NRC·CNRC

NRC Publications Archive
Archives des publications du CNRC

Quality of Service (QoS) for Video Transmission

Park, Shihyon; DeDourek, John

This publication could be one of several versions: author’s original, accepted manuscript or the publisher’s version. / La version de cette publication peut être l’une des suivantes : la version prépublication de l’auteur, la version acceptée du manuscrit ou la version de l’éditeur.

Publisher’s version / Version de l'éditeur:

Proceedings of the First International Conference on Ubiquitous and Future Networks (ICUFN 2009), 2009-06-09

NRC Publications Record / Notice d'Archives des publications de CNRC:
https://nrc-publications.canada.ca/eng/view/object/?id=2eeb86d1-e041-4fc0-b038-5b0dcbb12a6d
https://publications-cnrc.canada.ca/fra/voir/objet/?id=2eeb86d1-e041-4fc0-b038-5b0dcbb12a6d

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at
https://nrc-publications.canada.ca/eng/copyright
READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L’accès à ce site Web et l’utilisation de son contenu sont assujettis aux conditions présentées dans le site
https://publications-cnrc.canada.ca/fra/droits
LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D’UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n’arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

Quality of Service (QoS) for Video Transmission

Shihyon Park
National Research Council, Canada
Institute for Information Technology
46 Dineen Drive Fredericton, Canada
Shihyon.Park@nrc-cnrc.gc.ca

John DeDourek
University of New Brunswick
Faculty of Computer Science
540 Windsor Street Fredericton, Canada
dedourek@unb.ca

***Abstract*—This paper evaluates QoS techniques for transmission of real-time video that use a DiffServ mechanism that includes an efficient bandwidth agent to allocate bandwidth on heterogeneous networks, a Hierarchical Token Bucket (HTB) queuing scheme at the output interface of Linux-based routers, and a policing mechanism at the incoming interface of the edge router (ER) in a Linux-based testbed. The testbed consists of several different networks including IEEE 802.11b to find the characteristics of real-time video traffic using MPEG streaming video and VLC for server/client so that we will have a good idea what bandwidth and burst size is required to stream MPEG video through the QoS Diffserv domain.**

We set an efficient testbed to investigate how real-time streams behave in the QoS scheme, and to provide a realistic recommendation to manage the available bandwidth for both the QoS provider and clients.

***Index Terms*—Hierarchical Token Bucket (HTB), Policing, Shaping, DiffServ, Burst, Variable Bit Rate (VBR), MPEG.**

I. Introduction

Over the last few years, real-time Internet traffic such as audio and video streams have been growing in popularity. The number of Internet users is increasing rapidly and they request diverse services on the Internet. There are several reasons why real-time Internet traffic is increasing. First, wireless networking technology has been developed such as cellular phones, and PDAs. Second, the communication environment has changed in terms of wireless local area networking. The communication environment helps people to access to the Internet in public areas such as coffee shops, airports, and other public buildings. Third, multimedia applications are embedded in wireless environments; moreover, MPEG-1, MPEG-2, MPEG-4, H.263, and other digital multimedia content is standardized [2][3].

This environment enables applications such as video telephony, video conferencing, on-line lectures, on-line games, and Video on Demand (VoD). These real-time video streaming services have become important on the Internet, and have also been gradually expanding the service field. However, because of the Internet users, the Internet is inevitably growing, and Internet traffic reduces Internet services dramatically because more traffic cause traffic congestion at the router. It affects the real-time streaming video services which must provide Quality of Service (QoS). However, QoS is not guaranteed, because the Internet currently is designed for best effort delivery by default. Traditionally, different types of Internet traffic are not distinguished. This means that when congestion occurs all traffic is going to suffer the same impairment, i.e., increasing delay, more variable delay, and packet drops. Moreover, there are some critical issues that need to be dealt with for real-time video/audio transmission over a wireless network. The compressed video may generate burst traffic, and make it difficult for service providers to allocate bandwidth for videos. In order to solve these problems, we first need to have QoS mechanisms, and resource management to support QoS for real-time streaming video applications.

II. Differentiated Service (DiffServ) Domain Architecture

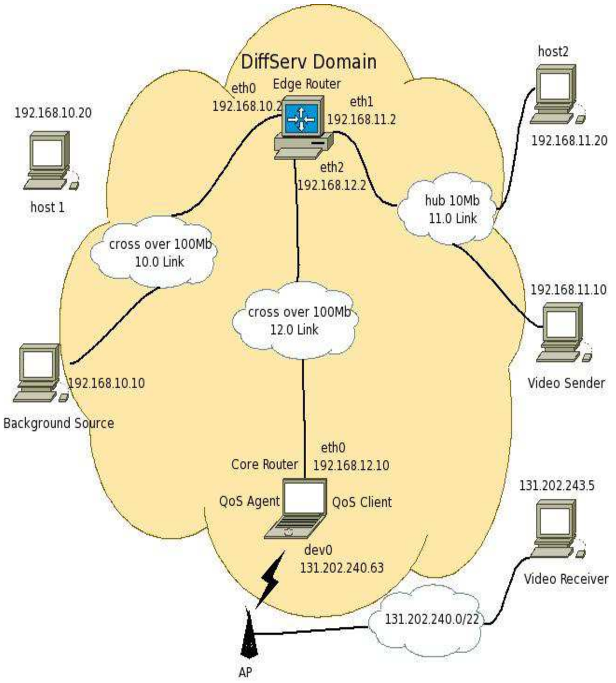

Fig. 1. DiffServ Testbed Architecture

This research involved the evaluation of a QoS technique for transmission of real time streaming video. The testbed shown in Figure 1 is used to investigate how real-time video/audio streams behave in a QoS domain and to provide recommendations on the methods to manage bandwidth. DiffServ (Differentiated Service) technology is used as the QoS mechanism. The DiffServ domain includes a bandwidth broker (BB) to manage bandwidth in heterogeneous networks. The Hierarchical Token Bucket (HTB) queuing scheme schedules packet transmission at the Linux based routers. A policing mechanism is also deployed at the Linux based routers. The policer is a mechanism by which a particular flow of packets can be limited to the rate specified by the BB. It accepts traffic if the traffic rate is below the specified rate but drops packets above that rate.

A QoS domain was designed by TungaBaskaraRao [6]. He mainly focused on investigating an architecture, designing a testbed for establishing QoS, and implementing a QoS agent and QoS client. The QoS architecture includes a signaling protocol between a client and QoS agent. When the QoS agent is executed, it configures the HTB at the egress interface of the routers. This way, the QoS model is scalable and allows aggregation of flows. The client sends a request to the QoS agent in the form of an XML message. The message contains a 6-tuple: a source IP address, a destination IP address, a source port number, a destination port number, the protocol, and the amount of bandwith. When the client sends a request to the QoS agent, the QoS agent processes the request based on the EF bandwith available in the QoS domain. In this way, the QoS agent controls admission. The QoS agent is the Bandwidth Broker (BB) in the DiffServ domain. The QoS domain includes a policer at the ingress interface of the edge router to control traffic and an HTB at the egress interface of each router to manage bandwidth. If there is enough EF DiffServ bandwidth available for the client's request, then the BB configures the policer to monitor packets and marks a packet DSCP-EF in the IP header when the packet matches the first five elements of the 6-tuple. However, if enough EF DiffServ bandwidth is not available for the client's request, then the QoS agent replies back to the client "Bandwith is not available", and "Reservation Rejection". The QoS client maintains the reservation by repeating the reservation request every 15 seconds until the reservation expires. If the reservation expires, the QoS agent deletes the reservation and disables the policer at the ingress interface of the router and removes the packet marker from the router. All of these actions take place by establishing a secured connection between the QoS agent and the routers in the DiffServ domain using the secure shell utility (SSH). SSH is used for establishing secured connections. The QoS agent establishes a connection with the routers through SSH. The QoS agent and the QoS client in the DiffServ domain are implemented using the Java language.

The testbed shown in Figure 1 contains four different links: the 192.168.10.0/24 link, the 192.168.11.0/24 link, the 192.168.12.0/24, and the 131.202.240.0/22 link. This testbed contains a wired network using a 100Mbps crossover cable, 10Mbit hub, and IEEE 802.11b wireless network.

A. Traffic Control in Linux

The testbed used in this work makes use of the Linux traffic control [8] features. Traffic control affects the entire packet queuing subsystem in a network or network device. Linux queuing disciplines include a packet scheduling scheme called Hierarchical Token Bucket (HTB). Any machine's outgoing interface is (First-In-First-Out) FIFO by default. FIFO treats all traffic on an interface equally by the first-in-first-out rule. However, the HTB has multiple classes in the queuing discipline (qdisc) so that the traffic flow may be differentiated in the qdisc. A difference between FIFO and the HTB is that one has a classless qdisc and the other has classful qdisc respectively:

- Classless qdiscs have no subdivisions within them. example include FIFO, Token Bucket Filter (TBF), Stochastic Fairness Queuing (SFQ), and Random Early Detect (RED).
- Classful qdiscs have multiple classes to categorize traffic. In order to classify the packets, the qdisc requires a filter. Packets are classified, then sent to the proper classes within the qdisc. Classes also can have internal qdiscs.

B. Linux Kernel Components

The QoS scheme used in this work utilizes a packet marker to categorize packets, a policer to limit traffic, and differentiated packet scheduling at the router egress interface. In this research, HTB is used to manage the bandwidth. Using HTB, the traffic flows in the qdisc can be differentiated and treated differently from each other. A Queuing Discipline is also used on the incoming (ingress) interface to limit flows to their allocated bandwidth. Policer is configured with a specified maximum rate and burst to regulate packets. The purpose of this research is to investigate the characteristics of QoS mechanisms when real-time video is steamed through them. Both the ingress queuing discipline, known as a policer, for incoming data, and the egress queuing discipline, known as shaper, for outgoing data, are used to provide QoS. The testbed consists of two routers: ER and CR. Figure 2 shows the Linux Kernel on the ER. Figure 3 shows the Linux Kernel on the CR at which there is no policer.

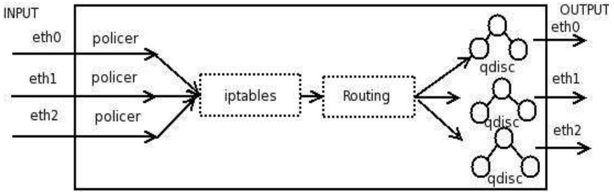

Fig. 2. IP packets in the Linux Kernel on the ER

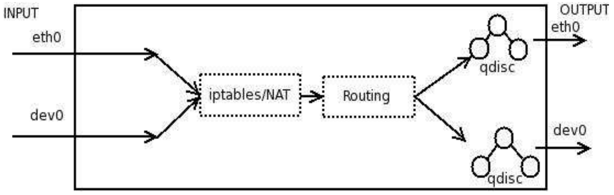

Fig. 3. IP packets in the Linux Kernel on the CR

Both the policer and shaper are basic elements of traffic control for limiting the bandwidth available to a particular flow or application or to reserve bandwidth for a particular flow or application:

The function of ER is to accept and filter the incoming packets' mainly characterizing, policing, and marking IP traffic. The core router (CR), shown in Figure 3, in the QoS domain does not perform any QoS control for admission control or any management functions, but carries on packet scheduling and forwarding. This is known as PHB (Per Hap Behavior) using a code-point in the IP header to select a PHB as a specific forwarding. EF (Expedited Forwarding) PHB is used for real-time video traffic in this research. Real-time streaming video is marked as EF. EF PHB provides low loss, low jitter, and low latency.

C. Queuing Discipline

The HTB queuing discipline as a classful qdisc was used to shape outgoing data. It is used to manage the bandwidth as a QoS solution. In order to understand how the HTB performs for QoS, token and bucket are explained.

1) Hierarchical Token Bucket (HTB): An HTB manages the outbound bandwidth on a given link. Dividing up a physical link and creating classes for each traffic type is a way to manage the bandwidth [8]. An advantage of using the HTB queuing discipline in this case, in Figure 4, is that even if the EF class is set to a specific bandwidth, but its bandwidth is not used up, then the BE class can share the excess bandwidth with the EF class. Because the EF class is assigned a maximum bandwidth that is less than the link bandwidth, some bandwidth is always available for BE traffic.

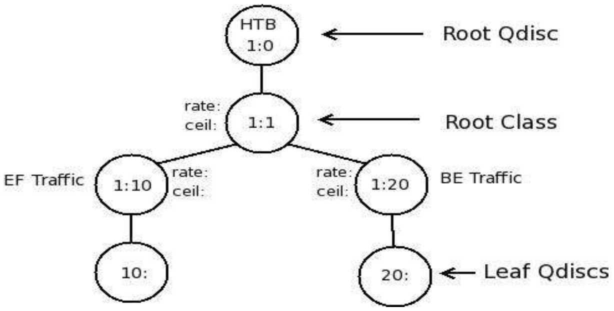

Fig. 4. The qdisc-class hierarchy of the configuration used for the research

In Figure 4, the qdisc-class hierarchy for this research is shown. In Figure 4, both the qdiscs and classes are identified by handles which have major and minor numbers. It is in the form of x:y. x represents the major number and y represents the minor number. The major number represents the parent qdisc of the class and the minor number represents the class within a qdisc. In Figure 4, the root qdisc is always numbered 1:0. However, it is not necessary to add '0' to represent the minor number for a qdisc handle [7]. The root qdisc represents the primary egress queuing discipline on any device. The root qdisc usually contains one class. The single HTB class has two parameters which are a rate and a ceil:

- Rate: The generated bandwidth available for a given class which represents minimum bandwidth,
- Ceil: The maximum bandwidth that a class is allowed to consume.

The root qdisc can have number of classes below it. This means that each of the classes can be allocated some of the available bandwidth from the parent class. These classes are called children classes, and the rate and ceil parameters do not need to be the same as the parents, so that we can reserve some bandwidth for a particular class. To configure this, we first need to know what throughput each link has. To measure the throughput for each link, Iperf was used.

TABLE I
LINK SPECIFICATION AT THE EDGE ROUTER

Link	Max BW(Mbps)	EF	BE
192.168.10.0/24	94.5	44.5	50
192.168.11.0/24	9.5	4.5	5
192.168.12.0/24	94.5	44.5	50

TABLE II
LINK SPECIFICATION AT THE CORE ROUTER

Link	Max BW(Mbps)	EF	BE
192.168.12.0/24	94.5	44.5	50
131.202.240.0/22	5.5	3.5	2

With the maximum throughput for each link shown in Tables 1 and 2, we need to allocate bandwidth for the EF class and BE class from the total throughput.

III. EXPERIMENTS

A. Testing the DiffServ (QoS) domain

The main issue is how compressed video can be transported effectively in heterogeneous networks. Therefore, when video and audio packets are sent over networks that do not have sufficient bandwidth and distribute insufficient bandwidth among packets, the video and audio packets will arrive late, out of order, or not at all. This causes delay, jitter, and packet loss and provides bad performance, for instance blank pieces within the video window, disruptions in audio, jittery picture, and/or talking over one another because of delay [4]. In order to improve quality it is essential to understand the characteristics of video traffic. The characteristics of video traffic are affected by the video source, the content of the video, and the coding algorithm [5].

The policer is a component of the DiffServ domain; the policer regulates the packets depending on the rate parameter. The following tests investigate a real-time streaming video transmission to find the actual required policing rate for a given clip. This experiment was conducted by sending a video stream called clip2.mpg. We first need an estimated average

Clip2.mpeg
MPEG2 compression video, 320×240, 29.970 fps,
MPEG layer-2, duration 92s.
File size = 8043364 bytes.

Fig. 5. clip2.mpg specification

TABLE III
TESTING RESULTS FROM WIRESHARK

Test No	Rate Kbit	Burst KB	Packet Count					
			Sender Out	ER In	ER Out	CR In	CR Out	Receiver In
1	760	40	6458	6458	6458	6458	6458	6458
2	760	30	6458	6458	6456	6456	6456	6456
3	760	5	6458	6458	6422	6422	6422	6422
4	760	33	6458	6458	6458	6458	6458	6458
5	760	31	6458	6458	6457	6457	6457	6456
6	760	32	6458	6458	6457	6457	6457	6456
7	700	32	6458	6458	5959	5959	5959	5959
8	700	90	6458	6458	6002	6002	6002	6002
9	700	150	6458	6458	6048	6048	6048	6048
10	700	200	6458	6458	6086	6086	6086	6086
11	700	400	6458	6458	6237	6237	6237	6237
12	1000	20	6458	6458	6457	6457	6457	6456
13	1000	15	6458	6458	5953	5953	5953	5953
14	1500	15	6458	6458	5954	5954	5954	5954

rate for clip2.mpg. The formula to estimate average rate is given in Figure 6.

Formula to get estimated average rate (bps)
= file size/duration.
(8043364×8)b/92s = 699422bps
699422bps / 1000 = 699kbits/sec
Estimated required average rate for clip2.mpeg
is 699kbits/sec

Fig. 6. Estimated required average rate for clip2.mpg

For the test the video sender is located at 192.168.11.10, and the video receiver is located at 131.202.243.5 in Figure 1. The EF traffic is captured at each interface of the video sender, the video receiver, the ER, and the CR . We also measure whether each component (the video sender, the video receiver, the ER, and the CR) in the QoS domain provides QoS for the EF traffic. Wireshark is configured to capture packets at each interface: the egress interface of the video sender, the ingress interface of the ER, the egress interface of the ER, the ingress interface of the CR, the egress interface of the CR, and the ingress interface of video receiver.

The number of packets at the video receiver, as shown in Table 3, is slightly below the number of packets at the CR because the video receiver does not belong to the QoS domain. This means that packets might be dropped when there is congestion in the 131.202.240.0/22 network. In Table 3, the number of packets are the same at both the egress interface of the ER and the ingress interface of CR. When Wireshark captures packets at the ingress interface of the ER, packets have not been processed by the policer yet. Therefore, the number of packets is same as the number of packets at the video sender egress interface. However, when packets are captured by Wireshark at the egress interface of ER, the result is different from the result from at the ingress interface, because when packets pass through the egress interface, they already have been processed by the policer and some packets have been dropped in those cases where the policer has been configured with too small a bandwidth for the flow.

IV. TESTING AND ANALYSIS

This section explains how much rate and burst are required for a given MPEG2 video through the QoS domain. Moreover, we need to investigate how the burst configuration affects the required bandwidth configuration. How do we know how much bandwidth a given MPEG video requires? Should a fixed bandwidth be assigned according to the estimated value, or should different amount of bandwidth be set than the estimated average rate depending on the type of video such as action movie, interview, concert, or sports? How much bandwidth would be sufficient for a given MPEG2 video? Is there any tradeoff between bandwidth and burst size? These issues will be discussed in this section. MPEG2 videos are considered by content. There are three types of MPEG2 video considered here. The interview clip has a person talking on the screen, the entertainment clip has lots of action and background changes, and the soccer clip has lots of movement and mainly focuses on players moving on the field.

A. Interview Clip

Interview.mpg has the following characteristics:

- MPEG 1/2 Video decoder,
- Resolution = 320 × 240,
- Frame rate = 25.000 fps,
- File size = 3.92 MB,
- Duration = 52.2s,
- Estimated average rate = 626 kbits/sec.

Interview.mpg has no background changes. It is a talking head video. This clip is selected because the characteristics of interview.mpg are similar to a video conference or online lecture.

1) Results and Analysis: Interview.mpg requires an estimated average rate of 626 Kbits/sec. Packet loss is measured at the policer. The policer was configured to a rate of 630 Kbits/sec and was tested with 10 KB, 15 KB, 20 KB, 25 KB, and 30 KB of burst. The policer rate was increased by 10 Kbits/sec and the process was repeated. This test was conducted three times: to investigate if the results are

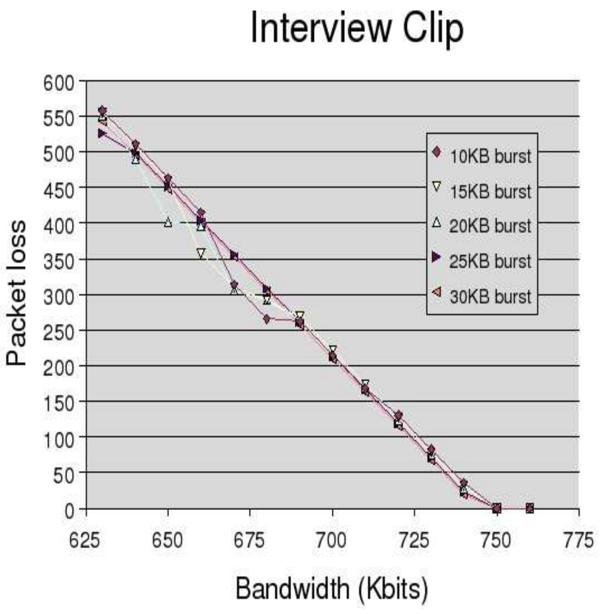

Fig. 7. decreasing packet loss as bandwidth increases

replicable, to find out how rate and burst affect packet loss, and to create a formula to calculate the rate and burst required to provide video quality for a clip with the characteristics of interview.mpg.

Most interesting result from each test is that burst size does not reduce packet loss. A real-time streaming video like interview.mpg requires a minimum amount of bandwidth rather than a minimum size of burst. When the policer rate hits 750 Kbits/sec, then packet loss is zero at any burst size. The content and the characteristics of interview.mpg explain the results. Interview.mpg does not include background changes or much action.

From these results, we can conclude that we do not need to specify a large burst, but rather need to specify a minimum amount of bandwidth. In this case, estimated average rate = 626 Kbits/sec and the actual average rate = 750 Kbits/sec with any burst. We can say that 120 Kbits/sec (750-630) needs to be added to the estimated average rate.

B. Card Tricks Clip

Card.mpg has the following characteristics:

- MPEG 1/2 Video decoder,
- Resolution = 320 × 240,
- Frame rate = 29.970 fps,
- File size = 10.8 MB,
- Duration = 62.6s,
- Estimated average rate = 1,448 Kbits/sec.

Card.mpg has lots of background changes during playing time. Card.mpg shows several card tricks in front of dynamic backgrounds. This clip is selected because the characteristics of card.mpg are similar to characteristics of entertainment videos. Entertainment videos like music videos, movies, and concerts have lots of background changes and lots of action involved.

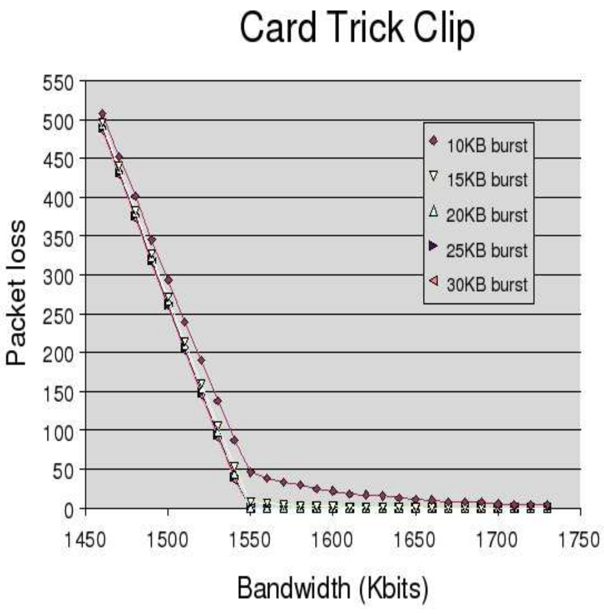

Fig. 8. decreasing packet loss with increased rate and burst

1) Results and Analysis: Packet loss in Figure 8 for card.mpg is affected by burst size. Packet drops occur with 10 KB burst even when the policer rate is 1,730 Kbits/sec. At 20 KB or more burst, zero packet drops occur when the policer rate is 1,550 Kbits/sec. This result explains that burst can help to reduce packet loss. This means that the VLC [1] application generates burst traffic when it transmits card.mpg. Therefore, as the test results show, if we do not consider the burst, then card.mpg requires allocating a large amount of bandwidth to the flow.

We observed that 10 KB burst does not eliminate packet loss. The required policer rate can be calculated as follows: Estimated average rate = 1,448 Kbits/sec, and the actual average rate = 1,550 Kbits/sec with 20 KB burst. We can say that 100 Kbits/sec (1,550 Kbits/sec - 1,450 Kbits/sec) needs to be added to the estimated required average rate.

C. Soccer Clip

Soccer.mpg has the following characteristics:

- MPEG 1/2 Video decoder,
- Resolution = 320 × 240,
- Frame rate = 30.000 fps,
- File size = 9.48 MB,
- Duration = 60.5s,
- Estimated average rate = 1,315 Kbits/sec.

Soccer.mpg has few background changes. Soccer.mpg shows players running and the camera angle often changes to close-ups of the players. Many people have started to watch sports on their personal computers. Therefore, this clip is a good choice to experiment with to measure rate and burst.

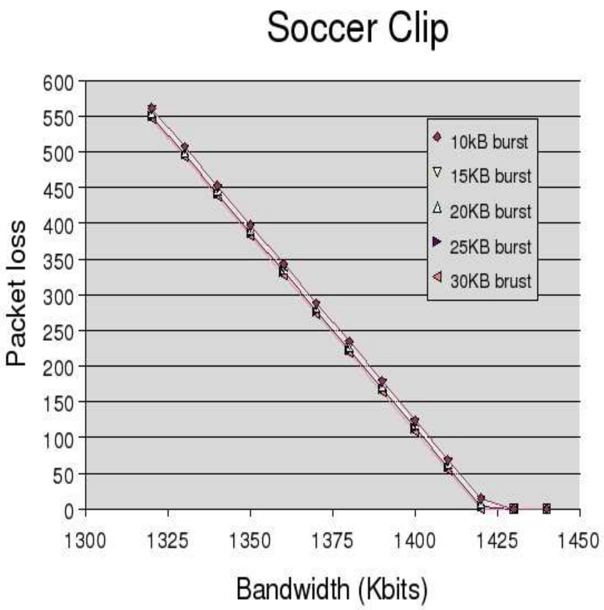

Fig. 9. decreasing packet loss with increased bandwidth

1) Results and Analysis: All the results from testing with soccer.mpg show that bandwidth is the most important parameter to consider when transmitting real-time streaming video like soccer.mpg. Our first expectation was that soccer.mpg has burst traffic so we may have needed to consider burst size. However, it turned out that in order to reach zero packet loss, a minimum bandwidth is required. The required policer rate can be calculated as follows: estimated average rate = 1,315 Kbits/sec, and the actual required rate = 1,430 Kbits/sec with any size of burst. We can say that 115 Kbits/sec (1,430 Kbits/sec - 1,315 Kbits/sec) needs to be added to the estimated average rate.

V. Future Work

Although this research investigated the characteristic of real-time MPEG2 compressed video through the QoS domain, a number of related issues need further research. This research can be extended in the following fields:

- QoS performance with different compression schemes, and with different video applications,
- signaling for multi-QoS domains,
- IPv4-IPv6 QoS performance.

VI. Acknowledgements

This CNSR research was made possible thanks to support from Bell Canada through its Bell University Laboratories R & D program, and from ACOA through an AIF grant.

References

[1] VLC media player http://www.videolan.org/vlc/, last accessed on May 20th.

[2] Cote, G., Wenger, S., & Kosentini, F., Standard-based system for robust video transmission over the internet, Canadian Conference on Electrical and Computer Engineering Vol 2, 1999, pp.739–744.

[3] Nahrstedt, K., Yuan, W., Shah, S., Xue, Y., & Chen, K., QoS Support in Multimedia Wireless Environments, http://eecs.vanderbilt.edu/people/yuanxue/publication-files/bookpre-nahrstedt-qos.pdf.

[4] Team Holt, Video conferencing February 1998 http://www2.cs.uh.edu/ lji/video.html.

[5] Nayak, C., & Ilyas, M., Characterization of video conferencing traffic in ATM networks, Conference Proceedings - IEEE SOUTHEASTCON, 1996, pp. 344–347.

[6] TungaBaskaraRao, V., End-to-End QoS (Quality of Service) Signaling for Real Time Data Streams, MCS Thesis Faculty of Computer Science, University of New Brunswick, 2007.

[7] Grewal, J., Design of testbed for evaluation of QoS (Quality of Service) on wireless networks, MCS Thesis Faculty of Computer Science, University of New Brunswick, April 2005.

[8] Brown, M., Traffic Control HOWTO, Oct 2006 http://www.tldp.org/HOWTO/TrafficControlHOWTO/intro.html.