All-IP Network Video Streaming through Interval Type-2 Fuzzy Logic Congestion Control

E. Jammeh, M. Fleury, Member IEEE, M. Ghanbari, Fellow IEEE
School of Comp. Science and Electronic Eng., University of Essex, Colchester CO4 3SQ, United Kingdom.
Emails: emmanuel.jammeh@plymouth.ac.uk;{fleum,ghan}@essex.ac.uk

Abstract

All-IP network delivery of video is a promising application of congestion control. This paper compares the advantage of a fuzzy logic controller over existing controllers for this type of network. The acceptance of fuzzy logic for video control has gained ground but its advantages will become even more apparent through interval type-2 (IT2) logic rather than traditional type-1 logic. This paper demonstrates that the new logic provides robustness to uncertainty both in modeling the network and in measuring congestion. The paper establishes that under conditions of heavy congestion, with multiple video sources, an IT2 fuzzy controller consistently outperforms traditional controllers, resulting in an improvement of several dB in video quality when streaming across a bottleneck link.

Keywords: fuzzy logic, interval type-2, video streaming, converged IP.

I. Introduction

Real-time video applications, such as video-on-demand (VoD), IPTV, video-clip-Web-click, and network-based video recorder, all interest telecommunication companies, because of their high bitrates, though they also risk overwhelming networks if it is not possible to control their flows. To avoid this threat, converged IP telephony networks, such as British Telecom's 21CN [1] or the all-IP network of KPN in the Netherlands, are moving towards a converged telephony network (CTN) (transporting data and voice alike) with Internet Protocol (IP) framing but low-blocking probability switching. Within the 21CN, video streaming is sourced either from proprietary servers or from an external Internet connection. It is not over-stating the case to say that all of these systems are primarily been brought into existence to support multimedia services [2], as these are best placed to take advantage of the inexorable rise in bandwidth capacity.

Unicast video streaming, which brings increased flexibility and choice to the viewer over multicast delivery, is achieved by determining the available bandwidth and adapting the video rate at a live video encoder or an intermediate transcoder. Fuzzy Logic Control (FLC) is suited to congestion control [3], because of the inherent looseness in the definition of congestion and the uncertainty in the network measurements available, together with the need for a real-time solution. Within video coding it has previously found an application [4] [5] in maintaining a constant video rate by varying the encoder quantization parameter according to the output buffer state. This is a complex control problem without an analytical solution. Fuzzy logic is gaining acceptance in the video community, witness [6] [7], but it turns out that further improvements are possible with interval type-2 (IT2) fuzzy logic.

In our application, FLC congestion control is a sender-based system for unicast flows. The receiver returns a feedback message indicating changes to the delay experienced by video stream packets crossing the Internet. This allows the sender to compute the network congestion level and from that the FLC estimates the response. The same controller also should be able to cope with a range of path delays and with video streams with differing characteristics in terms of scene complexity, motion, and scene cuts.

A traditional, type-1 FLC is not completely fuzzy, as the boundaries of its membership functions are fixed. This implies that there may be unforeseen traffic scenarios for which the existing membership functions do not suffice to model the uncertainties in the video stream congestion control task. An IT2 FLC can address this problem by extending a Footprint-of-Uncertainty (FOU) on either side of an existing type-1 membership function. In IT2 fuzzy logic, the variation is assumed to be constant across the FOU, hence the designation 'interval'. Though the possibility of type-2 fuzzy systems has been known for some time [8], only recently [9] has algorithms become available to calculate at video rate an IT2 output control value. The first IT2 controllers [9] are now emerging, in which conversion or retyping from fuzzy IT2 to fuzzy type-1 takes place before output. For video streaming there are important practical advantages. Not only does such a controller bring confidence that re-tuning will not be needed when arriving traffic displays unanticipated or un-modeled behavior but the off-line training period required to form the membership functions can be reduced in time.

This paper compares a type-1 FLC for congestion control of video streaming to an IT2 FLC and compares the performances in the presence of measurement noise, which is artificially injected to test the relative robustness. The delivered video quality in terms of Peak Signal-to-Noise Ratio (PSNR)[1] is equivalent to the successful type-1 FLC when the measurement noise is limited and under test results in a considerable improvement when the perturbations are large. The paper goes on to compare the IT2 FLC to a non-adaptive approach and to congestion control by two well-known controllers, TCP-friendly Rate Control (TFRC) [11] and TCP Emulation at Receivers (TEAR) [12], one sender-based and the other receiver based. These are tested by their ability to support multiple broadband connections over an all-IP network.

II. Related work

In surveys investigating the applications of computational intelligence to congestion control, it was noted that not much work has been reported on deploying computational intelligence algorithms to congestion control within the Internet [13], [14]. Asynchronous Transfer Mode (ATM) networks, which employ access control to virtual circuits, are one domain to which fuzzy logic has been more extensively applied [15], [16]. In fact, [15] reports a T2 fuzzy logic controller used for that purpose, though it should be noted that there is a strong trend amongst manufacturers to replace ATM networks with Ethernet and IP framing. Because of Bluetooth (IEEE 802.15.1)'s centralized scheduling, which resembles ATM admission control, fuzzy logic video bitrate control was applied in a similar manner to a Bluetooth wireless link [17]. For the same reason, in a number of papers, the authors of [18] have explored fuzzy logic to improve the performance of the Random Early Discard (RED) router queue algorithm and in [19] to improve the performance of DiffServ buffer occupancy for each class of layered video packets.

In [20], type-2 fuzzy logic was used to determine the size of different video frame type sizes and to classify the video genre. The intention was to allow modelling of variable bit-rate (VBR) video traffic without the need for video sources. Type-2 classification and modelling was found to be superior to using type-1 fuzzy logic [20]. Wireless networks represent a promising application of fuzzy logic [21], as not only are their uncertainties inherent in network traffic but the wireless channel is noisier and takes a wider variety of forms than a wired link. Additionally, the need to conserve battery power brings into play another resource to balance in the fuzzy logic model. In [22], type-2 fuzzy logic was indeed applied to modelling the lifetime of a wireless sensor network.

III. FLC congestion control

Fig. 1 is a block diagram of an FLC, with two inputs, the delay factor, *df*, and delay samples to form a trend. The formation of these inputs is described in Section III.b. These inputs are converted to fuzzy form, whereby their membership of a fuzzy subset is determined by predetermined membership functions. This conversion, which is described in Section III.a, takes place in the fuzzifier and trend test units of Fig. 1. The fuzzy outputs are then combined in the inference engine through fuzzy logic. Fuzzy logic is expressed as a set of rules which take the form of linguistic expressions. These rules express experience of tuning the controller and, in the methodology, are captured in a knowledge database. The inference engine block is the intelligence of the controller, with the capability of emulating the human decision making process, based on fuzzy-logic, by means of the knowledge database and embedded rules for making those decisions. Lastly, the defuzzification block converts inferred fuzzy control decisions from the inference engine to a crisp or precise value, which is converted to a control signal.

a) Fuzzy logic

In a fuzzy subset, each member is an ordered pair, with the first element of the pair being a member of a set S and the second element being the possibility, in the interval [0, 1], that the member is in the fuzzy

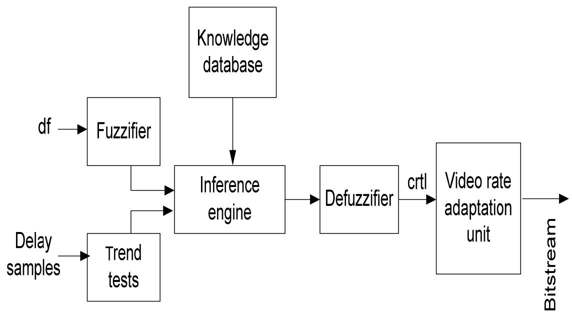


Fig. 1 FLC delay-based congestion controller

[1] PSNR = 10 $\log_{10}$ (MAX^2/ MSE) where MSE is Mean-Square Error summation of the pixel-wise difference between transmitted and received video frames, relative to the peak luminance value (MAX).

subset. This should be compared with a Boolean subset in which every member of a set S is a member of the subset with probability taken from the set {0, 1}, in which a probability of 1 represents certain membership and 0 represents non-membership.

b) Delay-based control

The FLC determines incipient congestion from one way queuing delay in intermediate router buffers. The queuing delay is a measure of network congestion, and the ratio of the average queuing delay to the maximum queuing delay is a measure of bottleneck link buffer fullness. For each received packet indexed by *i*

$$OWDi = Tr - Ts, \quad (1)$$

where *Tr* is the receive time of the current packet and *Ts* is the time the packet was sent. When it is appropriate, the computed *OWDi* updates the minimum and maximum one-way delays (*OWD*s), *OWDmin* and *OWDmax*, on a packet-by-packet basis. Subsequently, the maximum queuing delay is found as *maxQD = OWDmax − OWDmin*.

The queuing delay over the network path, *QDi* is computed from the measured delay and the minimum delay:

$$QDi = OWDi - OWDmin \quad (2)$$

and an exponentially-weighted average of the queuing delay for the ith received packet is formed by,

$$avgQDi = (1 - \alpha) \times avgQDi{-}1 + \alpha \times QDi \quad (3)$$

where $\alpha \leq 1$ is the forgetting constant. In tests, α was set to 0.1. A delay factor, *Df*, is computed from the average queuing delay and the maximum queuing delay,

$$D_f = avgQD_i / maxQD \quad (4)$$

where *df* ranges between [0,1] with 0 indicating no incipient congestion, 1 indicating full-blown congestion, with shades of incipient congestion between 0 and 1. *Df* is an early notification of congestion and is the first input to the FLC.
A trend analysis method is used to determine the general trend of the average delay. In each measurement epoch, a number *k* of queue delay samples are grouped into τ groups, where $\tau = \sqrt{k}$. We use the pairwise comparison test (PCT) to determine the overall trend of the queueing delay as shown in (5).

$$T_{PCT} = \frac{\sum_{i=2}^{\tau} I(M^{i} > M^{i-1})}{\tau - 1}, \quad (5)$$

where *Mi* is the median of group *i* and *I(X)* is 1 if *X* holds and 0 otherwise. The value of *TPCT* is sent back to the sender where a fuzzifier determines whether the level was increasing or not according to a membership function.

c) IT2 fuzzy logic control

IT2 input membership functions for *Df* and trend are constructed, Fig. 2, as an extension of the type-1 FLC through an FOU at the boundaries of the formerly crisp (fixed) membership functions. Assuming the usual singleton input of *Df* (or *TPCT*), an interval set requires just an upper and lower value to be resolved to form the resulting FOU in the corresponding output set. For example, Fig. 2 shows two IT2 membership functions for input sets A and B, each with an FOU. Singleton input X is a member of each with different degrees of membership. Strictly, an infinite number of membership functions (not all necessarily triangular) can exist within the FOUs of sets A and B, but IT2 sets allow the upper and outer firing levels to be taken, as shown in Fig. 3. The minimum operator (min) acts as a t-norm on the upper and lower firing levels to produce a firing interval.

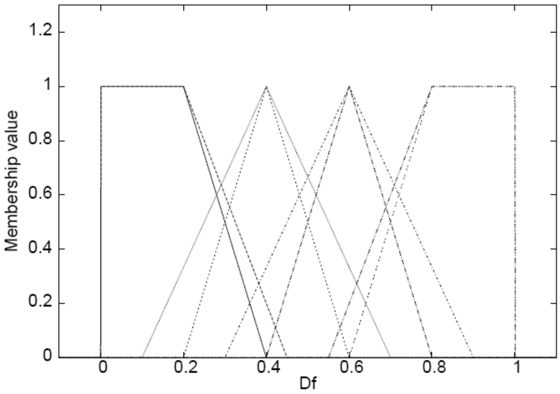
The firing interval serves to bound the FOU in the output triangular membership function shown to the right in Fig. 3. The lower trapezium outlines the FOU, which itself consists of an inner trapezoidal region that is fixed in extent. The minimum operator, also used by us as a t-norm, has the advantage that its implementation cost is less than a product t-norm. (A t-norm or triangular norm is a generalization of the intersection operation in classical logic.) Once the FOU firing interval is established, Center-of-Sets type reduction was applied by means of the Karnik-Mendel algorithm, which is summarized in [9]. Type reduction involves mapping the IT2 output set to a type-1 set. In practice, defuzzification of this type-1 output fuzzy set simply consists of averaging maximum and minimum values. The result of defuzzification is a crisp value that determines the change in the video rate.

d) All-IP network congestion control

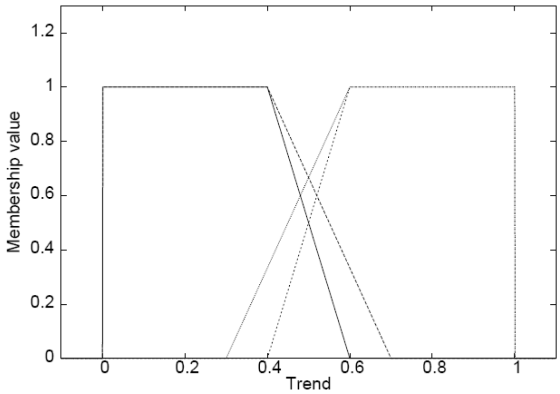
Fig. 4 shows the streaming architecture in which fuzzy logic controls the sending bit rate. The congestion level determination (CLD) unit finds the congestion state of the network from measured delay and delay variation made by the timer module. The congestion state data are relayed to the sender. FLC employs this 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. The video rate adaptation unit (either a bitrate 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

[23] but semantic filtering (changing the rate by altering the emphasis given to regions of interest or objects), or Fine-Grained Scalability or some other form of scalable video [24] are other possibilities.

Fig. 4 shows one instance of server and client. In VoD, IPTV or video clip services there are multiple video streams and multiple clients. Fig. 5 assumes a bank of such servers delivered over an access network such as Asymmetric Digital Subscriber Line (ADSL) or ADSL2+, with downstream rates to 24 Mbps and beyond, one of the passive optical network types (PON) terminating in 100 Mbps Ethernet or coaxial cable, or broadband wireless such as IEEE 802.16.



(a)



(b)

Fig. 2 IT2 FLCC (a) Delay factor (Df) (b) Trend membership functions.

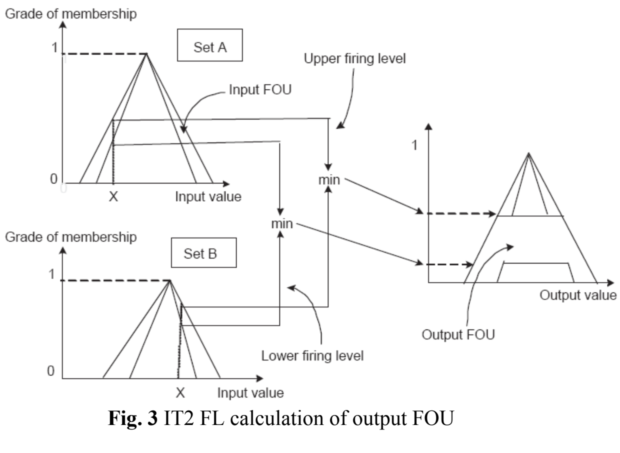


Fig. 3 IT2 FL calculation of output FOU

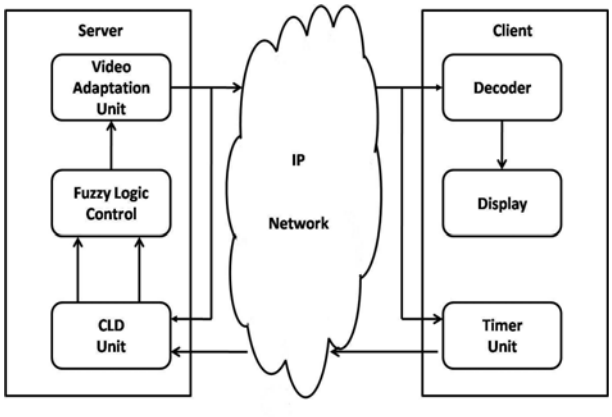


Fig. 4 Video server for all-IP network

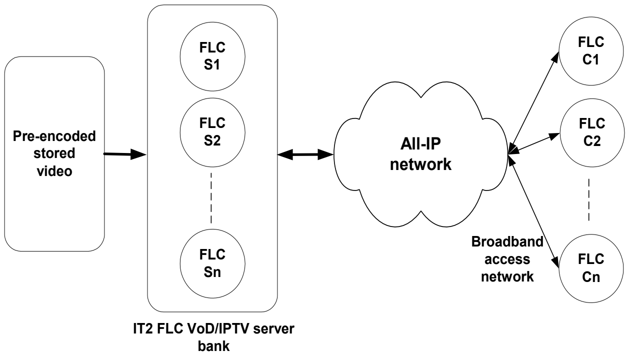


Fig. 5 VoD IPTV video delivery architecture

IV. Results

a) Comparing type-1 to IT2 FLC

An FLC congestion controller employs delay and its variation to gauge the state of the network. There is, however, inherent noise in the measurement of delay, including packet timestamps with limited resolution and unresolved clock drift between sender and receiver. These uncertainties in the input to an FLC will potentially impact its performance.

The well-known ns-2 network simulator (v. 2.32) was used, with the type-1 and IT2 FLCs implemented as new protocols within ns-2. A normal distribution generated a random noise value with zero mean and a specified standard deviation, determined by the level of noise required and dynamically adjusted relative to the measured (simulated) value. For each simulation the level of additional noise was incrementally increased. At each incremental step, the performance of the two controllers was compared in terms of rate adaptation accuracy, packet loss rate, and delivered video quality (PSNR). Input was a 40s MPEG-2 encoded video clip, showing a newreader with a changing backdrop, with moderate movement. The VBR 25 frame/s Source Input Format (SIF)-sized seuqence had a Group of Pictures (GOP) structure of N=12, M=3 [25]. For error resilience purposes, there was one slice per packet, resulting in 18 packets per frame. The FLC controllers adjusted their rate every frame. In this set of tests the encoded video was stored at a mean rate of 1 Mbps.

In Fig. 6, in separate tests both controllers vary the rate of a Constant Bit Rate (CBR) source, with a rate before control of 1 Mbit/s, Both controllers achieve minimal oscillations in the sending rate, which would not be the case for a bandwidth probing congestion controller, as was established in prior work [26] with a type-1 controller. There are small over-shoot peaks at the available bandwidth transition points but such drastic changes in background traffic rate are unlikely to occur across a live link. The similarity in response was expected given that the difference between the two controllers should be proportional to the degree of uncertainty in detecting a network's congestion and its trend. Put another way, the more certain the background, the more likely a type-1 FL can compete with an IT2 FLC.

The video streams were now passed across a bottleneck link restricted to 400 kbps in capacity. The results are gathered in Table 1 and Figs. 7–8. Below 30% additional noise, the two controllers do not significantly deviate. However, beyond 30% of additional noise, the IT2 FLC congestion controller showed significant improvement over the type-1 FLC in terms of reduced fluctuation in the sending rate and a reduced packet loss rate, both of which will be reflected in better average delivered video quality. The smoothness of the transmission rate (measured by a reduction in the standard deviation of the delay on a per-packet basis) is important in video transport as a fluctuating compressed bit-rate implies a fluctuation in video quality, which is more disconcerting to a viewer than a stream of consistent quality, even if that average quality was lower than that of a fluctuating stream. Fig. 7 confirms that delivered average video quality is improved, though, for very high levels of measurement noise, the encoded video stream is so corrupt it matters little which FLC is in control, the quality is very poor.

Looking more closely at the behavior of the two FLCs, in Fig. 9, it is clear that at stepped changes of the available bandwidth, with a 40% noise level addition to delay measurements the IT2 FLC is able more directly to follow the change in available bandwidth, resulting in less packet loss and consequently in higher video quality.

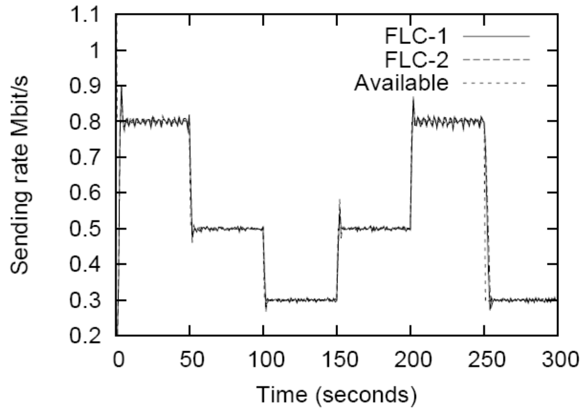


Fig. 6 IT2 and type-1 FLC sending rates for stepped available bandwidths.

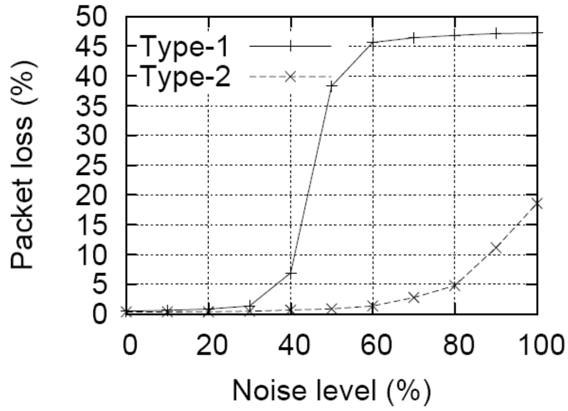


Fig. 7 Packet loss rate for an increasing noise level

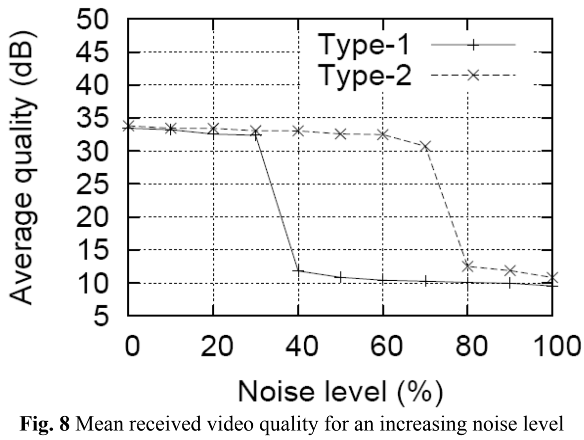


Fig. 8 Mean received video quality for an increasing noise level

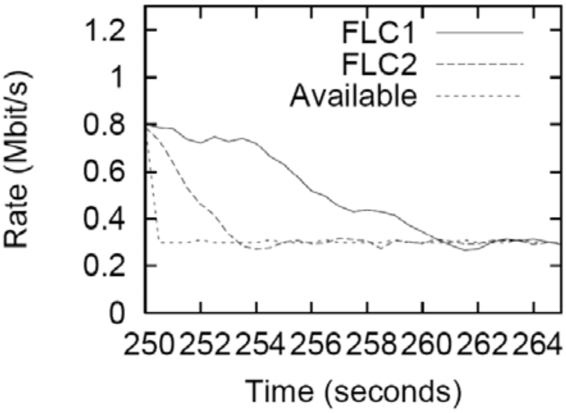


Fig. 9 Response of type 1 and IT2 FLC at a stepped edge after a 40% noise level addition to delay measurements.

Table 1 Standard deviation of FLC type-1 and type-2 sending rates (kbps)

Noise level (%)	Type-1	Type-2
0	77.527	76.722
10	78.192	76.607
20	78.986	77.098
30	80.281	77.677
40	109.927	77.747
50	193.612	78.244
60	227.173	80.238
70	230.016	84.294
80	230.651	93.822
90	230.924	113.355
100	231.082	124.652

b) Comparing IT2 FLC with traditional congestion controllers

Comparison was made with the TFRC protocol, the subject of an RFC [11] and a prominent method of congestion control from the originators of the 'TCP-friendly' concept. To ensure fairness the publicly available TFRC NS-2 simulator model (in the form of object tcl scripts to drive the simulator) was availed of from http://www.icir.org/tfrc/. In TFRC, the sending rate is made a function of the measured packet loss rate during a single round-trip time (RTT) duration measured at the receiver. Unfortunately, if the TFRC feedback frequency is reduced TFRC tends to dominate co-existing flows [12]. The sender then calculates the sending rate according to the TCP throughput equation given in [11]. As with IT2 FLC and TEAR, the UDP transport protocol is employed to avoid unbounded delays, which are possible with TCP transport.

Unlike TFRC, TEAR is based on the Arithmetic Increase Multiplicative Decrease (AIMD) algorithm of TCP. Unlike TCP, TEAR avoids the oscillatory behavior of TCP by averaging its sending rate over a round, based on the time to send a congestion window's packets. TEAR's sending rate approximates that of an equivalent TCP source. The default settings for TEAR were used [12], with publicly available ns-2 models.

Both TFRC and TEAR rely on measurements of the RTT, while TFRC is also adversely affected by inaccurate loss rate estimates [27]. As remarked in Section I, without a transcoder TFRC and TEAR require playout buffers to smooth out network delay. Therefore, PSNR is affected by loss rate only, assuming a large enough buffer to avoid overflow. FLC also reduces the video quality through transcoding if there is insufficient bandwidth, but this avoids the need for long start-up delays and allows smaller buffers on mobile devices.

In further comparison tests, the standard 'dumbbell' network topology was assumed with a bottleneck of 25 Mbps. The one-way delay, modeling the latency across the complete network path, was set to 40 ms, which is the same as the maximum delay across a country such as the U.K or France. Side link delay was set to 1 ms and the side link capacity was set to easily cope with the input video rate. The mean encoded video rate was again 1 Mbps. The buffer size on the intermediate routers was set to RTT×bandwidth, to avoid overflow through too small a buffer. The router queuing discipline was drop-tail.

The intention of these tests was to see how many video streams could be accommodated across the bottleneck link. In Table 2, the number of controlled video sources (replicating the source described in Section IV.a) was incrementally increased. The starting times of streaming the 'news clip' to each client was staggered, and then each clip was repeatedly sent over 200 ms. The first 40 s of results, was discarded as representing transient results. This method was chosen, rather than select from different video clips, because the side effects of the video clip type do not intrude.

As can be seen from Table 2, when there is no control, there is no packet loss until the capacity of the link is reached. Thereafter, the link utilization grows and, as might be expected, the packet loss rate rapidly climbs. Failure to estimate the available bandwidth causes both TFRC's and TEAR's mean link use to exceed the capacity of the bottleneck link. As the number of flows increases, it becomes increasingly difficult to control the flows and there is a steady upward trend in the overshoot. In respect to TEAR, this leads to considerable packet loss. The packet loss patterns are reflected in the resulting PSNRs, though there is no direct relationship because of the effect of motion estimation in the codec.

The packet loss reduction arises from a reduction in sending rate fluctuations in the case of the IT2 FLC.

This is illustrated in plots of instantaneous sent bitrate over the duration of the clip for the case of n = 2 in Fig. 9. Just two streams are plotted, as otherwise the behavior of individual steams is difficult to discern. Apart from initial settling down to steady state behavior, one of the TFRC plots varies around that of the other, whereas this is not the case for the IT2 FLCC streams. It is surprising in that TEAR was developed after TFRC and in part as a reaction to it [27]. However, subsequent to the development of TEAR, TFRC has undergone some refinements such as TCP's self-clocking. However, from Table 2 (a) it is apparent IT2 FLC congestion control does not suffer from the difficulties that TFRC and TEAR encounter. There is a very small loss rate due to moments when the time varying nature of VBR video results in the FLC overestimating the available bandwidth but this is significantly below the loss rates of the traditional controllers.

Table 2 Performance comparison of congestion controllers

No control			
No. of Sources	**Loss rate (%)**	**Link use (%)**	**PSNR (dB)**
25	0.0	100.0	–
30	16.66	120.0	–
35	28.56	140.0	–
40	37.49	160.0	–
45	44.44	180.0	–
50	49.99	200.0	–
TFRC			
No. of Sources	**Loss rate (%)**	**Link use (%)**	**PSNR (dB)**
25	1.50	101.48	36.08
30	1.81	101.80	35.11
35	2.11	102.80	33.78
40	2.39	102.44	33.07
45	2.65	102.78	31.34
50	2.91	102.96	30.18
TEAR			
No. of Sources	**Loss rate (%)**	**Link use (%)**	**PSNR (dB)**
25	2.50	102.52	33.27
30	3.51	103.60	32.34
35	4.61	104.80	31.56
40	5.75	106.08	30.70
45	6.86	107.36	29.61
50	7.91	108.56	28.78
IT2 FLC			
No. of Sources	**Loss rate (%)**	**Link use (%)**	**PSNR (dB)**
25	0.0	89.82	39.61
30	0.0016	99.96	37.90
35	0.0026	99.96	36.89
40	0.0029	99.96	35.44
45	0.0038	99.84	33.19
50	0.0048	99.82	31.40

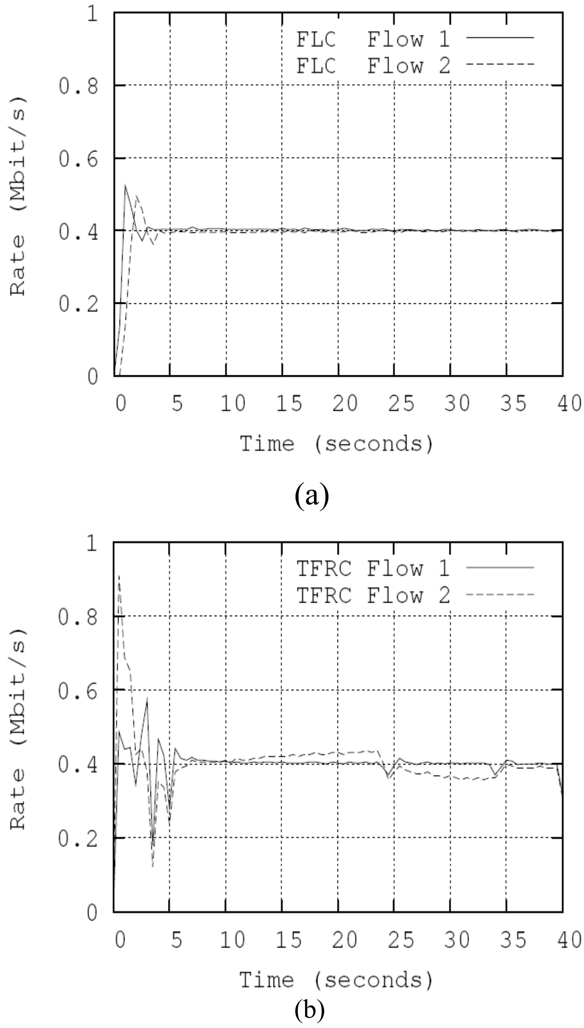


Fig. 10 Illustration of the behaviour of individual stream sending rates for (a) IT2 FLCC (b) TFRC.

V. Conclusion

Intelligent control of network traffic flows has been little explored, though policing of networks that have an access control mechanism has received some attention. However, streaming of encoded video clips is taking an increasing share of bandwidth on the Internet and as a result is being directed towards managed all-IP networks. Compressed video streams are sensitive to packet loss as the removal of temporal redundancy between video frames (along with entropic encoding as a final output stage) results in data dependencies.

Emulation of TCP at the application layer as a way of preserving its average behaviour is a popular approach but this still results in fluctuations in the sending rate and larger packet losses than are necessary. In this paper we have shown that an interval type-2 fuzzy logic controller preserves all the qualities of a traditional fuzzy logic controller but is also able to respond to uncertainty in the packet delay measurements that form the principle feedback to the controller. In fact, the ability to cope with considerable corruption of the input was quite dramatic in our results. It was also found that the interval type-2 fuzzy logic congestion controller was able to achieve minimal packet loss (which is highly desirable for delivered video quality) in comparison to the traditional TFRC and TEAR controllers. Both also are subject to uncertainties in round-trip estimation and for TFRC packet loss estimation.

Across an all-IP network bottleneck, FLC was shown to be clearly preferable. It was able to accurately estimate the available bandwidth across fifty competing flows without overshooting the link capacity, despite considerable delay in feedback. The average delivered video quality was around 2 dB better than the other controllers across all coexistent flows.

Acknowledgment

This work was supported by the EPSRC, UK under grant no. EP/C538692/1. The authors acknowledge assistance from C. Wagner and H. Hagras in applying interval-type 2 fuzzy logic to the authors' original type-1 controller.

References

[1] M. Reeve, C.E. Bilton, M. Holmes and M. Bross, "21CN", *IEE Comms. Eng.*, Oct., 2005.

[2] D. Greer, "Building converged networks with IMS technology", *IEEE Computer*, 38(11):14-16, 2005.

[3] E. A. Jammeh, M. Fleury, and M. Ghanbari, "Delay-based congestion avoidance for video communication with fuzzy logic control", *17th Int. Workshop on Packet Video*, November 2007.

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

[5] P. M. Grant, Y-S. Saw and J. M. Hannah, "Fuzzy rule-based MPEG rate prediction and control", *EURASIP ECAS Conf.*, June 1997, pp. 211-214.

[6] E. A. Jammeh, M. Fleury, and M. Ghanbari, "Fuzzy Logic Congestion Control of Transcoded Video Streaming without Packet Loss Feedback", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 3, pp. 387-393, 2008.

[7] M. Rezaei, M. H. Hanukse, and M. Gabbouj. "Semifuzzy rate controller for variable bit rate video", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 5, pp. 633-645, 2008.

[8] L. A. Zaddeh. The concept of linguistic variable and its application to approximate reasoning. *Information Sciences*, vol. 8, pp. 199–249, 1975.

[9] J. M. Mendel, "Type-2 fuzzy sets and systems: An overview", *IEEE Computational Intelligence*, pp. 20–29, Feb. 2007.

[10] H. Hagras. Type-2 FLCs: A new generation of fuzzy controllers. *IEEE Computational Intelligence*, pp. 30–43, February 2007.

[11] M. Handley, S. Floyd, J. Padyhe, and J. Widmer, "TCP friendly rate control (TFRC): Protocol specification," 2003, IETF RFC 3448.

[12] I. Rhee, V. Ozdemir, and Y. Yi. "TEAR: TCP emulation at receivers – flow control for multimedia streaming", NCSU Technical Report, April 2000.

[13] S. Ghosh, Q. Razouki, H. J. Schumacher, and A. Celmins, "A survey of recent advances in fuzzy logic in telecommunications networks and new challenges.", *IEEE Trans. on Fuzzy Systems*, vol. 6, no. 3, pp. 443–447, 1998

[14] A. Pitsillides and A. Şekercioglu, "Congestion control", In W. Pedrycz and A. Vasiliakos, editors, *Computational Intelligence in Telecommunications Networks*, pp. 109–158. CRC Press, Boca Raton, FL, Sept. 2000.

[15] Q. Liang, N. Karnik, and J. M. Mendel, "Connection admission control in ATM networks using survey-based type-2 fuzzy logic systems", *IEEE Trans. on Systems, Man, and Cybernetics Part C: Applications and Reviews*, vol. 30, pp. 329-339, 2000.

[16] A. Şekercioglu, A. Pitsillides, and A. Vasilakos, "Computional intelligence in management of ATM networks: A survey of current state of research", *Soft Computing Journal*, vol. 5, no. 4, pp. 257–263, 2001.

[17] H. B. Kazemian and L. Meng, "An adaptive control for video transmission over Bluetooth", *IEEE Trans. on Fuzzy Systems*, vol. 14, no. 2, pp. 263–274, 2006.

[18] L. Rossides, C. Chrysostemou, A. Pitsillides, and A. Sekercioglu, "Overview of Fuzzy-RED in Diff-Serv networks", *Soft-Ware 2002*, D. W. Bustard, W. Liu, and R. Sterritt, Eds., April 2002, pp. 2–14, LNCS # 2311.

[19] X. Wang, D. Ye, and Q. Wu, "Using fuzzy logic controller to implement scalable quality adaptation for stored video in DiffServ networks", *12th International PacketVideo workshop*, April 2002.

[20] Q. Liang and J. M. Mendel, "MPEG VBR video traffic modeling and classification using fuzzy technique", *IEEE Trans. on Fuzzy Systems*, vol. 9, no. 1, pp. 183-193, 2001.

[21] R. Razavi, M. Fleury, and M. Ghanbari, "Fuzzy logic control of adaptive ARQ for video distribution over a Bluetooth wireless link", *Advances in Multimedia*, 2007. 13 pages, online volume.

[22] H. Shu, Q. Liang and J. Gao, "Wireless sensor network lifetime analysis using interval type-2 fuzzy logic systems", *IEEE Trans. on Fuzzy Systems*, vol. 16, no. 2, pp. 416-427, 2008.

[23] 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.

[24] B. Pesquet-Poescu, S. Li, and M. van der Schaar, "Scalable video coding for adaptive streaming applications", In M. an der Schaar and P. A. Chou editors, *Multimedia over IP and Wireless networks,* pp. 154-158, Academic Press, Amsterdam, 2007.

[25] M. Ghanbari, "Advanced video compression", (in Persian), *IRIB publisher*, Tehran, 2008.

[26] E. A. Jammeh, M. Fleury, and M. Ghanbari. "Delay-based congestion avoidance for video communication with fuzzy logic control", *17*th *Int. Workshop on Packet Video*, November 2007.

[27] I. Rhee and L. Xu, "Limitations of equation-based congestion control", *IEEE/ACM Trans. on Networking*, vol. 15, no. 4, pp. 852-865, 2007.