Adaptive Feedback Algorithm for Internet Video Streaming based on Fuzzy Rate Control

Pavlos Antoniou, Andreas Pitsillides and Vasos Vassiliou
Networks Research Laboratory, Department of Computer Science, University of Cyprus*
75 Kallipoleos Street, P.O. Box 20537, 1678 Nicosia, Cyprus
e-mail: paul.antoniou@cs.ucy.ac.cy, andreas.pitsillides@ucy.ac.cy, vasosv@cs.ucy.ac.cy

Abstract

It is beyond any doubt that the unpredictable nature of the today's Internet has tremendous impact on the transmission of video streams. With respect to the real-time nature of video streaming, unpredictable bandwidth, end-to-end delay and packet loss, are all problems that can affect video streaming over the Internet. The problem is worsened when we consider wireless and mobile users. In this paper we adopt Network Adaptation Techniques applied together with Content Adaptation Techniques to achieve graceful performance degradation as network load increases. We present a novel feedback mechanism and a fuzzy oriented decision algorithm which collectively provide video adaptation to network parameters. Our preliminary performance evaluations indicate that the fuzzy-based algorithm can finely sense the available bandwidth of the network path and adapt the video transmission rate accordingly, maintaining acceptable video quality.

1. Introduction

In the last few years we have witnessed an ever-increasing prevalence of heterogeneous, video-enabled devices such as computers, mobile phones, and PDAs. The use of these devices has multiplied the need for efficient and effective techniques for adapting compressed video streams to suit better the different constraints, capabilities, and requirements of various transmission networks, applications, and end users.

One of the most significant problems that video communications face is the unpredictable nature of the Internet in terms of bandwidth, end-to-end delay and loss variation. Therefore, video streaming applications need to implement highly scalable and adaptive techniques in terms of content encoding and transmission rates in order to cope with the erroneous and time variant conditions of the network.

Since the bandwidth between two points in the Internet is generally unknown and time-varying, the goal is to estimate the available bandwidth and then match the transmitted video bit rate to it. Network Adaptation Techniques (NATs) deal with the end-to-end adaptation of real time multimedia application needs to the network parameters using algorithms which take into account the state and/or load of the network and the type of errors. Content Adaptation Techniques (CATs) deal with adaptation of content to the desirable transmission rate using primarily scalable video approaches.

We aim to address the problems arising in both fixed and mobile environments by combining NATs with CATs in order to finely adapt the video stream bit rate to the dynamically changing network parameters. In this paper, we propose a new feedback mechanism that works in co-operation with a fuzzy-based adaptation decision algorithm. Fuzzy control may be viewed as a way of designing feedback controllers in situations where rigorous control theoretic approaches can not be applied due to difficulties in obtaining formal analytical models.

The rest of the paper is organized as follows: Section 2 describes the fuzzy control system. Section 3 deals with the evaluation of the fuzzy rate controller. Section 4 presents the simulation model and QoS assessment framework for layered video transmission. Section 5 involves evaluation setup and scenarios. Section 6 presents some preliminary results. Finally, Section 7 concludes the paper and discusses future work.

2. Fuzzy Control System

Under rate-based control, a sender sends data packets strictly based on the estimated rate which is progressively tuned to avoid congestion. The estimation

*This work was partly funded by the UCY ADAVIDEO project.

of the available network bandwidth is based on feedback information. Rate-based control can be classified into two approaches, namely, the probe-based and the equation-based approach. A probe-based approach is based on probing experiments. Specifically, the source probes for the available network bandwidth by adjusting the sending rate so that some QoS requirements are met (e.g. packet loss fraction). Contrary to probe-based approach, where the sender implicitly estimates the available network bandwidth, the equation-based approach attempts to estimate the available network bandwidth explicitly. This can be achieved by using, for example, a throughput formula [6].

Our fuzzy rate control system follows the probe-based approach, using an adaptive feedback mechanism and a fuzzy-based adaptation decision algorithm. A video stream is transmitted over an RTP connection. RTP operates in cooperation with RTCP which provides the information regarding the connection quality.

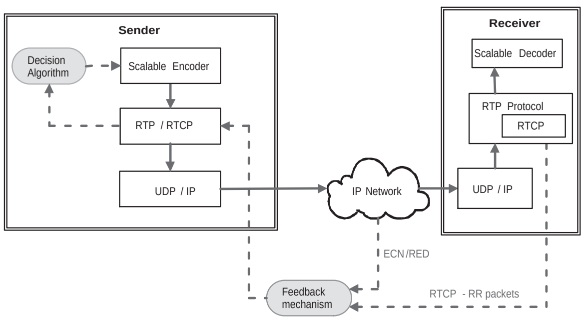


Figure 1. Fuzzy Rate-based System.

Figure 1 illustrates our unicast-oriented fuzzy rate control system. The two additional components, namely, feedback mechanism and decision algorithm, focus on the adaptation of the video content to the available network bandwidth. Dashed arrows track the path of control packets (RTCP packets) whereas solid arrows track the path of video data packets (RTP packets).

The feedback mechanism collects QoS information like loss rate and jitter from both the core network and the receiver that will be used for the evaluation of the available bandwidth of the path between the sender and a receiver. The decision algorithm which is implemented at the sender side, processes the feedback information and decides the optimum number of layers that will be sent. The role of the feedback and adaptation components is to link the quality demand of video-enabled applications to the underlying network state leading to network adaptation. Network adaptation should be assisted by a content adaptation technique which is carried out by a scalable video encoder.

The rest of this section is organized as follows: Section 2.1 deals with scalable encoding. Section 2.2 presents the feedback mechanism in detail. Section 2.3 analyzes the fuzzy decision algorithm.

2.1. Scalable Encoding

Scalable encoding is suitable for adapting the quantity of data transmitted by a video server to the capacity of a given network path. Video streams are encoded in a layered manner in a way that every additional layer increases the perceived quality of the stream. Usually a layered video stream consists of a base layer and several additional enhancement layers. Base layers should be encoded in a very low rate so as to accommodate for a large variety of mobile handheld devices as well as terminals connected to the Internet through low bandwidth modem connections. Additional enhancement layers are added, or dropped, in order to adapt the content rate to the desirable transmission rate.

2.2. Feedback Mechanism

As mentioned before, the feedback mechanism collects information from both the core network and the receiver in order to evaluate the available bandwidth. Each receiver sends reception statistics using RTCP packets. In accordance with [8], dedicated RTCP packets called Receiver Report (RR) packets are sent from participants that are not active senders and carry reception statistics. RRs provide (a) loss fraction which denotes the recent quality of the distribution, (b) cumulative number of packet lost (CNPL), (c) highest sequence number received (EHSR) and (d) inter-arrival jitter. The packet loss fraction within an interval is given by the number of packets expected divided by the number of lost packets during the interval. The loss rate per second (LRPS) can be obtained by dividing the loss fraction by the difference in RRs timestamps. The difference between two successive values of LRPS can be used in order to track the increasing or decreasing trend of packet loss percentage.

Additionally, network elements (i.e. routers within the network path) may explicitly notify the sender about the current status of congestion within the core network. These notifications can be efficiently used for the evaluation of the available bandwidth. The Explicit Congestion Notification (ECN) mechanism mentioned in [4] is used for the notification of congestion to the end nodes in order to prevent unnecessary packet drops. ECN option allows active queue management (AQM) mechanisms such as, for example RED [5] or Fuzzy-RED [3] to probabilistically mark packets. The number of marked packets within a given period may provide a meaningful reference about the congestion status. The receiver collects these data and sends them back to the sender using a dedicated field of the RR packets.

2.3. Fuzzy Decision Algorithm

Fuzzy control may be viewed as a way of designing feedback controllers in situations where rigorous control theoretic approaches can not be applied due to difficulties in obtaining formal analytical models.

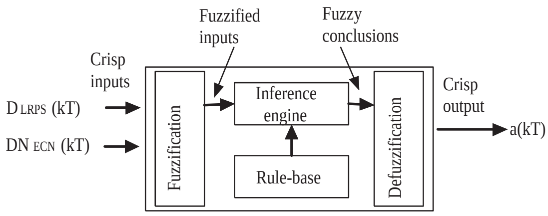


Figure 2. Fuzzy Logic Control System.

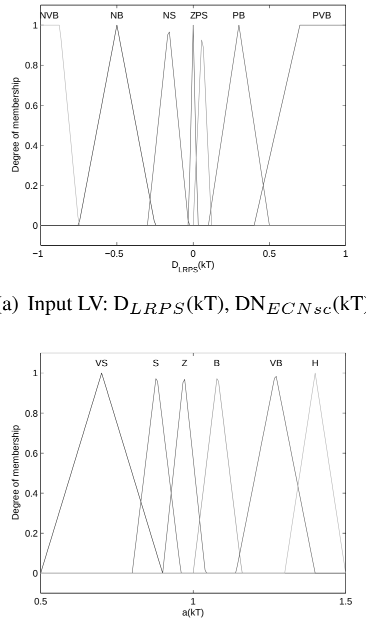
Our fuzzy system is shown in Figure 2. In general, a fuzzy logic controller consists of four modules, namely, fuzzification module, defuzzification module, fuzzy inference engine, and fuzzy rule-base. The key concepts in fuzzy logic are the linguistic variables (LV) and the membership functions. Linguistic variables take on linguistic values which are words (linguistic terms) that are used to describe characteristics of the variables. Each of these linguistic terms is associated with a fuzzy set defined by a corresponding membership function. Actually a membership function is a curve that defines how each linguistic term is mapped to a membership value (or degree of membership) between 0 and 1. Figure 3 illustrates the triangular membership functions used for the input and output linguistic variables. The choice of triangular membership functions is a common one, and is mostly based on their simplicity.

The fuzzification module transforms the crisp inputs into linguistic terms (fuzzy sets). After this, linguistic terms are processed by the inference engine based on the rule base. The process of fuzzy inference involves membership functions, fuzzy logic operators, and if-then rules. The defuzzification module provides the transformation of linguistic terms back to crisp values. Our fuzzy control system is based on two linguistic input variables and one linguistic output variable as shown in Figure 2. All quantities in our system are considered at the discrete instant kT, with T the decision period.

Our first linguistic input variable involves the LRPS parameter. LRPS(kT) is the loss rate per second at each decision period and LRPS(kT−T) is the loss rate per second with a delay T. The linguistic variable D_{LRPS}(kT) gives the increasing or decreasing trend of the LRPS and can be evaluated by:

$$D_{LRPS}(kT) = LRPS(kT) - LRPS(kT - T) \quad (1)$$

LRPS is lower and upper bounded by 0 and 1 respectively. Thus, D_{LRPS}(kT) ranges from -1 to $+1$.



(a) Input LV: D_{LRPS}(kT), DN_{ECNsc}(kT)

(b) Output LV: a(kT)

Figure 3. Membership Functions.

Furthermore, we considered the number of packets that have the ECN bit set within a period, as a strong indication for congestion. The receiver calculates periodically this number called N_{ECN}(kT). The sender extracts this value from an RR packet and calculates a scaled parameter, N_{ECNsc}(kT), which ranges from 0 to $+1$, and represents the percentage of packets marked within this period. Eq. 2 is used to obtain the scaled parameter N_{ECNsc}(kT):

$$N_{ECNsc}(kT) = \frac{N_{ECN}(kT)}{N_{ps}(kT)}, \quad (2)$$

where N_{ps}(kT) is the number of packets sent within the same period. Therefore, we are able to calculate the parameter DN_{ECNsc}(kT), which gives the increasing or decreasing trend of the number of marked packets. DN_{ECNsc}(kT) is upper and lower bounded by $+1$ and -1 respectively, and can be evaluated by:

$$DN_{ECNsc}(kT) = N_{ECNsc}(kT) - N_{ECNsc}(kT - T) \quad (3)$$

Our linguistic output variable, a(kT), is defined for every possible combination of inputs. The defuzzified crisp values of a(kT) can be used by the decision algorithm for the evaluation of the available bandwidth using the formula:

$$avail\_bw(kT) = a(kT) * avail\_bw(kT - T) \quad (4)$$

The defuzzified output value is selected to range from 0.5 to 1.5. Thus a 'gradual' increase is allowed when there is available bandwidth and reduced congestion, whereas quick action is taken to reduce the rate to half in case of severe congestion. The output of the fuzzy system can be used as input to a scalable video encoder (see Figure 1). Driven by the estimated available bandwidth, a scalable video encoder can encode a video stream in a fine-grained manner so as to meet the network requirements. Alternatively, the decision algorithm is able to choose one of several pre-encoded video streams and exclusively sends from that stream until it decides to change the video quality by adding/dropping layers.

Table 1 involves if-then rule statements which are used to formulate the conditional statements that comprise fuzzy logic.

Table 1. Linguistic Rules[1].

a(kT)		DN_{ECNsc}(kT)						
		NVB	NB	NS	Z	PS	PB	PVB
D_{LRPS}(kT)	NVB	H	H	B	B	Z	S	VS
	NB	H	VB	Z	Z	Z	S	VS
	NS	B	Z	B	Z	Z	S	VS
	Z	B	Z	Z	B	Z	S	VS
	PS	Z	Z	Z	Z	S	S	VS
	PB	Z	Z	Z	Z	S	S	VS
	PVB	S	S	S	S	VS	VS	VS

Our decision algorithm has to decide which layers should be sent according to the available network bandwidth, based on a non aggressive layer selection approach. The server will host an appropriate number of layers where each layer corresponds to a different transmission rate. To avoid ping-pong effects there should not be a transition to an upper level layer every time the available bandwidth exceeds the threshold of a specific transmission rate that corresponds to a higher video layer. Instead, a time hysteresis is introduced in order to avoid frequent transitions from one layer to another which may cause instability. In the case of a transition to a lower layer, the effect is immediate, as we seek quick relief from possible congestion.

The time hysteresis is equal to the time interval between the reception of two successive RR packets. If the available bandwidth exceeds the threshold of a specific transmission rate that corresponds to an upper level layer, then the hysteresis variable is set. When a new RR packet arrives, if the available bandwidth is still at the same levels, a transition occurs. This is shown in the pseudo-code of the decision algorithm below:

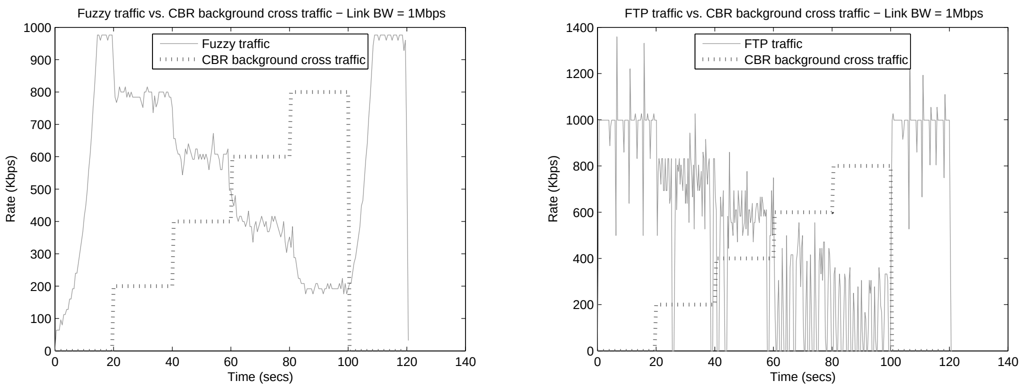
```
For all layers j up to MAX_LAYER
        If avail_bw ≤ BitRate(j) and hysteresis=false {
            If layer(j) = current_layer
                break;
            hysteresis = true;
            selected_layer = layer(j);
        }
        Else If avail_bw ≤ BitRate(j) {
            hysteresis = false;
            If selected_layer < layer(j) {
                current_layer = selected_layer;
                break;
            }
            else {
                current_layer = layer(j);
                break;
            }
        }
```

3. Fuzzy Rate Controller Evaluation

We investigate in ns2 [2] the ability of the fuzzy rate controller to sense the available bandwidth of a bottleneck link in the presence of multiple CBR connections which are superimposed progressively and FTP background cross traffic and adapt the transmission rate of a 1Mbps layered CBR non trace-based video stream. The propagation delay across the link was set to 10ms. We considered RED-enabled routers having buffer capacity of 50 packets (other AQM mechanisms could also be adopted, e.g., [3]). The min_{th} and max_{th} of each queue were set to 10 and 30 packets respectively and the p_{max} to 0.1. Moreover, the interval T between transmissions of RR packets was set to 0.3 seconds. The selection of 0.3 seconds is dictated by the desire to maintain responsiveness to changes in the network state. Further analysis of T, including sensitivity, responsiveness and signalling load, is planned for future work.

Figure 4(a) depicts the instantaneous transmission rate of the layered CBR video stream as the CBR cross traffic rate changes over the time. The bottleneck link bandwidth is 1Mbps and the CBR cross traffic rate increases from 200Kbps to 800Kbps. As can be seen, the

[1]Table Content Notations: Negative/Positive Very Big (NVB, PVB), Negative/Positive Big (NB, PB), Negative/Positive Small (NS, PS), Zero (Z), Very Small/Big (VS, VB), Small/Big (S, B), Medium (M), Huge (H).



(a) Fuzzy traffic rate

(b) FTP traffic rate

Figure 4. Instantaneous rate for 1Mbps bottleneck link with CBR cross traffic connections.

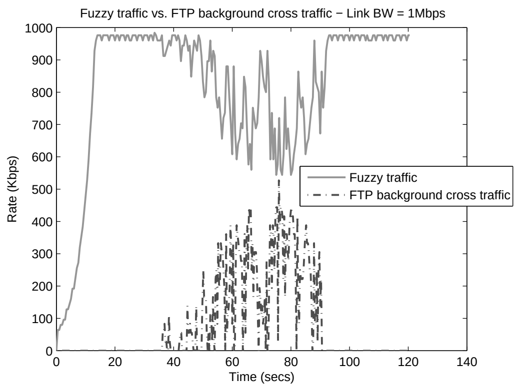


Figure 5. Instantaneous rate for 1Mbps bottleneck link with FTP cross traffic.

video transmission rate driven by the fuzzy rate controller, evolves at a slow and smooth pace in order to respond to the network and quality conditions, but also prevent unnecessarily many fluctuations. Figure 4(b) illustrates the FTP sending rate evolution under the same circumstances of CBR cross traffic rate. The FTP sending rate reveals the classic saw tooth pattern of TCP behavior and its inappropriateness to support real-time video streaming. On the other hand, our fuzzy rate controller estimates accurately the available bandwidth and then matches the transmitted video bit rate to it.

Figure 5 illustrates the transmission rate of the layered CBR video stream in the presence of FTP background cross traffic. Although the FTP cross traffic is more bursty than CBR cross traffic shown in Figure 4(a), the fuzzy controller senses the available capacity of the bottleneck link and finely adapts the video rate to it. The fuzzy-controlled flow appears to be TCP-friendly against an FTP flow, as it does not aggressively consume the available bandwidth.

4. Simulation Model and QoS Assessment Framework for Layered Video

The performance of our decision algorithm and its ability to support layered video streaming were investigated through simulations conducted using the ns2. Due to the inadequacy of the existing ns2 modules, we implemented some new software modules (Figure 6).

The four new modules that were implemented, namely, VideoRTPAgent, VideoRTCPAgent, VideoTrafficTrace, and RTPSession are shown in Figure 6 (shaded parts). VideoRTPAgent monitors the transmission of video streams in terms of packet processing. VideoTrafficTrace is used for the simulation of layered trace-based video transmission, working in co-operation with VideoRTPAgent and RTPSession. Every VideoTrafficTrace instance corresponds to a new layer. Every layer is actually a replicated instance of the raw video stream encoded (using FFmpeg [1]) in different bit rate. The encoded video stream is parsed and the trace file created is attached on a new VideoTrafficTrace instance. RTPSession module needs to monitor the layered video transmission mechanism. Thus, the decision algorithm is entirely embedded in RTPSession and all the VideoTrafficTrace instances must be attached on it.

The RTCP protocol functionality is divided between two modules: VideoRTCPAgent and RTPSession. The VideoRTCPAgent module implements the RTCP packet processing, while the RTPSession module implements the RTP session management, as mentioned before. The

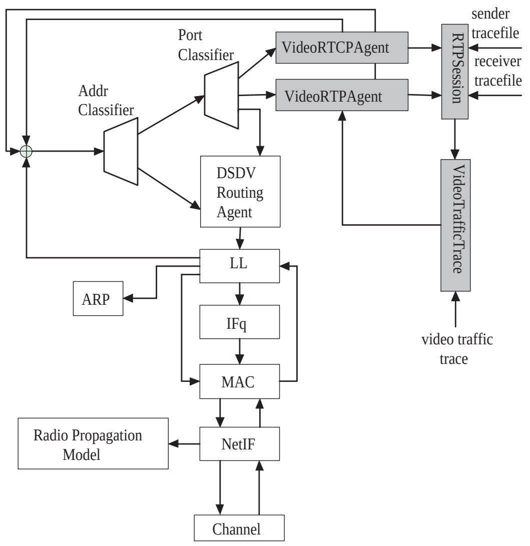


Figure 6. Enchanced Node Model in NS2.

RTCP protocol behavior is based on the periodic transmission of control packets. Every packet is prepared by the RTPSession in response to the VideoRTCPAgent request. The VideoRTCPAgent simulates the transmission and reception of the RTCP RR packets.

Simulation results have to be interpreted in terms of Quality of Service (QoS). For this reason we adopted Evalvid [7] which is a complete framework and tool-set for evaluation of the quality of video transmitted over a real or simulated communication network. Besides measuring QoS parameters of the underlying network, like loss rates, delays, and jitter, Evalvid also supports a subjective video quality evaluation in terms of PSNR.

5. Evaluation Setup and Scenarios

Figure 7 illustrates the topology we used in the performance evaluation. The topology consists of two routers directly connected with a link having variable characteristics. A video streaming server is attached to the first router. Mobile wireless clients are connected to the second router over wireless links. In order to make our scenarios more realistic we added background traffic initiated by the FTP server.

In order to simulate the video traffic patterns, we used a well known real test video sequence named Foreman which involves fair amount of movement and change of background. The sequence has temporal resolution 30 fps and spatial resolution 176x144. The video sequence was encoded in MPEG4 using the publicly available software tool called FFMPEG encoder [1]. We encoded this sequence in 8 different bit rates;

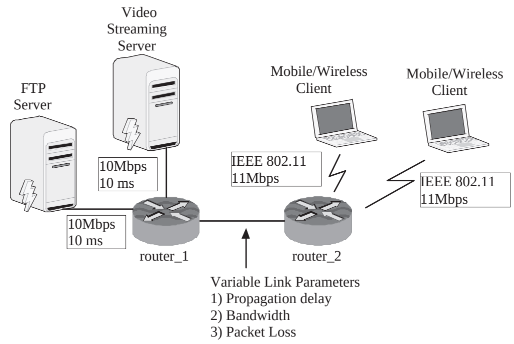


Figure 7. Evaluation Topology.

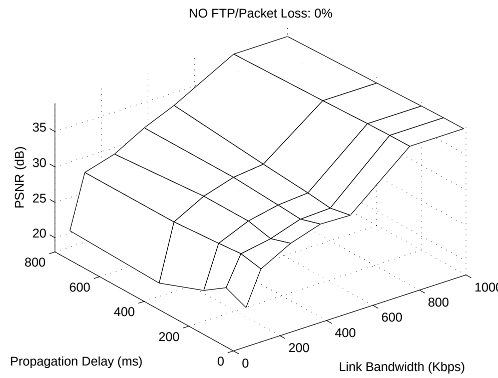
64, 96, 128, 192, 256, 384, 512 and 768Kbps. Each encoded video stream corresponds to a separate layer.

We set the maximum capacity of each buffer to 50 packets, the $\min_{th}$ and $\max_{th}$ of the queue as 10 and 30 packets respectively and the p_{max} to 0.1. The interval T between RR packets was set to 0.3 seconds.The bandwidth of the variable link was selected to range from 64Kbps to 1Mbps, while the propagation delay varies from 10ms to 800ms. In addition, we considered packet loss of 0% and 5%. The choice of the parameters used in the video quality evaluations was based on the representative characteristics of wired and wireless networks.

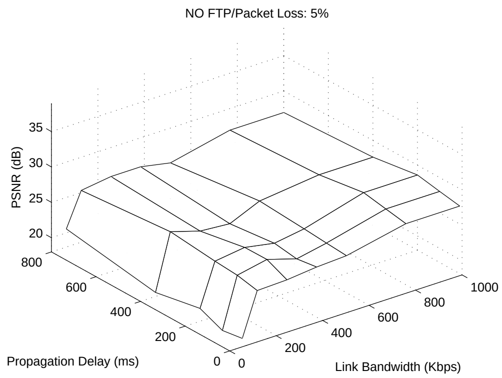
6. Results

In this section we present scenarios involving one and two mobile wireless users. Video quality is measured by taking the average of the Peak Signal-to-Noise Ratio (PSNR) over all the decoded frames.

The effect of propagation delay and link bandwidth on the PSNR in the absence of background traffic is presented in Figure 8. The results obtained by scenarios where the packet loss is 0% (Figure 8(a)) reveal that the PSNR values are increasing at a steady pace (up to 38dB) as the link bandwidth increases. PSNR values are decreased (less than 25dB) in scenarios where the link bandwidth is equal to the bit rate of the lowest layer (64Kbps), since there is a strong possibility of packet loss. Figure 8(b) presents the objective quality evaluations obtained by scenarios involving packet loss of 5%. Obviously the values of PSNR have been significantly decreased compared to those of Figure 8(a). This is because the decision algorithm recognizes the high packet drop rates and strives to maintain an acceptable level of video quality, whilst satisfying the worsening network state, by sending fewer layers, resulting in lower PSNR values. The PSNR metric partially ignores the effect of the propagation delay, but as it can be seen, the delay



(a) Packet Loss = 0%



(b) Packet Loss = 5%

Figure 8. Mean PSNR vs. Link BW and Prop. Delay, No FTP.

can indirectly influence the objective quality of a video stream. Actually, the larger the delay the larger the interval between reception of two successive RR packets. Under these circumstances, the system will experience delayed decision-making that will influence the quality of the video stream. As shown in Figure 8(a) and (b), PSNR values are slightly increased for low delay values especially regarding scenarios involving high bandwidth links. This is because the content adaptation to network parameters evolves at a faster pace. All in all, delay does not influence the values of PSNR in the same way as the link bandwidth does.

Figure 9 shows PSNR values for scenarios involving background FTP traffic while the packet loss is 0%. We observe a decrease in PSNR values for scenarios having link bandwidth less or equal to 256Kbps due to the excessive FTP traffic load. As the link bandwidth increases (more than 256Kbps), the quality of a video stream is not severely affected by the FTP traffic since the decision algorithm adjusts the number of layers sent according to the network conditions. The PSNR values are slightly fluctuating due to the saw-tooth behavior of the FTP sending rate evolution, which forces the video streaming server to add/drop layers accordingly. We perceive a lower objective quality for low delay values, because the FTP sending rate evolves at a faster and aggressive pace compared with scenarios with larger delay resulting in higher packet drop rates.

Figures 10(a) and (b) show the PSNR values for different values of propagation delay. Both figures reveal that in the case of two users, our algorithm provides fairness because no one of the two users takes advantage over the other, as both users perceive almost the same quality. Figure 10(a) shows that the PSNR values in case of one mobile user and loss of 0%, outperform all

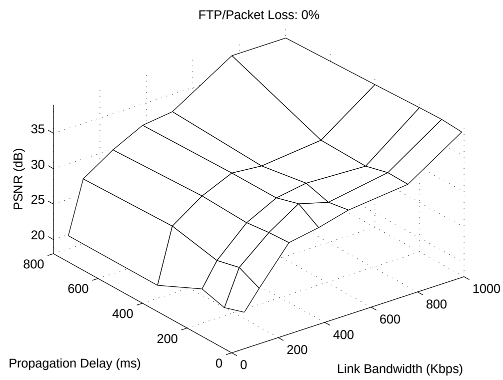
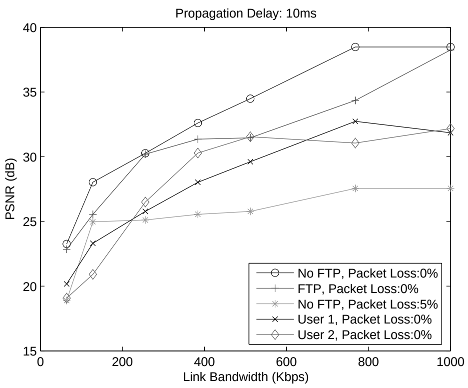


Figure 9. Mean PSNR vs. Link BW and Prop. Delay, FTP, Packet Loss = 0%.

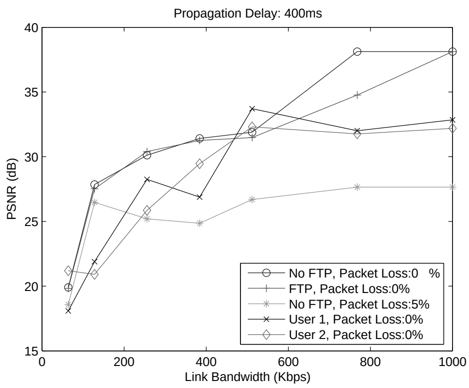
the others. Similarly, a mobile user in the presence of FTP traffic, exhibits higher PSNR values compared to the scenarios involving two users, because FTP sending rate is actually lower than the cumulative video sending rate. On the other hand, scenarios involving one user and loss of 0% (Figure 10(b)), exhibit lower PSNR values than before, because the adaptation evolves at a slower pace. The objective quality under FTP traffic ranges at slightly higher levels than before because FTP sending rate evolves at a slower pace, which means that the influence on the adaptive flow is lessened.

7. Conclusions and Future Work

In this paper we present an adaptive video transmission algorithm specifically designed for video stream-



(a) Prop. Delay = 10ms



(b) Prop. Delay = 400ms

Figure 10. Mean PSNR vs. Link BW.

ing over the Internet. Our main objective is to provide a framework that incorporates both CATs and NATs. Towards this direction, we introduce two new components; a feedback mechanism and a decision algorithm, that deal with layered video streams.

We evaluated our fuzzy rate control system under conditions of high congestion across a bottleneck link. Simulations showed that the fuzzy controller can finely adapt the video transmission rate to the available bandwidth of the link, based on loss rate per second and percentage of marked packets over a decision period T. Basically, fuzzy controller clearly detects the available bandwidth in the presence of CBR or FTP background cross traffic, and finely adapts the video transmission rate to it. Moreover, our fuzzy system achieves smooth rate change over the time, an appealing feature for video streaming over the Internet.

We evaluated our decision algorithm under error-free and error-prone environments in the presence of mobile wireless users and our preliminary results indicate that the algorithm can finely adapt the video stream bit rate to the available bandwidth, while providing high and stable objective quality of service at the same time. Moreover, simulations showed that the system performs best in the absence of background traffic like FTP but the objective quality remains acceptable in the presence of background FTP as well. Additionally, preliminary results indicate that our algorithm provides fairness, however, this is an issue which will be further investigated in the presence of multiple concurrent users.

For future work we would like to determine the sensitivity of our algorithm to various parameters (i.e time hysteresis, decision period T). To continue with further evaluation of our adaptation approach, we need to look at the interaction between our adaptive flow and other network flows sharing the same routers. In addition, the effect of delay variation (jitter) will be taken into consideration when designing the fuzzy inference engine. Moreover, subjective tests should be considered given the fact that PSNR is inappropriate for the evaluation of the actual user perceived quality of service because it is poorly correlated to human vision.

8. Acknowledgement

Many thanks to Dr. Chrysostomos Chrysostomou of UCY for assisting us in the implementation of the fuzzy logic system.

References

[1] FFmpeg site. http://ffmpeg.mplayerhq.hu/.
[2] NS2 site. http://www.isi.edu/nsnam/ns/.
[3] C. Chrysostomou, A. Pitsillides, G. Hadjipollas, A. Sekercioglou, and M. Polykarpou. Fuzzy Explicit Marking for Congestion Control in Differentiated Services Networks. *8th IEEE ISCC*, pages 312–319, 2003.
[4] S. Floyd. TCP and Explicit Congestion Notification. *ACM Comp. Comm. Review*, 24(5):8–23, Oct. 1994.
[5] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. on Networking*, 1:397–413, Aug. 1993.
[6] M. Handley, S. Floyd, J. Padhye, and J. Widmer. TCP Friendly Rate Control (TFRC): Protocol Specification. *RFC 3448*, Jan. 2003.
[7] J. Klaue, B. Rathke, and A. Wolish. Evalvid - A Framework for Video Transmission and Quality Evaluation. *13th Intl. Conf. on Modelling Techniques and Tools for Computer Performance Evaluation*, Sep. 2003.
[8] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. *RFC 3550*, Jul. 2003.