Avoiding Useless Packet Transmission for Multimedia over IP Networks: The Case of Multiple Multimedia Flows

Jim Wu and Mahbub Hassan
School of Computer Science & Engineering
The University of New South Wales
Sydney 2052, Australia
Email: {jimw,mahbub}@cse.unsw.edu.au

UNSW-CSE-TR-0216

November 2002

THE UNIVERSITY OF
NEW SOUTH WALES

SCIENTIA

MANU ET MENTE

SYDNEY · AUSTRALIA

Abstract

In this paper, we investigated UPT avoidance problem with multiple multimedia flows. We propose a management module, called *Unintelligible Flow Management* (UFM), to enhance UPTA in networks with multiple multimedia flows. We have proposed two different management policies for UFM, i.e. Random Select (RS) and Least Bandwidth Select (LBS). We have demonstrated incorporation of RS/LBS into WFQ, and evaluated the effectiveness of both RS and LBS under various network scenarios (e.g. single/multiple congested links, homogeneous/heterogeneous video applications, etc.). Simulation results show that UFM can significantly improve TCP throughput and average video intelligibility index, as compared with plain WFQ. On the other hand, our simulation results also suggest that RS and LBS have similar performance with homogeneous multimedia applications. However, with heterogeneous multimedia applications, LBS yields better performance, in terms of total number of video flows recovered and average intelligibility index of video applications.

Keywords: TCP, multimedia over IP, MPEG-2, Internet, Fair Packet Queueing Algorithm.

1 Introduction

The problem of *Useless Packet Transmission* (UPT) arises as a result of (i) rising popularity of audio/video applications over the Internet, and (ii) increasing deployment of fair packet queueing/scheduling algorithms (e.g. WFQ [11], FRED [8], CSFQ [12], etc.). Current Internet supports only "*best-effort*" network service, no *Quality of Service* (QoS) is guaranteed. Packets may be dropped by routers at times of congestion. Among the two transport services available on the Internet, TCP is adaptive while UDP is non-adaptive. Most multimedia applications over the Internet are UDP-based. These applications tend to grab all network bandwidth and force TCP applications to shut up. This problem is known as *fairness* problem. Over the past few years, there has been extensive research conducted to solve the fairness problem. Weighted Fair Queueing (WFQ) [11] and Core Stateless Fair Queue (CSFQ) [12] are two well-known fair algorithms. Other fair algorithms include BLUE [3, 4], CHOKe [10], and RFQ [1]. The fairness problem in the Internet is now well recognised. Many packet queueing and discarding algorithms have been proposed in the last few years to effectively address the issue of fairness. Some network equipment manufacturers have already implemented these algorithms in their latest products. For example, WFQ has been implemented in Cisco 2600/3600/3700 Series [6] and Nortel Networks Passport 5430 [9].

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 [2, 5, 7]. When packet loss rate exceeds this threshold, received audio and video become useless. We formally defined the UPT problem, and proposed an avoidance algorithm called *Useless Packet Transmission Avoidance* (UPTA) in [15]. We have discussed UPT avoidance with *single congested link* and *multiple congested links*, in [15] and [14] respectively. Another interesting issue is UPT avoidance in networks with *multiple multimedia flows*. In networks with large user base, there are usually more than one multimedia flows sharing the same bottleneck link. In this paper, we investigate UPT avoidance with multiple multimedia flows, and discuss issues related to management of multiple U flows.

The rest of the paper is organised as follows. In Section 2, we discuss problems arise with UPTA in networks with multiple multimedia flows. In Section 3, we propose two different management policies for UPTA: *Random Select* (RS) and *Least Bandwidth Select* (LBS). We describe our simulation configuration and performance metrics in Section 4, and present our simulation results in Section 5 and Section 6. Finally, we present our conclusions and discuss open issues in Section 7.

2 Scope of the Problem

UPTA attempts to drop all packets from a multimedia flow when it is in U intervals. In situations where multiple multimedia flows sharing the same bottleneck link are marked as U flows, all these flows will be cut off when UPTA is enforced. An undesirable effect of this is that all multimedia flows become useless, and the bottleneck link is under-utilised. This problem has adverse impacts on overall intelligibility of multimedia applications, as well as the efficiency of network resources.

We use an example as shown in Figure 1 to explain this problem. The figure shows that two MPEG-2 video flows share a bottleneck link of 2 Mbps in the network. Assume MPEG-2 video stream is encoded at a bit rate of 1.5 Mbps (CBR), and is transported in MPEG *Transport Stream* (TS) packets over UDP.

MPEG TS packets have a fixed packet size of 188 bytes. Taking TS header (4 bytes), UDP header (8 bytes), and IP header (20 bytes) into account, the total bandwidth required by an MPEG-2 video flow is about 1.76 Mbps. According to our experiment conducted in [15], the maximum tolerable packet loss rate for MPEG-2 video flows is 12% (i.e. $q = 12\%$). Assume UPTA is implemented in Router1, in

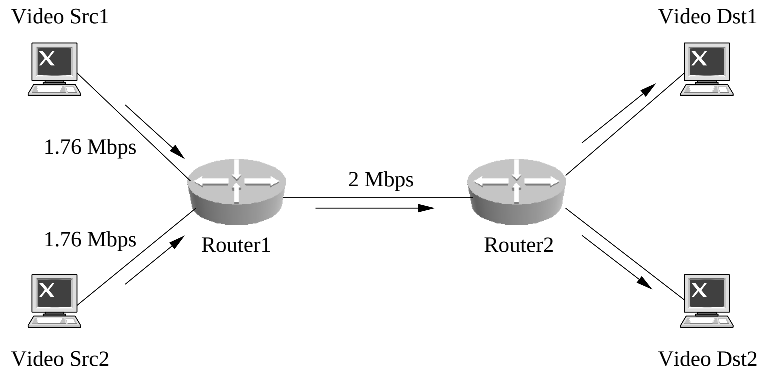


Figure 1: UPTA in networks with multiple U flows.

conjunction with WFQ. Obviously, both video flows in Figure 1 suffer a packet loss of over 40% (much higher than q), as the fairshare in the network is only 1 Mbps. Therefore, all packets from the two video flows will be dropped by UPTA, because they are all U flows (loss rate larger than 12%). This will lead to zero utilisation of the bottleneck link.

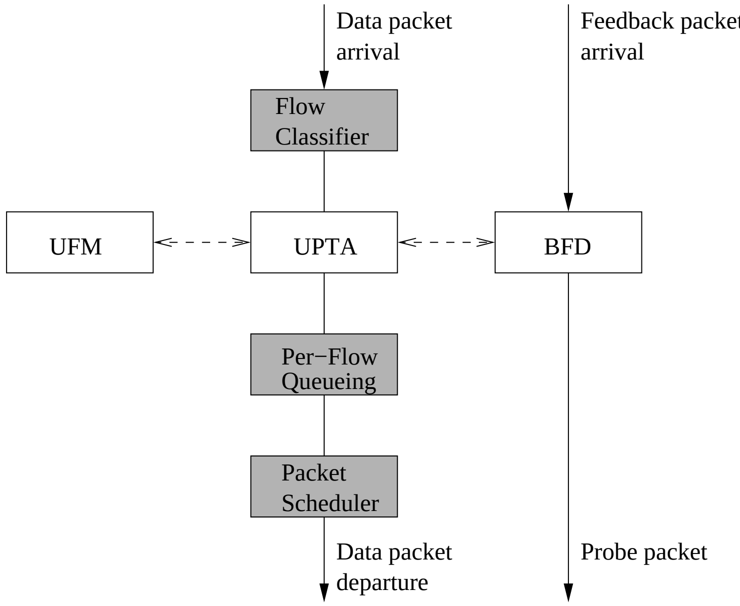
In the example shown in Figure 1, both video applications would be wasting network bandwidth without achieving meaningful communication if UPTA is not implemented. With UPTA implemented, UPT is eliminated, but the bottleneck link is under-utilised. In fact, the network can support one MPEG-2 video flow (but not two). The challenge here is how to minimise UPT in the network while maximising the number of intelligible video flows and high link utilisation. To address this problem, we propose a functional module called *Unintelligible Flow Management* (UFM) for UPTA, in this chapter.

3 Unintelligible Flow Management

Unintelligible Flow Management (UFM) is proposed to manage U flows efficiently in networks with multiple multimedia flows. It arbitrates which U flow should be converted into I flow based on bandwidth available. Figure 2 illustrates the high-level architecture of UPTA enhanced with UFM.

In Figure 2, we assume WFQ is implemented. The three shaded blocks (Flow Classifier, Per-Flow Queueing, and Packet Scheduler) represent functionalities already implemented by WFQ. BFD (Bottleneck Fairshare Discovery) is a feedback mechanism for Centralised UPTA (C-UPTA) (described in [14]). It is used to find out the bottleneck link (in a network) on an end-to-end basis, and enable UPTA to drop useless packets at network edge. UFM represents another enhancement to UPTA. It is proposed to address problems discussed in Section 2, by converting selected U flows into I flows when there are multiple U flows present in the network.

The core of UFM framework is a management policy which attempts to convert a U flow into I flow at a given interval. We propose two different UFM policies for this purpose: *Random Select* (RS) and *Least Bandwidth Select* (LBS). RS is a very simple algorithm which simply selects a U flow randomly from all U flows, if there are more than one U flow present in the network. LBS is a more intelligent algorithm. It always selects a U flow which requires the least bandwidth to survive (to switch from U to I). In this way, LBS can potentially recover more multimedia flows than simple RS. We discuss the operation of RS and LBS in following sections.



UFM: Unintelligible Flow Management
UPTA: Useless Packet Transmission Avoidance
BFD: Bottleneck Fairshare Discovery

Figure 2: UPTA enhanced with UFM.

3.1 Random Select (RS)

As a result of UPTA enforcement, there may be more than one multimedia flow marked as unintelligible (U flows). If the available bandwidth on the bottleneck link is large enough to accept at least one U flow, UFM will be called to select some U flows and convert them into I flows. The simplest way is to select a U flow randomly. We call such a management policy Random Select (RS). Figure 3 shows the pseudo-code of RS.

RS attempts to convert a U flow into I flow each time a router updates fairshare. As shown in Figure 3, RS is called at each fairshare update interval. RS randomly selects a U flow as "candidate" for conversion into I, if there are more than one U flow at the router. It then updates fairshare for I flows. To ensure that all I flows (accepted previously) are still intelligible even after the selected U flow is converted into I flow, RS performs an iterative check on all I flows maintained in its state table. If the new fairshare is less than the minimum required bandwidth of any I flow, the conversion attempt will be aborted. The minimum required bandwidth b is computed as per Eq. (1):

$$b = (1 - q)\lambda \tag{1}$$

in which λ is the arrival rate of a flow; and q is the maximum tolerable packet loss rate for the flow.

3.2 Least Bandwidth Select (LBS)

Least Bandwidth Select (LBS) is another management policy for UFM. It is more intelligent than RS in a sense that it always selects the flow which requires the least bandwidth to survive. An advantage of

```
On est_timer expiration:
if (M>1)
    select a U flow k randomly;
    /* Convert k into I */
    flow_state[k].int = I;
    /* Estimate fairshare for I flows */
    compute fs_I;

/* Loop through flow state table to see */
/* if the new fairshare will render any */
/* accepted I flow useless.             */
for (i=0;i<N;i++)
    if (flow_state[i].int == I &&
        fs_I < flow_state[i].b)
        /* Cannot convert the flow into I */
        flow_state[k].int = U;
        break;
```

Figure 3: Pseudo-code of Random Select (RS).

LBS is that it can potentially convert more U flows into I flows as compared with RS, because RS may select a U flow which requires more bandwidth to survive.

Figure 4 shows the pseudo-code of LBS. Like RS, LBS attempts to convert a U flow into I flow if there are multiple U flows. Unlike RS, LBS always selects the U flow which has the smallest Δ_{BW} value as the candidate. To accomplish this, LBS loops through the table (which contains all U flows) to find out the flow that requires the least bandwidth to survive (we call it *survival bandwidth*). LBS also performs an iterative check on I flows maintained by the router (same as RS). The conversion attempt will be aborted if the check fails.

The survival bandwidth (of a U flow) Δ_{BW} is defined as the bandwidth required by a U flow to switch from U to I (from unintelligible to intelligible). That is,

$$\Delta_{BW} = b - \alpha \tag{2}$$

Survival bandwidth Δ_{BW} is used by LBS to determine which U flow should be switched to I. We derive Δ_{BW} as below. Assume, for a given U flow f:

- α: current fairshare for U flows.
- λ: packet arrival rate.
- p: current packet loss rate.
- q: maximum tolerable loss rate.
- b: minimum required bandwidth (defined in Eq. (1)).
- δ: excess loss rate of a U flow ($\delta = p - q$).

As U flows are constrained flows, packet loss rate of a U flow can be expressed as:

```
On est_timer expiration:
if (M>1)
    /* Select the flow with least survival bandwidth */
    k = 0;
    for (i=1;i<M;i++)
        if (flow_state[i].survbw < flow_state[k].survbw)
            k = i;
    /* Convert flow k into I flow */
    flow_state[k].int = I;
    /* Estimate fairshare for I flows */
    compute fs_I;

/* Loop through flow state table to see */
/* if the new fairshare will render any */
/* accepted I flow useless.             */
for (i=0;i<N;i++)
    if (flow_state[i].int == I &&
        fs_I < flow_state[i].b)
        /* Cannot convert the flow into I */
        flow_state[k].int = U;
        break;
```

Figure 4: Pseudo-code of Least Bandwidth Select (LBS).

$$p = 1 - \frac{\alpha}{\lambda} \tag{3}$$

From Eq. (3) we obtain:

$$\alpha = (1 - p)\lambda \tag{4}$$

Substitute Eq. (1) and Eq. (4) into Eq. (2), we have:

$$\begin{aligned} \Delta_{BW} &= b - \alpha \\ &= (1-q)\lambda - (1-p)\lambda \\ &= (p-q)\lambda \\ &= \delta\lambda \end{aligned} \tag{5}$$

From Eq. (5) we can see that a U flow with smaller excess loss rate needs less bandwidth to switch to I flow, provided all contending U flows have the same arrival rate (e.g. all flows are running the same type of application).

4 Performance Evaluation

We have implemented the two different UFM policies (RS and LBS) using OPNET Modeler [13]. In following sections, we present our simulation study conducted to evaluate the performance of RS and LBS.

We conduct extensive simulations to evaluate these two management policies under various network conditions, as shown below:

- **Single congested link**: Simulates enterprise networks. There is only one congested link in the network. Basic UPTA (B-UPTA, discussed in [15]) is implemented.
- **Multiple congested links**: Simulates medium to large scale networks. There are multiple congested links in the network. Centralised UPTA (C-UPTA, discussed in [14]) is implemented.
- **Homogeneous multimedia applications**: All multimedia applications are of the same type. In our simulation, all MPEG-2 video applications have the same data rate.
- **Heterogeneous multimedia applications**: Multimedia applications have different requirements on bandwidth. In our simulation, the same MPEG-2 video is encoded at different data rates for different video sources.

In our simulation study, we use following performance metrics:

- **Number of video flows recovered**: The number of video flows (which would be marked as U flows by UPTA) converted from U flows into I flows.
- **TCP throughput improvement**: We compare TCP throughput achieved under RS/LBS against that achieved under WFQ.
- **Video intelligibility**: We use the intelligibility index (defined in [15]) to compare the overall intelligibility level of the received video.

5 Single Congested Link

First of all, we ran a set of simulations with single congested link. Figure 5 shows the network model used in the simulation. In the simulated network, there is only one congested link (l_1) which has a link speed of 15 Mbps, with a delay of 1 ms. All other links are 10 Mbps (Ethernet links), with a delay of 1 μs. As shown in the figure, A TCP source (S_{11}) shares the bottleneck l_1 with ten UDP-based MPEG-2 video sources (S_1–S_{10}, shaded). WFQ is implemented in routers R_1 and R_2 which have a buffer capacity of 100 packets. UPTA is implemented in R_1 to avoid UPT in the network. The MPEG-2 video stream used in the simulation is the same as that used in the previous chapters (`susi_015.m2v`). The TCP source is a greedy source (it always has data to send).

5.1 Homogeneous Video Applications

In this experiment, we try to simulate homogeneous multimedia applications. That is, all multimedia applications in the network are of exactly the same characteristics (i.e. same data rate, same loss threshold, etc.). In the simulated network, video sources S_1–S_{10} are sending video packets to their corresponding destinations at the same data rate. 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 data rate of each video application is about 1.76 Mbps. As link l_1 has a capacity of only 15 Mbps, obviously there is congestion at router R_1, and as a result packets will be dropped at R_1.

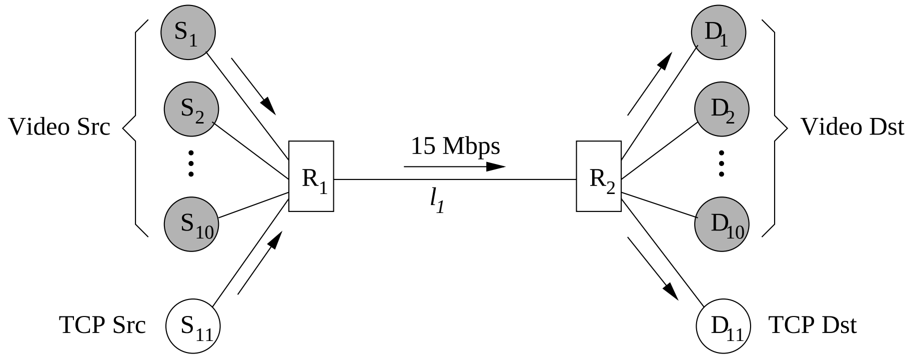


Figure 5: Simulation model for single congested link.

Table 1: Algorithms Implemented for Various Simulation Scenarios

Scenario	*WFQ*	*UPTA*	*UFM*	
			RS	*LBS*
WFQ	√	×	×	×
WFQ+	√	√	×	×
WFQ++(RS)	√	√	√	×
WFQ++(LBS)	√	√	×	√

5.1.1 Number of Video Flows Recovered

We ran four simulations with a duration of 15 seconds (corresponds to the duration of the MPEG-2 video stream). Different combination of algorithms was used in each simulation scenario, as shown in Table 1. In the first simulation, only WFQ was implemented (no UPTA and UFM). In the second simulation, UPTA was implemented in conjunction of WFQ (but no UFM). Random Select (RS) and Least Bandwidth Select (LBS) were implemented in the third and fourth simulations respectively.

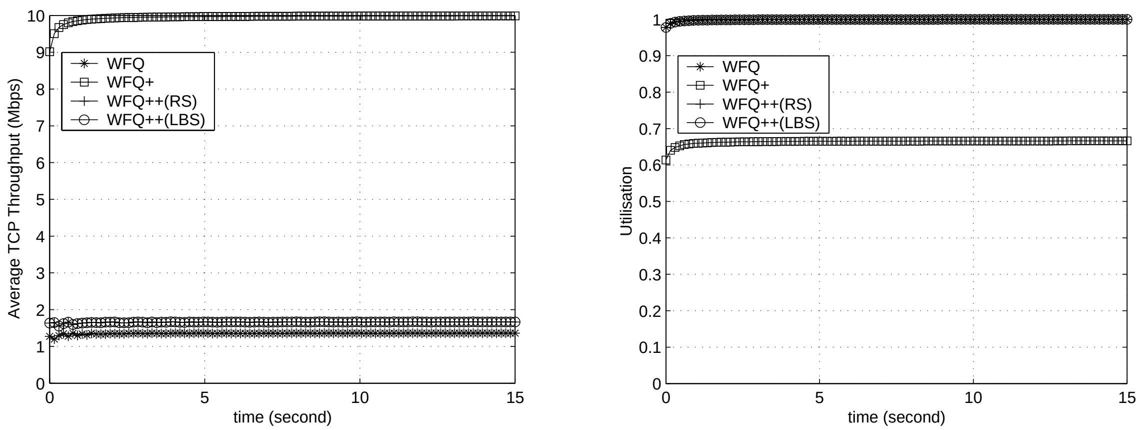
Table 2 shows the number of video flows accepted by router R_1. As shown in the figure, under WFQ, all video flows are accepted (but they are all unintelligible flows). With UPTA, no video flow is accepted. This can be explained as following. Because there are 11 contending flows in the network (10 video and 1 TCP), the fairshare of the bottleneck link (l_1) is about 1.36 Mbps. Video flows suffer a loss rate of more than 22%, much greater than the loss threshold of 12%. Therefore, all video flows are marked as U flows as per UPTA algorithm. As a result, all video packets (from all video sources) are dropped completely by R_1. When UFM policy RS/LBS is implemented, 8 video flows are accepted. The remaining two video flows are marked as U flows. In other words, RS/LBS recovers 8 video flows which would have been cut off by UPTA without UFM. Another observation we can make from Table 2 is that, for homogeneous video flows, RS and LBS have the same performance — they recover exactly the same number of video flows. This is because homogeneous video applications have exactly the same characteristics. There is no difference in survival bandwidth (Δ_{BW}) for homogeneous video flows.

5.1.2 TCP Throughput Improvement

Figure 6 shows both TCP throughput and utilisation of the bottleneck link measured in the four different simulation scenarios. As shown in Figure 6(a), the TCP flow achieves the lowest throughput under WFQ

Table 2: Number of Video Flows Accepted in Various Scenarios (single congested link; homogeneous video).

Scenario	*No. of Video Flows Accepted*
WFQ	10
WFQ+	0
WFQ++(RS)	8
WFQ++(LBS)	8



(a) TCP throughput

(b) Utilisation of bottleneck link

Figure 6: Comparison of TCP throughput and link utilisation (single congested link; homogeneous video).

(about 1.36 Mbps). This is because WFQ allocates bandwidth (of the bottleneck link) equally to all 11 contending flows, with each flow receiving the fairshare of 1.36 Mbps. When UPTA is implemented together with WFQ, all video flows are identified as U flows, because packet loss rate is as large as 22%. All 10 video flows are cut off by UPTA, leaving all bandwidth to the TCP flow. That is, the TCP flow can take up all bandwidth of l_1 (15 Mbps). However, the TCP flow is bottlenecked by its access link (10 Mbps). Therefore, in scenario WFQ+, TCP throughput achieved by the TCP flow is 10 Mbps, as shown in Figure 6(a). Although TCP throughput is improved greatly with UPTA implemented, it is not achieved at no cost. As we can see in Figure 6(b), the botttleneck link (l_1) is under-utilised under WFQ+ (utilisation less than 0.7). Obviously, there is a potential to support some video flows (although not all video flows can be accepted). This is completed by RS/LBS. As shown in Figure 6(b), utilisation of l_1 is brought to 1.0 when RS/LBS is implemented (scenarios WFQ++). The TCP throughput under both WFQ++(RS) and WFQ++(LBS) is about 1.67 Mbps, representing an increase of 25% over WFQ (Figure 6(a)). Again, RS and LBS yield similar performance, for the same reason we mentioned before. RS and LBS recover exactly the same number of video flows under homogeneous video, and all recovered video flows have the same requirement on bandwidth. Consequently, the TCP flow achieves the same throughput under RS and LBS. It should be noted, although TCP throughput under WFQ++ is not as large as that under WFQ+, there is improvement over plain WFQ. With RS/LBS, TCP throughput is improved, and more video flows are accepted while the bottleneck link is fully utilised.

Table 3: Comparison of Average Intelligibility Index for Video Applications (single congested link; homogeneous video)

Flow ID	*WFQ*	*WFQ+*	*WFQ++(RS)*	*WFQ++(LBS)*
1	0.0	0.0	0.0	0.9467
2	0.0	0.0	0.9333	0.9387
3	0.0	0.0	0.8960	0.9333
4	0.0	0.0	0.9173	0.9253
5	0.0	0.0	0.9467	0.9200
6	0.0	0.0	0.9040	0.9120
7	0.0	0.0	0.0	0.9067
8	0.0	0.0	0.9387	0.8986
9	0.0	0.0	0.9253	0.0
10	0.0	0.0	0.9093	0.0
Average	0.0	0.0	0.7371	0.7381

5.1.3 Video Intelligibility

Now let's have a look at improvement in intelligibility of MPEG-2 video flows. Table 3 shows the intelligibility index for each video flow in the four different simulation scenarios. The table shows that all video flows have an intelligibility index of 0 when RS/LBS is not implemented (scenarios WFQ and WFQ+). With RS/LBS in place (scenarios WFQ++), 8 of the total 10 video flows achieve good intelligibility. From Table 3 we can also see that, under RS, video flows are randomly selected. Video flows 2–6 and 8–10 are converted into I flows; Flows 1 and 7 remain unintelligible (due to lack of bandwidth). Under LBS, Flow 1–8 are selected, because all video flows have the same survival bandwidth requirement. Once again, RS and LBS yield essentially the same performance, with each scenario achieving an average intelligibility index of about 0.74. This is because all video flows have the same requirement on bandwidth (homogeneous video). As long as the number of video flows recovered is the same, bandwidth received by each recovered video flow should be the same for both RS and LBS.

5.2 Heterogeneous Video Applications

In the previous section, all video sources are sending video of the same data rate (1.5 Mbps). In this experiment, we also use the same video stream. However, we encoded the video stream with five different data rates, as shown in Table 4. When WFQ is implemented, the fairshare of the bottleneck link l_1 is 1.36 Mbps. We assume that the loss threshold q for all video applications remains unchanged (12%). Using Eq. (3) and Eq. (5), we can calculate packet loss rate p and survival bandwidth Δ_{BW} for each video flow, as shown in Table 4. We ran a set of similar simulations as we did for homogeneous video (Section 5.1). The results are presented in following sections.

5.2.1 Number of Video Flows Recovered

Table 5 compares the total number of video flows recovered in various simulation scenarios. For the same reason as we explained in Section 5.1.1, all video flows are accepted under WFQ, and no video flow is accepted under WFQ+. However, 3 and 5 video flows are recovered, for WFQ++(RS) and WFQ++(LBS) respectively. Obviously, LBS outperforms RS in terms of number of recovered video flows. To see why

Table 4: Characteristics of Heterogeneous Video

Flow ID	*Data Rate* (Mbps)	*Req. BW* (Mbps)	*Min Req. BW* b (Mbps)	*Loss Rate* (%)	*Survival BW* Δ_{BW} (Mbps)
1,2	1.5	1.76	1.55	22.5	0.19
3,4	2.0	2.35	2.07	42.0	0.70
5,6	2.5	2.93	2.58	53.5	1.21
7,8	3.0	3.52	3.10	61.3	1.73
9,10	3.5	4.11	3.62	66.8	2.25

Table 5: Number of Video Flows Accepted in Various Scenarios (single congested Link; heterogeneous video)

Scenario	*No. of Video Flows Accepted*
WFQ	10
WFQ+	0
WFQ++(RS)	3
WFQ++(LBS)	5

LBS could recover more video flows than RS, we analysed the simulation log. Based on the simulation log, we list the time (at which a video flow is recovered), the video flow ID, and fairshare for I flows (α') in Table 6.

As shown in Table 6, RS/LBS attempts to recover a video flow at an interval of 0.1 second (interval for fairshare update). Let's first have a close look at the operation of RS. At time 0.1 second, Flow 9 (3.5 Mbps video) is randomly selected by RS. I flow fairshare α' is then updated. The new α' becomes 10.89 Mbps which is large enough for the accepted video flow (Flow 9). Thus, Flow 9 is converted into I flow. Similarly, RS picks up Flow 5 (2.5 Mbps video) at time 0.2 second, and it selects Flow 7 (3.0 Mbps video) at time 0.3 second. α' is updated accordingly, as shown in the table. At time 0.4 second, RS tries unsuccessfully to recover Flow 2. In Table 6, we bracket Flow 2 to emphasise the fact that Flow 2 cannot be recovered although it is selected. This is because α' would drop to 3.44 Mbps if Flow 2 were accepted. This will render Flow 9 (which has already been accepted) useless, as Flow 9 has a minimum bandwidth requirement of 3.62 Mbps (see Table 4). For the same reason, Flows 10, 6, 3 are not accepted although they are all selected as "candidates". In contrast, LBS always selects the flow with least survival bandwidth. Therefore, it can recover more video flows than RS. This can be easily seen in Table 6. Therefore, LBS outperforms RS.

5.2.2 TCP Throughput Improvement

Figure 7 shows TCP throughput and link utilisation for various simulation scenarios. For WFQ and WFQ+, the results are exactly the same as that achieved with homogeneous video, as all video flows become U flows. Under WFQ, the TCP flow receives a throughput of 1.36 Mbps. Under WFQ+, useless packet transmission is eliminated (all video flows are cut off). The saved bandwidth is made available to the TCP flow. The TCP flow receives a throughput of 10 Mbps. However, the bottleneck is under-utilised (utilisation less than 0.7, see Figure 7(b)), because the TCP flow is bottlenecked by its access link. With RS and LBS, the bottleneck link is fully utilised (utilisation equal to 1.0). The TCP throughput is about

Table 6: Timeline of Video Flow Recovery (single Congested link)

Simulation Time (sec.)	*RS*		*LBS*	
	Recovered Video Flow ID	Fairshare for I Flows (α') (Mbps)	Recovered Video Flow ID	Fairshare for I Flows (α') (Mbps)
0.1	9	10.89	1	13.24
0.2	9,5	7.96	1,2	11.48
0.3	9,5,7	4.44	1,2,3	9.13
0.4	9,5,7,(2)	**3.44**	1,2,3,4	6.78
0.5	9,5,7,(10)	**3.02**	1,2,3,4,5	3.85
0.6	9,5,7,(6)	**3.02**	1,2,3,4,5,(6)	**2.30**
0.7	9,5,7,(3)	**3.24**	1,2,3,4,5,(6)	**2.30**
⋮	⋮	⋮	⋮	⋮
15.0	9,5,7	4.44	1,2,3,4,5	3.85

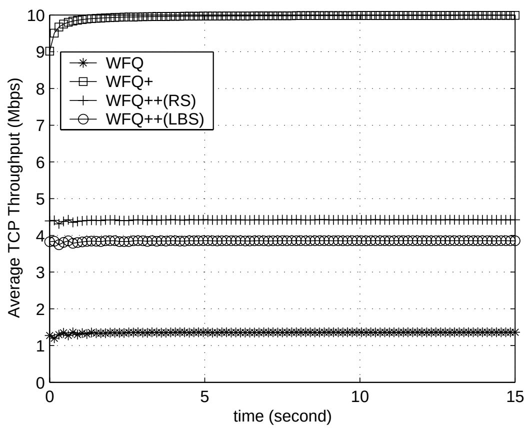
4.4 and 3.9 Mbps respectively for RS and LBS, representing an increase of 224% and 187% respectively over WFQ (see Figure 7(a)). The TCP throughput under RS is slightly larger than that under LBS, because fewer video flows are recovered by RS and thus more bandwidth are made available to the TCP flow under RS.

5.2.3 Video Intelligibility

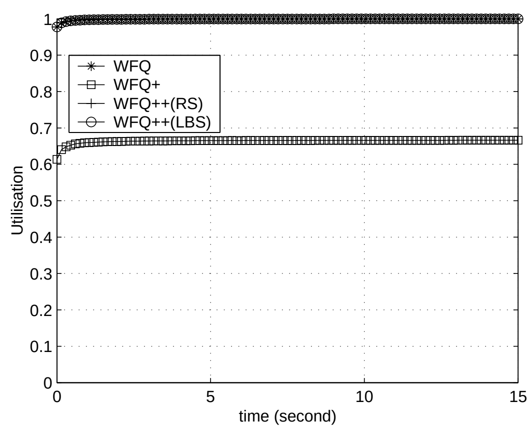
Intelligibility analysis results for the ten video flows are shown in Table 7. The table shows that no video flow is intelligible under WFQ and WFQ+, and some video flows are recovered when RS/LBS is implemented. RS recovers 3 video flows, with an average intelligibility index of 0.2939. Whereas LBS recovers 5 video flows, with an average intelligibility index of 0.4893. Needless to say, LBS outperforms RS in terms of video intelligibility. As we can see in Table 7, the recovered video flows have identical intelligibility indices for RS and LBS. However, LBS recovers more video flows than RS does. This explains why the average intelligibility index under LBS is higher than that under RS.

6 Multiple Congested Links

Finally, we ran a set of simulations with multiple congested links. Figure 8 shows the network model used in the simulation. There are two groups of traffic in the network. Traffic Group 1 (TG1) consists of 9 video sources (S_1–S_9) and one TCP source (S_{10}). Traffic Group 2 (TG2) also has 9 video sources (S_{11}–S_{19}) and one TCP source (S_{20}). Each traffic group uses different ingress/egress routers. For example, for TG1, router R_1 is ingress router, and router R_4 is egress router. For TG2, router R_5 is ingress router, and router R_6 is egress router. Routers R_2 and R_3 are core routers. All network links in the figure have a link speed of 20 Mbps; all access links have a link speed of 10 Mbps. S_{10} and S_{20} are two greedy TCP sources. Centralised UPTA (C-UPTA) is implemented in the network.



(a) TCP throughput



(b) Utilisation of bottleneck link

Figure 7: Comparison of TCP throughput and link utilisation (single congested link; heterogeneous video).

Table 7: Comparison of Average Intelligibility Index for Video Applications (single congested link; heterogeneous video)

Flow ID	*WFQ*	*WFQ+*	*WFQ++(RS)*	*WFQ++(LBS)*
1	0.0	0.0	0.0	0.9947
2	0.0	0.0	0.0	0.9867
3	0.0	0.0	0.0	0.9787
4	0.0	0.0	0.0	0.9733
5	0.0	0.0	0.9867	0.9627
6	0.0	0.0	0.0	0.0
7	0.0	0.0	0.9787	0.0
8	0.0	0.0	0.0	0.0
9	0.0	0.0	0.9947	0.0
10	0.0	0.0	0.0	0.0
Average	0.0	0.0	0.2960	0.4896

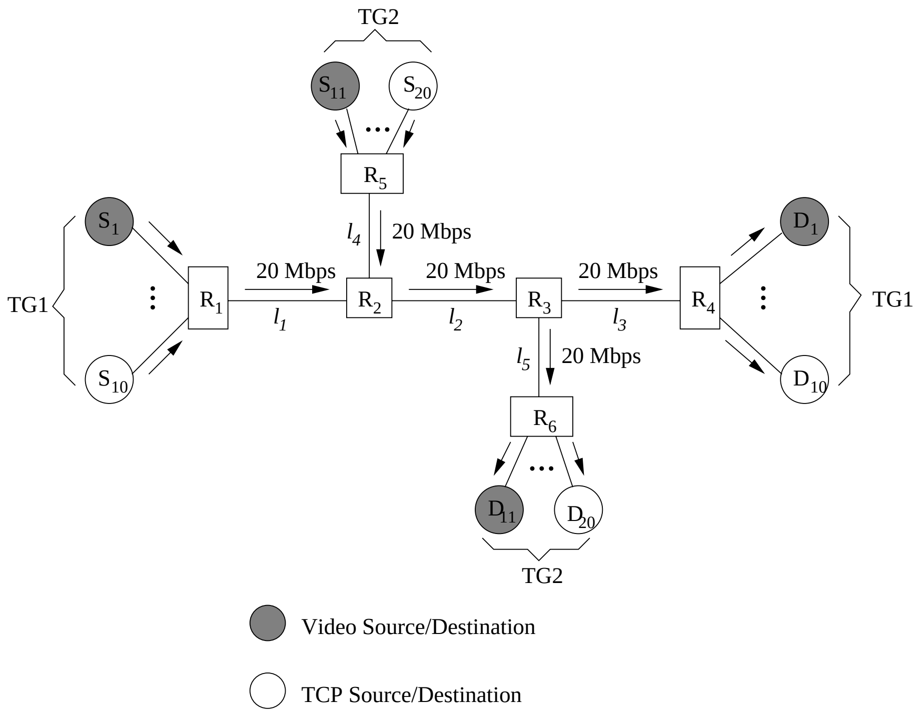


Figure 8: Simulation model for multiple congested links.

6.1 Homogeneous Video Applications

In this experiment, all video sources transmit MPEG-2 video (encoded at 1.5 Mbps) to their corresponding destinations. That is, all video applications are of the same type. We present our simulation results in following sections.

6.1.1 Number of Video Flows Recovered

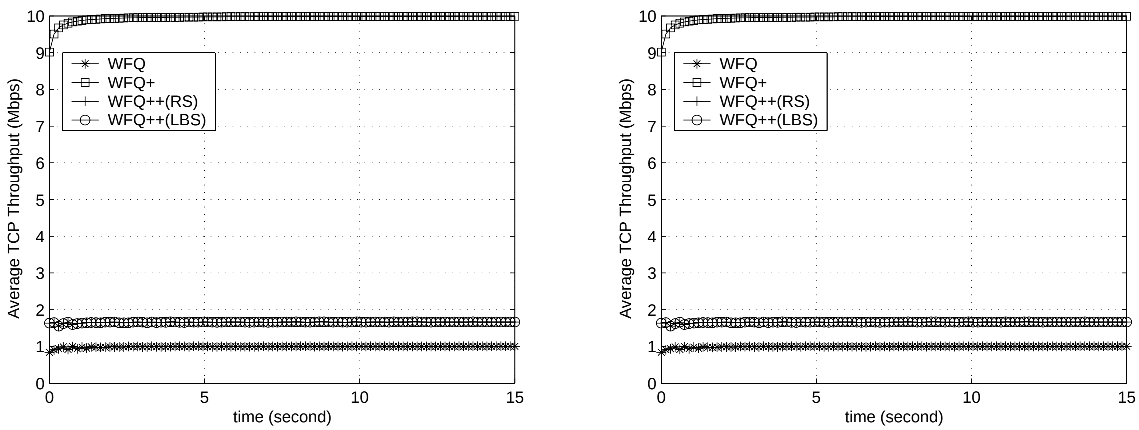
Table 8 shows the number of video flows accepted in each traffic group. Under WFQ, all video flows are accepted, in both TG1 and TG2. However, all of these video flows are U flows. Because in the simulated network, link l_2 is the bottleneck link (most congested link). The global fairshare in the network is 1 Mbps for all flows. All video flows suffer a packet loss rate of about 43%. Therefore, all video flows are transmitting useless packets. When UPTA is implemented with WFQ (WFQ+), no video flow is accepted (because all video flows are eliminated). For WFQ++(RS) and WFQ++(LBS), five video flows are accepted in each traffic group. In other words, RS/LBS successfully recovers five video flows. As we can see in Table 8, RS and LBS recover exactly the same number of video flows.

6.1.2 TCP Throughput Improvement

Figure 9 compares throughputs of the two TCP sources (S_{10} and S_{20}) achieved with various algorithms. Compare Figure 9(a) and Figure 9(b), we can see that, for a given algorithm, the two TCP flows have similar throughput. Under WFQ, both TCP sources achieve a throughput of about 1 Mbps, which equals to the global fairshare. Under WFQ+, the two TCP flows receive a throughput of 10 Mbps, representing an increase of 10 times over WFQ. Under WFQ++(RS) and WFQ++(LBS), the TCP throughput is about

Table 8: Number of Video Flows Accepted in Various Scenarios (multiple congested Link; homogeneous video)

Scenario	*No. of Video Flows Accepted*	
	TG1	TG2
WFQ	9	9
WFQ+	0	0
WFQ++(RS)	5	5
WFQ++(LBS)	5	5



(a) TCP1 (S_{10})

(b) TCP2 (S_{20})

Figure 9: Comparison of TCP throughput (multiple congested links; homogeneous video).

1.67 Mbps, representing an increase of 67% over WFQ. It should be noted that the two TCP flows receive larger throughput when RS/LBS is not implemented. This is because all video flows are cut off, all bandwidth of the bottleneck link is taken by the two TCP flows.

6.1.3 Video Intelligibility

Table 9 shows intelligibility indices of video applications in the network. From the table we can see that under WFQ, all video flows are unintelligible (although they are accepted). Under WFQ+, U flows are eliminated. The intelligibility index of each video flow is of course zero. When RS/LBS is implemented, five video flows become intelligible flows. Again, RS and LBS have essentially the same performance with homogeneous video applications, as we can see in Table 9.

6.2 Heterogeneous Video Applications

In this experiment, we introduce MPEG-2 video with different data rates, as we did for single congested link (Section 5.2). For simplicity, we assume there are only two different video applications in the simulated network. In TG1, video sources S_1–S_6 are sending MPEG-2 video encoded at 1.5 Mbps. Video sources S_7–S_9 are sending MPEG-2 video encoded at 2.5 Mbps. Similarly, in TG2, video sources

Table 9: Comparison of Average Intelligibility Index for Video Applications (multiple congested links; homogeneous video)

Flow ID	*WFQ*	*WFQ+*	*WFQ++(RS)*	*WFQ++(LBS)*
1	0.0	0.0	0.9227	0.9466
2	0.0	0.0	0.0	0.9387
3	0.0	0.0	0.9387	0.9307
4	0.0	0.0	0.9307	0.9227
5	0.0	0.0	0.0	0.9147
6	0.0	0.0	0.9466	0.0
7	0.0	0.0	0.9120	0.0
8	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.9440
12	0.0	0.0	0.0	0.9360
13	0.0	0.0	0.9093	0.9280
14	0.0	0.0	0.9360	0.9200
15	0.0	0.0	0.0	0.9120
16	0.0	0.0	0.9280	0.0
17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.9200	0.0
19	0.0	0.0	0.9440	0.0
Average	0.0	0.0	0.5160	0.5163

Table 10: Number of Video Flows Accepted in Various Scenarios (multiple congested Link; heterogeneous video)

Scenario	*No. of Video Flows Accepted*	
	TG1	TG2
WFQ	9	9
WFQ+	0	0
WFQ++(RS)	4	3
WFQ++(LBS)	5	5

S_{11}–S_{16} employ MPEG-2 1.5 Mbps video, and video sources S_{17}–S_{19} employ MPEG-2 2.5 Mbps video. Because there are multiple ingress/egress routers in the network, each ingress router is set to start sending QoS Probe messages randomly, so as to avoid synchronisation problem. QoS Probe messages are sent at an interval of 100 ms. Global fairshares are also updated at the same interval. Simulation results for heterogeneous video are presented in following sections.

6.2.1 Number of Video Flows Recovered

Table 10 shows the number of video flows accepted in each traffic group. In Traffic Group 1, 9 video flows are accepted under WFQ (but they are unintelligible flows). Under WFQ+, no video flow is accepted, as all video flows are marked as U flows when UPTA is activated. For the simulated network, fairshare of the bottleneck link (l_2) is 1 Mbps. The fairshare is much smaller than the minimum required bandwidth for both 1.5 and 2.5 Mbps MPEG-2 video (see Table 4). Therefore, all video flows become unintelligible flows (U flows) under WFQ. Under WFQ+, all video packets are dropped by UPTA (so no video flow is accepted). Under WFQ++(RS), 4 video flows are recovered. Under WFQ++(LBS), 5 video flows are recovered.

In Traffic Group 2, the number of video flows accepted under WFQ and WFQ+ is the same as TG1, i.e. 9 and 0 for WFQ and WFQ+ respectively. For above-mentioned reason, video flows accepted by WFQ are actually U flows. For WFQ++(RS) and WFQ++(LBS), the number of accepted video flows is 3 and 5 respectively. Table 10 suggests that LBS has a better performance than RS.

Timeline of video flow recovery is shown in Table 11. Let's first look at the operation of RS. At time 0.115 second, router R_1 (ingress router for TG1) receives the first feedback packet, RS randomly selects Flow 3 (1.5 Mbps video) and converts it into I flow. I flow fairshare (α') is then updated. At time 0.139 second, router R_5 (ingress router for TG2) receives the first feedback packet, it randomly selects Flow 18 (2.5 Mbps video) and updates α' to 7.66 Mbps. At time 0.215 second, R_1 receives its second feedback packet. It selects Flow 8 (2.5 Mbps video) and updates α' to 6.19 Mbps. R_5 receives its second feedback packet at time 0.239 second. It selects Flow 12, and so on... At time 0.439, R_5 tries to convert Flow 15 (1.5 Mbps video) into I flow. However, as the fairshare will drop to 2.24 Mbps, Flow 15 cannot be converted (we enclose it with brackets to emphasise that). This is because the fairshare is smaller than the minimum bandwidth required by 2.5 Mbps video (see Table 4). If Flow 15 were accepted, Flows 8, 9, 18 (2.5 Mbps video) would be rendered useless.

With LBS, the ingress routers (for both TG1 and TG2) always select flows with minimum survival bandwidth (i.e., 1.5 Mbps video). In Traffic Group 1, Flows 1, 2, 3, 4, 5 are converted into I flows respectively, each time R_1 receives a feedback packet. In Traffic Group 2, Flow 11, 12, 13, 14, 15 are converted into I flows, at time 0.139, 0.239, 0.339, 0.439, 0.539 second respectively. At time 0.615

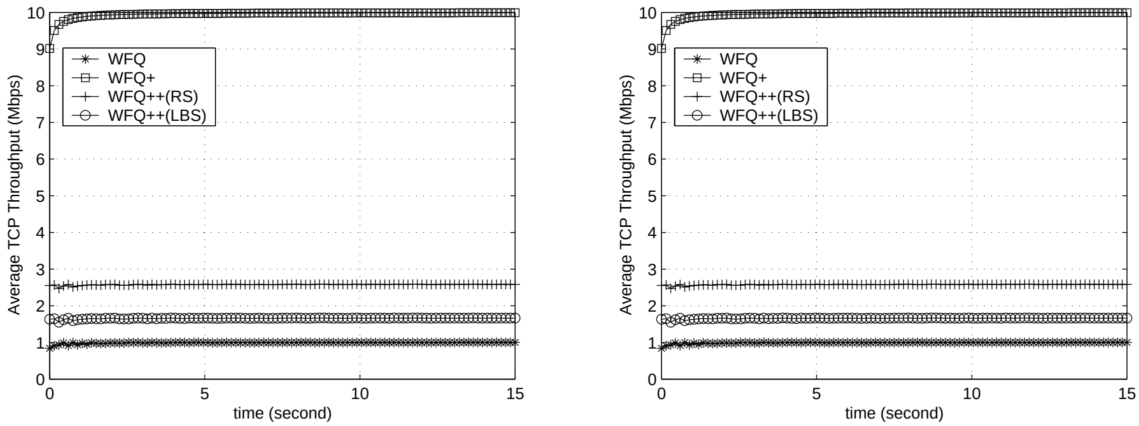
Table 11: Timeline of Video Flow Recovery (multiple congested links)

Simulation Time (sec.)	*RS* Flow ID TG1	TG2	α' (Mbps)	*LBS* Flow ID TG1	TG2	α' (Mbps)
0.115	3	–	19.25	1	–	19.25
0.139	3	18	7.66	1	11	8.24
0.215	3,8	18	6.19	1,2	11	7.36
0.239	3,8	18,12	5.31	1,2	11,12	6.48
0.315	3,8,5	18,12	4.43	1,2,3	11,12	5.60
0.339	3,8,5	18,12,11	3.55	1,2,3	11,12,13	4.72
0.415	3,8,5,9	18,12,11	2.59	1,2,3,4	11,12,13	3.84
0.439	3,8,5,9	18,12,11, (15)	**2.24**	1,2,3,4	11,12,13, 14	2.96
0.515	3,8,5,9, (2)	18,12,11	**2.24**	1,2,3, 4,5	11,12,13, 14	2.08
0.539	3,8,5,9	18,12,11, (19)	**2.24**	1,2,3, 4,5	11,12,13, 14,15	1.67
0.615	3,8,5,9, (7)	18,12,11	**2.24**	1,2,3,4, 5,(6)	11,12,13, 14,15	**1.54**
0.639	3,8,5,9	18,12,11, (13)	**2.24**	1,2,3,4, 5	11,12,13, 14,15,(16)	**1.54**
⋮	⋮	⋮	⋮	⋮	⋮	⋮
15.0	3,8,5,9	18,12,11	2.59	1,2,3,4, 5	11,12,13, 14,15	1.67

second, R_1 selects Flow 6. However, Flow 6 is not converted into I flow because the fairshare would drop to 1.54 Mbps (which is less than the minimum bandwidth required by 1.5 Mbps video). Similarly at time 0.639 second, R_5 tries unsuccessfully to convert Flow 16. Eventually, RS accepts 7 video flows in total whereas LBS accepts 10.

6.2.2 TCP Throughput Improvement

Figure 10 compares TCP throughput achieved with various algorithms. The two TCP flows (Flows 10 and 20) receives identical throughput under a given algorithm. The TCP throughput is about 1, 10, 2.59 and 1.67 Mbps respectively under WFQ, WFQ+, WFQ++(RS) and WFQ++(LBS). As we can see in the figure, there is significant improvement in TCP throughput when UPTA is implemented. The improvement is more significant without RS/LBS, at the cost of complete shut-down of all video applications. This is obviously against fair allocation. RS and LBS effectively recover some video flows that would be cut off by UPTA, while achieve an increase of 159% and 67% (for RS and LBS respectively) in TCP throughput over WFQ. TCP flows receive higher throughputs under RS than that under LBS, simply because RS fails to recover more video flows (while it is potentially possible). This allows the TCP flows to get more bandwidth unfairly.



(a) TCP1 (S_{10})

(b) TCP2 (S_{20})

Figure 10: Comparison of TCP throughput (multiple congested links; heterogeneous video).

6.2.3 Video Intelligibility

Table 12 shows intelligibility indices for all video flows. Under WFQ and WFQ+, no intelligible video transmission can be achieved. Under WFQ++(RS), 7 out 18 video flows are intelligible. Under WFQ++(LBS), 10 out of 18 video flows are intelligible. The average intelligibility index achieved under WFQ++(RS) and WFQ++(LBS) is about 0.36 and 0.52 respectively. The intelligibility analysis results suggest that LBS has a better performance than RS.

7 Conclusion

In this paper, we have investigated UPT avoidance with multiple multimedia flows. We proposed a framework called *Unintelligible Flow Management* (UFM) to address problems that arise with UPTA in networks supporting multiple multimedia applications. We have presented the UFM framework using WFQ as an example. We have proposed two different UFM policies in this paper. They are *Random Select* (RS) and *Least Bandwidth Select* (LBS). We have incorporated RS/LBS into WFQ. By using OPNET Modeler and MPEG-2 video, we have conducted extensive simulation study to evaluate the performance of RS and LBS. We have used three performance metrics: number of video flows recovered, TCP throughput improvement and video intelligibility. We have simulated four different network scenarios: single/multiple congested links and homogeneous/heterogeneous multimedia applications. Following conclusions can be drawn from our simulation results: (i) it is clear that the bottleneck link may be under-utilised without UFM (i.e. RS or LBS), and TCP sources may receive unfairly more bandwidth than video flows. UFM can effectively maximise link utilisation (full utilisation can be achieved) by converting selected U flows into I flows. (ii) both RS and LBS can significantly improve TCP throughput (up to 224% in our simulated networks), as compared with plain WFQ. (iii) both RS and LBS can effectively improve average intelligibility index of video applications. (iv) RS and LBS yield similar performance under homogeneous video. However, LBS out-performs RS under heterogeneous video.

Our simulation results also suggest that RS and LBS have identical performance with homogeneous multimedia applications. In such a situation, light-weighted RS may be sufficient. This will avoid

Table 12: Comparison of Average Intelligibility Index for Video Applications (multiple congested links; heterogeneous video)

Flow ID	*WFQ*	*WFQ+*	*WFQ++(RS)*	*WFQ++(LBS)*
1	0.0	0.0	0.0	0.9466
2	0.0	0.0	0.0	0.9387
3	0.0	0.0	0.9466	0.9307
4	0.0	0.0	0.0	0.9227
5	0.0	0.0	0.9307	0.9147
6	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0
8	0.0	0.0	0.9387	0.0
9	0.0	0.0	0.9227	0.0
11	0.0	0.0	0.9280	0.9440
12	0.0	0.0	0.9360	0.9360
13	0.0	0.0	0.0	0.9280
14	0.0	0.0	0.0	0.9200
15	0.0	0.0	0.0	0.9120
16	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.9440	0.0
19	0.0	0.0	0.0	0.0
Average	0.0	0.0	0.3637	0.5163

the complexity of LBS. However, with heterogeneous multimedia applications, it may be justifiable to implement LBS, as LBS out-performs RS in terms of number of video flows recovered as well as video intelligibility.

There remains several open issues. First, 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. Secondly, we only consider UPT due to excessive packet loss rate in this thesis. Other QoS parameters (e.g. delay and delay jitter) may also contribute to UPT. The impacts of delay and delay jitter have not been investigated in this thesis. To detect UPT due to unacceptable delay/jitter, routers are required to estimate a number of delay components, including propagation delay, transmission delay, processing delay and queueing delay. A challenging issue here is how to accurately determine the combined effect of delay/jitter at each individual router. Lastly, we assume that no error control or error concealment is implemented in our MPEG-2 video sources/destinations. If a video/audio host implements these techniques, it may be able to tolerate larger packet loss rate. A possible enhancement to UPTA is to make UPTA aware of the presence these video/audio repair algorithms, so UPTA can adjust its operation accordingly to secure optimum performance.

References

[1] Zhiruo Cao, Zheng Wang, and Ellen Zegura. Rainbow Fair Queueing: Fair Bandwidth Sharing without Per-Flow State. In *Proceedings of IEEE INFOCOM 2000*, volume 2, pages 922–931, Tel Aviv, Israel, March 2000.

[2] N. Chapman and J. Chapman. *Digital Multimedia*. John Wiley & Sons, 2000.

[3] Wu-chang Feng. *Improving Internet Congestion Control and Queue Management Algorithms*. PhD thesis, University of Michigan, 1999.

[4] Wu-chang Feng, Dilip D. Kandlur, Debanjan Saha, and Kang G. Shin. BLUE: A New Class of Active Queue Management Algorithms. Technical Report CSE-TR-387-99, University of Michigan, Ann Arbor, MI 48105, April 1999.

[5] F. Fluckiger. *Understanding Networked Multimedia: Applications and Technology*. Prentice Hall, 1995.

[6] Cisco Systems Inc. Advanced QoS Services for the Intelligent Internet. http://www.cisco.com/warp/public/cc/pd/iosw/index.shtml.

[7] James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison Wesley, 2001.

[8] Dong Lin and R. Morris. Dynamics of Random Early Detection. In *Proceedings of ACM SIGCOMM*, pages 127–137, October 1997.

[9] Nortel Networks. http://www.nortelnetworks.com/products/01/passport/wan/p5430/doclib.html.

[10] Rong Pan, Balaji Prabhakar, and Konstantinos Psounis. CHOKe, A Stateless Active Queue Management Scheme for Approximating Fair Bandwidth Allocation. In *Proceedings of IEEE INFOCOM 2000*, volume 2, pages 942–951, Tel Aviv, Israel, March 2000.

[11] A. Parekh and R. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: the Single-Node Case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

[12] I. Stoica, S. Shenker, and H. Zhang. Core-Stateless Fair Queueing: Achieving Approximately Fair Bandwidth Allocations in High Speed Networks. In *Proceedings of ACM SIGCOMM*, pages 118–130, Vancouver, Canada, September 1998.

[13] OPNET Technologies. OPNET Modeler 7.0.B. http://www.opnet.com/.

[14] Jim Wu and Mahbub Hassan. Avoiding Useless Packet Transmission for Multimedia over IP Networks: The Case of Multiple Congested Links. Technical Report UNSW-CSE-TR-0215, School of Computer Science & Engineering, The University of New South Wales, Sydney, Australia, November 2002. ftp://ftp.cse.unsw.edu.au/pub/doc/papers/UNSW/0215.ps.Z.

[15] Jim Wu and Mahbub Hassan. The Issue of Useless Packet Transmission for Multimedia over the Internet. *Computer Communications*, accepted for publication.