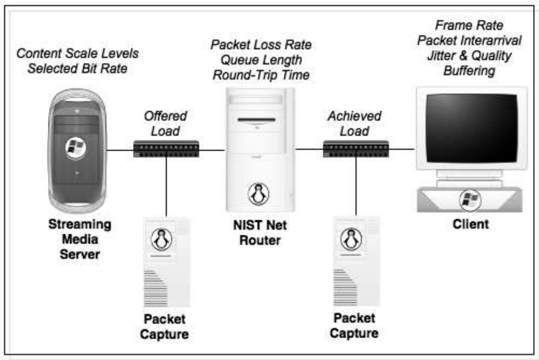


**Figure 1: Streaming Media Testbed**

## 2. METHODOLOGY

Figure 1 shows the testbed built to measure WSM under controlled, experimental conditions. On the left is a high-performance streaming media server running Windows Server 2003 and the latest release of Windows Media Services, version 9. The streaming content, encoded bitrate and encoding levels are controlled at the server. In the middle is a Linux router running the NIST Net network emulator[2] used to control the capacity, loss rates and latency characteristics of packets flowing between the client and server. On the right is a client running the latest Windows Streaming Media core, version 9, along with our custom MediaTracker software[3] that allows us to record client-side performance statistics, including frame rate, packet inter-arrival times, jitter, and media quality. At each side of the router is a Linux machine where tcpdump[4] captures packets to passively measure the offered and achieved network loads.

## 3. RESULTS

### 3.1 Single Bitrate Clips

Our first objective was to examine basic WSM behavior. Using the NIST Net router, the bottleneck capacity was constrained to 725 Kbps, a typical Internet broadband capacity and the latency was set to 45 ms, in the range of measured Internet latencies [5]. These are the canonical settings throughput this paper, unless otherwise indicated. A TCP transfer was started using iperf[5] 10 seconds before a WSM clip and ended 10 seconds after the WSM clip. The concurrent TCP flow is not meant to induce congestion, but rather provides a visual mechanism to judge when the WSM flow is TCP-friendly. The WSM server was provided with a 60 second video clip encoded at 340 Kbps, which is roughly equal to the streaming flow's fair share of the available capacity.

Figure 2 shows the TCP and WSM flows competing for available capacity over the life of the 60 second clip. The initialization of the WSM stream begins with a 5 second period where no data is sent and is followed by about a 10 second period where WSM streams the clip at 500 to 600 Kbps to fill a client-side playout buffer. Subsequently,

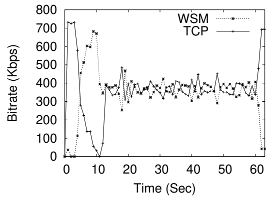


**Figure 2: Bitrate for 340 Kbps Clip**

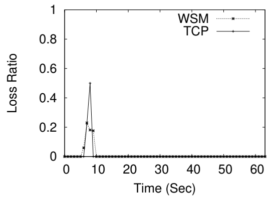


**Figure 3: Loss Ratio for 340 Kbps Clip**

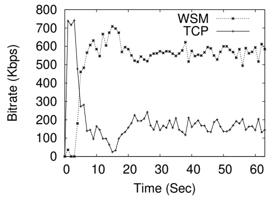


**Figure 4: Bitrate for 548 Kbps Clip**

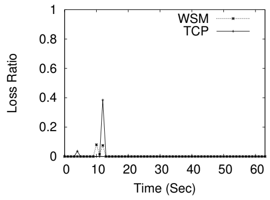


**Figure 5: Loss Ratio for 548 Kbps Clip**

WSM lowers its transmission rate to 340 Kbps. This experiment illustrates clearly that WSM behavior is characterized by two distinct periods - a buffering phase and a playout phase. This delineation guides subsequent analysis in this investigation.

Figure 3 reinforces the concept of two distinct WSM behavior periods by showing that the loss rate experienced by both the TCP and the WSM flows during the buffering period reaches anywhere from 20-40%. To ensure that the NIST Net router buffer was not responsible for such a high loss rates, a series of experiments (not shown here due to lack of space, but available in [8]) were conducted to systematically show that queue sizes larger than 60 packets resulted in the least amount of loss and the most consistent throughput. In all subsequent experiments, a queue size of 80 packets was used.

While Figure 2 suggests that the WSM flow is TCP-friendly during the playout period, this is not due to a deliberate decision by Intelligent Streaming. The results of repeating this experiment with a 540 Kbps clip presented in Figures 4 and 5 show that the streaming flow clearly consumes more than its fair share of the available capacity. The ideal streaming media server would have made use of the Intelligent Streaming feature to "thin" the stream and send fewer video frames to lower the bitrate in response to limited network resources. Instead, the TCP flow is denied its fair share of the available capacity. Furthermore, during the buffering period there are times when both flows experience heavy packet loss.

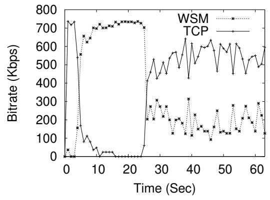


**Figure 6: Bitrate for 1128 Kbps Clip**

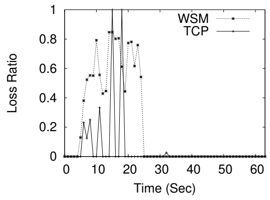


**Figure 7: Loss Ratio for 1128 Kbps Clip**

[2] http://snad.ncsl.nist.gov/itg/nistnet/
[3] See http://perform.wpi.edu/real-tracer/, used in [6]
[4] http://www.tcpdump.org/
[5] http://dast.nlanr.net/Projects/Iperf/

However, if a clip with a bitrate even *higher* than the available capacity (e.g, 1128 Kbps for our testbed) is streamed, WSM changes its transmission rate to a much lower bitrate, as shown in Figure 6. In fact, the chosen steady-state playout rate undershoots the fair-share bitrate. Examination of the offered load during this experiment reveals that WSM initially attempts to respond to packet loss during the buffering phase with a "fire-hose" approach that sends more traffic. As indicated by Figure 7, this yields a bitrate much higher then the content encoded rate and packet loss rates of over 80%!

## 3.2 Range of Single Bitrate Clips

Ten experiments, each with three replications, with a 60 second clip was run on the testbed where the clip was encoded with a different single bitrate for each experiment. The ten experiments where repeated using three distinct capacities: 250 Kbps (a low-speed broadband connection), 725 Kbps (a typical broadband connection), and 1500 Kbps (a high-end broadband or T1 connection). In these experiments, no loss is introduced by the NIST Net router. Figures 8-13 show bitrates and loss rates for these experiments during the buffering period.

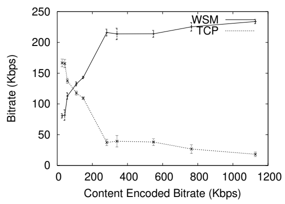


**Figure 8: Buffering Bitrate with 250 Kbps Capacity**

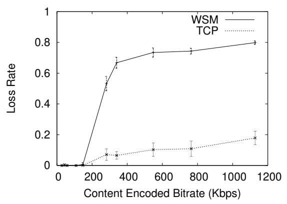


**Figure 9: Buffering Loss Ratio with 250 Kbps Capacity**

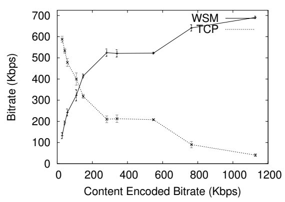


**Figure 10: Buffering Bitrate with 725 Kbps Capacity**

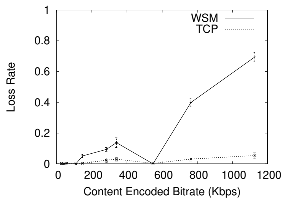


**Figure 11: Buffering Loss Ratio with 725 Kbps Capacity**

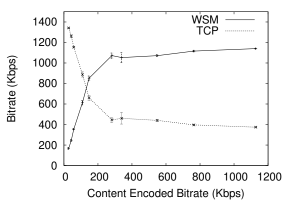


**Figure 12: Buffering Bitrate with 1500 Kbps Capacity**

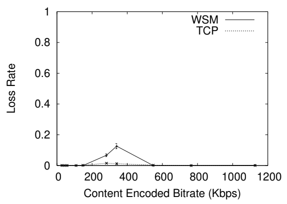


**Figure 13: Buffering Loss Ratio with 1500 Kbps Capacity**

Figures 8-13 clearly illustrate that WSM has little regard for network conditions during its buffering period. Buffering rates are proportional to the content encoding rate until the encoding rate exceeds the bottleneck capacity. Beyond this

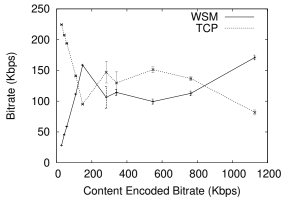


**Figure 14: Post-Buffering Bitrate with 250 Kbps Capacity**

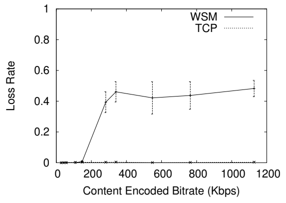


**Figure 15: Post-Buffering Loss Ratio with 250 Kbps Capacity**

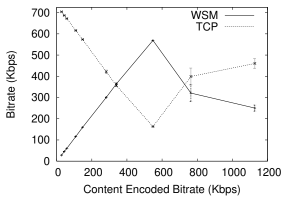


**Figure 16: Post-Buffering Bitrate with 725 Kbps Capacity**

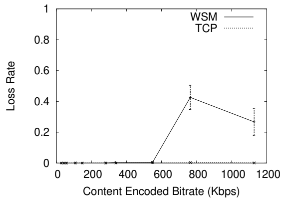


**Figure 17: Post-Buffering Loss Ratio with 725 Kbps Capacity**

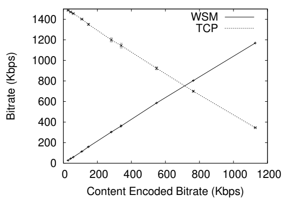


**Figure 18: Post-Buffering Bitrate with 1500 Kbps Capacity**

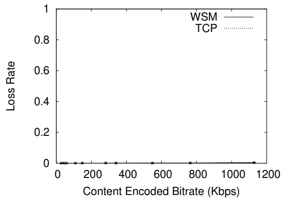


**Figure 19: Post-Buffering Loss Ratio with 1500 Kbps Capacity**

point, the loss rates are high, reaching almost 80% in some cases, with the WSM loss rate much higher than the TCP loss rate due to WSM's higher sending rate.

Figure 11 shows that as the encoding rate is increased, the WSM loss rate increases until a noticeable dip in the measured loss rate at the 548 Kbps encoding rate. Clearly, the behavior of WSM during buffering changes between the 340 Kbps and 548 Kbps encoding rates. The achieved bitrates during buffering for the 340 Kbps clip and 548 Kbps clip are similar. This is not what we would expect, as the trend for lower bitrates indicates that the buffering rate increases with encoding rate. For the clips encoded at rates below 548 Kbps, WSM buffers them at two to four times the encoding rate. However, for the 548 Kbps clip, it appears to buffer at the encoding rate. Perhaps, WSM has detected that the network cannot support a higher rate. Examination of the packet traces reveals that the server is sending packet-pair estimates right before starting buffering, presumably using techniques in [9] to determine capacity limits. We leave further exploration of this capacity estimation as future work.

Examination of packet level data reveals that WSM traffic is bursty in nature. This is evidenced visually by looking at the arriving packet sequence numbers versus time. Figure 20 shows the packet sequence number versus time for the 340 Kbps clip and clearly shows the packet bursts. The "spikes" in Figure 20 are out-of-order sequence numbers from the retransmission of packets lost at the NIST Net router because of the per-packet drop-tail queue. A possible explanation of this behavior is that at the lower bitrate, WSM simply shortens the transmission time between packets bursts to buffer more quickly.

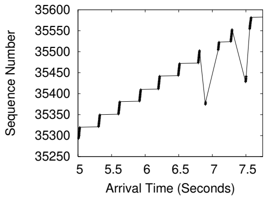


**Figure 20: Packet Sequence Number versus Time for 340 Kbps Clip**

These results show that during the buffering period, WSM is unresponsive to network conditions and congestion. However, the behavior of WSM is different during the period after buffering. The results during the post-buffering period are shown in Figures 14-19 where there is some response of WSM to limited network resources. In particular, in the case of a low-speed broadband connection (250 Kbps) shown in Figures 14-15, the bitrate of WSM during the post-buffering period is linear with the encoded bitrate up to the bottleneck capacity. After this point, for higher encoded bitrates the loss-rates can approach almost 40%, but the content is eventually thinned and uses less than its fair share of the capacity. In Figure 14 the data points for WSM and TCP show a different relationship than the preceeding bitrates. The explanation for this phenomenon is that the thinning of the stream does not take place directly after the buffering period. Thus, in the case of the highest encoded bitrate clip, the average over the entire post-buffering period is correspondingly higher, since the clip encoding is so much higher then the other clips. It is at this rate that WSM attempts to stream the clip before thinning.

Thus, measurement of a simple average over the entire post-buffering period misses some of the behavior of the WSM flow, particularly *when* this thinning occurs. In some experiments with a 548 Kbps bitrate clip streaming over a 250 Kbps link we see that the clip streams for almost 40 seconds at the 548 Kbps bitrate, inducing massive amounts of packet loss (over 80%), but eventually the stream is thinned, only requiring about 50 Kbps of capacity, which has significant implications for streaming clips with long duration. This behavior leads us to two conclusions. First, by looking at an average of bitrate over time, WSM may be appear to be TCP-friendly, but there are periods where it is unfriendly followed by periods where it is friendly. Second, the bitrate of the WSM flow must be examined along with the corresponding loss rate over the same period to seek insight into the true behavior of WSM. Thus, as in the case of Figures 14-15, we found that the while the bitrate was low and visually TCP-friendly there were high amounts of packet loss, meaning that there were times when WSM was un-friendly.

For a higher bottleneck capacity, such as 725 Kbps, Figures 16-17 show behavior similar to the lower capacity case above. The bitrate during the post-buffering period increases linearly as the encoded bitrate increases until the bottleneck capacity is exceeded, after which the loss rate increases and eventually the content is thinned in response to the network congestion. When the link capacity is even larger, 1500 Kbps, shown in Figures 18-19, there are not any clips in the experiments that exceed the bottleneck capacity. Thus, the post-buffering rate increases linearly with the content encoding rate.

## 3.3 Multiple Bitrate Clips

The documentation related to Intelligent Streaming [1] and prior work [2] suggests that the responsiveness of streaming media may be coupled with the number of encoded bitrates contained in the content. The next set of experiments examine the effects of multiple encoded bitrates contained in the clip.

The objective was to thoroughly explore the relationship between the number of bitrates contained in the clip and responsiveness. Two sets of multiple bitrate clips were created. The first set started with a clip that contained the highest bitrate (1128 Kbps). Then the next highest bitrate[6] was added to create a clip with two bitrates (1128 and 764) Kbps. This process was continued 10 times to yield 10 clips such that each successive clip had the next highest bitrate added to the clip. Thus, the last clip contained all 10 bitrates (1128, 764, 548, 340, 282, 148, 108, 58, 43, and 28) Kbps. Another set of clips was created in the opposite manner where the first clip began with the lowest bitrate and each subsequent clip iteratively added the next lowest bitrate clip available.

Experiments were run with three bottleneck capacities: 250 Kbps, 725 Kbps, and 1500 Kbps. Only the buffering period results are shown for the set of clips with the decreasing lowest encoded rate, shown in Figures 21-26. In these experiments, WSM chooses a bitrate that is lower than the capacity if it is available, otherwise it chooses the lowest capacity available. This is shown by the decrease in loss rate as the clip contains increasingly lower bitrates. Similar results for the set of clips with increasing highest encoded bitrate were observed. See [8] for more details.

We examined the packet traces during these experiments and found that after the initial RTSP SETUP message was sent, a SET_PARAMETER message was sent with the type field specifying "high-entropy-packetpair." Then, a few hundred milliseconds later, three large (1500 byte), RTP packets were sent closely together from the server to the client. Finally, a few seconds later the actual video and audio streams were initialized with SETUP messages. We conclude that WSM is using two packet-pair estimates to determine available network capacity. We examined all of the packet traces and found that these packet-pairs are only sent just prior to streaming, and not during steady state playout, thus WSM uses RTCP reports during the actual session. Examination of the packet traces reveal that these RTCP reports are normally sent infrequently. However, during periods of packet loss the reports are frequent, sometimes more than ten per second. Perhaps this is the mechanism the WSM client uses to request retransmissions of lost packets.

## 3.4 Induced Loss

In order to explore WSM responsiveness to Internet congestion that is not self-induced, the NIST Net router was set to induce loss in the testbed. The bottleneck capacity was again set at 725 Kbps. The 548 Kbps clip was streamed

[6]The bitrates selected for these studies are the defaults available from the WSM encoding interface.

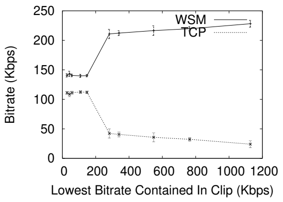


**Figure 21: Buffering Bitrate with 250 Kbps Capacity**

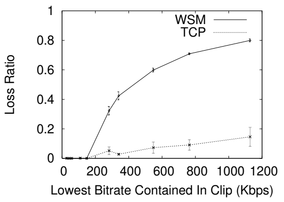


**Figure 22: Buffering Loss Ratio with 250 Kbps Capacity**

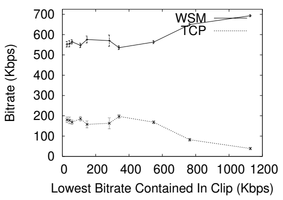


**Figure 23: Buffering Bitrate with 725 Kbps Capacity**

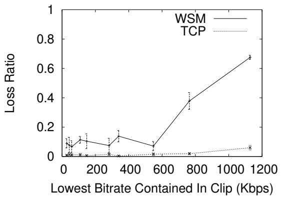


**Figure 24: Buffering Loss Ratio with 725 Kbps Capacity**

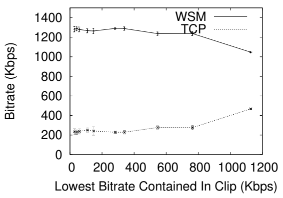


**Figure 25: Buffering Bitrate with 1500 Kbps Capacity**

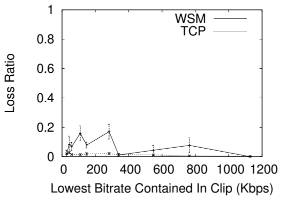


**Figure 26: Buffering Loss Ratio with 1500 Kbps Capacity**

because it is the highest single-encoded bitrate WSM will use in the presences of a 725 Kbps capacity without thinning the stream. The loss rate was constant for each experimental run and ranged from 0% to 20%. Figures 27-30 depict the results. The higher the loss rate, the more the TCP flow decreases its bitrate, while the WSM flow uses the additional available capacity during buffering and increases its sending rate to compensate for the high loss rate. The packet loss rates during buffering are high because they are the cumulative effect of WSM's self-inflicted packet loss in addition to the NIST Net imposed packet loss. During post-buffering, loss rates between 3% and 5% cause WSM to thin the stream in response to the higher loss rate. After 5%, additional packet losses do not effect the behavior of WSM.

The next experiment involved streaming a multiple bitrate clip containing the following bitrates: 548, 340, 282, 148, 106, 58, 43, and 28 Kbps. With these lower bitrates present, instead of thinning, WSM might choose one of these lower bitrates when the loss rate increases. Again, the loss rates range from 0% to 20% in Figures 31-34. The bitrate during the buffering stage is the same as in the prior experiment. However, during the post-buffering period WSM chooses a lower bitrate for loss rates greater then 5% instead of thinning the 540 Kbps rate. At 3% loss, WSM sometimes keeps streaming the higher bitrate, but other times chooses to send the lower rate.

For the final set of induced loss experiments the goal

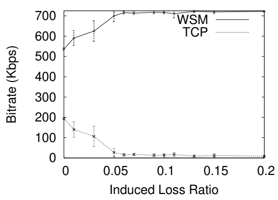


**Figure 27: Buffering Bitrate for SBR Clip with Induced Loss**

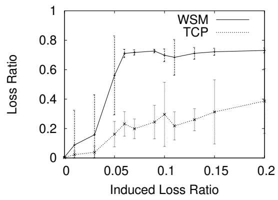


**Figure 28: Buffering Loss Ratio for SBR Clip with Induced Loss**

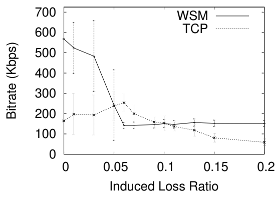


**Figure 29: Post-Buffering Bitrate for SBR Clip with Induced Loss**

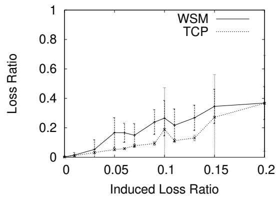


**Figure 30: Post-Buffering Loss Ratio for SBR Clip with Induced Loss**

was to examine the behavior of WSM when the loss rate suddenly changes during the post-buffering playout period. These experiments were similar to the previous streaming of 548 Kbps clip experiments except that instead of setting the loss rate at the beginning of the experiment, the NIST Net router was set to wait 15 seconds after the start of streaming before inducing loss. Figures 35-36 show that both during and after buffering, WSM does not react to the sudden increase in loss rate. Moreover, if loss is induced during buffering and for a short time into the playback of the clip and then stopped, Figures 37-38 show that WSM also does not respond.

Further experiments varying the induced latency over a considerable range were also conducted, but are omitted here due to lack of space. We refer the interested reader to [8] for these results. In general, WSM does not react to increased amounts of latency, except to achieve a higher bitrate when the TCP flow, having the same high latency, makes some capacity available.

# 4. CONCLUSIONS

Due to a prominent buffering period, WSM cannot be modeled as a simple CBR flow, as is common in many network simulations that include streaming media. In fact, looking at a simple average bitrate over time over the length of the entire clip may not reveal the true nature of WSM and may miss the buffering period where WSM can induce loss the network. An accurate bitrate distribution for WSM must include a buffering stage, whereby the sending data rate is 3-4 times the steady-state playout rate and a post-buffering stage whereby the actual bitrate is dependent on the encoding bitrate of the content and the network conditions. Moreover, an accurate model of WSM must include the bursty nature of packet transmission, especially during buffering for some encoded bitrates. The buffering behavior also leads us to conclude that there are periods when WSM can be TCP-unfriendly, while at some post-buffering

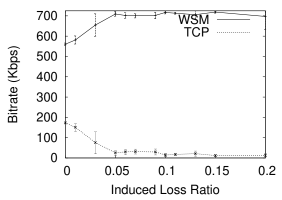


**Figure 31: Buffering Bitrate for MBR Clip with Induced Loss**

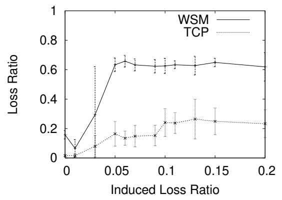


**Figure 32: Buffering Loss Ratio for MBR Clip with Induced Loss**

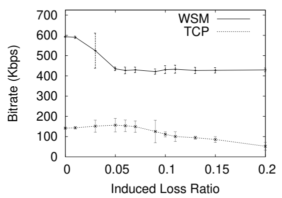


**Figure 33: Post-Buffering Bitrate for MBR Clip with Induced Loss**

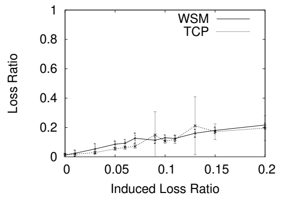


**Figure 34: Post-Buffering Loss Ratio for MBR Clip with Induced Loss**

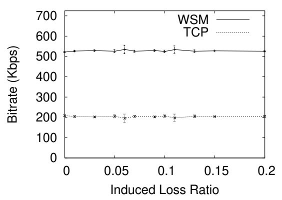


**Figure 35: Buffering Bitrate for SBR Clip with Induce Loss at 15 Seconds**

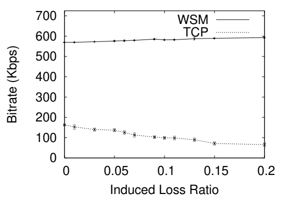


**Figure 36: Post-Buffering Bitrate for SBR Clip with Induce Loss at 15 Seconds**

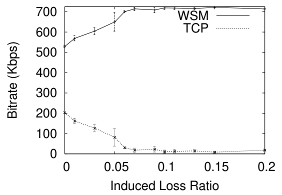


**Figure 37: Buffering Bitrate for SBR Clip with Induced Loss Stopped at 15 Seconds**

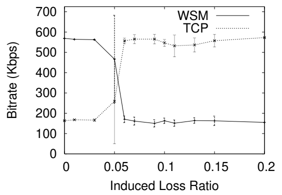


**Figure 38: Post-Buffering Bitrate for SBR Clip with Induced Loss Stopped at 15 Seconds**

times WSM is more than TCP-friendly, consuming less then the fair-share of capacity even when the encoded bitrate is higher than the capacity.

A main goal of this research was to examine the effects of the content encoding parameters on the responsiveness of WSM in order to understand what streaming server administrators and content producers themselves can do to influence the responsiveness of WSM. When streaming single bitrate clips,[7] WSM responds to congestion during buffering only when the encoding rate is less than the estimated capacity and will otherwise attempt to buffer at the encoding rate, thereby causes high loss rates. During playout, WSM responds to available capacity by thinning and discarding frames if necessary. Furthermore, if the encoded bitrate is less than capacity, WSM still responds to high loss rates (5%) as long as the loss is present at the start of streaming. However, encoding rates between 1/2 capacity and capacity will result in WSM taking more than a fair share of the available capacity if competing with TCP flows.

For multiple bitrate clips, WSM responds to capacity during buffering only when the content contains a suitable bitrate to choose. This chosen bitrate is the largest that capacity allows, which may mean WSM is unfair to competing TCP flows. If there is no encoded bitrate under capacity, WSM buffers at the smallest encoding bitrate available, again inducing high amounts of packet loss. During playout, WSM is responsive to available capacity, either because it chose the proper encoding rate, or because it thins if the proper rate is not encoded in the clip. Again, if there was a bitrate included in the clip that was less than the capacity, WSM chooses that rate, which may be unfair to competing TCP flows. Our results show that content producers can help WSM be more responsive to congestion by encoding several bitrates into their content. However, multiple encoded bitrates are not a panacea for making WSM TCP-friendly and ensuring fairness. But during the buffering period in particular, multiple bitrates do help WSM behave in a more network-friendly fashion.

[7]Based on [7], single bitrate clips are by far the most common.

# 5. REFERENCES


[1] B. Birney. Intelligent Streaming, Microsoft Corporation, Oct. 2003. Online at: http://msdn.microsoft.com/library/-en-us/dnwmt/html/intelligent_streaming__qwes.asp.

[2] J. Chung, M. Claypool, and Y. Zhu. Measurement of the Congestion Responsiveness of RealPlayer Streaming Video Over UDP. In *Packet Video Workshop (PV)*, Apr. 2003.

[3] J. V. der Merwe, S. Sen, and C. Kalmanek. Streaming Video Traffic: Characterization and Network Impact. In *Workshop on Web Content Caching and Distribution*, Boulder, CO, USA, Aug. 2002.

[4] S. Floyd and K. Fall. Promoting the Use of End-to-End Congestion Control in the Internet. *IEEE/ACM Transactions on Networking*, Feb. 1999.

[5] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley. Inferring TCP Connection Characteristics Through Passive Measurements. In *Proceedings of IEEE Infocom*, Mar. 2004.

[6] M. Li, M. Claypool, and R. Kinicki. MediaPlayer versus RealPlayer – A Comparison of Network Turbulence. In *ACM Internet Measurement Workshop (IMW)*, Nov. 2002.

[7] M. Li, M. Claypool, R. Kinicki, and J. Nichols. Characteristics of Streaming Media Stored on the Web. Technical Report TR-03-18, CS at Worcester Polythechnic Institute, May 2003.

[8] J. Nichols, M. Claypool, R. Kinicki, and M. Li. Measurements of the Congestion Responsiveness of Windows Streaming Media. Technical Report TR-04-07, CS at Worcester Polythechnic Institute, Mar. 2004.

[9] R. Prasad, M. Murray, C. Dovrolis, and K. Claffy. Bandwidth Estimation: Metrics, Measurement Techniques, and Tools. *IEEE Network*, November-December 2003.

[10] E. Veloso, V. Almeida, W. Meira, A. Bestavros, and S. Jin. A Hierarchical Characterization of a Live Streaming Media Workload. In *ACM Internet Measurement Workshop (IMW)*, Nov. 2002.