Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT•

ELSEVIER

Computer Communications 26 (2003) 1240–1254

computer communications

www.elsevier.com/locate/comcom

The issue of useless packet transmission for multimedia over the Internet

Jim Wu*, Mahbub Hassan

School of Computer Science and Engineering, The University of New South Wales, ANZAC PDE, Sydney 2052, Australia

Received 1 May 2002; revised 30 October 2002; accepted 7 November 2002

Abstract

When packet loss rate exceeds a given threshold, received audio and video become unintelligible. A congested router transmitting multimedia packets, while inflicting a packet loss rate beyond a given threshold, effectively transmits useless packets. Useless packet transmission wastes router bandwidth when it is needed most. We propose an algorithm to avoid transmission of useless multimedia packets, and allocate the recovered bandwidth to competing TCP flows. We show that the proposed algorithm can be easily implemented in well-known WFQ and CSFQ fair packet queueing and discarding algorithms. Simulation of a 15-s MPEG-2 video clip over a congested network shows that the proposed algorithm effectively eliminates useless packet transmission, and as a result of that significantly improve throughput and file download times of concurrent TCP connections. For the simulated network, file download time is reduced by 55% for typical HTML files, 36% for typical image files, and up to 30% for typical video files. A peak-signal-to-noise-ratio (PSNR) based analysis shows that the overall intelligibility of the received video is no worse than that received without the incorporation of the proposed useless packet transmission avoidance algorithm. Our fairness analysis confirms that implementation of our algorithm into the fair algorithms (WFQ and CSFQ) does not have any adverse effect on the fairness performance of the algorithms.

Keywords: TCP; Multimedia over IP; MPEG-2; Internet; Fair packet queueing algorithm

1. Introduction

Internet voice and video applications, such as IP telephony and video streaming, continue to gain popularity. A direct consequence is the increased congestion in the Internet routers. In the routers, multimedia packets, i.e. IP packets carrying voice or video traffic, compete head to head with traditional data packets for link bandwidth. Two important issues that need to be addressed when multimedia traffic is multiplexed with data traffic are: (i) fairness, and (ii) useless packet transmissions (UPT). The fairness issue arises due to the fact that the data applications use TCP, which cuts back the transmission rate when packets are discarded at the router due to congestion, but most multimedia applications, use UDP, which does not cut back transmission rates. As a result, multimedia flows unfairly gets more bandwidth than the congestion-responsive TCP flows. The fairness problem in the Internet is now well recognised. Several packet queueing and discarding algorithms, such as WFQ [1], CSFQ [2] and FRED [3], have been proposed in the last few years to effectively address the issue of fairness. Performance results confirm that these algorithms can fairly distribute link bandwidth among competing multimedia and data flows. Some router vendors have already started incorporating these algorithms in their latest products [4].

The issue of UPT, however, is less understood. UPT is based on the fact that for packetised audio and video, packet loss rate must be maintained under a given threshold for any meaningful communication [5–7]. When packet loss rate exceeds this threshold, received audio and video become useless. Thus a router transmitting multimedia packets at a fair rate (using WFQ for example), while inflicting a packet loss rate beyond the threshold actually transmits useless packets. These packets are useless, because they do not contribute to any meaningful communication. UPT effectively reduces the available bandwidth to competing TCP flows, which in turn increases file download times. Longer download times increase power consumption of battery-powered devices, and hence, have direct impact on mobile computing.

The UPT problem will be more significant in future due to the following trends: (i) rising popularity of voice and

* Corresponding author. Tel.: +61-2-9385-4358; fax: +61-2-9385-5995.
E-mail addresses: jimw@cse.unsw.edu.au (J. Wu), mahbub@cse.unsw.edu.au (M. Hassan).

0140-3664/03/$ - see front matter
PII: S0140-3664(02)00276-1

video applications, (ii) increasing deployment of fair packet queueing algorithms, and (iii) rapid proliferation of battery-powered mobile devices. Hence, there is a need to investigate mechanisms to effectively address the UPT issue in fair packet queueing algorithms. In this paper, we propose a UPT avoidance algorithm (we call it UPTA) that can be easily implemented in existing fair packet queueing and discarding algorithms. We have evaluated its performance using simulation of an MPEG-2 video stream competing with a TCP connection in a congested router. Simulation results show that with the proposed UPTA algorithm in place, we can effectively eliminate UPT, and significantly reduce file download times without deteriorating the perceived quality of the video stream.

The remainder of the article is organised as follows. We discuss related work in Section 2. Section 3 formally defines the UPT problem and derives the performance parameters. In Section 4, we propose an UPTA algorithm and show its incorporation in two well-known fair packet queueing algorithms, WFQ and CSFQ. In Section 5, we derive the packet loss rate threshold for intelligible communication for the MPEG-2 video clip used in the simulation. The simulation experiment is presented in Section 6, followed by the results in Section 7. Finally, we present our conclusions and discuss open issues in Section 8.

2. Related work

How to support multimedia over the Internet is a topic of intense research. In this section, we discuss the related work.

2.1. Adaptive multimedia

Although traditionally multimedia applications worked with a single bit rate, the trend is towards a multi-rate platform. These new applications can operate in several bit rates supporting different levels of QoS. The higher the bit rate, the better the quality of the audio or video. Multi-rate applications are capable of dynamically switching to a different rate upon receiving feedback of the current network condition. Adaptive multimedia [15] refers to multimedia applications adapting their rates according to network congestion.

RTCP [16] is a new protocol defined by IETF to assist multimedia applications to learn about network conditions, such as current packet loss rate and network delay. To achieve adaptive multimedia, an RTCP connection is set up between a receiver and a sender. The receiver periodically sends network statistics to the sender and the sender employs a control algorithm to adapt the rates of the multimedia according to the statistics received. TCP Friendly [17] is one such control algorithm proposed to adapt multimedia rates in a fashion similar to the TCP's Additive Increase Multiplicative Decrease (AIMD [18]) control algorithm.

Adaptive multimedia helps addressing the fairness issue when multimedia and TCP applications compete for common resources. However, if severe congestion persists, the network may still drop too many packets from a multi-rate multimedia flow even if the flow is operating at its lowest possible rate. Therefore, even with adaptive multimedia, UPT can still occur.

2.2. QoS in the Internet

IETF is working on two new QoS frameworks, namely Diffserv [19] and Intserv [20], to support quality of service in the Internet. The idea of QoS in IP networks is similar to the one in ATM networks. The main premise is to introduce more than one class of service in the Internet (currently *best-effort* is the only service available) and eventually guarantee the required QoS of all multimedia flows. However, a multi-service framework requires a significant upgrade of the existing Internet infrastructure. Given the scale of the existing user base, this type of upgrade is unlikely to take place in the near future.

Even in the future Internet with multiple service classes, a large number of users will continue to use the best-effort service for many multimedia applications, especially those not so business critical. There are at least two compelling reasons for not using the guaranteed or higher priority service classes. Firstly, these classes will attract much higher charges compared to the best effort service. Secondly, these users have already established a 'multimedia-over-best-effort' culture, and will be happy to continue using the same systems.

2.3. Forward error correction and loss concealment

Forward Error Correction (FEC) [21,22] and loss concealment techniques [23,24] can repair some of the damages in the received multimedia caused by lost packets. The net effect of employing such FEC and loss concealment techniques is to push the packet loss rate threshold for effective communication to a higher limit (more packet loss can now be tolerated). However, the fact that audio and video become unintelligible beyond a certain packet loss rate remains valid. UPT, therefore, remains a significant issue for multimedia over best-effort networks.

2.4. Partial packet discard in ATM networks

Useless cell transmission was considered in the context of IP packet transmission over ATM networks. One IP packet is usually transmitted over multiple ATM cells (due to very small cell size). Once an ATM switch drops an ATM cell due to congestion, the rest of the cells of the same IP packet become useless, as they will be all discarded at the destination (reassembly of the IP packet is not possible without receiving all cells). A technique called Partial Packet Discard (PPD) [25] was considered to drop all remaining cells of a packet as soon as one cell from the packet is dropped. Although this technique required ATM switches to detect IP packet boundaries, it was welcomed because of its potential to reduce bandwidth waste in ATM switches. Early Packet Discard (EPD) [26], a more advanced version of PPD, is now widely used in commercial ATM switches.

3. The UPT problem

This section details the UPT problem and formally defines the performance parameters. We make the following assumptions and observations for multimedia over best-effort network service:

- When packet loss rate exceeds a threshold q, the media, particularly voice and video, becomes unintelligible [5–7]. The exact value of this threshold may vary from application to application. For the video clip used in our experiment, we experimentally derive this threshold (see Section 5). As long as the packet loss rate remains below the threshold, the media is intelligible.
- If a multimedia connection is established over a best-effort network service (i.e. no guarantees on packet loss rate), packet loss rate can occasionally exceed the threshold. Therefore, the life-time of a multimedia connection can be described by a series of alternating intelligible and unintelligible intervals. In the best case, there will be no unintelligible intervals, the entire connection time makes up a single intelligible interval. Similarly, in the worst case, the entire connection time is composed of a single unintelligible interval.

We use the following notations:

- U: Denotes an unintelligible interval.
- I: Denotes an intelligible interval.
- Δ_U : Mean length of U intervals.
- Δ_I : Mean length of I intervals.
- η: Mean throughput of U intervals in bits per second (bps).

We define *unintelligible ratio*, i.e. the fraction of time a multimedia connection spends in U intervals, as:

$$\phi = \frac{\Delta_U}{\Delta_U + \Delta_I} \tag{1}$$

Note that in the best case, $\phi = 0$ (because $\Delta_U = 0$) and in the worst case $\phi = 1$ (because $\Delta_I = 0$). In other cases, $0 < \phi < 1$.

UPT is defined as transmission of packets during U intervals. These packets do not contribute to any meaningful communication, and hence, simply waste router bandwidth. The amount of bandwidth wasted by UPT can be obtained as:

$$\omega = \eta \frac{\Delta_U}{\Delta_U + \Delta_I} \text{bps} = \eta\phi \text{ bps} \tag{2}$$

For example, if a video connection has spent 20% of the total connection time in U intervals, with an $\eta = 1$ (this can happen for a 1.5 Mbps MPEG-2 video that becomes unintelligible when only 1 Mbps is allocated), then it has wasted 200 Kbps router bandwidth due to UPT.

4. UPT avoidance algorithm

There are two basic ways to avoid UPT. One way is to eliminate all U intervals; the other is simply to remove UPT from the U intervals. The former approach, which needs to reserve bandwidth for multimedia connections, is pursued by the guaranteed quality of service (QoS) efforts (see related work in Section 2). However, for best-effort network services, it is acceptable to expect occasional U intervals in multimedia connections. We propose an UPT avoidance algorithm (henceforth called UPTA), which aims to remove UPT from the U intervals without affecting the unintelligible ratio (ϕ) of the connection. Effectively, UPTA will reduce bandwidth waste by multimedia flows, and distribute the recovered bandwidth to competing TCP flows.

Fig. 1 illustrates the flow chart of the proposed UPTA algorithm. UPTA drops an arriving packet straightaway, if the current fairshare dictates a packet loss rate exceeding the threshold. Otherwise, UPTA has no effect on the usual processing of the packet. Some fair algorithms (e.g. CSFQ) compute a drop probability (current packet loss rate) for each arriving packet as part of the algorithm. Incorporation of UPTA in such algorithms would be straightforward. However, if a fair algorithm (e.g. WFQ) does not explicitly compute drop probability, UPTA will have to do that. Fig. 2 shows the implementation of UPTA in CSFQ. Since CSFQ already computes the drop probability of an arriving packet, UPTA simply compares this drop probability with threshold q and discards the packet if it is greater than q. Otherwise, it lets CSFQ to drop the packet probabilistically (usual CSFQ processing).

The implementation of UPTA in WFQ is shown in Fig. 3. WFQ has a built-in classifier that classifies arriving packets

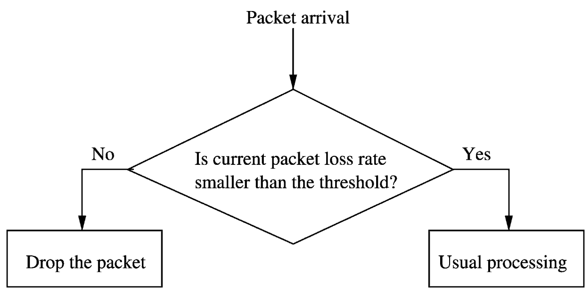


Fig. 1. Flow chart of UPTA.

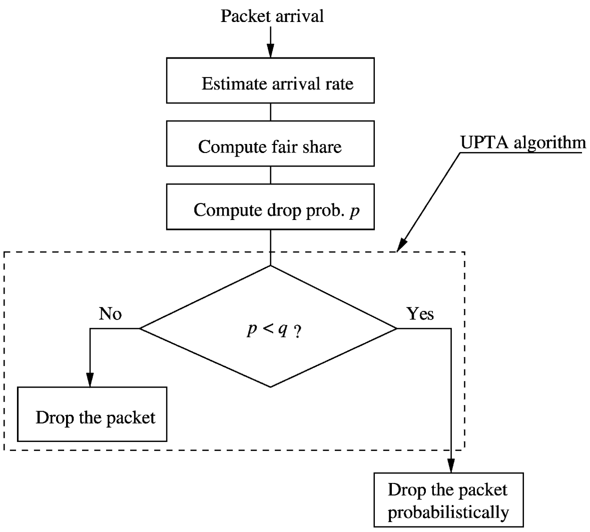


Fig. 2. Implementation of UPTA in CSFQ.

into flows. Each flow now goes through an UPTA controller. The UPTA controller estimates the arrival rate of the flow, computes the fair share, and based on the fair share, computes a drop probability. UPTA controller simply drops the packet if the drop probability is greater than threshold q, otherwise the packet is processed by WFQ (as per the flow chart shown in Fig. 1). CSFQ and WFQ implementations of UPTA (Figs. 2 and 3) are later used in the simulation experiments described in Section 6.

5. Packet loss threshold for the experimental video

Packet loss rate threshold q for intelligible communication may vary from application to application. We have carried out a series of experiments with different loss rates in the network to determine the threshold for the MPEG-2 video clip used in our simulation study. This section explains the experiment with a brief overview of the standards used for transporting MPEG-2 video over IP.

5.1. Overview of MPEG-2 over IP

MPEG-2 standard specifies two systems. The first simply multiplexes video, audio, and data of a single program together to form a program stream (PS). Program streams are intended for recording applications such as DVD. The other system defines transport stream (TS), a packet-based protocol for transmission applications such as cable TV, video on demand, interactive games, etc. TS packets have a fixed length of 188 bytes, including a minimum 4-byte header. For practical purposes, the continuous MPEG-2 bitstream, also called elementary stream (ES), from source encoder is broken into 18,800-byte segments [8]. These video segments are usually called packetised elementary stream (PES) packets. PES packets can be used to create program streams or transport streams. The Internet Engineering Task Force (IETF) has defined the protocol structure for delivery of MPEG-2 video over IP networks as illustrated in Fig. 4 [9]. In all our experiments and simulations, we use this protocol structure.

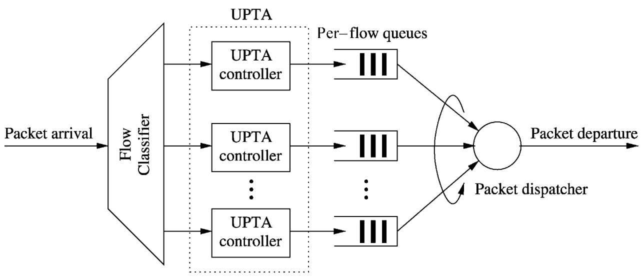


Fig. 3. Implementation of UPTA in WFQ.

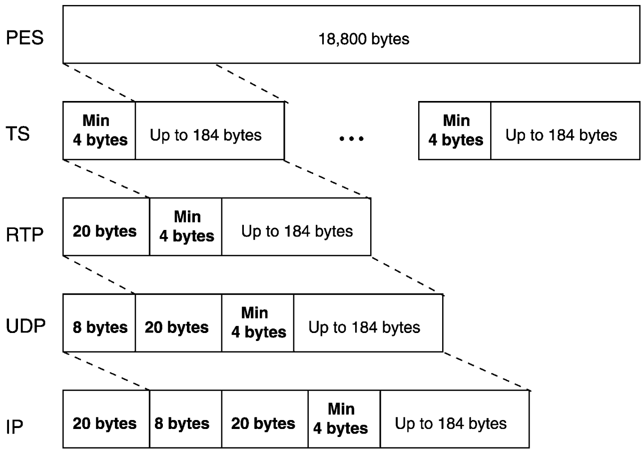


Fig. 4. Protocol structure for MPEG-2 over IP.

5.2. Loss threshold determination

The video stream we used in our simulations (see Sections 6 and 7 is an MPEG-2 test elementary stream called `susi_015.m2v` (provided by Tektronix [10]). Parameters of the video stream are given in Table 1. The experimental system used to determine the packet loss rate threshold for `susi_015.m2v` is shown in Fig. 5.

The system has five main components: system encoder, packet dropper, system decoder, source decoder, and PSNR calculator. A system encoder divides the original MPEG-2 elementary stream into a series of TS packets and sends them to the packet dropper. The packet dropper, which simulates a lossy channel, randomly drops arriving TS packets with a probability p $(0 < p < 1)$. System decoder reassembles TS packets into MPEG-2 elementary stream

Table 1
Parameters of MPEG-2 test stream

Parameters	Duration (s)	Bit rate (Mbps)	Frame rate (fps)	Frame size (pixels)
Values	15	1.5	25	352×288

Parameters	Duration (seconds)	Bit Rate (Mbps)	Frame Rate (fps)	Frame Size (pixels)
Values	15	1.5	25	352 x 288

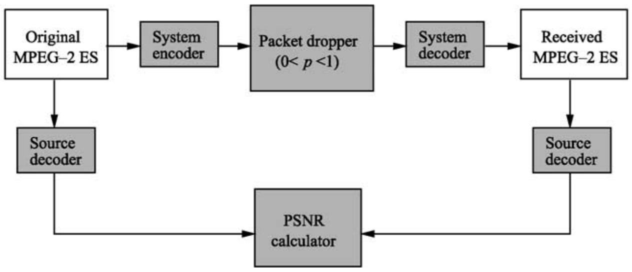


Fig. 5. Block diagram of intelligibility analysis system.

(received MPEG-2 ES). Source decoder decodes elementary streams into series of video frames. PSNR calculator compares a received frame with the corresponding original frame, and calculates PSNR for the frame as [11]:

$$PSNR = 10\log_{10}\left(\frac{P^2}{e}\right) \tag{3}$$

where P is the maximum value for a pixel, and e is the Mean Square Error (MSE) between two frames. For pictures employing 8-bit colour, $P = 255$. Given frame size of $M \times N$ pixels, the MSE is expressed as:

$$e = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(p_{(i,j)} - p'_{(i,j)})^2 \tag{4}$$

where $p_{(i,j)}$ represents value for pixel (i,j) in the original frame, and $p'_{(i,j)}$ represents value for pixel (i,j) in the corresponding received frame. We implemented the PSNR calculator using MATLAB, and the packet dropper using C. For the system encoder/decoder and the source decoder, we used the MPEG-2 codec developed by MPEG Software Simulation Group (MSSG) [12].

We did several experiments by varying p from 0 to 25%. For simplicity, we did not employ error recovery or error concealment in our system; the payload of a lost video packet is replaced by all 0s. Fig. 6 shows a decoded frame, along with its PSNR, under different loss rates.

As expected, the video quality degrades with the increase of packet loss rate. In the first three experiments ($p = 0.05, 0.10, 0.12$), the video is still intelligible. In the later three experiments ($p = 0.15, 0.20, 0.25$), the video quality is too poor to be intelligible. Therefore, we conclude that for this video stream, the video quality is unintelligible when packet loss rate exceeds 12% (similar results were reported in Ref. [13]). This packet loss rate threshold will be used by our UPTA algorithm in Section 6 to detect U intervals and avoid useless packet transmission in the router. We also note, that for this video, the frames become unintelligible for PSNR less than 65 dB. This

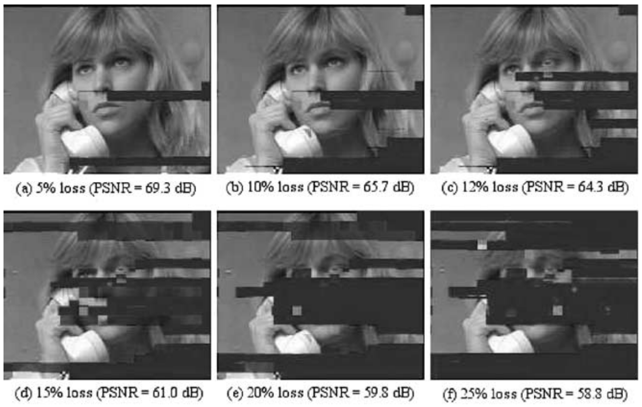


Fig. 6. PSNR under different packet loss rates.

PSNR threshold will be used in Section 7 to assess the impact of our UPTA algorithm on the overall intelligibility of the received video.

6. Simulation

This section details the network model and the performance metrics used in the simulation study.

6.1. OPNET model

Using OPNET Modeler 7.0.B [14], we simulate a well-known single bottleneck network topology ('dumbbell') as shown in Fig. 7. This topology simulates an intranet over two long-distance sites interconnected by a (virtual) leased line. Three workstations (simulating senders) are connected to Router1 through 10 Mbps access links. Their corresponding receivers are connected to Router2, also using 10 Mbps links. Router1 is the congested router; it has a buffer size of 100 packets. The two routers are connected by a 4 Mbps link (e.g. $2 \times E1$ lines), with a propagation delay of 1 ms. All other links have a propagation delay of 1 μs.

Among the three senders, there are a TCP source, a video source, and a ON–OFF source. TCP source simulates file transfers. The video source simulates the transmission of

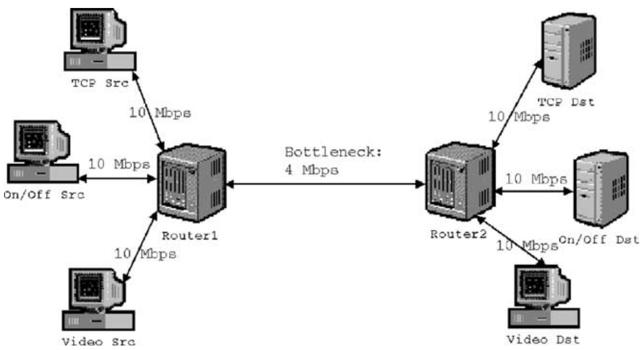


Fig. 7. OPNET simulation model.

the MPEG-2 video clip (`susi_015.m2v`) described in Section 5. The video source, which operates at 1.5 Mbps, employs a packet size of 188 bytes (MPEG-2 TS packet size); the TCP and the ON–OFF sources use 512-byte packets. The ON–OFF source transmits at 2 Mbps during ON period, and stops transmitting in OFF periods. The ON–OFF source simulates the dynamics of background traffic over the congested link.

We induce and control the length of U intervals in the video connection by controlling the ON–OFF source as follows. The MPEG-2 video is encoded at a bit rate of 1.5 Mbps. Taking TS header (4 bytes), UDP header (8 bytes), and IP header (20 bytes) into account, the total bandwidth required by the video application is about 1.76 Mbps. When the ON–OFF source is switched off, the fair share of the bottleneck link is 2 Mbps (there are only two flows); the video connection does not suffer any packet loss. However, when the ON–OFF source is turned on, the fair share drops to about 1.33 Mbps (there are three flows), and the video connection suffers a packet loss rate beyond the threshold. Therefore, by turning the ON–OFF source on, we can effectively induce a U interval in the video connection. For these simulated U intervals, we get $\eta = 1.33$ Mbps.

The length of the simulation is 15 s, which corresponds to the length of the video clip. To simulate different levels of UPT, we varied the length of the ON period of the background traffic, resulting in five different simulation scenarios. For scenarios 1–5, the ON period was set to 3, 6, 9, 12, and 15 s, respectively (starting at 0 s). Therefore, in each scenario, we have one U followed by one I interval in the video connection, where Δ_U is simply the length of the ON period, and Δ_I is obtained as $15 - \Delta_U$. Using Eqs. (1) and (2), we can compute the theoretical unintelligible ratios and bandwidth wastes for different scenarios (see Table 2).

6.2. Performance metrics

We measure the following performance metrics:

- *TCP throughput*: For TCP connections, we measure the throughput at the destination as bytes received per second. This measure includes all bytes received at the destination, including the packet header bytes and the retransmitted bytes. We also measure TCP *goodput* that excludes the overhead and retransmitted bytes. Throughput and goodput are measured for persistent TCP connections, connections that do not terminate before the end of simulation.
- *File download time*: For non-persistent TCP connections, i.e. connections that complete before the end of simulation, file download time represents the total connection time. Smaller the file size, shorter the file download time.
- *Video intelligibility*: To measure the level of intelligibility of the received video, we define an intelligibility index as:

$$\text{intelligibilityindex} = \frac{N_{in}}{N_{tx}} \tag{5}$$

where N_{in} is the total number of intelligible video frames received, and N_{tx} is the total number of video frames transmitted. A received video frame is considered intelligible if it has a PSNR greater than or equal to 65 dB (see Section 5). Intelligibility index is a dimensionless fraction between 0 and 1. Closer the value to 1, the higher the level of intelligibility. A value of 1 means all received frames are intelligible. We use this index to measure the impact of our proposed UPT avoidance algorithm on the quality (in terms of overall intelligibility) of the received video.

- *Throughput fairness*: We measure throughput fairness among competing flows using the well-known Jain's fairness index as:

$$F = \frac{\left(\sum_{i=1}^{N} x_i\right)^2}{N \sum_{i=1}^{N} x_i^2} \tag{6}$$

where x_i is the normalised throughput of flow i, and N is the total number of competing flows. Normalised throughput refers to the ratio of actual measured throughput to the maximum throughput possible in a system.

Table 2
Levels of bandwidth waste for different simulation scenarios

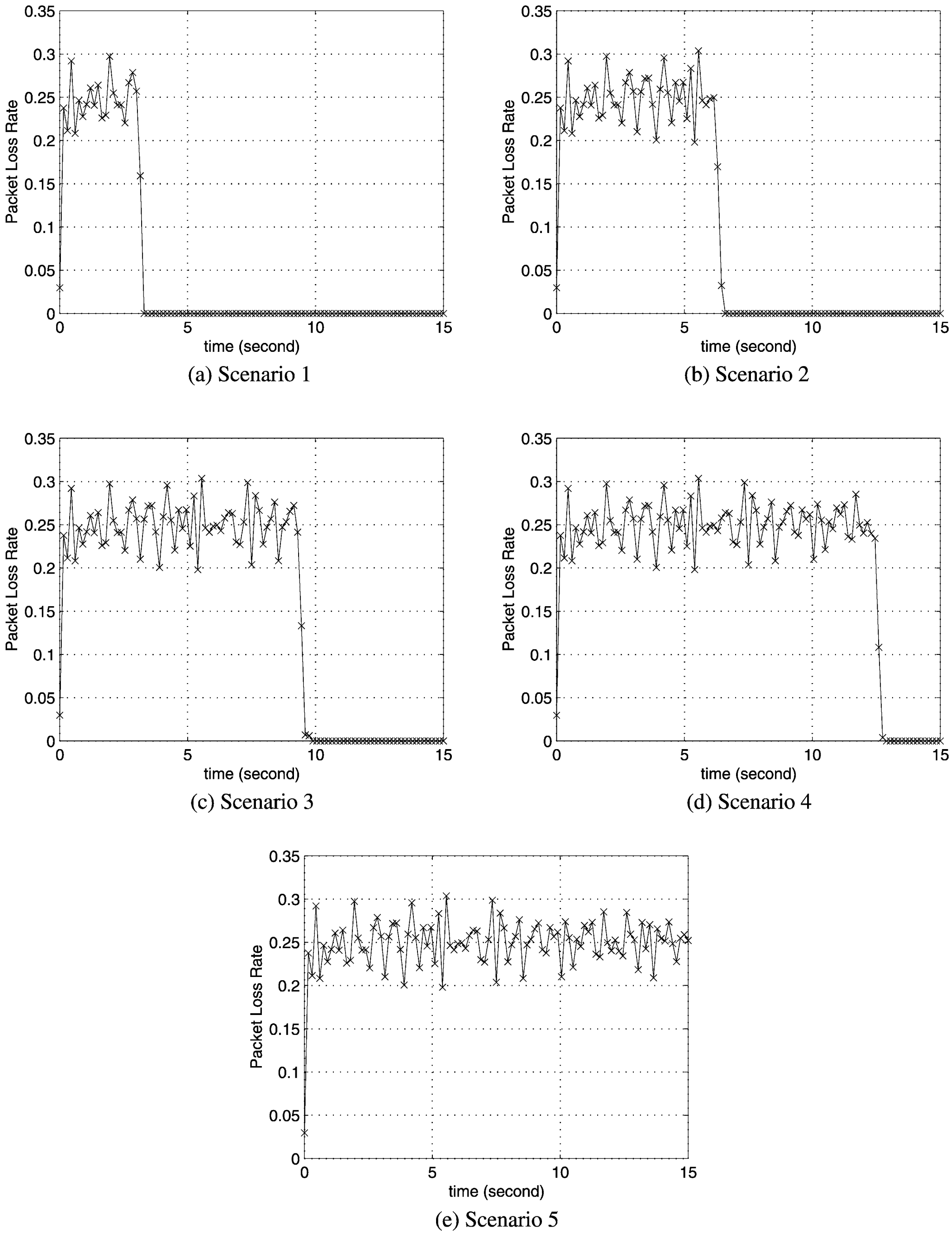
Scenario	Unintelligible ratio (ϕ)	Bandwidth waste in Mbps (ω)
1	0.2	0.266
2	0.4	0.532
3	0.6	0.798
4	0.8	1.064
5	1.0	1.333

7. Results

In this section, we present the results obtained from the simulation experiments.

7.1. TCP throughput

To assess the effect of UPTA on TCP throughput, we have simulated a persistent TCP connection that continues to transfer data for the full duration (15 s) of the simulation. In each of the five scenarios, we have a U interval (ON–OFF source is turned on) followed by an I interval (ON–OFF source is switched off) in the video connection. Fig. 8 shows packet loss rate of the video connection under WFQ in five different scenarios (similar results for CSFQ are



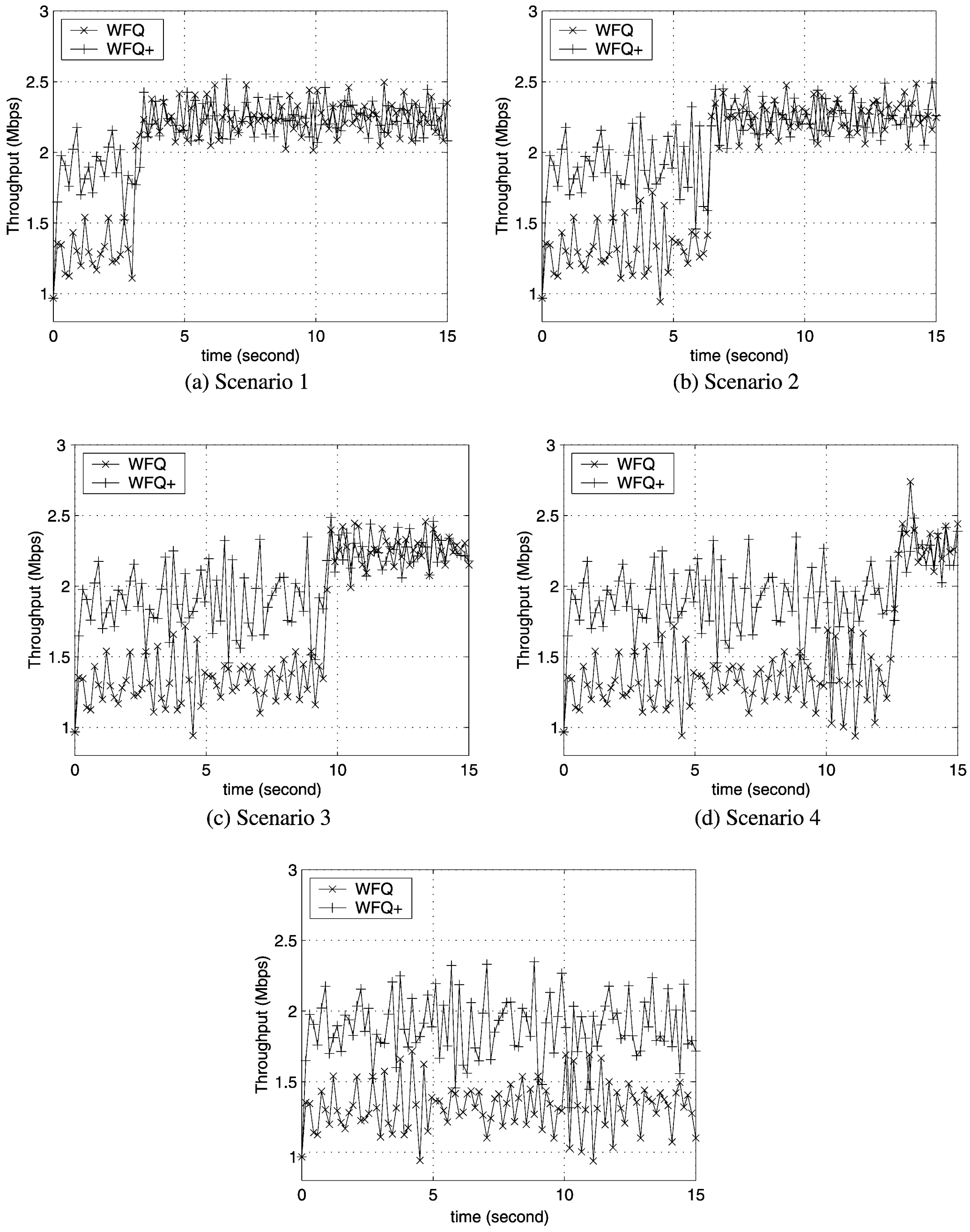
(a) Scenario 1 (b) Scenario 2 (c) Scenario 3 (d) Scenario 4 (e) Scenario 5

Fig. 8. Packet loss rate of video flow for WFQ.

given in Appendix A). The figure shows that the video connection suffers a packet loss rate of around 25% in *U* intervals, and no packet loss in *I* intervals.

To analyse TCP throughput dynamics in different intervals, we have used OPNET's 'bucket' option to plot throughput in consecutive buckets of 100 packets received at the destination. This short-term throughput analysis allows us to observe any variation of TCP throughput over short intervals. Fig. 9 shows short-term TCP throughput received under WFQ for all five scenarios. It is evident, that with our proposed UPTA in place, we receive higher TCP throughput during the *U* interval. The proposed UPTA effectively recovers bandwidth that would have been otherwise wasted in UPT, and distributes the recovered bandwidth to other competing connections. We can also see that UPTA has no effect on TCP throughput during the *I* interval, substantiating the fact that our UPTA can successfully switch to 'normal' processing mode when there is no UPT in the system. We have received similar results for CSFQ (see Appendix A).

Table 3 shows the average TCP throughput achieved over the entire duration (15 s) of the video connection. In



(a) Scenario 1
(b) Scenario 2
(c) Scenario 3
(d) Scenario 4
(e) Scenario 5

Fig. 9. Short-term TCP throughputs for WFQ, with and without UPTA.

this table, CSFQ + and WFQ + represent incorporation of our proposed UPTA in these fair algorithms. A quick glance through the table reveals that incorporation of UPTA increases TCP throughput in all scenarios. The magnitude of throughput increase is a direct function of the amount of bandwidth waste incurred in each scenario. For different scenarios, Table 2 shows the theoretical values for bandwidth waste (ω). Since there are two other competing connections (TCP and ON–OFF) during the U interval, UPTA would distribute ω equally between them.

Table 3
Comparison of TCP throughputs with and without UPTA

Scenario	TCP throughput (Mbps)			
	CSFQ	CSFQ +	WFQ	WFQ +
1	1.79	1.93	1.86	1.99
2	1.69	1.92	1.75	1.97
3	1.54	1.90	1.60	1.95
4	1.40	1.88	1.46	1.93
5	1.25	1.86	1.33	1.91

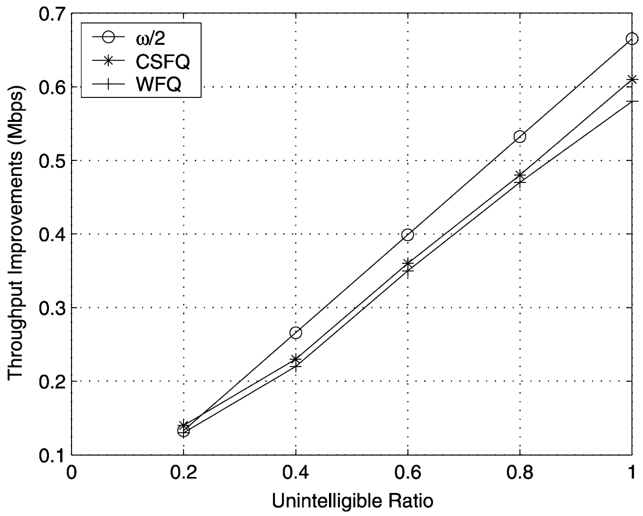


Fig. 10. TCP throughput improvement with UPTA.

Therefore, the ideal TCP throughput improvement would be $\omega/2$. Fig. 10 shows TCP throughput improvement with UPTA against CSFQ and WFQ, under various unintelligible ratios. We can see that throughput improvement achieved by UPTA is very close to the ideal values, for both CSFQ and WFQ implementations.

Finally, Table 4 shows application level throughput improvement (excluding TCP/IP header and any retransmissions) achieved with UPTA. In consistence with TCP throughput, we can see that UPTA increases application throughput under all five scenarios, for both CSFQ and WFQ.

7.2. File download time

To observe the impact of UPTA on file download time, we have simulated file transfers (between TCP source and destination) of three different file sizes, each representing a different type of object on the Web. A 15 KB file represents HTML Web pages; a 150 KB file represents compressed image objects; and finally, a 1.5 MB file represents compressed video objects or clips [5]. Table 5 shows the improvements in file download times achieved with our proposed UPTA under scenario 1.

Table 4
Comparison of application throughputs with and without UPTA

Scenario	Application throughput (Mbps)			
	CSFQ	CSFQ +	WFQ	WFQ +
1	1.51	1.61	1.54	1.77
2	1.34	1.52	1.49	1.65
3	1.18	1.51	1.33	1.62
4	1.02	1.48	1.17	1.59
5	0.94	1.38	1.13	1.58

Table 5
Comparison of download time with and without UPTA (Scenario 1)

File size	Download time (s)			
	CSFQ	CSFQ +	WFQ	WFQ +
15 KB	0.23	0.14	0.18	0.08
150 KB	1.62	0.96	1.29	0.82
1.5 MB	8.85	7.93	8.31	7.05

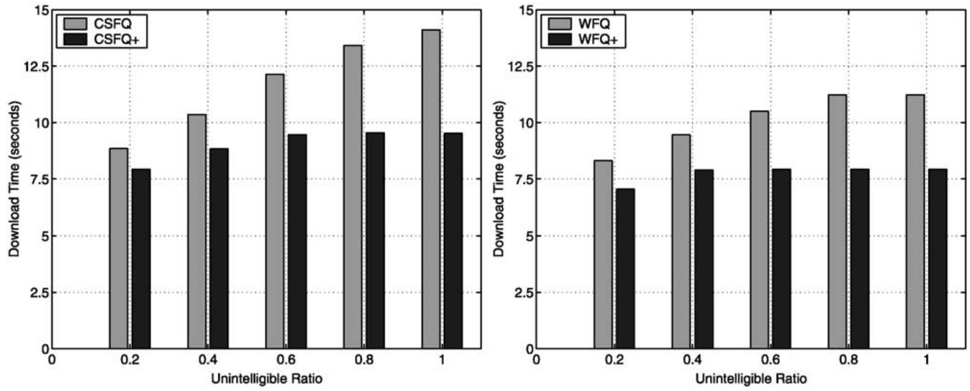
We can see that UPTA significantly reduces file download time for all file sizes, with both CSFQ and WFQ implementations. For Scenario 1, percentage savings in download times are more significant for shorter files (HTML and image objects). For example, with UPTA incorporated in WFQ, file download time is reduced by 55% for 15 KB file, and 36% for 150 KB file, but only about 15% for 1.5 MB file. This is because, with Scenario 1, we only have a modest amount of bandwidth waste in the system.

Fig. 11 graphically compares file download times for 1.5 MB file under all five scenarios. We can see that with more UPT and bandwidth waste present in the system, UPTA can reduce file download time more significantly, both in absolute and percentage scales. For example, under Scenario 4, with UPTA incorporated in WFQ, download time for 1.5 MB file is reduced from 11.23 to a mere 7.93 s, a saving of 3.3 s or 30%. Savings of such magnitudes are useful not only in traditional computing and communication environments, but also in mobile and wireless environments where battery power consumption is directly affected by the connection times.

7.3. Impact on video intelligibility

As stated in Section 4, the proposed UPTA algorithm aims to recover bandwidth wasted by multimedia connections (during U intervals) without inflicting any further damage to the overall intelligibility of the communications. Using the intelligibility index defined in Section 6, Table 6 shows the overall intelligibility of the received video at the destination under all five scenarios. As we can see, UPTA has very little effect on the overall intelligibility of the video connection. This result substantiates that our proposed UPTA is capable of improving TCP performance without inflicting noticeable damage on multimedia.

The good performance of UPTA with respect to intelligibility index (Table 6) is due to its success in maintaining the number of intelligible frames, and reducing only unintelligible frames. For WFQ implementation, the distribution of intelligible and unintelligible frames under all five scenarios are shown in Fig. 12 (similar results were obtained for the CSFQ implementation as well, see Appendix A). Fig. 13 shows the quality of some of the unintelligible frames received at



(a) CSFQ (b) WFQ

Fig. 11. Comparison of download time with and without UPTA.

Table 6
Comparison of video intelligibility

Scenario	Intelligibility index			
	CSFQ	CSFQ +	WFQ	WFQ +
1	0.74	0.74	0.76	0.76
2	0.53	0.52	0.58	0.56
3	0.36	0.34	0.38	0.35
4	0.13	0.11	0.15	0.14
5	0.0	0.0	0.0	0.0

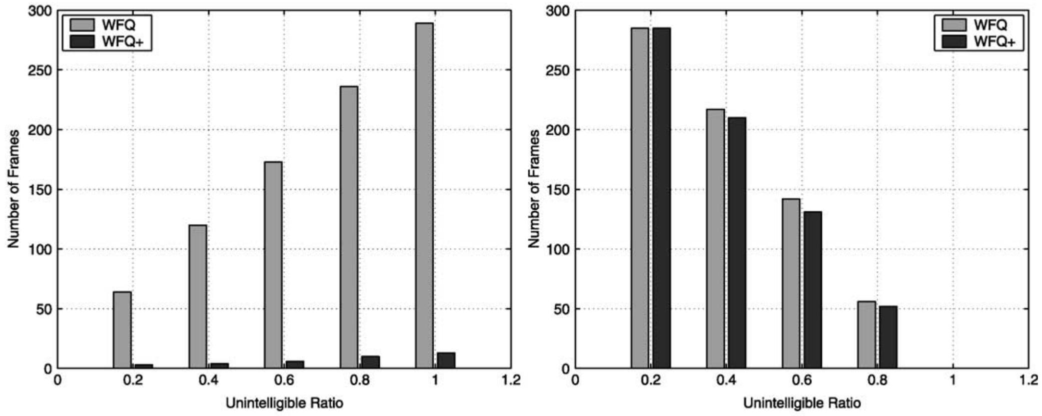
the destination with standard CSFQ (without UPTA). Clearly, by not receiving these frames (when UPTA is implemented), the users would not lose much.

7.4. Throughput fairness

During *U* intervals, fairness is computed over only two connections, TCP and ON–OFF, as the video connection is not supposed to receive fair throughput (and waste bandwidth) under UPTA. However, during *I* intervals, all three connections are considered. Table 7 shows throughput fairness achieved with and without UPTA under all five scenarios. It is encouraging to see that our proposed UPTA can maintain the same level of fairness obtained with CSFQ and WFQ. In other words, the implementation of UPTA into fair algorithms does not have any adverse effect on the fairness performance of the algorithms.

8. Conclusion

We have investigated the issue of UPT in IP routers when multimedia applications use best-effort network services. With best-effort service, available bandwidth is equally allocated to all competing flows using fair packet queueing and scheduling algorithms (e.g. WFQ and CSFQ). TCP-based data applications can tolerate any bandwidth, but multimedia become unintelligible when allocated bandwidth is too low and packet loss rate exceeds a given threshold. Multimedia packets transmitted by the router during such intervals are basically useless, and effectively reduce throughputs and increase file download times of competing TCP connections. We have



(a) Unintelligible frames (b) Intelligible frames

Fig. 12. Distribution of unintelligible and intelligible frames for WFQ.

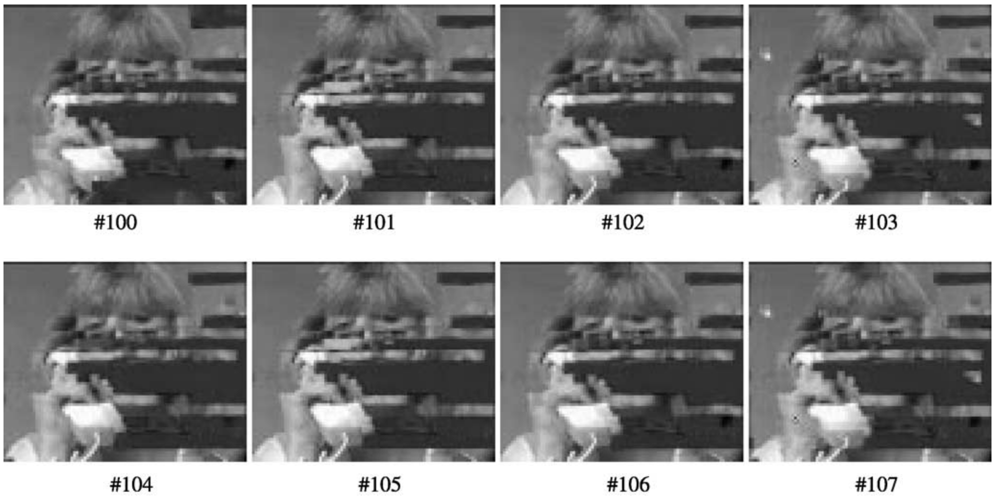


Fig. 13. Snapshots of unintelligible video frames.

Table 7
Throughput fairness

Scenario	Fairness index in U interval				Fairness index in I interval			
	CSFQ	CSFQ +	WFQ	WFQ +	CSFQ	CSFQ +	WFQ	WFQ +
1	0.9923	0.9972	0.9987	0.9989	0.9987	0.9987	0.9998	0.9998
2	0.9956	0.9971	0.9996	0.9994	0.9983	1.0000	0.9999	0.9999
3	0.9966	0.9982	0.9998	0.9997	0.9986	1.0000	0.9999	1.0000
4	0.9963	0.9985	0.9999	0.9998	0.9988	1.0000	1.0000	1.0000
5	0.9965	0.9995	1.0000	0.9998	N/A	N/A	N/A	N/A

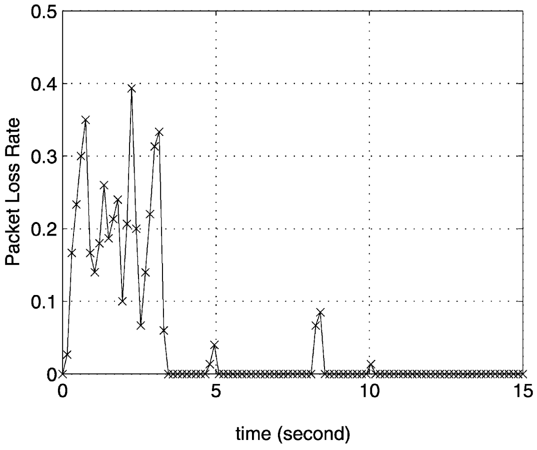
proposed a UPT avoidance algorithm, which prevents a router from transmitting useless multimedia packets, and allocate the recovered bandwidth to competing TCP flows. We have implemented our algorithm in two well-known fair packet queueing algorithms: WFQ and CSFQ. Simulation of a traffic mix consisting of MPEG-2 video and TCP over a typical enterprise network topology reveals that our proposed algorithm can effectively eliminate useless packet transmission in congested routers, and as a result increase TCP throughput and decrease file download time. Our UPT avoidance algorithm can operate unobtrusively; it has noticeable effect neither on the overall intelligibility of the multimedia applications, nor on the fairness performance of the 'host' packet queueing algorithms (WFQ and CSFQ) on which we 'mount' our algorithm.

There remains two open issues. We have assumed that routers have the knowledge of packet loss rate thresholds (for intelligible communication) for all multimedia applications. This threshold information is used by our UPTA algorithm to detect the unintelligible intervals where UPT occurs. How applications can pass this information to the network remains an open issue, and has not been explicitly addressed in this paper. Generally speaking, a signalling mechanism (between users and the network) is needed for this purpose. A user will tell the network the maximum loss rate the user can tolerate for a particular video. The maximum tolerable loss rate will vary from user to user, video to video. The signalling mechanism can be as simple as using the *Type of Service (ToS)* byte of IP header to carry loss rate thresholds, or as complex as periodic ICMP messages. With the first approach, routers do not maintain per-flow states, and no additional control traffic is generated. The second approach is more complex, as routers have to keep per-flow states. We took the first approach in our simulations, and hence no per-flow table was maintained to keep such information in the router.

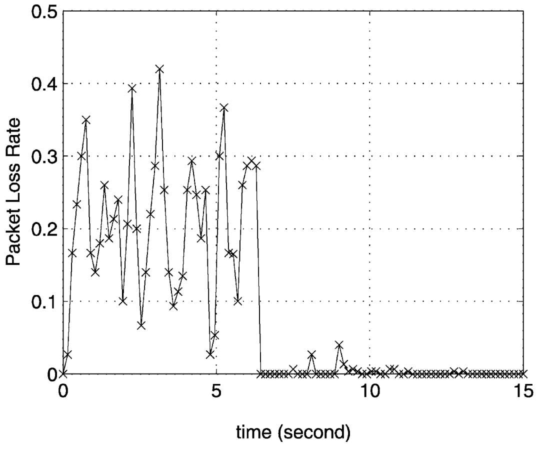
The other issue not addressed in this paper is the possibility of multiple congested routers in the end-to-end path of a large IP network. In such scenarios, upstream routers can still cause UPT if they do not have information on the bottlenecks at the downstream routers. We are currently investigating this issue.

Appendix A. CSFQ simulation results

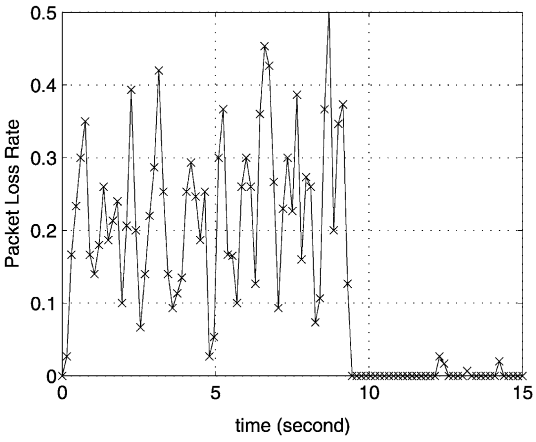
Figs. A1–A3.



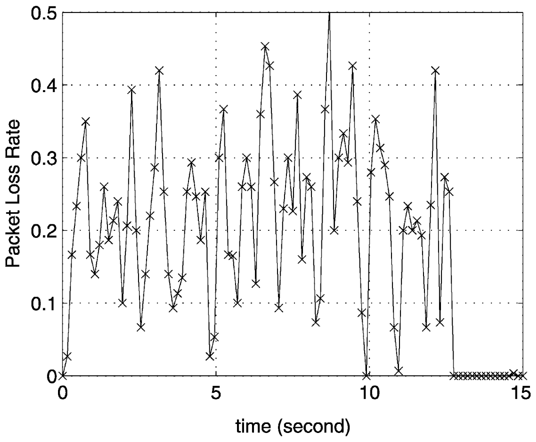
(a) Scenario 1



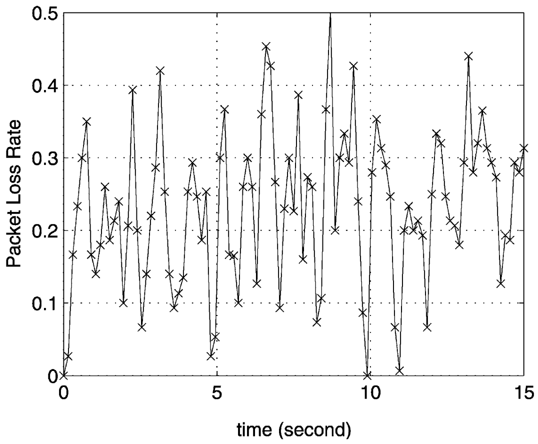
(b) Scenario 2



(c) Scenario 3



(d) Scenario 4



(e) Scenario 5

Fig. A1. Packet loss rate of video flow for CSFQ.

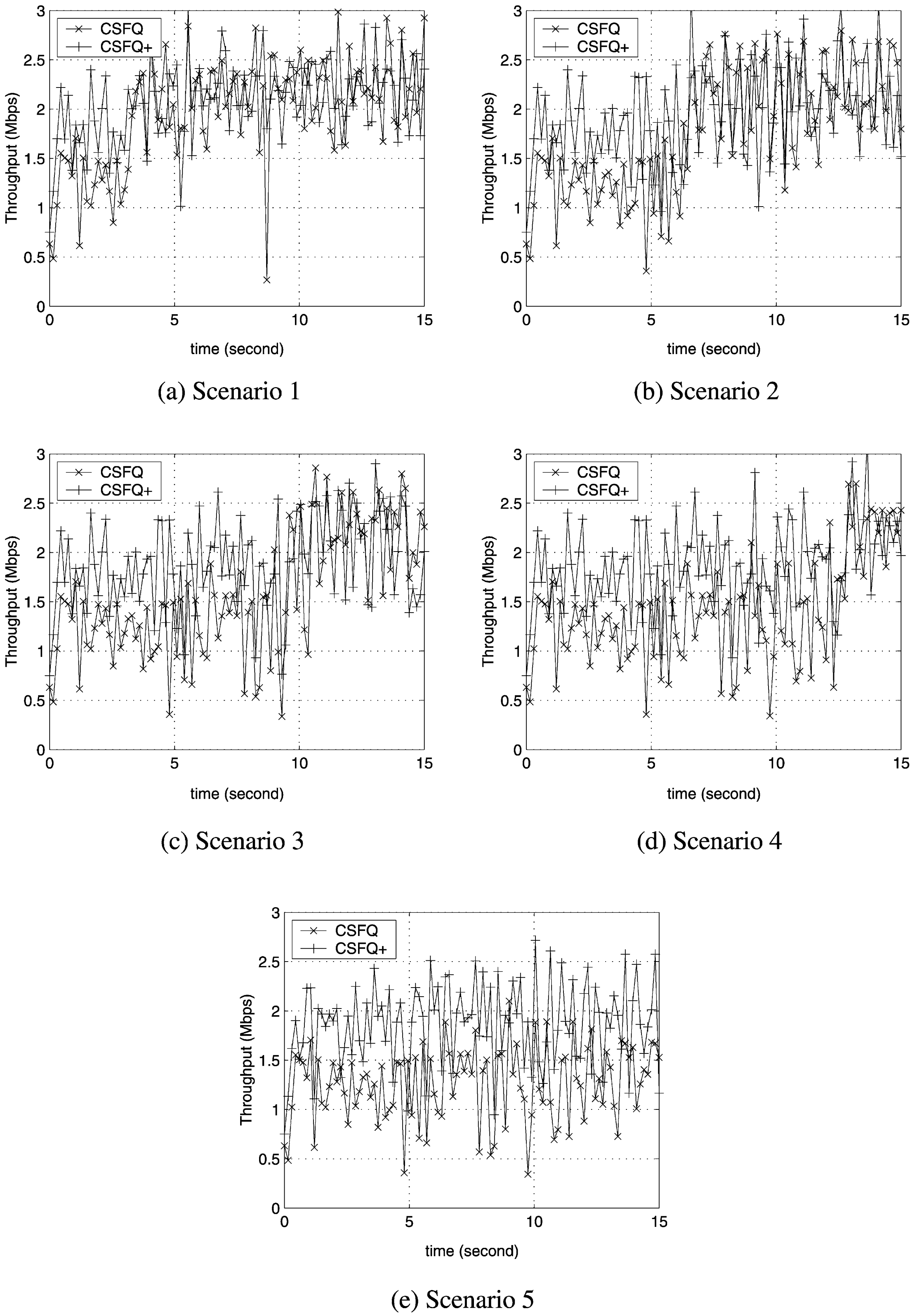


Fig. A2. Short-term TCP throughputs for CSFQ, with and without UPTA.

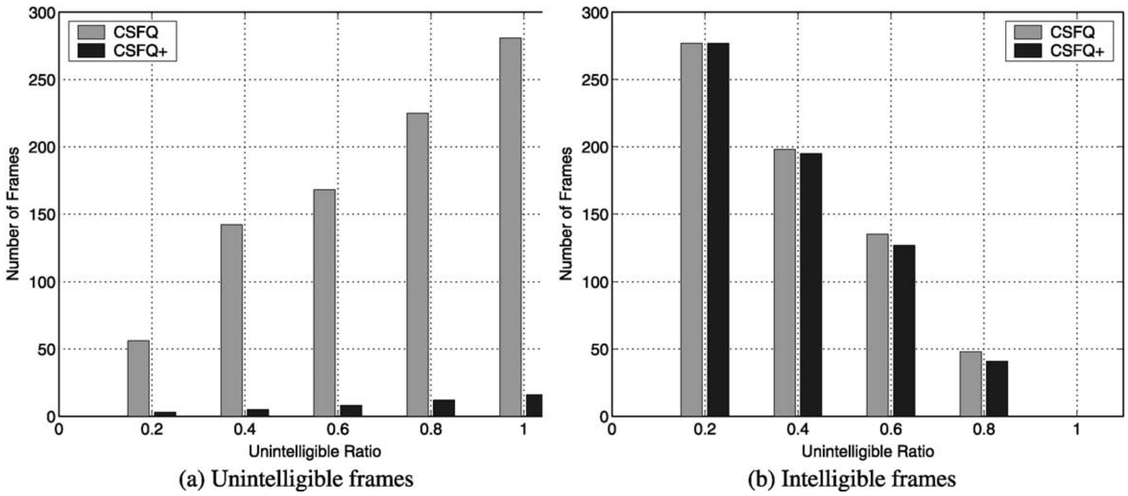


Fig. A3. Distribution of unintelligible and intelligible frames for CSFQ.

References

[1] A. Demers, S. Keshav, S. Shenker, Analysis and simulation of a fair queueing algorithm, Internetworking: Research and Experience 1 (1) (1990) 3–26.

[2] I. Stoica, S. Shenker, H. Zhang, Core-stateless fair queueing: achieving approximately fair bandwidth allocations in high speed networks, Proceedings of ACM SIGCOMM, Vancouver, Canada (1998) 118–130.

[3] D. Lin, R. Morris, Dynamics of random early detection, Proceedings of ACM SIGCOMM (1997) 127–137.

[4] C.S. Inc, Advanced QoS Services for the Intelligent Internet, http://www.cisco.com/warp/public/cc/pd/iosw/index.shtml.

[5] N. Chapman, J. Chapman, Digital Multimedia, Wiley, London, 2000.

[6] F. Fluckiger, Understanding Networked Multimedia: Applications and Technology, Prentice Hall, Englewood Cliffs, NJ, 1995.

[7] J.F. Kurose, K.W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, Addison-Wesley, Reading, MA, 2001.

[8] O. Verscheure, X. Garcia, User-oriented QoS in packet video delivery, IEEE Network 12 (6) (1998) 12–21.

[9] D. Hoffman, G. Fernando, V. Goyal, M. Civanlar, RTP payload format for MPEG1/MPEG2 video, RFC 2250 January (1998).

[10] Tektronix, ftp://ftp.tek.com/tv/test/streams/Element/index.html.

[11] K. Rao, J. Hwang, Techniques and Standards for Image, Video, and Audio Coding, Prentice Hall, Englewood Cliffs, NJ, 1996.

[12] M.S.S. Group, MPEG-2 Video Codec, http://www.mpeg.org/MPEG/MSSG/.

[13] L. Cottrell, W. Matthews, Perception of packet loss, ESSC Review Meeting, Berkeley May (1998) http://www.slac.stanford.edu/grp/scs/net/talk/essc-may98/sld024.html.

[14] O. Technologies, OPNET Modeler 7.0.B, http://www.opnet.com/.

[15] M. Atiquzzaman, M. Hassan, Guest editorial: adaptive real-time multimedia transmission over packet switching networks, Real-Time Imaging 7 (2001) 219–220.

[16] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: a transport protocol for real-time applications, RFC 1889 January (1996).

[17] S. Floyd, M. Handley, J. Padhye, J. Widmer, Equation-based congestion control for unicast applications, Proceedings of ACM SIGCOMM 2000, Stockholm, Sweden (2000) 43–56.

[18] D. Chiu, R. Jain, Analysis of the increase/decrease algorithms for congestion avoidance in computer networks, Computer Networks and ISDN Systems 17 (1) (1989) 1–14.

[19] I.D.S.W.G. homepage, http://www.ietf.org/html.charters/diffserv-charter.html.

[20] J. Wroclawski, The use of RSVP with IETF integrated services, RFC 2210 September (1997).

[21] J.-C. Bolot, S. Fosse-Parisis, D. Towsley, Adaptive FEC-based error control for interactive audio in the Internet, Proceedings of IEEE INFOCOM, New York 3 (1999) 1453–1460.

[22] M. Podolsky, C. Romer, S. McCanne, Simulation of FEC-based error control for packet audio on the Internet, Proceedings of IEEE INFOCOM, San Francisco, California, USA, vol. 2, 1998, pp. 505–515.

[23] C. Perkins, O. Hodson, V. Hardman, A survey of packet loss recovery techniques for streaming audio, IEEE Network 12 (6) (1998) 40–48.

[24] H. Sanneck, A. Stenger, K.B. Younes, B. Girod, A. New, A new technique for audio packet loss concealment, IEEE Global Internet (1996) 48–52.

[25] G.J. Armitage, K.M. Adams, Packet reassembly during cell loss, IEEE Network 7 (5) (1993) 26–34.

[26] A. Romanov, S. Floyd, Dynamics of TCP traffic over ATM networks, IEEE Journal on Selected Areas in Communications 13 (4) (1995) 633–641.

Jim Wu is a full-time PhD student at School of Computer Science & Engineering, The University of New South Wales. He submitted his PhD thesis for examination on 15 November 2002. Mr. Wu received his B.S. degree (in Electronics & Information Systems) from Zhong Shan University (China) in 1988, and his Graduate Diploma and Master degree (both in Digital Communications) from Monash University, in 1995 and 1997 respectively. He worked as a system engineer at No. 34 Research Institute of Electronic Industry Ministry (Guilin, China) before he came to Australia in 1994. He worked as an associate lecturer from 1998 to 2002, at Monash University and The University of New South Wales. Mr. Wu is currently employed as a research assistant at Network Research Laboratory, School of Computer Science & Engineering, The University of New South Wales. His research interests include congestion management, multimedia over IP, QoS for IP networks, etc.

Dr. Mahbub Hassan is an Associate Professor in the School of Computer Science and Engineering, University of New South Wales, Sydney, Australia, where he directs advanced research and development activities in the Network Research Laboratory. He received PhD from Monash University, Melbourne, Australia and MSc from University of Victoria, Canada. He serves in the Editorial Advisory Board of Computer Communications journal (Elsevier Science) and previously served as an Associate Technical Editor for IEEE Communications Magazine. He was Guest Editor for Real Time Imaging Journal and Journal of Supercomputing. He has near 100 refereed publications, including four books. Dr. Hassan was the recipient of the Teaching Excellence Award from the School of Computer Science, Monash University, 1999. He has chaired SPIE conference on QoS, 2001 (Denver) and 2002 (Boston). He is a Senior Member of IEEE. Dr. Hassan also serves as consultant to industry and government. Consulting services range from advanced short course development and delivery to network analysis and design. Past and current clients include NEC, Telstra, Lucent and Canon