DELAY AND LOSS FUZZY CONTROLLED VIDEO OVER IP NETWORKS

E Jammeh, M. Fleury and M. Ghanbari

University of Essex
ESE Department
Colchester
UK

ABSTRACT

The performance of rate adaptive multimedia streaming applications is impacted by the accuracy of determining the level of network congestion, and by the control mechanism used to adapt the sending rate to the available network bandwidth. The focus of this paper is on the use of fuzzy logic for congestion avoidance and control of networked video, in combination of using delay as a network congestion level indicator, and packet loss as an indicator of full blown congestion. It is shown through simulation results that using delay to gauge the level of network congestion, and packet loss to signal full blown congestion, in combination with fuzzy logic control, can enhance the performance of video streaming, under varying network congestion states. Furthermore, it is shown that this scheme enables rate adaptive video streaming applications to optimally adapt to the available network bandwidth.

1. INTRODUCTION

The performance of networked video communication applications is highly dependent on a) the accuracy with which the available network bandwidth, or network congestion level is determined b) the accuracy of the rate controller that takes the measured network status as input, and compute a new sending rate that accurately reflects the network state with respect to the current sending rate and, c) the efficiency and performance of the rate adaptation unit that changes the video quality to match the desired sending rate.

The accuracy and timeliness of determining the state of the network, is one of the key components of an adaptive video streaming architecture. Several metrics such as buffer fullness, round trip times and packet loss, have all been used for network congestion level determination. However, it is packet loss that has traditionally been used by TCP to signal congestion with remarkable success in avoiding excessive Internet congestion [1].

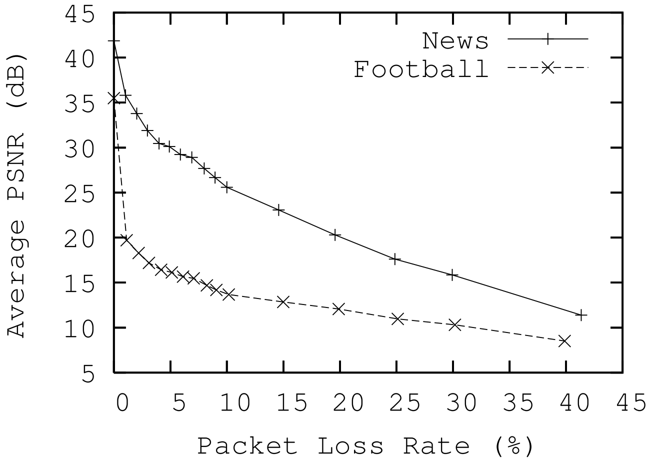

Fig. 1. The effect of packet loss on transported video quality

The current Internet have had an explosive growth in size, speed, distance, and types of applications. The limitation of packet loss for congestion notification, in the currently deployed Internet, has generally been accepted, especially for the newer high-bandwidth delay product networks. It has also been suggested that packet delay is a better gauge of incipient congestion than packet loss which has a one-bit information on network congestion; congested or not congested. Consequently, many enhancements to the TCP protocol have been proposed, with packet delay as a multi-bit congestion level indicator [2][3][4].

Packet loss drastically impacts the quality of networked video, especially since any reference picture packet loss is significant, as the effects are felt subsequently until the next intra-coded picture. Figure 1 shows what effect an increasing packet loss rate has on networked video. For a packet loss rate of 5% the video quality for a low motion news clip dropped by over 10dB and by nearly 20dB for a high motion sports clip.

It is evident that rate adaptive video streaming appli-

This work was supported by the EPSRC, UK under grant no. EP/C538692/1.

cations should reduce the video quality in response to network bandwidth constrictions rather than incur packet loss. Nonetheless, TCP emulators [5] for video transport over UDP do include a packet loss factor in their models. Furthermore, the subjective video quality suffers with huge variations in video quality (rate), and any control strategy that employs the Additive Increase Multiplicative Decrease (AIMD) control will have negative impact on the subjective video quality.

This paper investigates the effectiveness of using delay (OWD) as a multi-bit congestion state signal for rate adaptive video transport, augmented by the one-bit packet loss information that signals full blown congestion. Central to the proposed control algorithm is the use of fuzzy logic to control the sending rate of the networked video. The uncertainties and possible measurement noise with the need for a real-time solution, make fuzzy logic control attractive.

1.1. Delay-based congestion level determination

Assume an application with input sending rate R_i and tight link capacity C_b. For duration δt, the total incoming traffic, T_i into a tight link is given by

$$T_i = (R_i + R_c) \times \delta t \tag{1}$$

where R_c is the cross-traffic rate.

There will be no backlog of data at this link if the tight link capacity is greater than the total incoming traffic. However, excess data will be buffered by the tight link router when the incoming rate is greater than the outgoing rate. This buffering results in a queuing delay, q_d, which is a function of the difference between the input and output rates of the tight link:

$$q_d = \frac{R_o - T_i}{C_b}. \tag{2}$$

where R_o is the output rate of the tight link.The queueing delay q_d, which adds to the OWD of packets, is an therefore indicator of network congestion.

Minimum OWD in theory occurs when there is no queuing delay on the network path, with propagation delay being the only contributor. There will normally be some buffering delay included in the OWD measurement but q_d at a tight link will change as the link becomes congested. Increase in OWD is a function of an increase in buffer fullness, which in turn is a function of the level of network congestion. Maximum queueing delay occurs when the buffer at the tight link becomes full and packets start to be dropped: full blown congestion. The queuing delay of packets with respect to the minimum measured OWD is a measure of the tight link's buffer fullness which we call multi-bit information, as it has a level which ranges between the minimum measured and maximum possible delay.

1.2. Fuzzy Logic Congestion Control

FLC [6] is a convenient tool for handling un-modeled network congestion states. It allows the intuitive nature of congestion reduction to be applied through linguistic variables [8]. Within video coding it has found an application [9][10] in maintaining a constant video rate by varying the encoder quantization parameter according to the output buffer state, which is a complex control problem without an analytical solution.

Rather than a single, invariant throughput equation to emulate TCP's behavior [11], FLC provides a flexible method of adapting to evolving Internet traffic patterns, as precise mathematical representations are difficult to devise for such a complex, non-linear control problem [12]. For example, the same controller should be able to cope with a range of Internet path delays and with video streams with differing characteristics in terms of scene complexity, motion, and scene cuts.

The remainder of this paper is organized as follows. As the key to successful closed loop congestion control is the nature of the feedback information Section 2 examines previous delay-based mechanisms. Section 3 principally presents the FLC methodology, while Section 4 details our simulation experiments to date. Finally, Section 5 draws some conclusions.

2. RELATED WORK

Several measurement schemes have investigated delay as a way of determining available network bandwidth. Train of packet pairs (TOPP) [15] is an active probing scheme that injects multiple trains of packets pairs with differing intra-pair separations into the network under test. The sending rate of each trains is linearly increased by reducing the inter-pair separations. The one-way packet pair dispersions are measured as an indication of delay experienced in queues along the network path. Pathload [16] measures the trend of OWD of each packet within a long packet train to determine a self-induced loading effect to determine the available bandwidth. PathChirp [17] and Pathload measure OWD but, as these methods induce congestion, they are not suitable for streaming applications. LDA [18] uses periodic pairs of video packets to estimate available bandwidth through packet dispersion at the tight link. However, LDA actively increases the video sending rate according to the available bandwidth until packet losses occur, rather than set the rate from the estimated delay.

The main intention of delay-based control is to avoid the need to rely on detrimental packet losses, and, therefore, delay-based control is a form of congestion avoidance [19][20]. In [21], delay-based congestion avoidance, using the delay gradient [22], was applied to interactive applications such as video conferencing to find the minimum possible delay without overly restricting throughput. Measurements of packet loss and delay across a typical tight [24] indicate that packet loss may be a weak indicator of congestion level because, when cross-traffic across the link is 'bursty', losses may not be evenly distributed between the competing flows.

There has been a recognition of the benefits of employing delay and delay variation as an early network congestion indicator. TCP Santa Cruz [3] measures relative delay between packet pairs on their forward path. FAST TCP [4] finds changes in RTT to determine the onset and level of network congestion. Synch-TCP [25] is another proposed enhancement to TCP with early network congestion detection. The main difference between Synch-TCP and FAST TCP is that Synch-TCP measures delay from one-way transit time, while FAST TCP uses RTT instead.

A survey of congestion control through computational intelligence [26] observes that not much work has been reported on deploying natural algorithms within the Internet. The authors of [27] have explored fuzzy logic to improve the performance of the Random Early Discard (RED) Internet router queue algorithm, including Explicit Congestion Notification (ECN), and research in [28] considers DiffServ buffer occupancy for each class of layered video packets.

3. FUZZY LOGIC CONGESTION AVOIDANCE, AND CONTROL

Fig. 2 shows the streaming architecture in which fuzzy logic controller computes a control signal that forces the video rate adaptation module to change the video quality to match the dsiered sending bit rate. The congestion level determination (CLD) unit finds the congestion state of the network from measured delay and delay variation made by the timer module. FLC employs the multi-bit delay information to compute a new sending rate that is a reflection of the current sending rate and the level of network congestion, and the current sending rate.

The video rate adaptation unit (either a transcoder adapting pre-encoded video or an encoder) changes the sending rate to that computed by the fuzzy controller. The current implementation changes the quantization level of a frequency-domain transcoder [29], but transmoding (changing the rate by altering the emphasis given to regions of interest or objects), or Fine-Grained Scalability are other possibilities.

3.1. Congestion Detection

For each received packet indexed by i

$$OWD_i = T_r - T_s, \tag{3}$$

where T_r is the receive time of the current packet and T_s is the time the packet was sent. The queuing delay over the network path, QD_i is computed from the measured delay and the minimum delay:

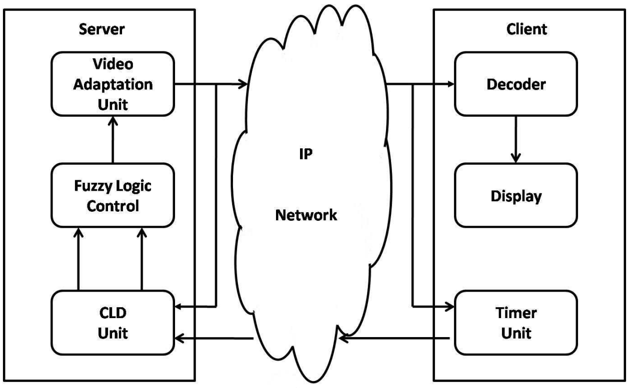

Fig. 2. Block diagram of video streaming architecture

$$QD_i = OWD_i - OWD_{min} \tag{4}$$

and an exponentially weighted average of the queuing delay for the i^{th} received packet is formed by,

$$avgQD_i = (1-\alpha) \times avgQD_{i-1} + \alpha \times QD_i \tag{5}$$

where $\alpha \leq 1$ is the forgetting constant. In simulations, α was set to 0.1.

A delay factor (d_f) is computed from the average queuing delay and the maximum queuing delay,

$$d_f = \frac{avgQD_i}{maxQD} \tag{6}$$

where d_f ranges between [0,1] with 0 indicating no incipient congestion, 1 indicating full blown congestion, with shades of incipient congestion between 0 and 1.

It is difficult to determine the average queuing delay and whether it is increasing or not, given that background cross traffic can cause the queueing delay to fluctuate around the average [25]. A trend analysis method [16] is used to compute a trend value which is fuzzified by the FLC to determine whether the queueing delay is increasing (ITR) or decreasing (DTR).

3.2. Fuzzy Congestion Avoidance and Control

The input and output variables for this FLC are summarized in Table 1. The input variables were fuzzified using triangular membership functions, being a compromise between reduced computation time at the expense of a sharper transition from one state to another. Choosing the number of membership functions is important, since it determines the smoothness of the bit-rate. The delay factor is partitioned using four membership functions separated by 20% increments as shown in Figure 3. The variation on the queuing delay is variable and a 20% gap is sufficient avoid oscillations caused by spurious measurements but enough to capture relative increase of delay and yet responsive to delay variations caused by network load conditions. The trend analysis method determines whether a trend is increasing or not, and Figure 4 shows the used of two membership functions to ensure a smooth decision on whether the delay trend was decreasing or not. The loss rate is fuzzified with the effect of loss on packet loss shown in Figure 1 and the fuzzification is as shown in Figure 5

Table 1. Linguistic variables for FLC input and resulting output

d_f		output	
L	Low	*NL*	Negative Low
M	Medium	*NM*	Negative Medium
H	High	*NH*	Negative High
VH	Very High	*NVH*	Negative Very High
EH	Extremely High	*NEH*	Negative Extremely High
		Z	Zero
		PL	Positive Low
		PM	Positive Medium
		PH	Positive High
		PVH	Positive Very High
		PEH	Positive Extremely High

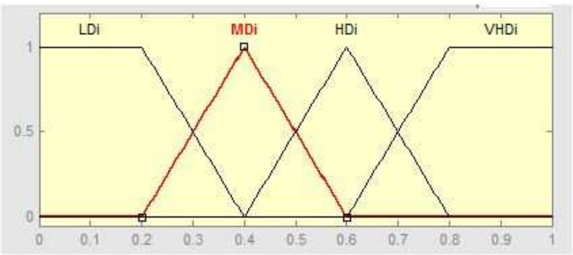

Fig. 3. Delay factor membership functions

Dividing the congestion levels into a manageable set ranging from low to extremely high (according to delay) makes for a simple number of decision rules. The fuzzy inference rules are in the same form as the following examples:

```
if df is H and T is ITR  and LR is L then S is NVH
if df is L and T is DTR  and LR is L then S is PVH
```

where S is the fuzzified output from the controller, DTR is a decreasing trend (T) and ITR is an increasing trend. The complete set of rules for the evaluation of the control output capture in a concise form, the information contained in English sentences constrained in the manner of the previous two examples.

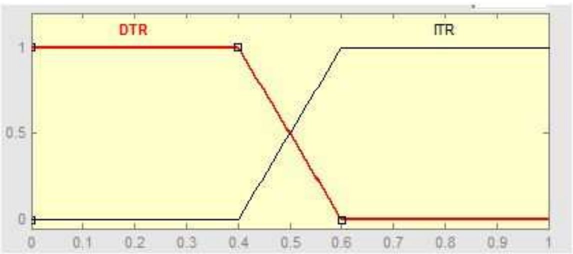

Fig. 4. Trend membership functions

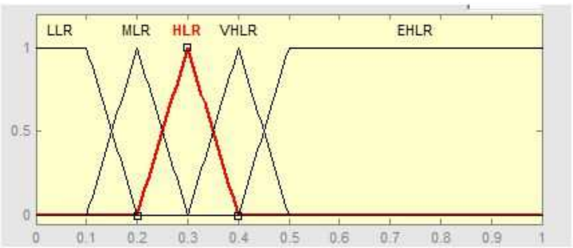

Fig. 5. Loss membership functions

The FLC employs a simple Mamdani inference model [30] and center-of-gravity defuzzification method. Eqn. (7) maps the input to the output of the controller:

$$Ctrl = \frac{\sum_{i=1}^{M} S_i K_i}{\sum_{i=1}^{M} K_i} \tag{7}$$

where M is the number of rules, S_i is the value of the output for rule *i*, K_i is the inferred weight of the i^{th} output membership function. More specifically, S_i is the value at the middle of the range of data values that are possible members of the i^{th} fuzzy subset. K_i is the area under the i^{th} output membership function, clipped by the minimum possibility of membership of d_f in the input membership function of the i^{th} rule.

Figure 6 is the output surface of the FLC. The control signal *Ctrl*, as specified in (7), is normalized to the range (0.1, 0.9] to avoid unacceptable low quality video output. For input bitrate R_{in}, the target output bitrate is R_{out} is given by,

$$R_{out} = (1 + Ctrl) \times Rin. \tag{8}$$

4. EXPERIMENTS

The algorithm was simulated with the well-known ns-2 network simulator. The simulated network, with a typical dumb-

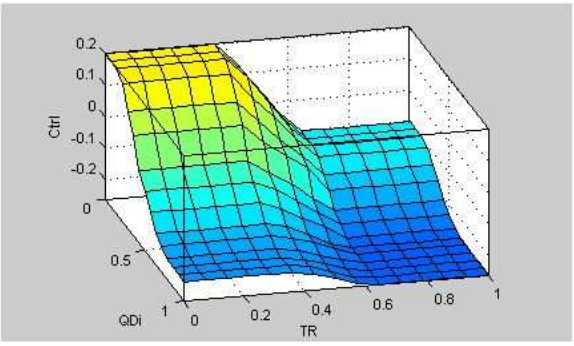

Fig. 6. Output control surface (with fuzzy 'trend' input)

bell topology, had a tight link between two routers and all side link bandwidths were provisioned such that congestion would only occur at the tight link. The one-way delay of the tight link was set to 5 ms and the side links delays were set to 1 ms. The tight links queueing policy was defaulted to be FIFO and the queue size was set to twice the bandwidth-delay product.

4.1. Tracking Bandwidth

The control algorithm was tested for it ability to track bottleneck bandwidths fixed at various rates without any background traffic. Several tests were conducted for each bottleneck and the average results were computed. The sending rate and the received video quality were measured for each bottleneck bandwidth. Figure 4.1 shows the response of the control algorithm in response to a 0.4 Mbit/s, 0.6 Mbit/s and a 0.8 Mbit/s bottleneck bandwidth. It is clear that the control optimally adapts to the available network bandwidth in a timely manner. The quality of the sent video was also calculated for each bottleneck bandwidth and the result is shown in Figure 4.1. It is clear from this result that the quality of the video is adapted to the available bandwidth and that there is a graceful change in the video quality in sympathy with the available bandwidth.

4.2. Tracking Bandwidth

The control algorithm was also tested for it ability to track available network bandwidths. The sending rate and the received video quality were measured for the duration of the streaming session. Figure 4.2 show the result of the sending rate tracking a changing available network bandwidth and it is clear that the sending rate accurately tracks the available network bandwidth, albeit with a slow edge when tracking an increase in the available bandwidth. The video quality of this tracking of the available bandwidth is shown on Figure 4.1. It shows that the quality of the video is adapted to the available bandwidth.

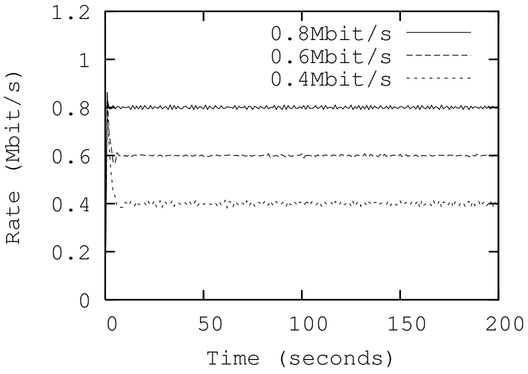

Fig. 7. FLC rate for a 400 kbit/s bottleneck bandwidth

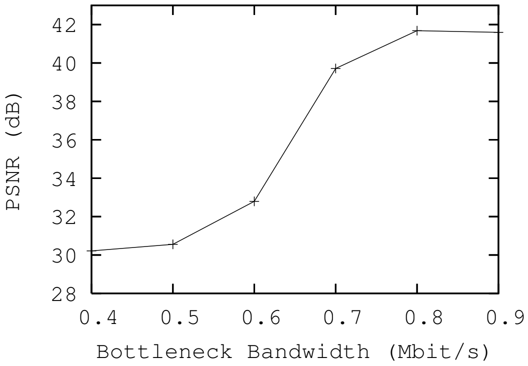

Fig. 8. FLC rate for a 400 kbit/s bottleneck bandwidth

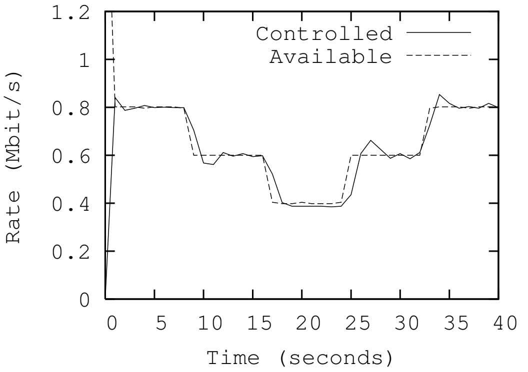

Fig. 9. FLC rate for a 400 kbit/s bottleneck bandwidth

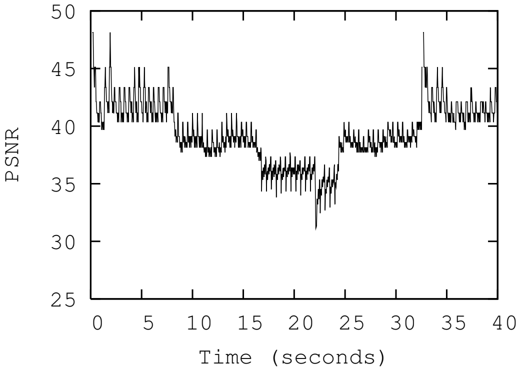

Fig. 10. FLC rate for a 400 kbit/s bottleneck bandwidth

4.3. Tracking Bandwidth

A set of tests were also performed to establish whether available bandwidth tracking was maintained in typical Internet conditions. Internet measurement studies [31][32] have demonstrated a typical Internet traffic mix to consist of longer term flows, 'Tortoise', representing file transfers, and transient HTTP connections, 'Dragonflies'. In our set of tests, one FLC video source and ten TCP sources were passed across the link. The first five TCP sources were configured as 'tortoise', with an on duration of between five and twenty seconds and an off duration between one and five seconds, all also randomly generated from a uniform distribution. The remaining five TCP sources were 'dragonflies' with a random duration of between one and five seconds. These sources were generated from a uniform distribution and with an off duration of between one and five seconds, also randomly generated from a uniform distribution.

Ten experiments were conducted for a bottleneck capacity of 1 Mbit/s, with a 5 ms delay across the link. In the first experiment, only one TCP 'tortoise' source was present as background traffic, in the second two TCP 'tortoise' sources acted as background traffic and so on until the 'dragonfly' sources are eventually introduced, so that all ten TCP sources were on as background traffic for the tenth experiment.

5. CONCLUSIONS

The contributions of this paper is to show that fuzzy logic control, rather than a modelling aproach can be an effective congestion avoidance and control mechanism for network aware networked video communication applications. Furthermore, it shows that delay information, when added, to packet loss information is a good indicator of network congestion state. It is shown that, fuzzy control in combination with delay and packet loss information is a good congestion avoidance and controller for networked video. The implication is that a stable set of fuzzy models can form part of a hardware congestion controller, which could work for a large range of networks conditions.

6. REFERENCES

[1] M. Allman, V. Paxon, and W. R. Stevens, "TCP congestion control," *RFC 2581*, April 1999.

[2] L. Brakmo, W. W. O'Malley, and L. L. Petersen, "TCP Vegas: New techniques for congestion detection and avoidance", *Proceedings of ACM SIGCOMM*, pp. 24-35, Aug. 1994.

[3] C. Parsa and J. J. Garcia-Luna-Aceves, "Improving TCP congestion control over internets with heterogeneous transmission media", *7^{th} IEEE Int. Conf. on Network Protocols*, pp. 213–221, Oct. 1999.

[4] D. X. Wei, C. Jin, S. H. Low, and S. Hedge, "FAST TCP: Motivation, architecture, algorithm and performance", *IEEE/ACM Trans. on Networking*, vol. 4, no. 6, pp. 1246–1259, 2006.

[5] J. Widmer, R. Denda, and M. Mauve, "A Survey on TCP-friendly Congestion Control", *IEEE Network*, vol. 15, no. 3, pp. 28–37, 2001.

[6] L. A. Zadeh, "Fuzzy sets", *Information and Control*, vol. 8, pp. 338–353, 1965.

[7] H. Takagi, "Application of neural networks and fuzzy logic to consumer products", *IEEE Technology Updates Series: Fuzzy Logic Technology and Applications*, vol. 1, pp. 8-12, 1994.

[8] C. C. Lee, "Fuzzy logic control systems: Fuzzy logic controller – part 1", *IEEE Trans. Sys. Man and Cybern.*, vol. 20, no. 2, pp. 404–418, 1990.

[9] A. Leone, A. Bellini, and R. Guerrieri, "An H.261 fuzzy-controlled coder for videophone sequences", *IEEE World Conf. on Computational Intelligence*, pp. 244–248, June 1994.

[10] P. M. Grant, Y.-S. Saw, J. M. Hannah, "Fuzzy rule based MPEG video rate prediction and control", *Eurasip ECASP Conf.*, pp. 211–214, June 1997.

[11] J. Padyhe, V. Firoiu, D. Towsley, and J. Krusoe, "Modeling TCP throughput: A simple model and its empirical validation", *Proceedings of ACM SIGCOMM*, pp. 303-314, Sept. 1998.

[12] C. E. Rohrs, R. A. Berry, and S. J. O'Halek, "A control engineer's look at ATM congestion avoidance", *IEEE GLOBECOM*, vol. 2, pp. 1089-1094, Nov. 1995.

[13] R. Rejaie, "On integration of congestion control with Internet streaming applications", *13th Int. Packet Video Workshop*, April 2003.

[14] I. Baturone, A. Barriga, S. Sánchez-Solano, C. Jiménez, and C. López, *Microelectronic Design of Fuzzy Logic-based Systems*, CRC Press, Baton Rouge, FO, 2000.

[15] B. Melander, M. Bjorkman, and P. Gunningberg, "A new end-to-end probing and analysis method for estimating bandwidth bottlenecks", *Global Internet Symposium*, pp. 415-420, Nov. 2000.

[16] M. Jain and C. Dovrolis. "Pathload: A Measurement Tool for End-to-End Available Bandwidth". *Passive and Active Measurements Workshop*, March 2002.

[17] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell. "pathChirp: Efficient available bandwidth estimation for network paths", *Passive and Active Measurements Workshop*, April 2003.

[18] D. Sisalem and H. Schulzrinne, "The loss-delay based adjustment algorithm: a TCP-friendly adaptation scheme", *Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, pp. 215–226, Jul. 1998.

[19] R. Jain, "A Delay-based Approach for Congestion Avoidance in Interconnected Heterogeneous Computer Networks", *Computer Communications Review*, vol. 19, no. 5, pp. 56-71, 1989.

[20] C. Jin, D. X. Wei, and S. H. Low, "The Case for Delay-based Congestion Control",*IEEE Annual Workshop on Computer Communications*, pp. 99-104, 2003.

[21] W. Dabbous, "Analysis of Delay Based Congestion Avoidance Algorithm", *High-Performance Networking IV*, pp. 283-298, 1992.

[22] Z. Wang and J. Crowcroft, "Eliminating periodic packet losses in the 4.3-Tahoe BSD TCP congestion control algorithm", *Computer Communications Review*, vol. 22, no. 2, pp. 9-16, 1992.

[23] S. Floyd and V. Jacobsen, "Traffic phase effects in packet-switched gateways", *Computer Communications Review*, vol. 22, no. 2, pp. 26–42, 1991.

[24] M. Paredes Farrera, M. Fleury, E. Jammeh, M. Ghanbari, and K. Guild, "A Measurement Study of Delay-based Congestion Level Detection in Video Streaming", submitted to *16th Int. Packet Video Workshop*, 2007.

[25] M.C. Weigle, K. Jeffay, and F.D. Smith, "Delay-based early congestion detection and adaptation in TCP: Impact on Web performance", *Computer Communications*, vol. 28, no. 8, pp. 837-850, May 2005.

[26] A. Pitsillides and A. Sekercioglu, "Congestion Control", *Computational Intelligence in Telecommunications Networks,* W. Pedrycz and A. Vasiliakos, Eds. CRC Press, Boca Raton, FL, Sept. 2000, pp. 109-158.

[27] L. Rossides, C. Chrysostemou, A. Pitsillides, and A. Sekercioglu, "Overview of Fuzzy-RED in Diff-Serv networks", *Soft-Ware 2002*, pp. 2-14, LNCS # 2311, 2002.

[28] X. Wang, D. Ye, and Q. Wu, "Using fuzzy logic controller to implement scalable quality adaptation for stored video in DiffServ networks", *12th Int. Packet Video Workshop*, April 2002.

[29] P. A. A. Assunção and M. Ghanbari, "Buffer analysis and control in CBR video transcoding", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 1, pp. 83-92, 2000.

[30] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller", *Int. J. of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.

[31] N. Brownlee and K. C. Claffy, "Understanding Internet traffic streams: Dragonflies and Tortoises", *IEEE Communications*, vol. 40, no. 10, pp. 110-117, 2002.

[32] J. S. Marron, F. Hernandez-Campos, and F. D. Smith, "Mice and Elephants visualization of Internet traffic", *15th Conference on Computational Statistics*, Aug. 2002.