A Novel Wide-Band Audio Transmission Scheme over the Internet with a Smooth Quality Degradation

Fulvio Babich
Dipartimento di Elettrotecnica, Elettronica ed Informatica
Università di Trieste
Via A. Valerio 10, I-34127 Trieste, Italy
babich@stelvio.univ.trieste.it

Marko Vitez
Dipartimento di Elettrotecnica, Elettronica ed Informatica
Università di Trieste
Via A. Valerio 10, I-34127 Trieste, Italy
vitez@stelvio.univ.trieste.it

ABSTRACT

Real-time delivery of multimedia information over the Internet is finding increasing interest. This paper considers wide-band audio transmission utilizing a priority scheme. The proposed scheme complies with both the new Internet Protocol Version 6 (IPv6) and the current Internet Protocol Version 4 (IPv4), providing that, in the latter case, routers are set to manage priority. A new queuing algorithm, namely Priority Weighted Fair Queuing (PWFQ), is defined and evaluated. A scalable audio encoder is adopted to perform audio transmissions over an emulated network. Background traffic is emulated, employing a traffic generator that adopts a self-similar model. Objective and subjective quality tests are performed, using a set of musical excerpts. Quality is evaluated as a function of Internet traffic. In the paper it is shown that, by adopting an encoding technique with scalable bit-rate, and a prioritized transmission algorithm, a smooth degradation of quality may be obtained during network congestion periods. This technique shows better performance than feedback-based algorithms, in which the delayed responses cause the core stage packets to be lost in low-to-high traffic transitions and the enhancement packets not to be transmitted in high-to-low traffic transitions.

Keywords

Technical subjects: Internet applications, Multimedia, Weighted Fair Queuing, Audio on Demand, Quality of Service.

1. INTRODUCTION

Real-time delivery of multimedia information over the Internet is finding increasing interest. New services, such as audio and video-conferencing, audio and video-broadcasting, or audio and video on demand may become popular and world-wide spread within a short period of time. This paper considers wide-band audio transmission over the Internet. More precisely, audio on demand distribution is taken into consideration. Most of existing audio on demand applications have been designed for the current version of the Internet protocol, whose *best effort* operation does not fit well with the stringent quality requirements of real-time services. In this paper, the benefits that may be achieved by adopting a prioritized transmission scheme are discussed. To this end, a new queuing algorithm, namely Priority Weighted Fair Queuing, is defined and compared with the previously experimented techniques. The proposed technique complies with the new version of Internet protocol, IPv6, which is explicitly designed for prioritized transmission schemes [1]. However, the routers can be configured to consider the Type of Service field of IPv4 as a priority field and, therefore, the IPv4 protocol can be used too.

To fully exploit the benefits of a prioritized transmission scheme, a suitable audio encoding algorithm is also required. The Mobile Audio Visual Terminal (MAVT) audio encoder-decoder, chosen for the analysis presented in this paper, adopts an object oriented coding scheme, which is very flexible in its configuration [2]. The used coding scheme allows bit-rate scalability, and is suitable for a straightforward prioritized packetization of audio information. In this paper it is shown that, by adopting a scalable encoding technique, and a prioritized transmission algorithm, a smooth degradation of quality may be obtained during network congestion periods, without using any feedback message. Objective and subjective quality tests are performed, using a set of musical compositions. Quality is evaluated as a function of Internet traffic. To perform a realistic performance evaluation, a network emulator is developed and used. The emulator adopts a traffic generator based on a self-similar traffic model. This model suits very well the typical Internet traffic patterns, which are characterized by traffic bursts that can be found at any time scale, from milliseconds to hours [3]. A new queuing algorithm, PWFQ, is introduced and used in the network emulator. It implements the priority mechanism and guarantees fair network utilization, when the network is congested. It allows the sender not to bother about network congestion, making any feedback mechanism unnecessary.

The most important feature of the proposed PWFQ scheme, which is not a congestion avoidance technique, is the smooth degradation of quality that may be obtained during network congestion periods. Moreover, PWFQ has the advantage to be fair among different streams. Given that PWFQ does not adopt any feedback mechanism, it does not suffer from the drawbacks caused by the delayed effects of feedback-based techniques. It has not been tested in a multicast environment, but it is nevertheless compatible with it.

The paper is organized as follows. Section 2 illustrates some background work and the motivations of the work presented in this paper. Section 3 describes the chosen audio encoder-decoder and the prioritization scheme. Section 4 introduces the traffic emulator.

The quality evaluation results are presented in Section 5. Finally, Section 6 summarizes the most important conclusions.

2. BACKGROUND AND MOTIVATIONS

Various techniques have been previously proposed, which use scalable encoders to transmit audio or video information in differentiated flows. They focus on multicast environments and use different multicast groups for different layers of the data stream. Some type of feedback mechanism is usually adopted to test for available bandwidth [4]. In [5], authors base their considerations on the TCP-Friendly Rate-Based Flow Control, which tries to be fair with competing TCP traffic. Given that the multicast group *leave* operation is quite lengthy and its effects are delayed, the *join* attempts should be infrequent. In [6] an interesting scheme is proposed for testing the available additional bandwidth without trying to *join* a group: the sender transmits occasional bursts, which cause packet losses in case of a lack of bandwidth. In [7], authors concentrate more on the quality of the received audio, than on the bandwidth fairness or congestion avoidance, by transmitting redundant low-bandwidth data.

Some efforts have been done in the networking society to develop priority-enabled protocols, but the priority feature has never been fully implemented for Internet applications. Some discussions and experiments on priority transmission can be found in [8]. As the authors in [8] correctly point out, when priority based mechanisms are adopted, there is no performance incentive for the applications to transmit less data during the network congestion periods. Therefore, the bad behavior of one sender would prevent other well-behaved flows (like all the TCP flows, which accomplish congestion control) from transmitting. Thus, a queuing algorithm that enforces fairness between flows is appealing. The WFQ algorithm has been adopted by router manufacturers and a significant amount of work and research has been done in the field of fair queuing algorithms ([9], [10], [11]). Therefore, the combination priority + fair queuing + scalable encoding seems to be a valid idea for audio and video transmission over the Internet. Some critics have been moved toward fair queuing algorithms because of their intrinsic complexity, but they are continuously evolving and new techniques are being discovered and implemented, such as the core-stateless fair queuing [12], which can lighten the processing load of the more critical backbone routers. Thus, fair queuing may become fundamental in preventing network collapse, because of increasing multimedia traffic that does not accomplish congestion control, and is overloading the network. The proper combination of layered source coding, priority dropping of packets in the network, and fair queuing, is the subject of the present work.

3. AUDIO ENCODER

The Mobile Audio Visual Terminal (MAVT) audio encoder-decoder, chosen for the analysis presented in this paper, adopts an object oriented coding scheme, which is very flexible in its configuration [2]. It consists of a low band core algorithm, which can contain any speech or music compression scheme (standard or non standard), and of an arbitrary number of low band and high band enhancement stages. The number of enhancement stages allocated to a certain band depends on the result of energy and Signal to Noise Ratio (SNR) evaluations and on the available bit-rate. Core bits identify the minimum amount of information needed to provide an output signal. Enhancement bits are relevant to the additional information, which will provide the maximum quality, if delivered to the destination. However, they can be discarded with no need to change the core information. This coding scheme allows complexity scalability and interoperability with other standards, because a plurality of core low-rate algorithms/tools can be used by the algorithm. For music encoding, MAVT adopts a subdivision into 20 sub-bands, in the range 0-20 kHz. The sampling rate may be selected among the following values: 8, 12, 16, 20, 24, 28, 32, 40, and 48 kHz.

3.1 Bit-stream description

The bit-stream generated by MAVT audio encoder consists of a collection of different audio objects. The first byte of the stream, named OVH0, specifies the number of objects, the input and the output sampling rates. Audio samples are encoded on a 32 ms frame basis. For every frame, the stream includes the parameters described in Table 1.

3.2 Packetization and Priority scheme

Audio transmission utilizes a set of User Datagram Protocol (UDP) sockets. Different sockets correspond to different priority levels. MAVT parameters are included in the payload of audio packets, each of which consists of a length field (8 bits), a sequence number field (16 bits) and a payload of variable length. The highest priority packets include also a field specifying the maximum suggested packet size. This size may not be satisfied by the highest priority packets, which must include in every case all the parameters described in Table 1, excluding the music core and enhancement stages. Packets belonging to lower priorities must satisfy the maximum suggested packet size. The high priority packet structure is shown in Fig. 1.

The MAVT encoded stages do not contain any priority information. Thus, priority scheme allocates music core bits and enhancement stages among packets of different priorities in an equilibrated fashion, in order to distribute the distortion among different frequency bands. The prioritization unit operates as follows.

1. Given the selected maximum bit rate, it determines the maximum number of bits per frame;
2. Given the selected maximum packet size, b, it determines the allowed bandwidth, W, given by

$$W = \begin{cases} b/128 \text{ kHz}, & b \leq 512, \\ 4 + (b-512)/128 \text{ kHz}, & 512 < b \leq 768, \\ 8 + 3(b-768)/320 \text{ kHz}, & \text{otherwise}; \end{cases} \tag{1}$$

3. All core end enhancement stages in bands exceeding the allowed bandwidth are discarded;
4. From the highest allowed band, it discards the highest level enhancement stages; this operation, which may be repeated for the following levels, if necessary, and may include the core levels, ends when the allowed number of bits is obtained;
5. The core and enhancement stages that have been discarded at a certain priority level may be included in a lower priority level, if any.

As a consequence first priority allocation scheme performs a bandwidth limitation and then it de-allocates the enhancement stages belonging to the allowed bands. A simplified de-allocation example is summarized in Fig. 2.

The number of priority levels can be chosen arbitrarily. In our simulations we adopt, as an example, 3 priority levels with the same bit rate for every level (later named flow).

Description	Bits	Symbolic Name
Abstract class	1	OVH1
Concrete class (sub-frame 1)	3	OVH2
Concrete class (sub-frame 2)	3	OVH2
Used sub-bands (0 to 20)	5	OVH2
Stages per band (1-4 bands)	3 per band ($\leq$ 12 bits)	OVH3
Stages per band (5-20 bands)	2 per band ($\leq$ 32 bits)	OVH3
Speech core bits	0 if Abstract class = 0 (music encoding: core not used) else 4 for Concrete class = 0 32 for Concrete class = 1 89 for Concrete class = 2 133 for Concrete class = 3 76 for Concrete class = 4 128 for Concrete class = 5 172 for Concrete class = 6	
Music core and enhancement stages	56 + 49 * (enh. stages - 1) per band or 0 if there are not enhancement stages for a given band.	

Table 1: MAVT frame parameters

Packet length in bytes (8)	Sequence number (16)	Maximum suggested size (8)	MAVT data (Payload, variable length)

Figure 1: High priority audio packet

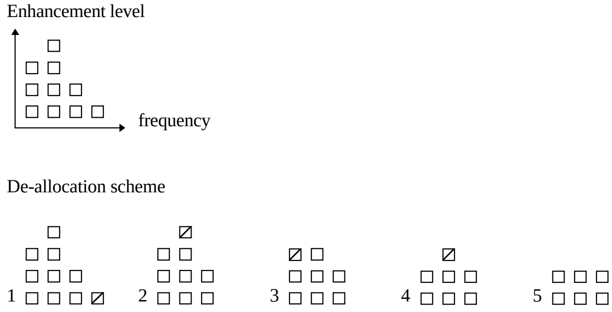

Figure 2: Example of de-allocation procedure

3.3 Interleaving

Given the bursty nature of packet discarding due to congestion, the use of interleaved transmission of packets may be convenient to spread losses among non consecutive packets. In this paper a block interleaved scheme is evaluated. The block length is kept small, to avoid the introduction of an excessive delay in the decoding procedure.

Assume that b is the block size and i is the interleaving step, being b/i an integer number. The block interleaved scheme reorders the packets starting the transmission with the first packet in the block and continuing with successive packets in the block, skipping$i-1$ packets every step. Once the process reaches the block end, the cycle restarts with the second packet in the block, continuing in the same way. When all the packets of the block have been transmitted, the next block is processed. It is simple to show that this gives a maximum jitter of $(i-1)(b/i-1)$ packets, which will be removed at the receiver, by adopting a sufficiently long buffer. In our simulations we use $i=2$, $b=64$.

4. NETWORK EMULATOR

A network emulator has been developed for evaluating system performance in a given traffic condition. The emulator, that is depicted in Fig. 3, consists of two programs: a traffic generator and the network emulator itself that allows one to model a specific network configuration. Internal nodes may operate in First In First Out (FIFO) mode, or in Priority Weighted Fair Queuing (PWFQ) mode. The first mode can be used to emulate the most common operation mode of routers.

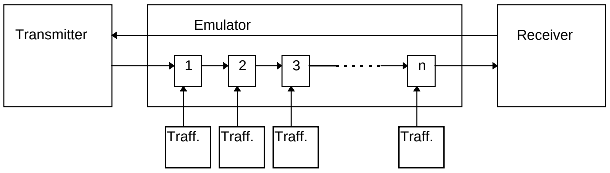

Figure 3: Traffic emulator

4.1 Traffic generator

In this paper a traffic generator based on a self-similar traffic model has been adopted. This model, which fits very well to typical patterns of traffic that can be found in Internet, is characterized by traffic bursts that can be found at any time scale, from milliseconds to hours. Self-similarity may be characterized by the parameter H (Hurst), defined as follows. Consider a sequence $\{Y_k\}_{k=1}^n$, with average value $\bar{Y}(n)$ and variance $S^2(n)$. The rescaled adjusted range statistic is given by $R(n)/S(n)$, where

$$\begin{aligned} R(n) &= \max\left\{\sum_{i=1}^{k}\left(Y_i-\bar{Y}(n)\right):1\leq k\leq n\right\}- \\ &\quad \min\left\{\sum_{i=1}^{k}\left(Y_i-\bar{Y}(n)\right):1\leq k\leq n\right\}. \end{aligned} \qquad (2)$$

It has been shown that, as $n\to\infty$, it is $E[R(n)/S(n)]\approx n^H$ for many measured sequences. In the Poisson traffic case it is$H=0.5$, while Internet exhibits a self-similar traffic with $H\approx 0.8$ [3].

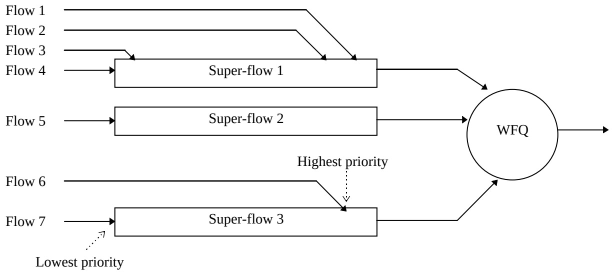

Figure 4: Queue model

For this reason, the aggregation of different flows of Internet traffic may be impulsive at a higher degree than a single flow.

4.2 Priority WFQ router

WFQ discipline has been proposed first in [13] and, independently, in [14], [15]. The algorithm implemented in the emulator is a modified version, for accommodating priority, of the algorithm in [16], named Self-Clocked Fair Queueing (SCFQ). SCFQ algorithm is slightly less complex and less fair in respect of the original WFQ algorithm. The algorithm operates as follows. In a WFQ router, each input data flow has an associated weight of Φ_k, and a relevant bandwidth share given by $\frac{\Phi_k}{\sum_j \Phi_j}$. A WFQ router has the following characteristics:

- it is work conservative: the server must be busy if there are packets waiting in the system;
- each data flow has a guaranteed bandwidth given by $\frac{\Phi_k}{\sum_j \Phi_j} r$, where r is the output rate.

In the priority based transmission scheme, the packets belonging to a given flow are ordered and transmitted according to their priority. Priority management is independent for every input flow (i.e. priorities of different super-flows are not comparable). More precisely, define *flow* a set of data with a given service specification (e.g. with a given priority), and *super-flow* a WFQ flow, consisting of a collections of different flows. Each flow is characterized by its super-flow k, and a priority level p, which specifies the relative importance of the flow within the super-flow it belongs to. Each super-flow is characterized by a label k, a weight Φ_k, and the set U_k, of the included flows. The queue model is shown in Fig. 4.

The PWFQ router has the following operation mode. Every packet is marked with a timing label, namely the *service tag*, that is used to determine the next packet to be transmitted. The packets in the queue are picked up for service in increasing order of the associated service tags. The service tag is evaluated according to the following steps.

1. When a new packet arrives, it is identified with the labels k, p of the super-flow, and the priority level it belongs to; this couple of numbers identifies the packet flow, too;
2. For each super-flow k, the service tags of the arriving packets are iteratively computed as
$$\hat{F}_k^i = L_k^i \frac{\sum_j \Phi_j}{\Phi_k} + max(\hat{F}_k^n, F_{\rm cur}), \qquad i = 1, 2, ... \qquad (3)$$
where $\hat{F}_k^0 = 0$, L_k^i is the packet length (the $i-th$ packet of the superflow k), $F_{\rm cur}$ is the packet currently in transmission, and n identifies the most recent packet (that is the packet with the highest service tag) of the same super-flow, that has an equal or higher priority of the i-th packet.

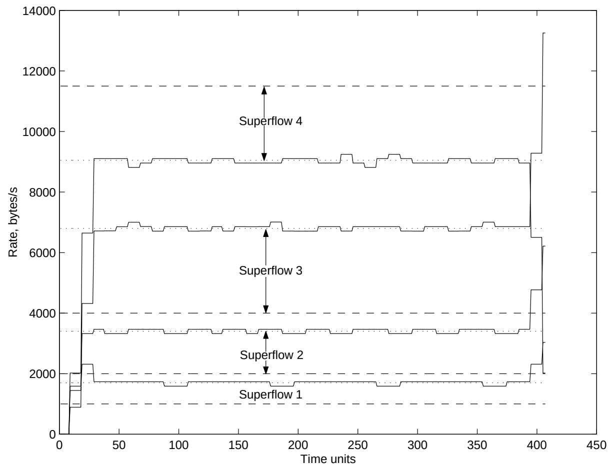

Figure 5: Throughput of superflows in a PWFQ system, for the case described in the text: guaranteed bandwidth (dashed lines), available bandwidth (super-flow 1, 2, and 3) or requested bandwidth (super-flow 4)(dotted lines), used bandwidth (solid lines).

3. Once a busy period is over and there are no more packets in the queue, the algorithm is reinitialized setting to zero $\hat{F}_k^i$ and the packets counts i for each super-flow k.
4. Moreover, every time a packet is received, all the lower priority packets belonging to the same super-flow increment their service tag by adding $L_k^i \frac{\sum_j \Phi_j}{\Phi_k}$.
5. Packets are re-ordered, according to their service tags, and packets exceeding queue capacity are discarded.

The proposed PWFQ algorithm extends the original SCFQ by adding the priority mechanism. If the priority is not used (there are only single flows in every super-flow), their behavior is exactly the same. The fairness of the SCFQ algorithm has been demonstrated in [16]. For completeness, some experimental results are given here.

Fig. 5 shows the behavior of the PWFQ algorithm in the most general situation. Assume that there are five superflows, each of which has an available input rate of about 9000 bytes/s. Four superflows consist of a single flow, and one super-flow consists of three flows, of different priorities. The total available bandwidth is 30 kbytes/s. The bandwidth weights of the five superflows are 10 (super-flow 1), 20 (super-flow 2), 40 (super-flow 3, the composite one), 115 and 115 (super-flow 4 and 5; only one of these two superflows is shown in the figure, for clarity). Thus, the guaranteed bandwidths are 1000, 2000, 4000 and 11500 bytes/s (dashed lines in the figure). Given that the last two superflows do not use all their available bandwidth, there remains some bandwidth that is distributed among the three superflows, proportionally to their weights (dotted lines). The solid lines represent the actual throughput. The fairness of the algorithm is manifest.

Assume that the composite super-flow consists of three flows, each of which has an input rate of 3000 bytes/s, and assume that the third flow has the lowest priority. Fig. 6 shows the performance of the composite super-flow. The allocated bandwidth is 40/70 * (30000 - 2 * 9000) = 6857 bytes/s. The two higher priority flows pass through completely (only one is shown, for clarity), while the third

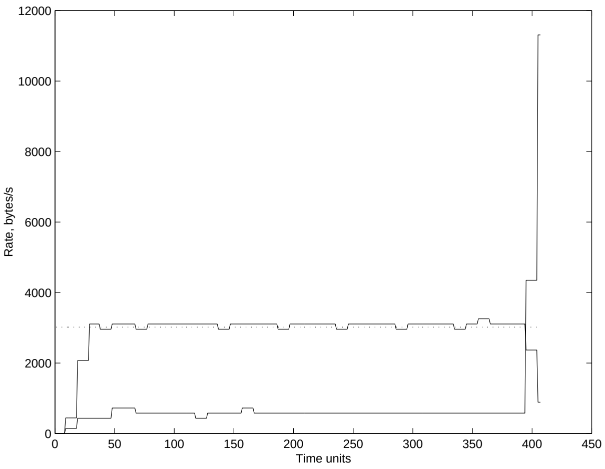

Figure 6: Throughput of two flows of the composite super-flow

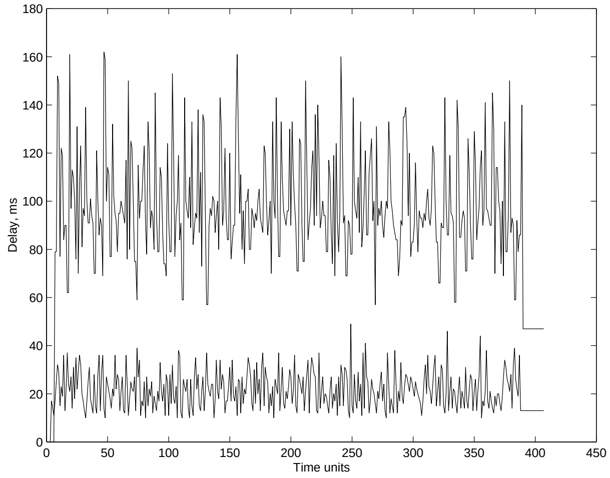

Figure 7: Delay of two different flows that are transmitted completely

one has a throughput of only ≈ 800 bytes/s, which is the remaining available bandwidth for the composite super-flow.

Fig. 7 shows the delay of two flows. The lower one is the delay of the single flow belonging to the super-flow 4. As shown before, the input rate of this flow is lower than the guaranteed rate. Its delay is, therefore, very low. The upper one is the second flow of the composite super-flow (the flow with intermediate priority). The flows that exceed their guaranteed (and therefore experience packet dropping) exhibit all much higher delays, which depend on the buffer length (not depicted in the figure).

Packet reordering ensures that the packets are serviced following a prioritized queuing discipline. This way, the delay of the high priority packets is kept low, being not affected by the low priority packets in the buffer. This improves the probability of timely delivering of high priority packets. As a consequence, some packets of the lowest priority can be delayed for a high amount of time. The delayed low priority packets may have some chances of being transmitted if the bandwidth becomes available within a reasonable time.

5. RESULTS

Performance is evaluated by starting five parallel music transmission sessions, one of which is monitored for quality measurement. Background traffic is added by using the traffic generator. Quality is determined for the following musical compositions.

- Antonio Vivaldi: da “Le quattro stagioni”, Concerto n.2 in sol minore, “L’estate: tempo impetuoso d’estate”,
- Heroes del Silencio: Entre dos tierras,
- John Lee Hooker: The Healer,
- Franz Liszt: Der blinde Sänger.

The pieces have been chosen to test different kinds of music (classical/rock, with/without voice) and of varying bandwidth and coding complexity (last two pieces). The first two pieces are encoded at a fixed rate of 64 kbit/s per second, while the other two ones are encoded at a variable rate, with an average value of 50 kbit/s per second for the third composition, and of 36 kbit/s for the fourth one. Three different transmission algorithms are compared using three different queuing disciplines, so that the following cases hold:

1. Basic, FIFO queue,
2. Interleaving, FIFO queue,
3. Congestion avoidance by using feedback, FIFO queue,
4. Basic, WFQ queue,
5. Interleaving, WFQ queue,
6. Congestion avoidance by using feedback, WFQ queue,
7. Basic, PWFQ queue,
8. Interleaving, PWFQ queue,
9. Congestion avoidance by using feedback, PWFQ queue.

Priority-based schemes adopt three priority levels. For each musical excerpt, and each network and traffic condition, 3 simulations are performed. The self-similar traffic is generated by using the value $H = 0.8$. 4 different network and traffic conditions are evaluated by varying the total average traffic and the WFQ weights Φ_i. In order to assign WFQ weights that are significant for a comparison with the FIFO discipline, not requiring weights, some assumptions about the number of active data sessions generating the self similar traffic, must be done: first, the bit rate per session is established, and then, knowing the total bit rate, the number of sessions is calculated. 5 sessions are audio sessions, identified by 5 different flows in the emulator, while the other are data sessions, grouped in the self similar traffic as a single emulated flow. The mean bit rate of the self similar traffic is imposed as the number of data sessions multiplied by the bit rate per session. The WFQ weights are assumed to be equal to the bit rate per session.

5.1 Objective Quality Measurements

For audio on demand and audio broadcasting applications, some usual performance parameters, such as delay and jitter, are of minor interest, because a long delay is not annoying for the user. The jitter can be eliminated simply by increasing the receiving buffer and the delay. On the contrary, audio distortion, which depends on packet dropping rate, is the main factor that affects the user perceived quality. Objective quality is evaluated by using two different distortion measures, selected among the wide amount of techniques proposed in the literature (for speech quality evaluation, see for instance [17]).

The selected measures operate in time domain:

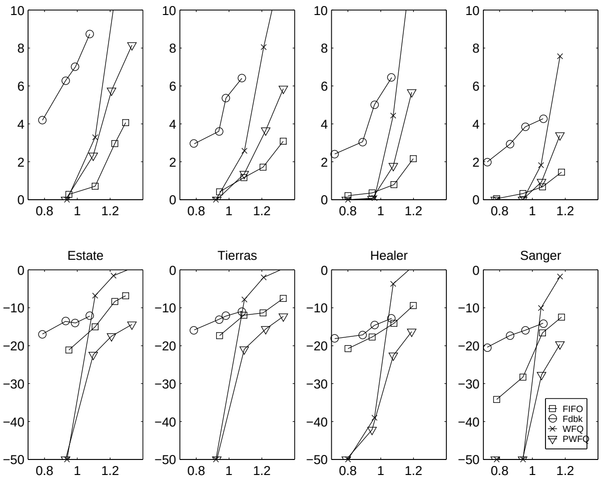

Figure 8: Signal to noise degradation and signal to noise ratio (with inverted sign for clarity) as a function of network traffic G

- The first technique evaluates the degradation of the segmental Signal to Noise ratio. For each frame, the decoded file is compared with the original one, to evaluate the difference, which is the noise. The obtained signal to noise ratio, expressed in dB, is compared with the signal to noise ratio that is obtained without any packet loss, and the difference is averaged over the whole file.

- The second technique evaluates the average signal to noise ratio, by comparing, for each frame, the received file and the file that is obtained after decoding, but without any packet loss. The resulting noise to signal ratio is averaged over the whole file, to determine the (linear) Average Segmental Signal to Noise Ratio.

Figure 8 shows both the segmental SNR degradation (upper part) and the average SNR (lower part) as a function of the normalized network traffic, G. Results are shown for a traffic range, such that packet loss rate may be significant. Only the four most significant transmission schemes are shown for brevity (FIFO basic, FIFO with feedback, WFQ basic and PWFQ basic).

All the measures are summarized in Table 2. According to measure B, a network with PWFQ routers offers better audio quality than FIFO and, moreover, it guarantees bandwidth fairness. According to measure A, only bandwidth fairness is achieved. WFQ alone offers the needed bandwidth for audio at lower loads only, while at higher loads WFQ is useless, because WFQ nodes would not give enough bandwidth to allow an acceptable audio transmission. Feedback may be used in a FIFO network to achieve fairness, but from the objective measurements PWFQ appears as offering a better performance than FIFO discipline with feedback. According to objective measure results, the utilization of interleaving does not lead to significant improvements.

5.2 Subjective Quality Measurements

Objective measurements allow one to perform a statistically significant number of tests. However, they do not provide a reliable distortion characterization from the user's viewpoint. Objective tests show that interleaving alone may be useless or even counterproductive, while priority based schemes may allow a significant reduction of the distortion introduced by packet dropping at high traffic rates. Subjective tests may allow a better evaluation of the distortion from the user's viewpoint [18]. To assess the correctness of previous conclusions, informal subjective tests are performed by employing 10 listeners. All the algorithms and queue disciplines under test are evaluated using two of the four musical pieces. For every combination two different traffic levels are employed, both of them resulting in a clearly audible distortion level. Table 3 summarizes the average traffic levels, and the relevant dropping rates. The dropping rates are shown for the non-feedback case only, because the feedback makes them negligible.

5	Perfect	Transparent
4	Good	Audible but non annoying distortion
3	Fair	Annoying distortion: acceptable for a limited period of time
2	Poor	Severe distortion
1	Unusable	Music intelligibility is compromised

Table 4: Subjective test rating scale

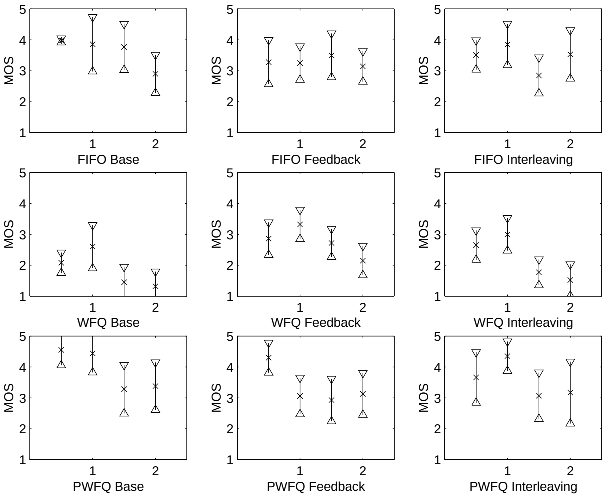

Figure 9: Subjective quality evaluation; ‘1’: Vivaldi, low traffic, ‘2’: Heroes del Silencio, low traffic, ‘3’: Vivaldi, high traffic, ‘4’: Heroes del Silencio, high traffic

During the test different files, corresponding to different algorithms, are reproduced by following a random order, and listeners are requested to give an opinion, among 5 five possible choices, as summarized in Table 4. Some choices result from some specific utilization of audio on demand service on Internet. For instance, a musical song with a poor quality may be still useful for a user that has to decide about a possible purchase.

Results of subjective tests are summarized in Fig. 9. The figure shows the score dispersion, highlighting some significant difference among the opinions of the listeners. However, some general conclusions may be derived from the results. From the subjective viewpoint, the spreading effect deriving from interleaving has controversial effects: it shows benefits when the WFQ queue is used, but it is harmful in the other cases. When priority is used, the interleaving may produce a continuously varying bandwidth effect, that may be more annoying than a bandwidth reduction. In all the circumstances, priority allows a significant improvement in the perceived quality.

6. COMMENTS AND CONCLUSIONS

	FIFO	WFQ	PWFQ
Base	1 Highly unfair, penalizes data traffic.	4 Higher distortion than FIFO at high loads (more lost packets due to fair bandwidth allocation), better quality at low loads (no lost packets when load $<$ 1).	7 Distortion measures do not agree; measure A favors dropouts (FIFO), measure B favors bandwidth reduction (PWFQ), better quality than FIFO at low loads (no lost packets when load $<$ 1); always better quality than simple WFQ.
Interleaving	2 Highly unfair, penalizes data traffic; slightly perceptible quality loss (to be verified with subjective tests).	5 Little or no difference compared to the non-interleaved case.	8 Little or no difference compared to the non-interleaved case.
Feedback	3 Lower quality, but bandwidth fairness is achieved; because total load is lower, there is a slightly perceptible improvement in quality at high data traffic loads, according to measure B (less packets are lost and therefore less dropouts experienced); the quality is always lower according to measure A.	6 Lower quality at low loads compared to the non feedback case (the congestion avoidance algorithm has a soft start); better quality at high data traffic loads (only measure B), because there are less lost packets. Better quality than FIFO at low loads, worse at high loads.	9 Useless. Lower quality than PWFQ/Base; lower quality than WFQ/Feedback al low data traffic loads, better at high (the lower quality is caused by the congestion avoidance algorithm, which is not tuned to priority transmission, where there are more flows with different delays).

Table 2: Objective quality measurements summary

Piece and traffic	G wo/ feedback	G w/ feedback	Dropping rate FIFO	D.r. (P)WFQ
Estate, low	1.04	0.90	0.02	0.08
Tierras, low	1.11	0.96	0.06	0.11
Estate, high	1.18	0.94	0.05	0.27
Tierras, high	1.23	0.99	0.07	0.33

Table 3: Average traffic G and relevant dropping rate for the subjective quality tests

This paper examines wide-band audio transmission utilizing a priority-enabled protocol. A scalable bit-rate audio encoder is utilized, and a set of priority, feedback and interleaving based transmission algorithms is defined and used for the performance evaluation. Three different queuing disciplines are considered. Objective and subjective quality tests are performed, using a set of musical excerpts. All the tests show that interleaving alone may be useless or even counterproductive, while priority based schemes may allow a significant reduction of the distortion introduced by packet dropping at high traffic rates.

Feedback-based mechanisms are useful when used in a network without priority queuing, because they may lower the packet loss rate and they may avoid congestion. A simple audio transmission scheme, without any feedback-based congestion control mechanism, may introduce severe performance degradation in a network adopting a FIFO discipline, because it could prevent other well behaved data traffic from transmitting. On the contrary, feedback is not necessary in a WFQ network, because the queuing mechanism guarantees fairness among different traffic sources. Nevertheless, it is useful in a WFQ network without priority, because it causes a reduction of packet loss. A PWFQ network does not require any further loss reduction or congestion avoidance mechanism.

The proposed technique does not require any additional protocol for the set-up. The WFQ is used for achieving fairness and not for performing reservations, so the application does not need to configure the routers along the path. The routers will identify different flows from their headers. Moreover the fairness provided by the PWFQ algorithm is useful for TCP traffic, too [12].

Given that PWFQ does not adopt any feedback mechanism, it does not suffer from the delayed effects of feedback-based techniques. The main drawback of PWFQ could be the computational complexity. However, it must be pointed out that it is not necessary to have PWFQ nodes everywhere throughout the path for allowing the system to work. The performance worsen only when some highly loaded nodes do not manage the priority, bringing the system to behave like a non priority system. Because of this, and because the system does not require any new protocol to be invented, the Internet can be upgraded gradually.

The final conclusion is that by adopting a scalable encoding technique, and a prioritized transmission algorithm, a smooth degradation of quality can be obtained during network congestion periods. The combination of three mechanisms (layered encoding, priority and fair queuing) has been proposed in the paper as a base for the transmission of multimedia data over the Internet and some practical results for the audio case have been presented, though the details can be changed. For instance, MAVT coding may be not the best choice and the presented PWFQ implementation is not computionally very efficient, even if results are encouraging. Better encoding schemes and a more efficient implementation will be object of future research.

7. ACKNOWLEDGMENTS

The authors want to thank CSELT S.p.A. for having supplied the MAVT encoder, Alessandro Cobalti for the self-similar traffic generator, and Dr. Francesca Vatta and Dr. Giancarlo Lombardi for many fruitful discussions.

8. REFERENCES

[1] W. Stallings, “IPv6: The New Internet Protocol”, IEEE Communication Magazine, vol. 34, No. 7, pp. 96-108, July 1996.

[2] C. Dorize, J-M. Muller, D. Sereno, “Detailed description of the MAVT audio candidate to MPEG-4”, ISO/IEC JTC1/SC29/WG11, MPEG95/0412, November 1995.

[3] W.E. Leland, M.S. Taqqu, W. Willinger, D.V. Wilson, “On the Self-Similar Nature of Ethernet Traffic (Extended Version)”,

IEEE/ACM Transactions on Networking, Vol. 2, No. 1, pp. 1-15, February 1994.

[4] S. McCanne, V. Jacobson, M. Vetterli, “Receiver-driven Layered Multicast”, ACM SIGCOMM ’96, August 1996.

[5] T. Turletti, S. F. Parisis, J. Bolot, “Experiments with a Layered Transmission Scheme over the Internet”, INRIA, Rapport de Recherche n^o 3296 - Novembre 1997.

[6] L. Vicisano, L. Rizzo, J. Crowcroft, “TCP-like congestion control for layered multicast data transfer”, UCL-CS Research Note RN/97/75, extended version of the INFOCOM’98 paper.

[7] J.C. Bolot, A.V.García, “Control Mechaninsms for Packet Audio in the Internet”, Proceedings of IEEE Infocom ’96.

[8] S. Bajaj, L. Breslau, S. Shenker, ”Uniform versus Priority Dropping for Layered Video”, SIGCOMM ’98, Computer Communications Review, October 1998

[9] J. C. R. Bennett, H. Zhang, “Hierarchical Packet Fair Queuing Algorithm”, IEEE/ACM Transactions on networking, Vol. 5., No. 5, pp. 675-689, October 1997.

[10] P. Goyal, H. M. Vin, H. Cheng, “Start-time Fair Queuing: A Scheduling Algorithm for Integrated Services Packet Switching Networks”, IEEE/ACM Transactions on Networking, Vol. 5, No. 5, pp. 690-704, October 1997.

[11] D. Stiliadis, A. Varma, “Efficient Fair Queuing Algorithms for Packet-Switched Networks”, IEEE/ACM Transactions on Networking, Vol. 6, No. 2, pp. 164-174, April 1998.

[12] I. Stoica, S. Shenker, H. Zhang, ”Core-Stateless Fair Queuing: Achieving Approximately Fair Bandwidth Allocations in High Speed Networks”, SIGCOMM ’98, Computer Communications Review, October 1998

[13] A. Demers, S. Shenker, S. Keshav , “Analysis and Simulation of a Fair Queuing Algorithm””, Internet. Res. And Exper., vol. 1, 1990.

[14] A.K. Parekh, R.G. Gallager, “A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case”, IEEE/ACM Transactions on Networking, Vol. 1, No. 3, pp. 344-357, June 1993.

[15] A.K. Parekh, R.G. Gallager, “A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case”, IEEE/ACM Transactions on Networking, Vol. 2, No. 2, pp. 137-150, April 1994.

[16] S. J. Golestani, “A self-clocked fair queueing scheme for high speed applications”, in Proc. INFOCOM ’94, Apr. 1994.

[17] S. Dimolitsas, “Objective speech distortion measures and their relevance to speech quality assessments”, IEEE Proceedings, Vol. 136, Pt. I, No. 5, October 1989.

[18] D. Meares, K. Watanabe, E. Scheirer, “Reports on the MPEG-2 AAC Stereo Verification Tests” ISO/IEC JTC1/SC29/WG11 N2006, February 1998.