Service Differentiation in the Internet to Support Multimedia Traffic

Fouad Tobagi, Waël Noureddine, Benjamin Chen, Athina Markopoulou, Chuck Fraleigh, Mansour Karam, Jose-Miguel Pulido, and Jun-ichi Kimura

Department of Electrical Engineering
Stanford University, Stanford CA 94305
Contact author: tobagi@stanford.edu

Abstract. The current best-effort infrastructure in the Internet lacks key characteristics in terms of delay, jitter, and loss, which are required for multimedia applications (voice, video, and data). Recently, significant progress has been made toward specifying the service differentiation to be provided in the Internet for supporting multimedia applications. In this paper, we identify the main traffic types, discuss their characteristics and requirements, and give recommendations on the treatment of the different types in network queues. Simulation and measurement results are used to assess the benefits of service differentiation on the performance of applications.

1 Introduction

The Internet is seeing the gradual deployment of new multimedia applications, such as voice over IP, video conferencing, and video-on-demand. These applications generate traffic with characteristics that differ significantly from traffic generated by data applications, and they have more stringent delay and loss requirements. Voice quality, for example, is highly sensitive to loss, jitter, and queueing delay in network node (i.e., switch or router) buffers, and the quality of video traffic is significantly degraded during network congestion episodes. The current best-effort infrastructure of the Internet is ill-suited to the quality of service requirements of these applications.

In addition to emerging streaming applications, the Internet must also support interactive data applications. Good user-perceived performance of these applications, such as telnet, video gaming, and web browsing, requires short response times and predictability. However, these requirements are often not met, due to the interaction of TCP with packet loss during network congestion periods.

In this paper, we identify the main traffic types, discuss their characteristics and requirements (Sec. 2), and examine the degree of separation of traffic types necessary to provide adequate user-perceived performance (Sec. 3). In addition, we describe a conceptual model for a network node port that provides service differentiation and discuss the different required functionalities (Sec. 4). Within this model, we demonstrate that the provision of multiple packet drop priorities

S. Palazzo (Ed.): IWDC 2001, LNCS 2170, pp. 381–400, 2001.

within a queue, in association with appropriate packet marking, can further enhance performance (Sec. 5). Throughout, simulation and measurement results support our proposals.

2 Multimedia Traffic Characteristics and Requirements

In this section we present the characteristics and requirements of voice, video, and interactive data applications. For each application, we discuss its characteristics in terms of data rate and variability, and we describe its requirements in terms of delay, jitter, and packet loss. These have to be well understood in order to determine the appropriate treatment to give each application in the network.

2.1 Voice

Voice connections generate a stream of small packets of similar size (a few tens of bytes) at relatively low bit rates. Typical stream rates range from 5 Kbps to 64 Kbps, depending on the encoding scheme, to which header overhead adds a few tens of Kbps. Therefore, voice stream rates remain on the order of tens of Kbps, regardless of the encoding scheme. For example, G.711, a simple pulse code modulation encoder, generates evenly spaced 8-bit samples of the voice signal at 125 msec intervals, resulting in a 64 Kbps stream. It is possible to reduce the rate through voice compression schemes and silence suppression, at the expense of increased variability. The suppression of samples corresponding to silence periods (which account for 45% of total time in typical conversations [6]), leads to substantial average rate reduction. For example, G.729A generates an 8 Kbps stream, while G.723 generates a 5.33 Kbps stream.

For the Internet to provide toll quality voice service, packet delay and loss must meet stringent requirements. Interactivity imposes a maximum round trip time of 200-300 msec. That is, the one-way delay incurred in voice encoding, packetization, network transit time, de-jittering, and decoding must be kept below 100-150 msec. Jitter must be limited (e.g., less than 50 msec) to ensure smooth playback at the receiver. Subjective tests have shown that periods of lost speech (clips) larger than 60 msec affect the intelligibility of the received speech [9]. Since packet loss in the Internet is bursty [2,22], the probability that consecutive voice packets are lost, resulting in long clips, is significant. Therefore, loss rates have to be kept at very low levels unless packet loss concealment is used.

2.2 Video

Video traffic is stream-oriented and spans a wide range of data rates, from tens of Kbps to tens of Mbps. The characteristics of encoded video (data rates and variability in time) vary tremendously according to the content, the video compression scheme, and the video encoding scheme.

In terms of content, more complex scenes and more frequent scene changes require more data to maintain a certain level of quality. For example, video streams of talking heads are lower-rate and less variable than those of motion pictures and commercials.

Different video compression schemes, such as H.261, H.263, MPEG-1, and MPEG-2, are designed to meet different objectives and therefore have different bit rates and stream characteristics. For instance, the applications of H.261 include video conferencing; consequently, the rates are multiples of 64 Kbps, up to 2 Mbps, and the coding is designed to achieve a fairly uniform bit rate across frames. On the other hand, prerecorded movies using MPEG-2 may have several times the picture resolution and typically require several Mbps.

The video characteristics are also affected by the video encoding control scheme used. For a given content and a given compression scheme, constant bit rate (CBR) video maintains a streaming rate that varies little over time. By contrast, variable bit rate (VBR) video traffic has been shown to be self-similar and may have a peak rate which is many times the average. Typically, VBR aims to achieve a more consistent quality for the same average bandwidth, and it is more commonly employed in practice.

The latency requirements of video depend on the application. Like voice, interactive video communication requires low delay (200-300 msec round-trip); however, one-way broadcast and video-on-demand may tolerate several seconds of delay. As is the case for voice, a packet delayed beyond the time when it needs to be decoded and displayed is considered lost. Furthermore, it has been observed that packet loss rates as low as 3% can affect up to 30% of the frames, due to dependencies in the encoded video bit stream [4]. Therefore, the packet loss rate and delay in the network should be kept small [7].

2.3 Interactive Data Applications

Data applications still account for the large majority of Internet traffic [25] and have significantly different characteristics and requirements. We focus here on interactive data applications, namely telnet (remote login) and the web.

Typical telnet sessions consist of characters being typed by a user at a terminal, transmitted over the network to another machine (server), which echoes them back to the user's terminal. The packet stream being generated consists of small datagrams (typically less than 50 bytes). Occasionally, the results of commands typed by the user are sent back by the server. This results in asymmetric traffic, with server to user terminal traffic on average 20 times the user to server traffic [21]. Packet interarrival times have been found to follow a heavy-tailed (Pareto) distribution, resulting in somewhat bursty traffic [23]. However, the inter-packet time is normally limited by the typing speed of humans, which is rarely faster than 5 characters per second [13], giving a minimum 200 msec average inter-packet time. Consequently, the traffic generated by telnet is of relatively low bandwidth and low burstiness.

Telnet is a highly interactive application and, similarly to voice over IP, has strict delay requirements on individual packets. Echo delays start to be notice-

able when they exceed 100 msec, and in general, a delay of 200 msec is the limit beyond which the user-perceived quality of the interactivity suffers [13]. Furthermore, telnet traffic is highly sensitive to packet loss, since the retransmit timeout required for recovery has a minimum value that exceeds the maximum acceptable echo delay. Therefore, telnet packet loss needs to be kept at a minimum for best user-perceived performance.

Web traffic, carried by HTTP over TCP, is closely tied to the contents of web pages and to the dynamics of TCP. Trace studies of web traffic have shown that the majority of HTTP requests for web pages are smaller than 500 bytes. HTTP responses are typically smaller than 50 KB, but may also be very large when HTTP is used to download large files off web pages [18]. Indeed, HTTP responses have been found to follow a heavy tailed distribution, corresponding to that of web files in the Internet. Moreover, the aggregate traffic generated by many users of the WWW has been shown to exhibit self-similarity [5,17].

In general, short page download times (less than 5 seconds) are required for good user-perceived performance. In addition, users highly value the predictability of web response times. In other words, not only does the average download time need to be small, but so does the variance of download times. In this context, it is important to distinguish between the interactive use of HTTP, i.e. for downloading actual web pages (html file and images) which tend to be short transfers (and therefore have low rate), and the non-interactive FTP-like use, where HTTP is employed to download large files.

3 Mixing vs. Separating Traffic Types

As discussed above, voice, video, and data applications differ significantly in their traffic characteristics and requirements. Naturally, we are interested in understanding how we can support the various traffic types in a single network, such that the user-perceived performance is maximized. It appears reasonable that identifying different classes of traffic in order to separate them at the queues in the network and to treat them appropriately would achieve this goal. The questions that arise are: Where in the network would differential handling be necessary, if at all? Which types need to be separated, and which types can be safely mixed? What is the appropriate treatment of each class in its queue? In this section we attempt to answer the first two questions, and we address the third in Sec. 4 and 5.

3.1 Mixing Voice and Data

In [15], we investigate the effect of mixing and separating traffic types, i.e., which types can be mixed together in the same queue without incurring a significant loss in throughput, and which types need to be separated to meet performance objectives. We first consider the impact of data traffic on voice traffic by mixing 1 Mbps of voice traffic (11 streams) with TCP data traffic. We determine the maximum number of data sources that can be mixed with voice traffic if voice

packets are to satisfy a 10 ms delay budget allocated to the section of the path being studied. We consider paths composed of either T1, 10Base-T, or 100Base-T links. Simulations reveal that mixing voice and data traffic is impossible for T1 links (1.5 Mbps). In the case of 10Base-T links, it is only possible for less bursty data flows, at the expense of a significant throughput reduction. Mixing FTP traffic with voice is possible only on 100Base-T links, in which case the link utilization must be kept below 20%. This result shows that data traffic is incompatible with voice traffic and should be separated from it.

To corroborate these simulation results, we examine [19] the transmission of voice on a VPN path between two Internet POPs in the U.S., coast-to-coast during business hours. Using delay measurements from this path, we simulate the quality of G.711 VoIP calls. In Fig. 1, we show the voice quality experienced, quantified by the MOS (Mean Opinion Score) scale. There are several series, each corresponding to different echo loss (echo cancellation) capabilities. EL=inf corresponds to perfect echo cancellation, while EL=31 corresponds to poor echo cancellation. Given that toll quality voice has a MOS of 4–5, we observe that even for good echo loss (EL=41, 51), there are many times in the course of the hour when quality is poor, due either to packet loss or excessive delay. We also note the importance of echo cancellation capabilities in contributing to voice quality.

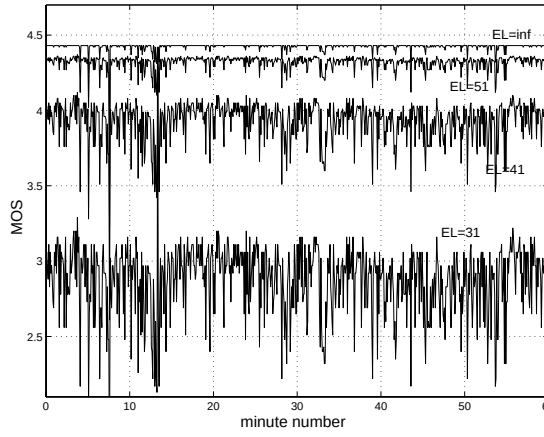

Fig. 1. MOS, averaged over 5-second intervals, for different echo cancellation capabilities.

We now simulate 1000 G.711 calls of exponential duration in the same environment. The playout deadline is optimized every 15 sec, and the echo cancellation is very good (EL=51). Even with this favorable setup over the path, we observe in Fig. 2 that the quality experienced is much worse than toll quality: around 10% of the calls experience at least one minute of poor quality (the MOS drops below 4), and 4% of all calls experience poor quality in at least 10% of

their minutes. Note that these results account only for voice transit between two POPs; end-to-end, the quality suffers yet further. Clearly, voice and data traffic should be separated into different traffic classes.

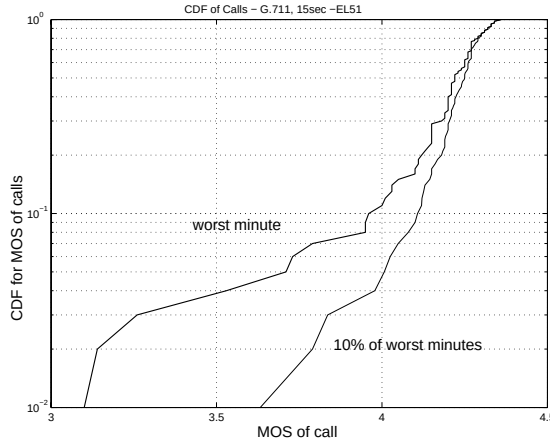

Fig. 2. Percentage of calls with MOS less than a particular value.

3.2 Mixing Voice and Video

Now that we have established the need for at least two queues, we consider whether voice and video can be mixed. First, we note that CBR and VBR video have very different characteristics, so we consider them separately.

When CBR video is mixed with voice, the increase in voice delay is contained because the CBR video streams are well behaved. Our work [15] shows that since voice and CBR video have similar characteristics and real-time delay requirements, we may allow them to be handled together. Consider a scenario in which CBR video streams are added to a 7-hop 10Base-T network carrying a 450 Kbps aggregate voice load. Voice delay exceeds 10 ms only when the link approaches full utilization. However, on a T1 link (1.5 Mbps), to achieve the target performance, only one video stream and two voice streams can be admitted, resulting in utilization as low as 55%. Therefore, unless the rate of the video stream is large compared to the link bandwidth, CBR video does not hurt voice traffic.

Consider the same scenario, with CBR video replaced by VBR video. As expected, given the characteristics of VBR video, the results of its mixing with voice traffic differ greatly from those of CBR video. Figure 3 shows the distribution of delay of voice and video packets when they are mixed and when they are separated. As we increase the number of multiplexed VBR video streams, both voice and video delays increase rapidly because of the long bursts that

get injected into the queue. This implies that if latency constraints are to be met, only a small number of video streams may be mixed with voice, resulting in low network utilization. In general, the achievable throughput depends on the voice/video mix, the burstiness of the video streams, and the link speeds; however, it is clear that VBR video cannot be mixed successfully with voice.

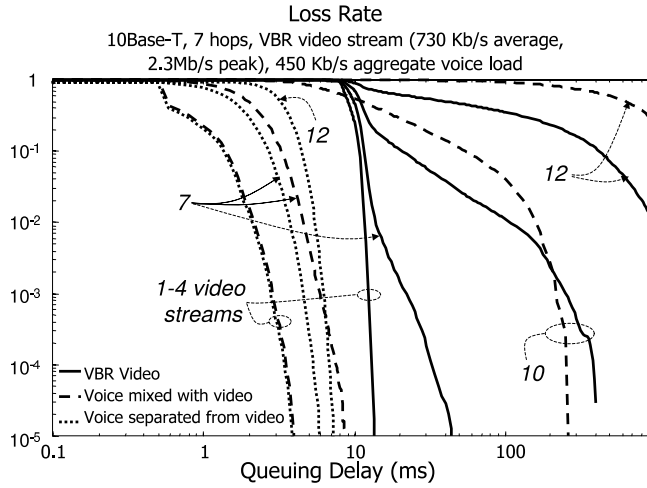

Fig. 3. Delay distributions for voice and VBR video. Video streams are added to the network.

3.3 Mixing Video and Data

We have shown that VBR video and voice are incompatible. The next question we attempt to answer is: can data and VBR video be mixed? It turns out that the answer depends on what delay can be tolerated by the video and what aggregate throughput is considered acceptable. In general, when the delay bounds are tighter, the interfering traffic load must be kept lighter. In addition, there is another important factor, which is the buffer size. Indeed, there is a tradeoff between a large buffer size, with a correspondingly large queueing delay, and a small buffer size, with the increased possibility of packet loss due to buffer overflow and the resulting decrease in throughput.

In Table 1, we show the results of a scenario in which a constant video load and FTP streams are mixed, and a video packet loss rate of 10^{-3} is tolerated. Considering the large dependence of the results on the buffer size, we experiment with a range of buffer sizes which scale according to the link speeds considered. We determine, for network delay requirements of 100 msec and 500 msec for the video stream, the total achievable aggregate throughput. We find that the buffer size which maximizes the throughput increases proportionally to the delay

bound, and the total achievable throughput can be increased if the delay bound is relaxed. Lower speed links are also more sensitive to proper buffer sizing, so it is not advisable to mix video with data even when interactivity is not required. To summarize, video should be mixed with data only on high bandwidth links.

Table 1. Video mixed with FTP traffic: maximum achievable throughput for 100 msec and 500 msec end-to-end delay bounds for video (tolerable loss rate: 10^{-3}). $D(Q\max)$ is the maximum buffer delay, $Q\max$ is the buffer size.

	10Base-T			**T3**			**100Base-T**		
$D(Q\max)$	$Q\max$ (KB)	throughput 100 ms	throughput 500 ms	$Q\max$ (KB)	throughput 100 ms	throughput 500 ms	$Q\max$ (KB)	throughput 100 ms	throughput 500 ms
40.1 ms	50	0%	0%	225	49%	49%	500	61%	61%
81.9 ms	100	29%	29%	450	75%	75%	1000	82%	82%
123.9 ms	150	20%	37%	675	56%	90%	1500	82%	82%
491.5 ms	600	0%	96%	2700	0%	96%	6000	0%	90%

3.4 The Case for Three Classes of Service

From the results above, we conclude that separating multimedia traffic into a minimum of three queues—voice, video, and data—is necessary for good performance for a range of network conditions. Interactive voice, with its high expectations, requires a queue of its own, for protection from the bursty traffic of VBR video and TCP data applications. Video may be mixed with data under particular conditions, but other considerations also compel us to give it its own queue. The real-time constraints of video call for higher priority in scheduling compared to data. Furthermore, the reliability levels required by video are close to 100%, whereas TCP is designed to recover from loss. Finally, in contrast to data traffic, video streams should be subjected to admission control due to their large and predictable rates. The nature of CBR video allows it to be combined with voice or with VBR video.

Yet we must be sensitive to the two extremes. High speed links tolerate better the mixing of disparate traffic types, such that on very high speed backbone routes, differential handling may be unnecessary. Likewise, very low speed links may require finer grain distinction of packets and/or packet preemption mechanisms.

4 Network Node Structure

From the discussion in the previous section, we can consider a model for the internal structure of a network node (output) port, which is shown in Fig. 4. This structure is to be used at places where packet buffering is performed within the switching node, which could be at the input ports, the output ports, or both.

We assume here that the node has an output queued implementation to avoid having to go into the details of the different possible switching architectures. We describe the components of the port (classifier, traffic conditioner, buffer management, and scheduler) in more detail below.

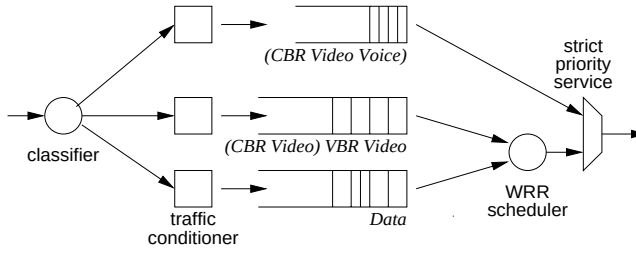

Fig. 4. The structure of a network node port containing three queues.

4.1 Packet Classification

The first step in providing differentiated services is enabling network nodes to identify the class of service of each packet they receive, possibly through a special marking carried by the packet. For example, the DiffServ architecture [10] uses the byte in the IP header previously allocated to TOS and renamed DS Field, as a priority code. In the IEEE 802.1 LAN realm, packet identification is done through a field in the recently adopted VLAN tag (added to the MAC frame header) that indicates the class of service of each packet. The priority code is checked by the classifier upon reception of a packet to determine the queue in which the packet is to be placed. Packets carrying the same marking expect to receive the same treatment in the network. The discussion in the previous section suggests that 3 queues are necessary: one for voice, another for video, and the third for data. However, allowing for more queues (e.g., 8 as in IEEE 802.1D) increases flexibility in assigning traffic to appropriate classes.

Before a packet is enqueued, it goes through traffic conditioners, which perform functions such as metering and policing. The policer ensures that the traffic entering a queue does not exceed a certain limit, determined by the queue's allocation of the link resources. This functionality is particularly needed for queues that are serviced with high priority in order to avoid starvation of lower priority traffic.

4.2 Scheduling

With traffic separated in multiple queues, a scheduler is required to service them. The scheduler's service discipline needs to be carefully designed in order to provide the appropriate delay through the node for each traffic type. In this

section, we discuss the appropriate service discipline for each queue and the supporting mechanisms needed, such as admission control.

Voice. In Sec. 3, we argued for separating voice traffic in a queue of its own. The next step is to determine the appropriate scheduling discipline for providing voice with the required quality of service. In [14], we show that a strict high priority service is appropriate for handling voice traffic. Through modelling and simulation, we find that, considering the conservative 99.999^{th} percentile of packet delays, priority queueing does limit the delay and jitter of voice packets for typical link speeds. Refer to Fig. 5, where we plot the complementary cumulative distribution function (ccdf) of voice packet delays for a voice load of 1.1 Mbps over five 45 Mbps hops. We compare different link scheduling schemes, namely, priority queueing with preemption of low priority transmissions, priority queueing (PQ), weighted round robin (WRR) with a weight of 1.5 Mbps, and WRR with a weight of 10 Mbps for the voice queue. We also plot the ccdf for the case where voice packets are given a separate 10 Mbps circuit. The graphs show that, as would be expected, priority queueing with preemption achieves negligible queueing delays over the 5 hops. In addition, non-preemptive priority queueing still results in low queueing delays (the 99.999^{th} percentile is smaller than 2 msec, ignoring switching time through the node). WRR scheduling requires a large weight for the voice traffic (10 Mbps, more than 9 times the actual load) for it to compare to PQ. Note that the round robin scheduler insures that a large weight for the voice queue does not translate into wasted resources, since low priority traffic can utilize any unused bandwidth. In contrast, providing a 10 Mbps dedicated circuit for voice results in delays that are better than those of non-preemptive priority queueing and WRR, at the cost of wasted resources.

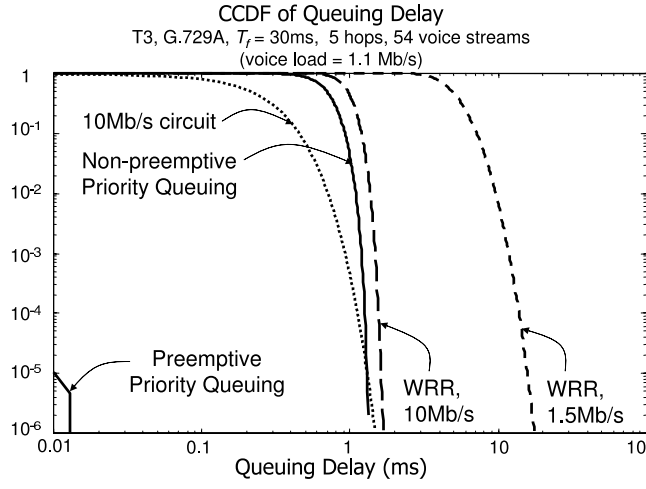

Fig. 5. Voice delay distributions for different link scheduling schemes.

While it may appear that both priority queueing and WRR can be used, the results in this graph consider only one low priority queue. Therefore, one would expect worse results if the round robin scheduler services more queues, where a voice packet may have to wait for more than one low priority packet transmission. While WRR has the well known advantage of preventing starvation of low priority queues, we believe that in the context of the Internet, no special precaution has to be taken to prevent voice traffic from starving others. Indeed, voice traffic volume is limited, and its growth rate is significantly smaller than that of data and other applications. Therefore, its share of the total traffic is decreasing. This means that not only would it not starve other traffic, but also that voice traffic may not require per-flow admission control. Rather, appropriate provisioning of the network would allow it to be serviced at highest priority.

Video. As discussed in Sec. 3, CBR video traffic is well behaved and can be mixed with voice traffic. If CBR video is mixed with voice traffic in the same queue, admission control would be needed, given the higher rates of video streams. On the other hand, VBR video streams need to be mapped to a queue separate from voice. This queue has to be serviced using round robin scheduling; otherwise, it may starve lower priority classes due to the burstiness of its traffic. Similarly to CBR video, VBR video streams may need to be subjected to admission control.

Data. Data applications could be all mapped to the same queue or, given the wide range of characteristics and requirements among them, may benefit from being mapped to multiple queues. Thus, if more than one data queue is available, low rate interactive data applications can be shielded from other data applications by separating them. In particular, telnet would benefit from having its own queue, serviced with high weight, especially on low speed links.

If the buffer sizes are chosen small enough to restrict the queueing delay of interactive packets to acceptable levels, all data applications may share the same queue. Then, differentiation among the different applications can be provided by assigning the packets to different drop priority levels within that queue.

Given that most data flows are of the short lived type [25], it is impractical to perform per-flow admission control. In addition, TCP generates bursty traffic, and therefore it is not possible to guarantee congestion-free service for data traffic without significant over-provisioning. Thus, the data queue should be serviced with a round robin scheduler to avoid starving lower-priority queues, if any.

4.3 Buffer Management

To enable high speed processing, the buffers would most likely have to be implemented as FIFO queues. Therefore, any action to be taken has to be performed before the received packet is enqueued. Possible actions are dropping or marking the packets, e.g. if explicit congestion notification (ECN) is implemented [8].

Early dropping/marking schemes such as Random Early Detection (RED) and its derivatives, aim at providing early notification of congestion before the buffer gets full, and bursty packet loss becomes necessary. Such schemes assume that an end-to-end congestion control mechanism, such as the one implemented in TCP, will react to the congestion signals. Therefore, they may not be effective or appropriate for applications which do not use such mechanisms. Indeed, different buffer management schemes should be used for different classes. Moreover, the size of the buffers should be tailored to suit the application types. Thus, small buffers would be used for packet-delay sensitive traffic to limit queueing delays, while large buffers can be used for applications that are insensitive to per-packet delay, but are sensitive to packet loss.

The priority code of each packet may indicate, in addition to the queue where it is to be placed, one of several drop priorities within that queue. This approach is used in the DiffServ AF class. Providing several drop priorities within each queue allows further differentiation among packets and may be used to achieve significant improvements in quality degradation during congestion episodes, as discussed in Sec. 5.

5 Improving Resilience to Congestion for Video and Data Applications

With traffic types appropriately mapped to different queues in the network, the dynamics of each queue depend on the particular traffic it is serving. Adequately serviced voice traffic would see little queueing delay and loss in the network, and its treatment within the queue does not require further refinement. This is not the case for video and data traffic. Since it is not possible to guarantee congestion-free delivery to unshaped bursty traffic, performance degradation may occur for such traffic. In this section, we show how providing several packet drop priorities within one queue can be used to significantly improve the user-perceived quality of applications during congestion episodes.

First, for video applications, we show that layered video in association with priority dropping is a simple, yet effective technique for providing graceful degradation in the event of packet loss. Then, we address data applications, which themselves span a wide range of requirements and characteristics. We show how identifying and prioritizing different applications can improve user experience.

5.1 Addressing Packet Loss with Layered Video

Let us consider the transmission of digital video over packet networks. One particular characteristic of video is its high sensitivity to packet loss. Video quality is greatly eroded when there is loss of data which contribute heavily to quality, such as low frequency DCT coefficients, motion vector information, or start codes needed for synchronization. In Fig. 6 we illustrate the drastic quality degradation of a video sequence resulting from random packet loss. The line segment on the left corresponds to a video sequence encoded with P frames

and B frames; for the line segment on the right, only I frames were used in the encoding of the same sequence. When the sequence contains P and B frames, 1% packet loss can lead to poor quality because there is interdependence among pictures—the loss of a single packet can have an effect on multiple subsequent P and B pictures. We observe in the plot that when this interdependence is removed, such that all frames are encoded as I frames, the quality is less affected by packet loss.

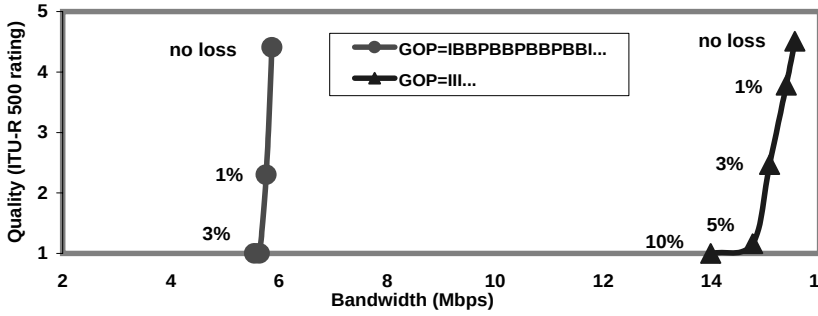

Fig. 6. Random packet loss is applied to an MPEG-2 video stream packetized using RTP. Video quality degrades sharply.

There are several possible ways to deal with packet loss. Adaptive encoding could be used, in which feedback from the receiver or a network node provides the source with information to adapt the transmission rate by modifying encoding parameters. However, this is fairly complex to implement and is limited by feedback delay. It is not suitable for networks with high variability, for multipoint communications, or for stored video. The technique of smoothing or shaping can limit the variability of the stream's traffic, and the use of admission control can limit the variability in the aggregate traffic. This, too, introduces complexity in the nodes, and it curtails statistical multiplexing gain, decreasing overall throughput. Furthermore, smoothing or shaping introduces delay, clearly undesirable when latency is of concern.

Because loss is inevitable, we must limit its effect by dealing with it intelligently, protecting important data and dealing with loss where and when it occurs. This leads us to consider layered video and priority dropping [16,24]. Simply put, layered video prioritizes information according to its importance in contribution to quality. In conjunction with priority dropping, layered video is a powerful technique for maintaining quality in the presence of loss. We show that it offers graceful quality degradation rather than the sharp drop we saw in Fig. 6, and we show how to divide a video stream into layers to maximize the perceived video quality for a particular range of network conditions.

Video layering using data partitioning. Layering mechanisms define a base layer and one or more enhancement layers that contribute progressively to the quality of the base layer. A base layer can stand alone as a valid encoded stream, while enhancement layers cannot. The key observation is that some bits are more "important" than others; we identify their importance by placing them into different layers, thus allowing a node to drop packets with discrimination.

The MPEG standards [11,12] specify four scalable coding techniques for the prioritization of video data: temporal scalability, data partitioning (DP), SNR scalability, and spatial scalability. We consider layering based on data partitioning, treating temporal scalability as a special case of it. One advantage of data partitioning is that the overhead incurred by layering is negligible. Another advantage is that it is performed after the encoding of the stream, allowing it to be easily used with pre-encoded video.

Data partitioning divides the encoded video bit stream into two or more layers by allocating the encoded data to the various layers. Naturally, the data with the most impact on perceived quality should be placed in the base layer. To indicate the portion of the data which is included in the base layer, we define a drop code for each picture type, I, P, and B. The drop code takes on a value from 0 to 7, where 0 indicates that all of the data are included in the base layer, and 7 indicates that only the header information is placed in the base layer. The partitioning of the stream data into the base and enhancement layers is completely specified by a drop code triplet, e.g., (036). There is a correspondence between the drop code value and a header field defined by the MPEG standards that can be used for data partitioning [16].

Temporal scalability is a special case of data partitioning, where the drop code is 007. In temporal scalability, entire B pictures are dropped prior to dropping any information in I or P pictures.

We illustrate in Fig. 7 the advantage of using 2-layer data partitioning in a network which supports priority dropping. In this figure, we show the quality of video using temporal scalability, data partitioning, and no layering. With temporal scalability, the dropping of B picture data degrades quality. If DC coefficients and motion vector information from B frames were not dropped, the decoder could have reconstructed enough of these frames to significantly improve the perceived quality. Therefore, not all B frame data should be assigned to the enhancement layer. Thus, data partitioning using a well-chosen drop code triplet (036) allows for graceful quality degradation as enhancement packets are randomly dropped. In the region where no base layer packets need to be dropped (above 3.8 Mbps), the quality degradation incurred varies almost linearly with the rate of packet loss. However, once we start losing base layer packets, the quality falls sharply. This establishes the need to protect the base layer from network loss with an appropriate nodal structure.

We have shown the graph for drop code triplet (036). Other layering structures (i.e., other triplets) will place the knee at different points while exhibiting the same two-piece behavior. We have, through simulation, identified the layering structures which achieve the highest quality for a given bandwidth. These

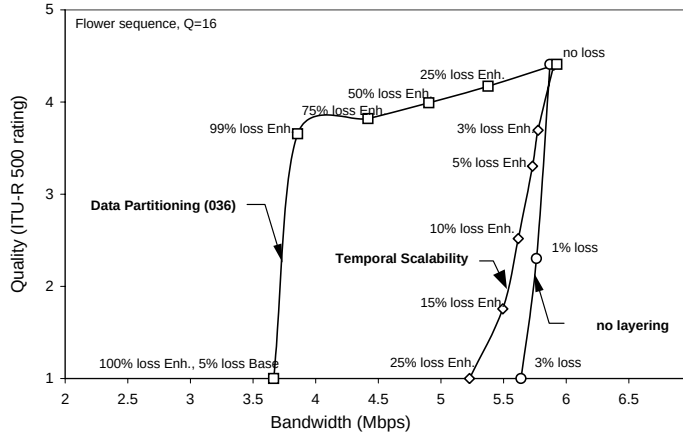

Fig. 7. Bandwidth-quality tradeoff curves for 2-layer DP, temporal scalability, and no layering.

dominant structures are the same for all sequences studied, and they correspond to the following triplets: 003, 014, 005, 015, 016, 036, 136, 046, 146, and 156 [16]. In practice, it is desirable to choose from these triplets the proper layering structure such that the linear portion of the graph is large enough to just cover the expected bandwidth range delivered by the network.

Quality degradation can be further improved with 3-layer data partitioning. In the example shown in Fig. 8, we create three layers by keeping the enhancement layer the same as in 2-layer DP with drop triplet (036). We then make the break between the middle layer and the base layer at the point where 2-layer

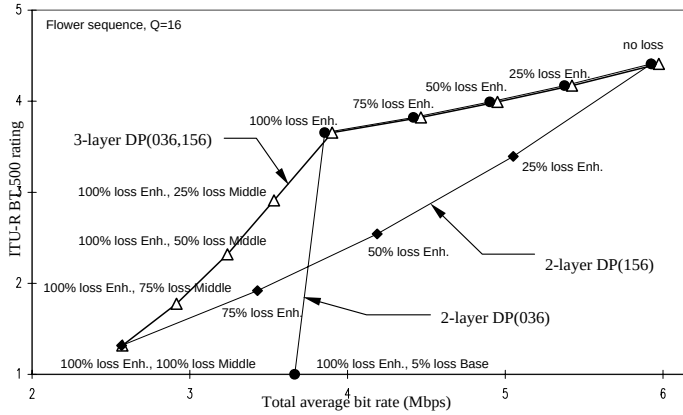

Fig. 8. Bandwidth-quality tradeoff curves for 3-layer DP compared to 2-layer DP schemes.

DP(156) does. Thus, in the 3.8–6.0 Mbps range (only the enhancement layer is dropped), the behavior parallels that of the 2-layer DP(036). The only difference is a slight increase in overhead. In the region where the middle layer is being dropped (2.5–3.8 Mbps), the quality is superior to the same region for DP(156), because the least important data have been identified and dropped first. As one would suppose, additional layers do continue to improve video quality, though with limited incremental improvement beyond 4 layers.

Multiplexing layered streams. So far, we have examined a single video stream with prioritized random loss in the enhancement layers. We now consider the case where several layered streams share a limited resource. We have observed graceful quality degradation as the number of streams is increased, as shown in Fig. 9. We have also studied SNR scalability [16] but do not make further comment here, except to remark that it performs similarly to DP, but the latter is preferred because of its negligible overhead and ease of implementation.

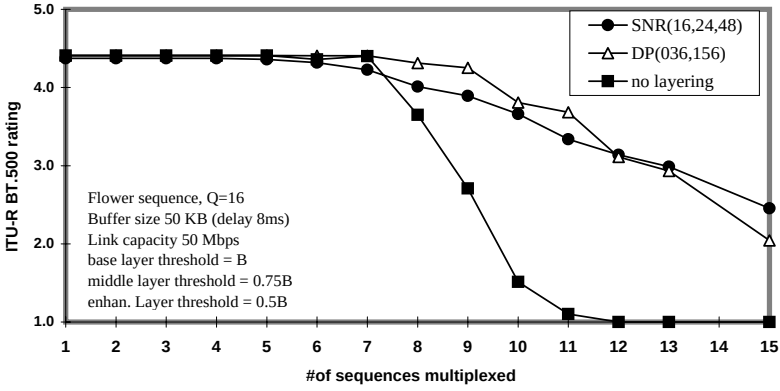

Fig. 9. Multiple video streams sharing a 50 KB buffer, for three schemes: SNR scalability, DP, and no layering.

We conclude that the combination of a simple layering of video data and a simple priority dropping mechanism at network nodes, appropriately employed, can have a significant effect on sustaining video quality in the face of packet loss.

5.2 TCP Applications

Until now, all data applications have used best-effort service for lack of an alternative. However, most Internet users have experienced times where severe quality loss is suffered. Such degradation is most distinctly perceived when associated with interactive applications. Hence, telnet interactivity is severely hindered,

and web page download times become excessive during congested hours. This is particularly unacceptable for business applications which require predictable service. While large page download times can be attributed in part to server overload, we focus here on the delays caused by the interaction of TCP's congestion control mechanisms with packet loss in the network.

We observe that, similar to layered video, some data packets contribute more to user perceived quality than others. With large window sizes, the TCP fast retransmit mechanism can be used to minimize the impact of packet loss. However, when the congestion window is small, there is a much longer delay to recover from a lost packet.

Here we illustrate, using simulation results, the benefits that can be achieved for interactive applications when their packets are appropriately prioritized in the network. We consider that all data applications share one queue. Packets are marked at the source with one of 3 drop priorities. For FTP and HTTP, the SYN packet and the packets sent when the connection is operating with a small congestion window are marked as high priority because of the penalty in recovering from their loss. For telnet, all packets are marked with high priority. The aggregate high and medium priority traffic generated by each station is shaped to conform to two token bucket profiles, the goal of which is to limit the amount of high and medium priority packets injected by each user. The access to high priority tokens at the source is prioritized based on the application, with telnet receiving the highest access priority, and FTP the lowest.

The topology used for the simulations consists of 800 user stations, organized into 400 source-destination pairs of different round trip times (ranging from 20 msec to 200 msec), and connected by a symmetric tree with hierarchical link speeds (starting at 1.5 Mbps for user links, with a bottleneck of 100 Mbps). The router buffers are appropriately sized to provide low delay for telnet, while giving good performance to HTTP and FTP traffic. A randomized dropping function similar to RED is used for dropping packets for each of the three priorities. As the queue size increases, low priority packets are dropped first, followed by medium priority packets. High priority are only dropped when the queue size gets close to the maximum buffer size. Traffic consists of 1 telnet and 1 web connection per source-destination pair, with background traffic of repetitive short FTP file transfers (200 KB) in both directions.

In the following, we illustrate how the performance of interactive applications can be improved by appropriate service differentiation, at a modest cost to non-interactive applications. In Fig. 10, we plot the complementary cumulative distribution function of web download times[1] for different service differentiation scenarios. The curves marked DT and RED correspond to scenarios without service differentiation (best effort), with queues managed using Drop Tail and RED, respectively. As can be seen from the plot, 10% of the page downloads for both Drop Tail and RED suffer a delay in excess of 19 seconds.

[1] We show results for HTTP/1.0 traffic here, for a web page with eight 10 KB imbedded images. Up to 4 connections are opened in parallel to download the page components. Similar results were obtained for HTTP/1.1.

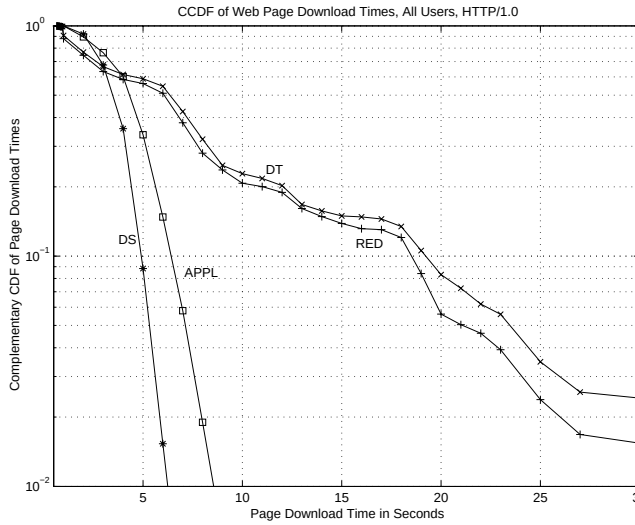

Fig. 10. The complementary cumulative distribution function of web page download times with (DS and APPL) and without (DT and RED) service differentiation.

In contrast, we show the results with service differentiation, where packets are marked at the source based on the application and TCP connection state. The corresponding curve, marked DS, clearly shows a significant improvement in terms of download times, where all downloads take between 3 and 6 seconds.

In Fig. 11, we plot the ccdf of short file transfer times, which shows that the improvement in page download times was obtained at little cost to the background traffic. A simpler form of differentiation would be to base the packet drop priority marking solely on the generating application type. Thus, packets belonging to telnet and similar low-bandwidth and delay sensitive applications such as Internet gaming would be marked high priority, and those belonging to short web page downloads would be marked medium priority. Packets generated by other, non-interactive applications such as FTP, would be marked low priority. The aggregate traffic of each priority is again shaped to limit its rate. The curves marked APPL in Figures 10 and 11 show that this technique can improve the performance of web page downloads, but rather less effectively than the more intricate method (DS) and at a higher cost in performance loss to the background traffic. Similar results can be shown for telnet in this scenario, where appropriate differentiation is provided through marking all of its packets at high priority. This allows the elimination of excessive delay of character echoes (1 sec), which result from retransmits due to packet loss. Without service differentiation, these delays occur for about one out of ten typed characters (more details can be found in [20]). In conclusion, it is possible to use multiple drop priorities to the advantage of interactive applications, in order to improve the user-perceived performance of such applications.

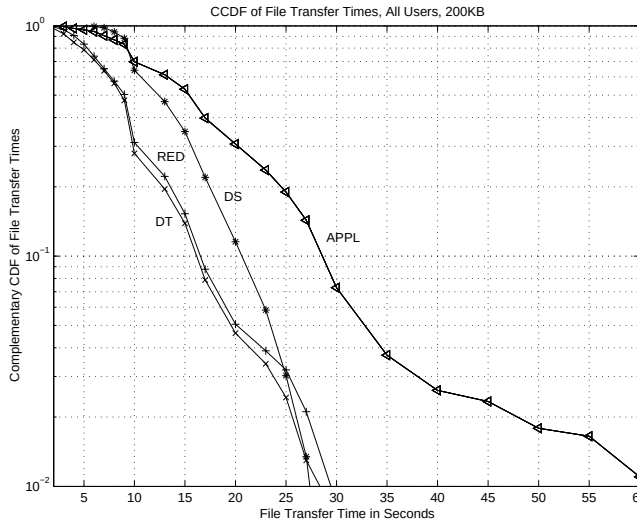

Fig. 11. The complementary cumulative distribution function of file transfer times with (DS and APPL) and without (DT and RED) service differentiation.

6 Conclusion

For the Internet to support multimedia applications, service differentiation is needed. In this paper, we describe the characteristics and requirements of voice, video, and interactive data applications, and we demonstrate the peformance improvements achieved by providing different treatments for each of these three types of traffic. We propose a three queue model for network nodes with one queue for voice traffic, one queue for video, traffic, and one queue for data traffic. The voice queue is served with strict priority, and the video and data queues share the remaining capacity using weighted round robin scheduling. We also show that the performance of the video and data queues may be further improved by using multiple levels of drop precedence within each queue and by marking packets according to their importance.

References

1. Bhatti, N., Bouch, A., and Kuchinsky, A.J. Integrating User-Perceived Quality into Web Server Design. Presented at *WWW'00*, Amsterdam, 2000.
2. Bolot, J.-C. End-to-End Packet Delay and Loss Behavior in the Internet. in *Proceedings of SIGCOMM'93.*
3. Bouch, A., Sasse, M., and DeMeer, H.G. Of Packets and People: A User-Centered Approach to Quality of Service. Submitted to *IWQoS'00.*
4. Boyce, J.M. and Gaglianello, R.D. "Packet loss effects on MPEG video sent over the public Internet. In *Proceedings of ACM MULTIMEDIA '98*, pages 181–190, Bristol, England, September 1998.

5. Crovella, M. and Bestavros A. Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. *IEEE/ACM Transactions on Networking*, December 1997.
6. Daigle, J. and Langford, J. Models for Analysis of Packet Voice Communications Systems. in *IEEE Journal on Selected Areas in Communications*, Volume 4, Number 6, September 1986.
7. Dalgic, I. and Tobagi, F. "Glitches as a Measure of Video Quality Degradation Caused by Packet Loss," In *Packet Video Workshop '96*, Brisbane, Australia, March 1996.
8. Floyd, S. TCP and Explicit Congestion Notification. In *ACM Computer Communication Review*, Volume 24, Number 5, October 1994.
9. Gruber, J. and Strawczynski, L. Subjective Effects of Variable Delay in Speech Clipping in Dynamically Managed Voice Systems. In *IEEE Transactions on Communications*, Volume 33, Number 8, August 1985.
10. IETF, DiffServ Working Group, http://www.ietf.org/html.charters/diffserv-charter.html.
11. ISO/IEC, "Generic coding of moving pictures and associated audio information" (MPEG-2), ISO/IEC 13818-2, 1995.
12. ISO/IEC, "Generic coding of audio-visual objects" (MPEG-4), ISO/IEC 14496-2, 1999.
13. Jacobson, V. Compressing TCP/IP Headers for Low-Speed Serial Links, RFC 1144, February 1990.
14. Karam, M. and Tobagi, F. Analysis of the Delay and Jitter of Voice Traffic Over the Internet. In *Proceedings of INFOCOM'01.*
15. Karam, M. and Tobagi, F. On Traffic Types and Service Classes in the Internet. In *Proceedings of GLOBECOM'00.*
16. Kimura, J., Tobagi, F., Pulido, J-M., and Emstad, P. "Perceived Quality and Bandwidth Characterization of Layered MPEG-2 Video Encoding" in *Proceedings of the SPIE International Symposium on Voice, Video and Data Communications.* Boston, Mass, September 1999.
17. Leland, W.E., Taqqu, M.S., Willinger, W., and Wilson, D.V. On the self-similar nature of Ethernet traffic, *IEEE Transactions on Networking*, Vol. 2, No. 1, Feb. 1994.
18. Mah, B. An Empirical Model of HTTP Traffic. In *Proceedings of INFOCOM'97.*
19. Markopoulou, A. and Tobagi F. Assessment of Perceived VoIP Quality Over Today's Internet. TR-CSL work in progress, Stanford University.
20. Noureddine, W. and Tobagi, F. Improving the User-Perceived Performance of TCP Applications with Service Differentiation. TR-CSL work in progress, Stanford University.
21. Paxson, V. Empirically-Derived Analytic Models of Wide-Area TCP Connections. In *IEEE Transactions on Networking*, 2(4), August 1994.
22. Paxson, V. End-to-End Internet Packet Dynamics. *IEEE/ACM Transactions on Networking*, Vol.7, No.3, pp. 277-292, June 1999.
23. Paxson, V. and Floyd, S. Wide-Area Traffic, the Failure of Poisson Modeling. In *ACM Computer Communication Review*, October 1994.
24. Pulido, J.-M. *A Simple Admission Control Algorithm for Layered VBR MPEG-2 Streams*, Engineer's thesis, July 2000. Available as Stanford University's Computer Systems Laboratory Technical report CSL-TR-00-806.
25. Thompson, K., Miller, G.J., and Wilder R. Wide-Area Internet Traffic Patterns and Characteristics. In *IEEE Network*, November/December 1997.