

## Research Article

# Power-Constrained Fuzzy Logic Control of Video Streaming over a Wireless Interconnect

Rouzbeh Razavi, Martin Fleury, and Mohammed Ghanbari

*Department of Computing and Electronic Systems, University of Essex, Colchester CO4 3SQ, UK*

Correspondence should be addressed to Martin Fleury, fleum@essex.ac.uk

Received 29 September 2007; Accepted 6 May 2008

Recommended by David Bull

Wireless communication of video, with Bluetooth as an example, represents a compromise between channel conditions, display and decode deadlines, and energy constraints. This paper proposes fuzzy logic control (FLC) of automatic repeat request (ARQ) as a way of reconciling these factors, with a 40% saving in power in the worst channel conditions from economizing on transmissions when channel errors occur. Whatever the channel conditions are, FLC is shown to outperform the default Bluetooth scheme and an alternative Bluetooth-adaptive ARQ scheme in terms of reduced packet loss and delay, as well as improved video quality.

Copyright © 2008 Rouzbeh Razavi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. INTRODUCTION

Preservation of battery power is an essential feature of mobile devices, to reduce the frequency of recharges. Though Bluetooth (IEEE 802.15.1) [1] devices have hold, park, and sniff low activity modes, and the transceiver is designed to minimize power [2], it is still important that an application reduces the total data transmitted, as there is approximately a linear relationship [3, 4] between bit rate and energy consumption. A number of authors, for example [4–8], have investigated ways to manage power in a wireless network when streaming video. Although the enhanced data rate (EDR) of Bluetooth version 2.0 [9] now has a peak user payload of 2.2 Mbps (gross air rate 3.0 Mbps), which is the same average rate offered by some implementations of IP-TV, it must still compete with lower power alternatives, such as Wibree from Nokia, intended for button cell batteries, with a gross air rate of 1.0 Mbps. However, compared to IEEE 802.11 (Wi-Fi)'s [10] typical current usage of 100–350 mA, Bluetooth's consumption is 1–35 mA, implying that for mobile multimedia applications with higher bandwidth capacity requirements, Bluetooth is a preferred solution.

Many cellular phones are also equipped with a Bluetooth transceiver and larger resolution screens of CIF (352 × 288) and QCIF (176 × 144) pixel size. However, as in a group of

pictures (GOP), slices within one picture are predicted from previous ones, noise and interference on the wireless channel may corrupt slice-bearing packets, as they make the final hop before decoding and display on a mobile device. This suggests retransmission of corrupted packets should occur, which automatically increases the power budget, quite aside from the possibility for video of missed display deadlines. This is unfortunate, as in general automatic repeat request (ARQ) has proved more effective than forward error correction (FEC) [11] in ensuring statistically guaranteed quality-of-service (QoS) over wireless networks. FEC imposes an ongoing overhead, adding to the power budget, whereas typical channel errors come in bursts, with the channel state alternating between good and bad states. For example, in an indoor environment, fast fading occurs when persons walk across the line-of-sight between the communicating devices. Hybrid ARQ [12], in which reply packets advise the sender of errors, is complex to implement at the data link layer and, owing to the volatility of the wireless channel, may impose too great a latency if adaptive error control occurs at the application layer, at a remote encoder. In Bluetooth, fast ARQ comes for free by virtue of time-division duplex (TDD) polling, which is necessary for transmit/receive recovery, allowing a single-chip implementation, whereas data link layer FEC is only possible at the legacy basic rate (1.0 Mbps gross air rate).

Effective ARQ management is the key to both power management and ensuring acceptable video quality at the receiver device. However, it is a multifaceted control problem, as account must also be taken of wireless channel conditions, and of the display/decode deadlines of the picture type slices being conveyed. This paper proposes fuzzy logic control (FLC) of ARQ, as a way of combining all three factors: (1) channel state; (2) display/decode deadline; and (3) power budget. In our earlier work [13], we did not consider the need to meet a power budget. We have adopted a modular scheme whereby a two-input FLC stage with a single output is concatenated with a second FLC stage, with the output from the original FLC and an additional “remaining power” input. The two inputs to the first FLC stage are buffer fullness and the deadline margin of the packet at the head of the Bluetooth send queue, which gives a direct measure of delay. Assuming a fixed power budget for the duration of a video clip streaming session, the declining power budget as the stream progresses has the effect of modulating the ARQ retransmission count. A modular scheme reduces the construction complexity of the design and allows for future enhancements.

FLC, which has from its inception [14] been extensively used for industrial and commercial control applications [15], is a convenient tool for real-time control as unlike genetic algorithms or neural networks there is no long period of convergence or online training. Two factors imply that a mathematical model is unsuitable: the inputs are dependent on the outputs as there are feedback channels, implying that the problem is nonlinear; and the complexity of multiple constraints is an obstacle. Within video coding, FLC has already found an application [16, 17] 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. Therefore, FLC is a natural candidate for the solution of this problem. In general, a fuzzy scheme is easily tuned by adjustment of its membership functions. A fuzzy scheme is also well suited to the implementation on a mobile device, because not only are the decision calculations inherently simple (and can be made more so by adoption of triangular membership functions) but also, by forming a look-up table (LUT) from the fuzzy control surface, its operation can be reduced to simple LUT access. There is also a range of hardware designs [18] for FLC to aid real-time operation.

As is well known, real-time delivery of video is delay-sensitive, as a frame cannot be displayed if its data arrive after their decode deadline. A further deadline exists for reference picture types if their presence contributes to decoding of future frames [19]. In practice, a playout buffer exists on a mobile device to account for startup delay and also absorbs delay jitter (variation of delay). Therefore, the maximum delay permissible corresponds to the startup delay deemed tolerable to the user. Packets may arrive too late for the frame to be displayed and, as error concealment at the decoder is implementation dependent, the net result is poor quality video. Not only do packets arrive after their display deadline, but while retransmission takes place, other packets may either wait too long in the send buffer or in the extreme case

arriving packets may find the send buffer full. ARQ adds to delay and, therefore, the number of retransmissions should be minimized even before taking into account the impact on the power budget.

Adaptive ARQ is not a complete solution, as it fails to account for deadline expired packets remaining in the send buffer while retransmission takes place. The danger is that these packets will then be transmitted simply to be discarded at the receiver. The presence of expired packets in the send buffer, just like excessive ARQ delay, contributes to the queuing delay of other packets and possibly to buffer overflow. Therefore, an active discard policy for deadline expired packets is required as an addition to adaptive ARQ. In our system, the active discard policy is implemented as a deadline-aware buffer (DAB) and is also based on picture type. Picture type can be ascertained by inspection of application packet headers, whereas accounting for picture content rather than picture importance may require intervention at a source encoder. The DAB introduced by us has a threefold advantage: (1) queuing time of packets in the send buffer is reduced; (2) the possibility of send buffer overflow is effectively removed, except for the smallest of buffer sizes; and (3) power is conserved as deadline expired packets are no longer needlessly transmitted.

The remainder of this paper is organized as follows. Section 2 is a survey of related work, with a concentration on power-aware video streaming. Section 3 contributes background material on Bluetooth and explains the FLC in detail. The research methodology is also detailed. Section 4 contains our simulated results, while Section 5 summarizes and draws some general conclusions.

## 2. RELATED WORK

In [20], it was shown that transmission accounts for more than a third of the total energy consumption in communication on a mobile device. In [3], 78% of power consumption is attributed to transmission and playback at the receiver. In general, transmission consumes more power than reception, but this does not necessarily imply that in Bluetooth a master consumes more power than a slave receiver, because a receiver is unable precisely to anticipate when a transmission will occur. Thus a Bluetooth slave receiver on average consumes 46 mA, [21] as opposed to a master transmitter's 17 mA consumption.

In [4], assuming the aforementioned linear relationship between energy consumption and bit rate, within a GOP, B-pictures are first discarded, while if this does not succeed in reducing the bit rate then P and even I pictures are discarded. The authors propose spreading the discards to allow easier reconstruction at the decoder. However, this is an early work that gives no account of the impact on video quality of this rather simple policy. In [5, 6], the decoding capability of the receiver is signaled to the transmitter, which subsequently adjusts its transmission accordingly through fine-grained scalability. The transmitter encoder power budget is taken into account in [22], varying the power allocation between source and channel coding. However, the former approach apparently does not consider the effect of the channel,

whereas the latter is inappropriate for preencoded video. A transcoder at the wireless transmitter is assumed in [3] and the rate is controlled according to a linear model of power consumption, together with a piecewise linear model of playback power consumption. In [23], an energy constraint is introduced into a rate-distortion encoding model. In [24] also, content importance is factored in by annotating video segments through MPEG-7. Moderate improvements in user perception were reported. Despite the title in [8], the video content itself does not determine the transmit rate so much as the length of MPEG (*sic*) packets. The lengths are used to determine a packet burst profile for IEEE 802.11 networks. Depending on the video clip, approximately 60% energy savings are reported for this technique.

Our scheme considers a fixed playout buffer at the receiver and assumes single-layered video. Fixed-size playout buffers at the receiver are liable to underflow given that variable bit rate (VBR) encoded video is inherently “bursty.” The burstiness occurs at multiple time scales, owing to changes in picture type within a GOP, within a scene with variable motion, and between scene cuts. Though in fixed networks large playout buffers (at up to several seconds of startup delay) may be applied in video-on-demand applications, web-based video clip distribution with click-level interactivity is less tolerant of startup delay. On a mobile device, memory contributes significantly to the power budget [25], resulting in relatively small buffers. For example, the experiments in [26] assumed a send buffer size of fifty packets, as also assumed in our experiments. In [26] also, selected packets are given priority transmission, rather than enforce rate changes at the encoder, which discriminates against preencoded video. However, layered encoding is assumed, while much content exists in nonlayered format.

For single-layer video, the packet type is a simple way of applying either a delay or a loss priority packet transmission. The packet type indicates content importance without the need for content awareness at the link layer. In [27], simple packet type discrimination is proposed as a means of implementing differentiated services QoS on the fixed Internet.

Varying the number of retransmissions as part of ARQ management is a feature of IEEE 802.11 wireless networks and in IEEE 802.11e it is also possible to set a maximum limit to the time spent in the transmitter buffer [28]. In [9], the packet loss rate over the wireless link is balanced with the loss rate from buffer overflow by incremental adjustments to the retry limit. Packet purging is also employed in [9], whereby packets dependent on lost packets are removed from queues. The problem with purging, as opposed to deadline-aware active discard (as in our paper), is that it appears only actionable when I-picture packets have been lost. The scheme in [9] was tested for a six-layered video stream, which increases the time taken in searching queues for packet purging, while the computational cost is less for the single queue nonscaleable video. Both IEEE 802.11’s Point Coordination Function and IEEE 802.11e’s Hybrid Coordination Function allow for centralized packet scheduling and, hence, techniques applicable to Bluetooth are to some extent

transferable to these. IEEE 802.11e has a variable set of ARQ modes but a management policy is not part of the standard.

### 3. METHODOLOGY

#### 3.1. Bluetooth background

Bluetooth is a short-range (less than 10 m for class 2 devices), radio frequency interconnect. Bluetooth employs robust frequency-hopping spread spectrum (FHSS). It also has centralized medium access control through time division multiple access and TDD. These features indicate that Bluetooth is less prone to interference than from other Bluetooth networks. Bluetooth employs variable-sized packets up to a maximum of five frequency-hopping time slots of 625  $\mu$ s in duration. Every Bluetooth frame consists of a packet transmitted from a sender node over 1, 3, or 5 timeslots, while a receiver replies with a packet occupying at least one slot, so that each frame has an even number of slots. Therefore, in master to slave transmission, a single slot packet serves for a link layer stop-and-go ARQ message, whenever a corrupted packet payload is detected.

The timeout or retransmission limit value by default is set to an infinite number of retransmissions. On general grounds, this is unwise in conditions of fast fading caused by multipath echoes, as error bursts occur. Another source of error bursts is cochannel interference by other wireless sources, including other Bluetooth piconets, IEEE 802.11b,g networks, cordless phones, and even microwave ovens. Though this has been alleviated to some extent in version 1.2 of Bluetooth by adaptive frequency hopping [29], this is only effective if interference is not across all or most of the 2.402 to 2.480 GHz unlicensed band. However, both IEEE 802.11b and g may occupy a 22 MHz subchannel (with 30 dB energy attenuation over the central frequency at  $\pm 11$  MHz) within the 2.4 GHz band. Issues of interference might arise in apartment blocks with multiple sources occupying the 2.4 GHz band or when higher power transmission occurs such as at WiFi hotspots.

For Bluetooth, an ARQ may occur in the following circumstances [30]: (a) failure to synchronize on the access header code; (b) header corruption detected by a triple redundancy code; (c) payload corruption detected by cyclic redundancy check; (d) failure to synchronize with the return packet header; and (e) header corruption of the return packet. Notice that a faulty ARQ packet can itself cause retransmission. The main cause of packet error [30], however, is (c) payload corruption, which is the simplified assumption in this paper.

#### 3.2. Analysis of ARQ impact

Given the probability of bit error,  $P_e$ , then  $P_s$ , the probability of a successful packet transmission is defined as

$$P_s = (1 - P_e)^L, \quad (1)$$

where  $L$  is the bit length of a packet. Variations of the following analysis (1) to (5) are well known, occurring,

for example, in [31]. Furthermore, the expected number of retransmissions,  $N$ , under the default ARQ scheme is

$$E[N] = 0 \times P_s + 1 \times P_s \times (1 - P_s) + 2 \times P_s \times (1 - P_s)^2 + \dots,$$

$$E[N] = \frac{1 - P_s}{P_s} \quad (2)$$

which implies that the expected total number of transmissions,  $E[T]$ , is simply

$$E[T] = E[N] + 1 = \frac{1}{P_s}. \quad (3)$$

More interestingly, for a maximum number of retransmissions  $M$  the expected number of retransmissions is

$$E[N] = P_s \times \sum_{n=1}^{M-1} n \times (1 - P_s)^n + M$$

$$\times \left( 1 - \left[ P_s \times \sum_{n=1}^{M-1} (1 - P_s)^n \right] \right), \quad (4)$$

$$E[N] = \frac{(1 - P_s)(1 - (1 - P_s)^M)}{P_s},$$

and again  $E[T] = E[N] + 1$ .

The mean packet departure rate,  $S$  packet/s, from the Bluetooth send buffer is given by

$$S = \frac{1}{(n + 1) \times 625 \mu\text{s} \times E[T]}, \quad (5)$$

where  $n$  is the number of slots occupied by a Bluetooth packet. Assume that packets are fully filled (refer to Section 3.7) and, to find an upper bound on waiting time, that the buffer is fully occupied in a bad state. This means that a simple scaling may be applied to (5) based on the packet bit length. Figure 1 plots packet delay against the probability of a bit error for various retransmission policies. In Figure 1, the buffer size is set to 50 packets, assuming that just one picture type packet, I-picture, is in use. In practice, the buffer will not become fully occupied immediately and the effect of a DAB is to remove packets from the buffer but the plots in Figure 1 present the general situation for  $n = 5$  (packet payload 1021 B). Clearly, delay climbs more rapidly under infinite ARQ within a critical region around  $P_e = 10^{-4}$ .

### 3.3. Fuzzy logic control

A fuzzy subset is expressed as a set of rules which take the form of linguistic expressions. These rules express experience of tuning the controller and are captured in a knowledge database. An inference engine 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, defuzzification converts inferred

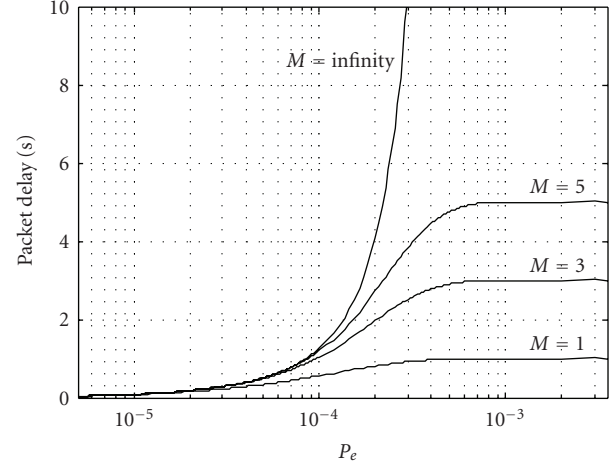


FIGURE 1: Packet delay against  $P_e$  (logarithmic horizontal scale) for varying values of  $M$  (max number of retransmissions).

fuzzy control decisions from the inference engine to a crisp or precise value, which is converted to a control signal.

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 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 nonmembership.

As a simple example, in a fuzzy subset of (say) “tall,” the possibility that a person with a given height taken from the set  $S$  of heights may be called tall is modeled by a membership function, which is the mapping between a data value and possible membership of the subset. Notice that a member of one fuzzy subset can be a member of another fuzzy subset with the same or a different possibility. Membership functions may be combined according to a set of “if ... then” rules to make inferences such as if  $x$  is tall and  $y$  is old then  $z$  is happy, in which tall, old, and happy are membership functions of the matching fuzzy subsets and  $x$ ,  $y$ ,  $z$  are linguistic variables (names for known data values).

In practice, the membership functions are applied to the data values to find the possibility of membership of a fuzzy subset and the possibilities are subsequently combined through defuzzification to provide a precise output. We have applied a semimanual method of deriving the rules, combining human knowledge of network behavior with testing by simulator.

The fuzzy model behavior itself was examined through Matlab fuzzy toolbox v. 2.2.4. This results in a widely applicable but static set of rules. The FLC’s behavior can be predicted from its output surface, formed by knowledge of its rule table and the method of defuzzification. For example, Matlab’s toolbox allows a set of output data points to be calculated to a given resolution, allowing interpolation of the surface.

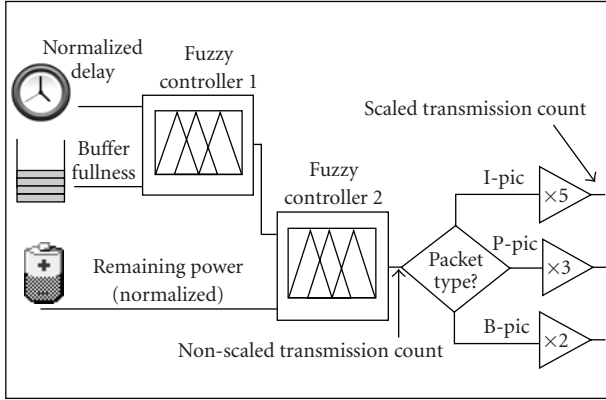


FIGURE 2: Overview of the FLC of ARQ system.

### 3.4. Fuzzy logic control of ARQ

Figure 2 shows the complete two-stage FLC adaptive ARQ system. For the first stage, there are two inputs: buffer fullness and the normalized delay of the head of the queue packet. Bluetooth buffer fullness is a preferable measure (compared to delay or packet loss) of channel conditions and of buffer congestion, as was established in [32]. Buffer fullness is available to an application via the host controller interface (HCI) presented by a Bluetooth hardware module to the upper layer software protocol stack. As an FLC input, buffer fullness is normalized to the size of the send buffer.

The retransmission count of the packet at the head of the Bluetooth send queue will affect the delay of packets still to be transmitted. Retransmissions overcome the effect of noise and interference but also cause the send buffer queue to grow, with the possibility of packet loss from send buffer overflow, which is why it is necessary also to introduce a DAB. The second FLC input modulates the buffer fullness input by the already experienced delay of the head of queue packet.

The output of the first stage FLC forms the input of the second stage FLC. The other input to the second stage is normalized remaining power, assuming a predetermined power budget for streaming of a particular video clip, which diminishes with time and retransmissions. The output of the second stage is a transmission count, which is subsequently scaled according to picture type importance. Though it might be possible to modify the first stage output by non-fuzzy logic means, by keeping the whole within an FLC framework, the possibility of complex power models is allowed for.

The assigned membership functions, which were achieved heuristically, are shown in Figures 3(a) and 3(b), and once found remain fixed. The buffer fullness range in Figure 3(a) is [0–1] corresponding to a percentage fullness. In Figure 3(b), the horizontal axis represents the delay time of the packet at the head of the queue divided by the display deadline. In Figure 3(b), unit delay corresponds to expiration of playout deadline. It is important to note that any packet in the send buffer is discarded if its deadline has expired. However, this takes place after the fuzzy evaluation of the

desired ARQ retransmission count. In practice, the inputs to the FLC were sampled versions of buffer fullness and packet delay deadline, to avoid excessive ARQ retransmission count oscillations over time. The sampling interval was every 20 packets. Table 1 shows the “if ... then” rules that allow input fuzzy subsets to be combined to form an output from stage one and an input to stage two. Notice more than one rule may apply because of the fuzzy nature of subset membership. The output of stage one is combined with a fuzzy input for “remaining power,” and the “if ... then” resulting in the final nonscaled transmission count in Table 2.

The inputs were combined according to the well-known Mamdani model [33] to produce the output values for each stage. The standard center of gravity method was employed to resolve to a crisp output value, according to the output membership functions shown in Figures 3(c) and 3(e). The fuzzy control surfaces are represented in Figure 4, as derived from the Matlab Fuzzy Toolbox v. 2.2.4. As mentioned in Section 1, by means of an LUT derived from the surface, a simple implementation becomes possible.

Clearly a packet can only be transmitted an integer number of times but the final crisp output may result in a real-valued number. This difficulty was resolved by generating a random number from a uniform distribution. If the random number was less than the fractional part of the crisp output value then that value was rounded up to the nearest integer, otherwise it was rounded down. Notice that this means that, for (say) a less important B-picture packet very close to its display deadline, a packet at the head of the queue may never be transmitted because of the impact upon more important packets still remaining in the send buffer. The advantage of the randomization procedure over simple quantization is that, in the long term, the mean value of the output numbers of transmissions will converge more closely to a desired output level. The output value was subsequently scaled according to the priority of the packet’s picture type. The complete algorithm including randomization and scaling is summarized in Figure 5.

A simple scaling of 5 : 3 : 2 was applied, respectively, for I-, P-, B-pictures, giving up to a maximum of five transmissions. The value of five retransmissions was selected to be inline with the experiments reported in [26]. Subsequently, the retransmission limit for the other picture types was scaled accordingly. In practice, the scaling was applied to the crisp value output after defuzzification. For example, if the crisp output value was 0.7, and a P-picture packet was involved then the value after scaling is  $0.7 \times 3.0 = 2.10$ . Then, the random-number-based resolution results in three transmissions if the random number is less than or equal to 0.10 and two transmissions otherwise.

### 3.5. Deadline-aware buffer

In the conservative send buffer discard policy of this paper, all packets of whatever picture type have a display deadline, which is the size of the playout buffer expressed as a time beyond which buffer underflow will occur. In a conservative policy, the deadline is set as the maximum time that the playout buffer can delay the need for a packet. In the

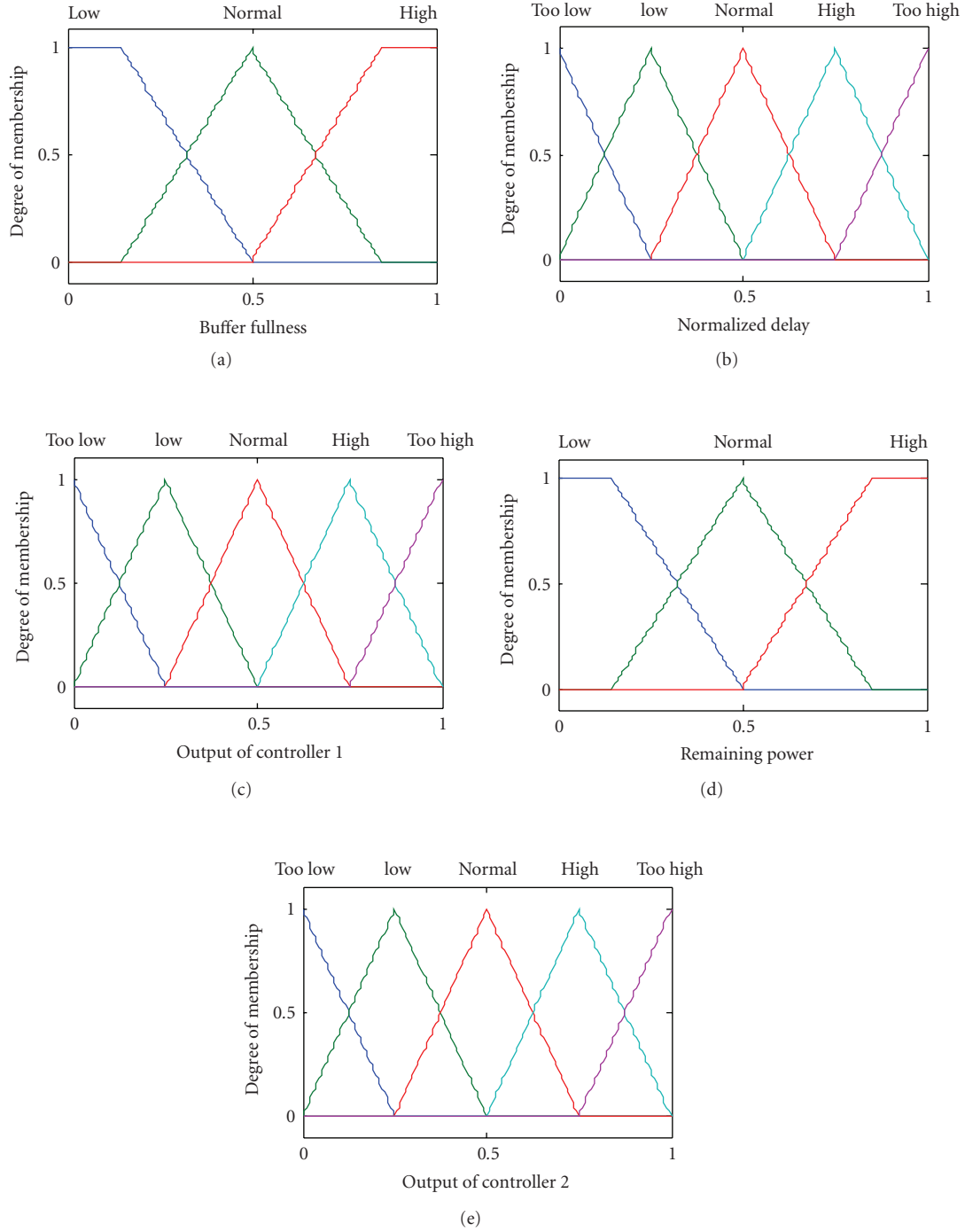


FIGURE 3: Fuzzy membership functions: (a) stage one, input buffer fullness; (b) stage one, input delay deadline; (c) output of stage one controller; (d) stage two input remaining power; (e) stage two output transmission count.

TABLE 1: FLC stage one if ... then rules used to identify output fuzzy subsets from inputs.

		Delay/deadline				
		<i>Too low</i>	<i>Low</i>	<i>Normal</i>	<i>High</i>	<i>Too high</i>
Buffer fullness	<i>High</i>	Normal	Normal	Low	Too low	Too low
	<i>Normal</i>	Too high	High	Normal	Low	Too low
	<i>Low</i>	Too high	Too high	High	Low	Too low

TABLE 2: FLC stage two *if ... then* rules used to identify output fuzzy subsets from inputs.

		Output1				
		<i>Too low</i>	<i>Low</i>	<i>Normal</i>	<i>High</i>	<i>Too high</i>
Remaining power	<i>High</i>	Too low	Low	High	Too high	Too high
	<i>Normal</i>	Too low	Low	Normal	High	High
	<i>Low</i>	Too low	Too low	Low	Low	Normal

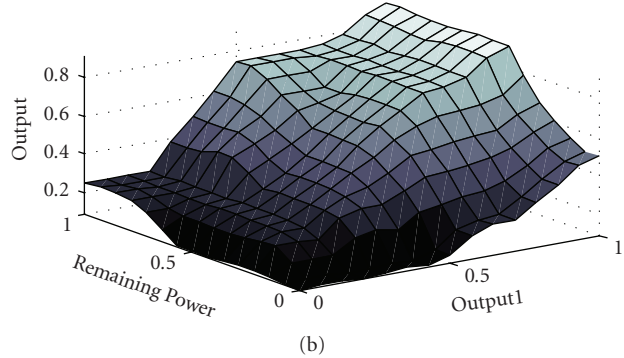
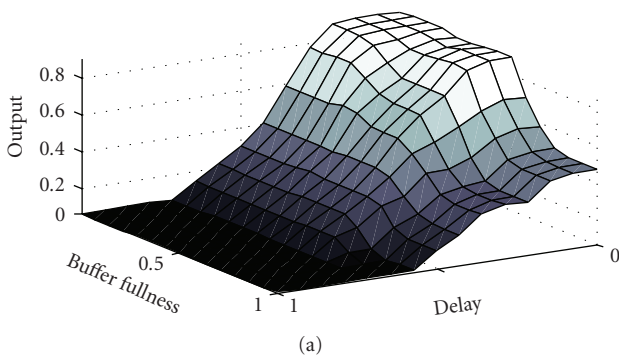


FIGURE 4: (a) Stage one, FLC control surface resulting from FLC ARQ; (b) stage two, control surface giving the transmission count output (before subsequent scaling).

simulations of Section 4, the display deadline was set to 0.10 second.

In addition to the display deadline, all I-picture packets have a decode deadline, which is the display time remaining to the end of the GOP. This is because reference pictures (I- or P-) are still of value to the receiver as they serve in the decoding of subsequent pictures, even after their display deadline has elapsed. Thus, for a 12-picture GOP, this is the time to display 11 frames, that is, 0.44 second at 25 frame/s. For P-picture packets, the decode deadline will vary depending on the number of frames to the end of the GOP. For B-pictures the decode deadline is set to zero.

The decode deadline is added to the display deadline and a packet is discarded from the send buffer after its total deadline expires. By storing the GOP end time, an implementation performs one subtraction to find each decode deadline. Account has been taken of I- B- P-picture reordering at encode and send buffer output, which has an effect on buffer fullness. Reordering is introduced to ensure that reference pictures arrive and can be decoded before the dependent B-pictures. In the discard policy, packet handling and propagation delay are assumed (optimistically) to be constant. In all experiments, the buffer queue discipline is assumed to be first-in, first-out.

**3.6. Channel model**

Wireless channel errors are usually bursty and dependent in time, rather than independent and identically distributed. For this reason, we adopt a Gilbert-Elliott [34, 35] two state discrete-time, ergodic Markov chain to model the wireless

channel error characteristics between a Bluetooth master and slave node. By adopting this model, it is possible to simulate burst errors of the kind that cause problems to an ARQ mechanism. The Gilbert-Elliott model was, in [36], applied to the same version of Bluetooth as herein.

The mean duration of a good state,  $T_g$ , was set at 2 seconds and in a bad state,  $T_b$ , was set to 0.25 second. In units of  $625 \mu s$  (the Bluetooth time slot duration),  $T_g = 3200$  and  $T_b = 400$ , which implies from

$$T_g = \frac{1}{1 - P_{gg}}, \quad T_b = \frac{1}{1 - P_{bb}}, \quad (6)$$

that, given the current state is good ( $g$ ),  $P_{gg}$ , the probability that the next state is also  $g$ , is 0.9996875 and, given the current state is bad ( $b$ ),  $P_{bb}$ , the probability that the next state is also  $b$ , is 0.9975. The transition probabilities,  $P_{gg}$  and  $P_{bb}$ , as well as the bit-error rate (BER) are approximately similar to those in [37], but the mean state durations are adapted to Bluetooth. At 3.0 Mbps, the BER during a good state was set to  $a \times 10^{-5}$  and during a bad state was set to  $a \times 10^{-4}$ , where  $a$  is a scaling factor and is subsequently referred to as the channel parameter.

**3.7. Bluetooth adaptive ARQ schemes**

Unfortunately, in respect to Bluetooth, we are not aware of other adaptive ARQ that would form a direct point of comparison to our FLC scheme, particularly if a power budget is factored in. As an alternative Bluetooth comparison, an adaptive ARQ scheme designed for audio streaming [34] was

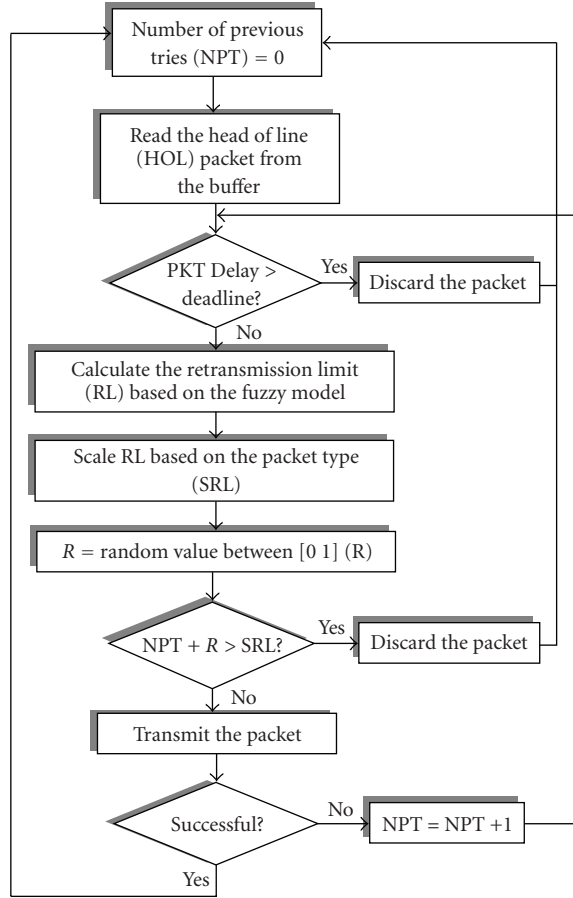


FIGURE 5: FLC algorithm for processing a packet.

considered. For ease of reference, the details are summarized in this section.

In [38], the round-trip time (RTT) was measured at the link layer. The RTT was then smoothed over time, using a forgetting constant  $\gamma$  to form the smoothed RTT (SRTT). From these values, a retransmission timeout (RTO) was formed. The RTO forms a threshold on the number of ARQ retransmissions,

$$\begin{aligned}
 \text{SRRT} &= (1 - \gamma) \times \text{SRTT} + \gamma \times \text{RTT}, \\
 \text{RTO} &= \begin{cases} \alpha \times \text{RTO}, & \text{if } \text{RTT} < \text{SRTT}, \\ \beta \times \text{RTO}, & \text{if } \text{RTT} > \text{SRTT}, \\ \text{RTO}, & \text{if the previous packet was lost.} \end{cases} \quad (7)
 \end{aligned}$$

In simulations, the values of  $\gamma = 0.25$ ,  $\alpha = 1.1$ ,  $\beta = 0.9$  were adopted from [34] as bounds on RTO, namely  $\text{RTO}_{\min}$  was set to the total time to send a packet,  $T_{\text{Packet}}$ , which is the Bluetooth packet length divided by the arrival rate at the Bluetooth sender of the data forming that packet. The upper bound was set as follows:

$$\text{RTO}_{\max} = T_{\text{packet}} \times \text{Max}(\text{AvailBuff} \times 75\%, 2), \quad (8)$$

where AvailBuff is the remaining free space in the buffer.

Because this adaptive ARQ algorithm relies on a calculation of the available buffer space in the Bluetooth send buffer,

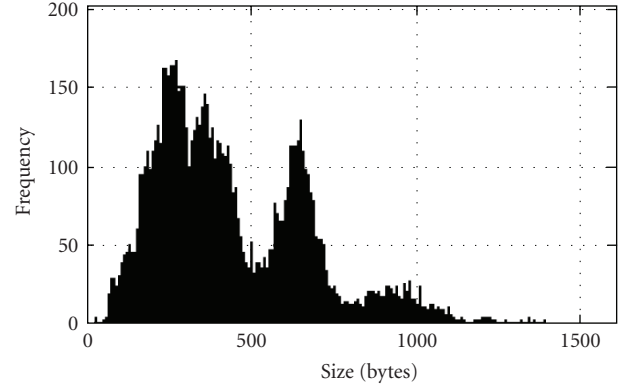


FIGURE 6: Distribution of slice sizes for the encoded video clip.

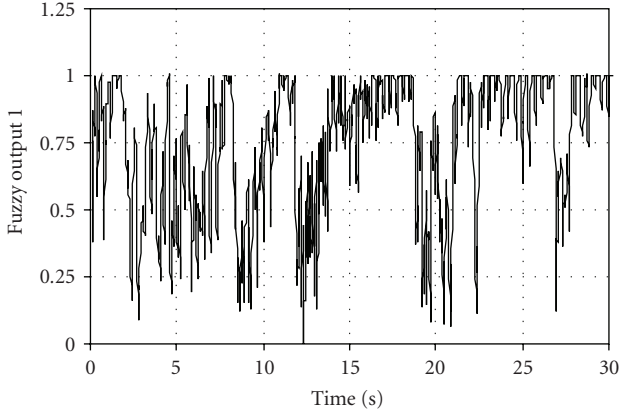
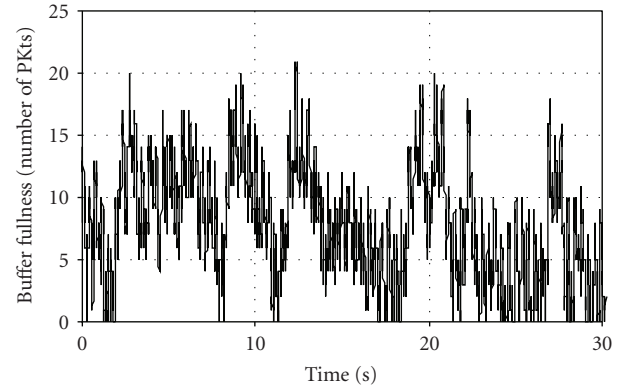
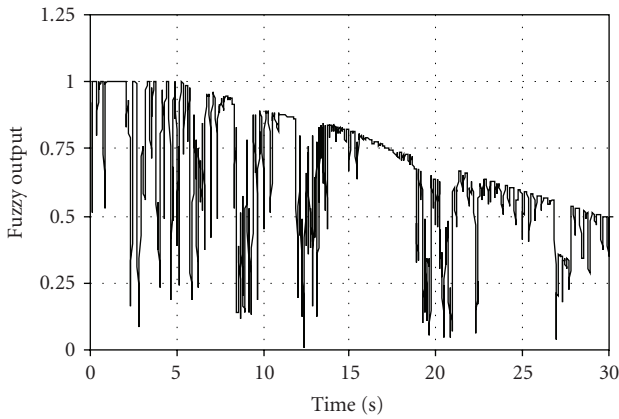
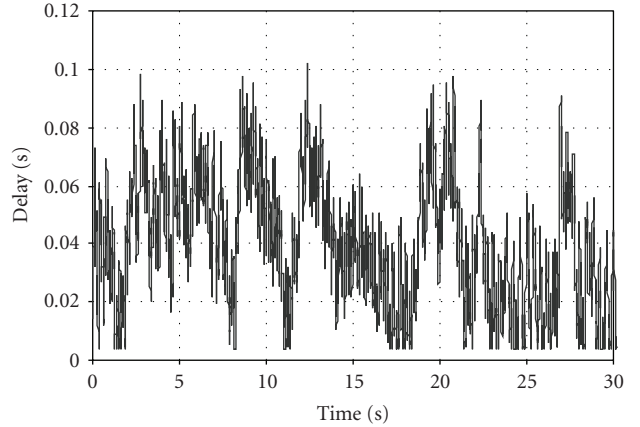
it is not possible to combine this algorithm with the use of a DAB. As the adaptive ARQ system relies on buffer fullness to adjust the number of retransmission, if a DAB is employed, expired packets will be actively removed from the buffer, keeping the buffer fullness at a low level. This will mislead the algorithm as it will interpret this low buffer fullness as a sign of the available capacity in the network and increase the number of retransmissions. Because our purpose was to make a fair comparison and because the absence of a DAB unfairly increases packet delays compared to default ARQ and FLC ARQ, in simulations with this adaptive ARQ algorithm, packets were not dropped at the receiver if their frame had missed its display deadline at the receiver. This compensates the calculated PSNR for this algorithm in the results in Section 4.

### 3.8. Simulation setup

This research employed the University of Cincinnati Bluetooth (UCBT) extension (a download is available from <http://www.ececs.uc.edu/~cdmc/ucbt/>) to the well-known ns-2 network simulator (v. 2.28 used). The UCBT extension supports Bluetooth EDR but is also built on the air models of previous Bluetooth extensions such as BlueHoc from IBM and Blueware. The Gilbert-Elliott channel model was coded in C++ to be called by an ns-2 object tcl (otcl) script. All links were set at the maximum EDR 3.0 Mbps gross air rate. Each of the simulation runs was repeated twenty times and the results were averaged to produce summary statistics.

The simulations were carried out principally with input from an MPEG-2 encoded bitstream at a mean rate of 1.5 Mbit/s for a 30-second video clip with moderate motion, showing a newsreader and changing backdrop, which we designate “News.” (Other video inputs are summarized in Section 4.) PSNR was found by reconstructing with a reference MPEG-2 decoder. The display rate was 25 frames/s, resulting in 750 frames in each run. The source video was common intermediate format (CIF)-sized ( $366 \times 288$  pixels) with a GOP structure of  $N = 12$ , and  $M = 3$  (where in standard codecs  $N$  designates the GOP length and  $M$  is the number of pictures between anchor pictures). The slice size distribution of the input video clip is shown in Figure 6.



FIGURE 7: Output from stage one of the FLC, with  $a = 2$ .FIGURE 9: Buffer fullness input to stage one of the FLC, with  $a = 2$ .FIGURE 8: Output from stage two of the FLC, with  $a = 2$ .FIGURE 10: Delay input to stage one of the FLC, with  $a = 2$ .

In [39], fully filled Bluetooth packets were formed using maximal bandwidth five time slot packets, regardless of slice boundaries. These packets carry a 1021 B payload. While this results in some loss in error resilience, as each MPEG-2 slice contains a decoder synchronization marker, in [39], it is shown that the overall video performance is superior to choice of smaller packet sizes.

## 4. RESULTS

### 4.1. Fuzzy logic model response

Figure 7 shows the output of stage 1 of the FLC as the “News” video clip of Section 3.7 was passed across a Bluetooth link with channel parameter  $a$  set to two. The high variability of the output is due to the repeated onset of bad states occasioned by the Gilbert-Elliott channel model (Section 3.5).

The normalized power budget for the clip declines with the number of bits passed across the link and the loss is exacerbated by repeated retransmissions during bad states. As the power budget changes linearly, this has the effect of modulating the original input, as illustrated in Figure 8, again with channel parameter set to two.

After the removal of deadline expired packets, through operation of the deadline aware buffer (DAB) described in Section 3.4, the buffer fullness input to stage one of the send buffer oscillates around a level well below the 50-packet maximum, Figure 9. Head-of-line packet delay, Figure 10, acts as a typical trimming input to the FLC stage one unit, as its pattern resembles that of buffer fullness over time. Notice that for the default ARQ scheme, Figure 11, delay is frequently over the 0.10 second display deadline and, therefore, B-picture packets face the possibility of being dropped without transmission if they have already spent longer than that time in the send buffer, while I- and P-picture packets have the grace arising from their extra decode deadline time.

### 4.2. Response of FCL, default ARQ, and adaptive ARQ

A comparison was made between the default scheme with infinite ARQ, the adaptive ARQ scheme of Section 3.6, and the FLC scheme. These schemes were all allocated an infinite power budget. The FLC scheme with power control was then introduced. To improve the comparison, the default static ARQ scheme was compared with a DAB in place, though,

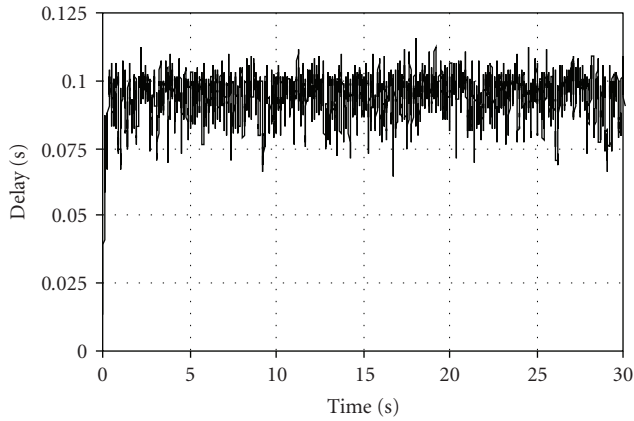


FIGURE 11: Delay in default ARQ with DAB, with  $a = 2$ .

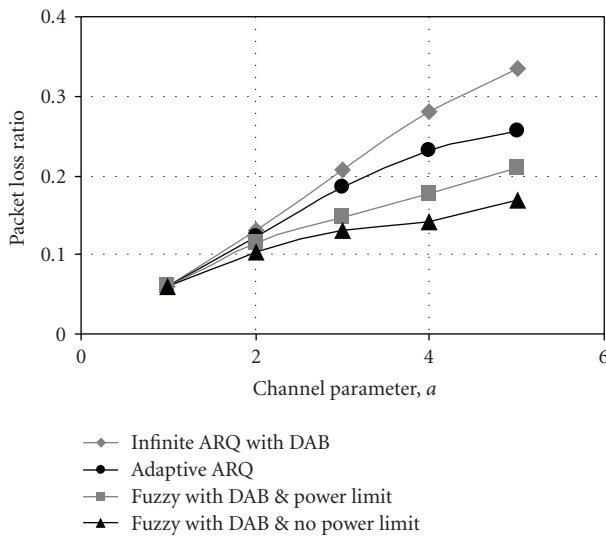


FIGURE 12: Packet loss during transmission of the “News” video clip, with the default scheme and the FLC power-aware scheme.

of course, a DAB is not a feature of the Bluetooth standard. The channel parameter,  $a$ , was varied in the tests to show the impact of differing channel conditions.

Figure 12 compares the ratio of packets lost to total packets arriving in the send buffer. The FLC ARQ is superior in worsening channel conditions both to default static ARQ and the adaptive scheme [34]. Even when compensating for a diminishing power budget, the FLC scheme shows a clear improvement. By monitoring the local (sender) buffer fullness and reducing the number of retries in the event of congestion, packet loss due to buffer overflow is reduced. In addition, as delay is also considered by the FLC, it is less likely that a packet’s delay exceeds the display deadline (and therefore, removed by the DAB scheme). Therefore, the total packet loss rate is reduced when the proposed scheme is employed. Of course, when a power constraint is also considered, the packet loss rate will be compromised but as the Figure 12 shows the FLC still outperforms the other schemes.

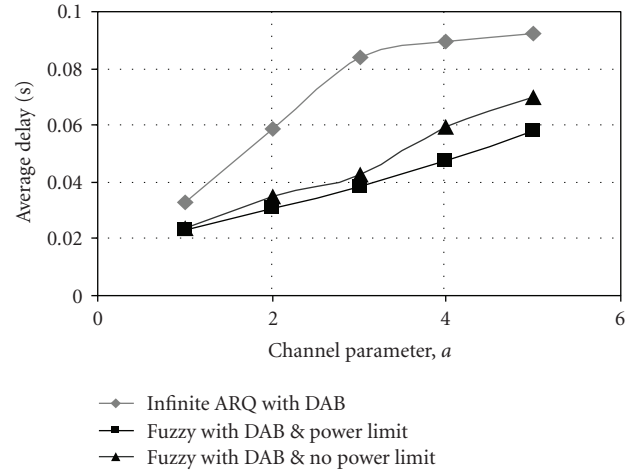


FIGURE 13: Average packet delay during transmission of the “News” video clip, with the default scheme and the FLC power-aware scheme.

The average delay of successfully transmitted packets was also considerably reduced under the FLC schemes, Figure 13, while the default ARQ scheme results in a more rapid climb to its peak average value. Larger average delay will impact start-up time in one-way streaming and will add to overall delay in a two-way video exchange, such as for a videophone connection. Notice that removing the power budget results in more delay for the FLC scheme than with a power budget because the scheme is not handicapped by the need to reduce transmissions for power considerations. Either way the scheme is superior to default ARQ in delay (and also in reduced packet loss). As remarked in Section 3.6, the adaptive ARQ scheme is disadvantaged by the lack of a DAB and for that reason its results are not plotted in Figure 13.

Crucially, the FLC is able to save power over both the nonpower-aware default ARQ and the adaptive scheme, Figure 14. The impact is clearly greater as channel conditions worsen. Closer inspection of the distribution of packet losses between the picture types shows the advantage of FLC ARQ, Figure 16, as less B-picture packets and more reference picture packets are lost under default ARQ, Figure 15.

In fact, the loss pattern of the default ARQ replicates the distribution of packet types within the input video clip, Table 3, whereas FLC does not, as is clear by comparing the final two columns of Table 3. This is because the FLC is able to take account of packet type through the delay deadline of the head-of-line packet and because the number of transmissions output is scaled according to the picture type.

Considering the packet loss statistics of Figure 12 and the distribution of those packet losses between packet picture types from Figures 15 and 16, it is not surprising, Figure 17, that the mean PSNR of FLC ARQ is better than that of the other schemes and the relative advantage becomes more so as the channel conditions worsen. A significant part of that advantage is also due to the superiority of FLC ARQ and there is little difference between FLC ARQ with and without a power budget in better channel conditions. Notice that for

TABLE 3: Percentage distribution of input and lost packets by picture type.

Picture type	Packets in input video bitstream (%)	Lost packets in infinite ARQ scheme (%)	Lost packets in fuzzy ARQ scheme (%)
I	17.97	17.29	7.50
P	37.93	36.92	22.91
B	44.10	45.79	69.59

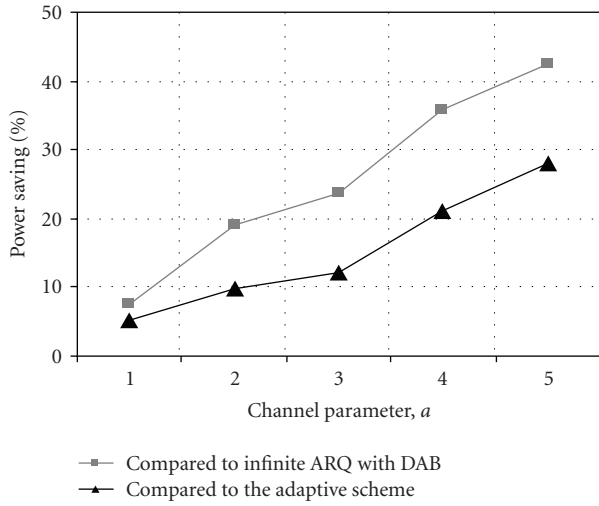


FIGURE 14: Relative power saving of the FLC power-aware ARQ scheme compared to that of the default ARQ and the adaptive ARQ [34] schemes.

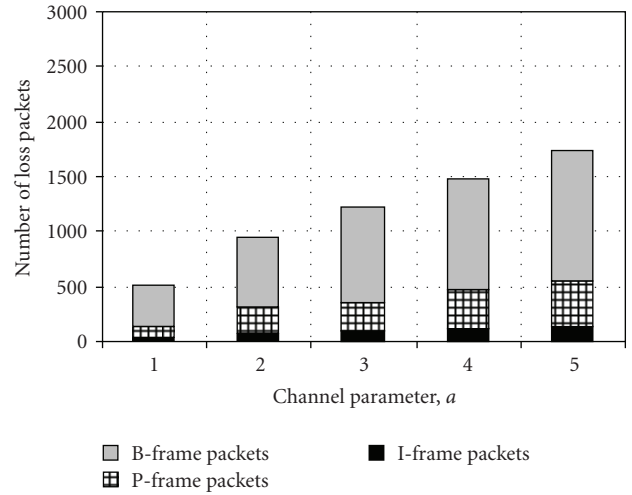


FIGURE 16: Contribution of I-, P-, and B-frame packet losses to total packet loss in the FLC with DAB and power-aware scheme.

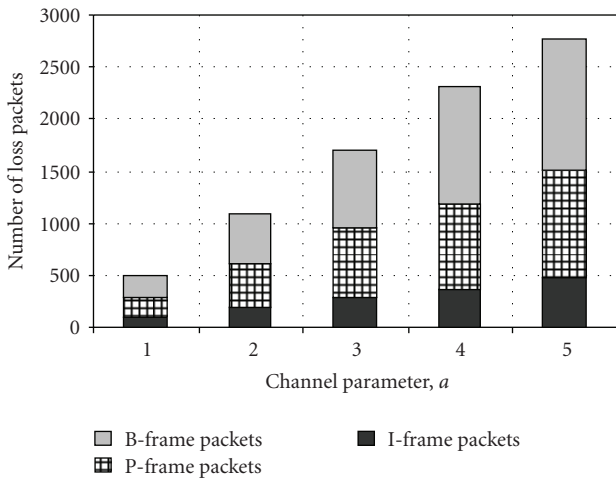


FIGURE 15: Contribution of I-, P-, and B-frame packet losses to total packet loss in the infinite ARQ with DAB scheme.

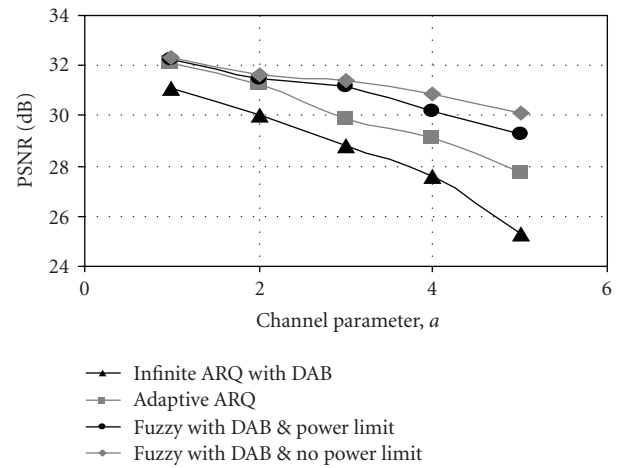


FIGURE 17: Comparison of PSNR for the “News” video clip between the Bluetooth default and FLC ARQ schemes with DAB.

power-aware control averaged PSNR, figures do not “show the whole story,” as the achievable PSNR will deteriorate over time, as the available power becomes less. This confirms previous experience [40] that for the very worst channel conditions shown in Figure 17, that is,  $a = 5$ , then the mean PSNR is improved by around 3 dB if an infinite power budget

is assumed. Thus, power conservation comes at a cost to the receiver in reduced video quality and is a trade-off that might be open to user configuration. The contribution of the current paper is that power-awareness has been realistically factored in, resulting in over 40% saving in power, Figure 14, in the same conditions.

TABLE 4: Comparison of video quality between power-aware FLC ARQ with DAB, and default ARQ with DAB for various video clips.

	$a = 1$		$a = 2$		$a = 3$		$a = 4$		$a = 5$	
	FLC	Inf ARQ	FLC	Inf ARQ	FLC	Inf ARQ	FLC	Inf ARQ	FLC	Inf ARQ
News	32.22	31.12	31.45	30.05	31.18	28.78	30.19	27.56	29.29	25.33
Football	31.32	29.91	30.85	28.81	30.01	27.55	28.95	26.31	28.19	24.88
Friends	31.88	30.09	30.94	28.94	30.14	27.77	29.32	26.67	28.34	24.29
Italian job	30.97	30.41	30.39	29.01	30.02	27.62	29.19	26.55	28.45	24.09

### 4.3. Detailed comparison of video quality

As an illustrative example of the fluctuations of video quality over time, for the purposes of this test only, the power budget was artificially set to  $60 \times 10^6$  bits for both the Bluetooth default ARQ scheme and power-aware FLC ARQ. The choice of the power budget is arbitrary and the reader can refer back to Figure 14 for a quantitative comparison of the relative power savings, with no specific power budget imposed. Again, a DAB was in place for both tests. Figure 18 demonstrates a falling trend in PSNR quality over the course of the video clip stream, though quality is generally high for a wireless channel (channel parameter  $a = 2$ ). In contrast, under the default ARQ scheme (see Figure 19), not only is the video quality lower but also there is an abrupt end to transmission as the bit budget runs out at about 652 frames. Figure 20 shows the equivalent result for adaptive ARQ of [38], when it will be seen that the objective video quality is more variable than FLC.

To verify the generality of the results, a comparison was made across a variety of video clips. Table 4 provides summary statistics (mean of 20 runs) for the different schemes, with four input video sequences: (1) “News,” as in previous experiments in this section, (2) “Football” with rapid movement, (3) “Friends” from the well-known American situational comedy, with more “action” than in “News” and equally, (4) “Italian job,” with an extract from the well-known film including car chases. The additional clips had the same GOP structure as the “News” sequence and similarly were CIF-sized at 25 frames/s and were encoded at the same rate. The default ARQ with DAB scheme once again was deployed with an infinite power budget, whereas once again the FLC ARQ regulated its power allocation for each video clip over the course of the streaming session. Over changing channel conditions, Table 4, the results for video quality are broadly similar, except that the greater motion in the other clips results in a lower received video quality.

## 5. CONCLUSIONS

Power usage becomes an important factor when mobile stations are employed in ad hoc mode or when the receiver is a mobile device. The proliferation of such devices implies that any reduction in the recharge frequency is a welcome development. Certainly, alternatives to Bluetooth are considered in power terms, whether Wibree or as a commercial sensor network IEEE 802.15.4 (Zigbee). Transmission of higher quality video over a Bluetooth interconnect has been

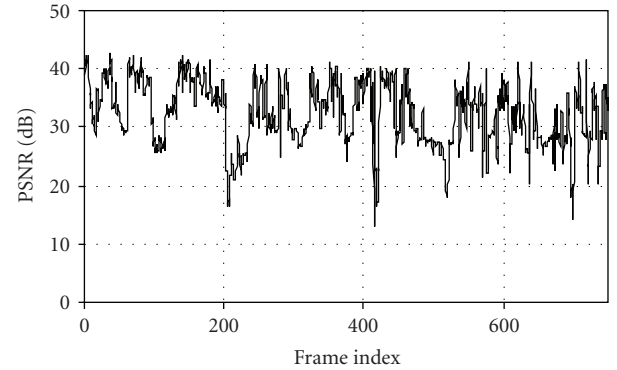


FIGURE 18: PSNR over the 750 frames, showing a falling trend in PSNR over time, for the power-aware FLC ARQ with DAB.

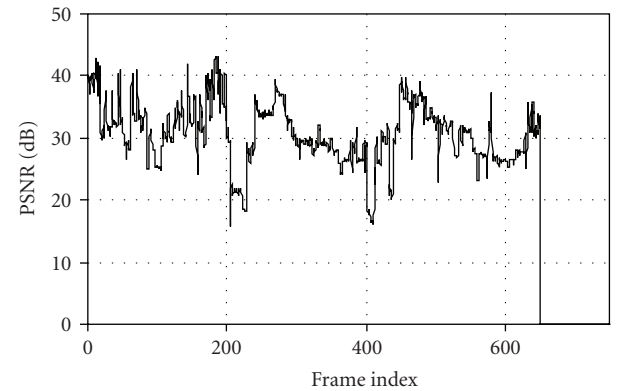


FIGURE 19: PSNR over the 750 frames of the input video, under a fixed power budget with the default ARQ with DAB.

long sought. However, it is important to factor in power usage and not simply regard a wireless channel as a fixed channel with the addition of errors, to caricature one view. In this paper, fuzzy logic control of ARQ is able to respond to a fixed power budget, which diminishes over time. Other factors included are packet delay deadlines (both display and, for anchor picture packets, decode deadlines), and send buffer congestion. Because an ARQ management system is not also able to manage the send buffer size, a deadline aware buffer (DAB) removes expired packets. This avoids retransmission attempts on these packets and more importantly

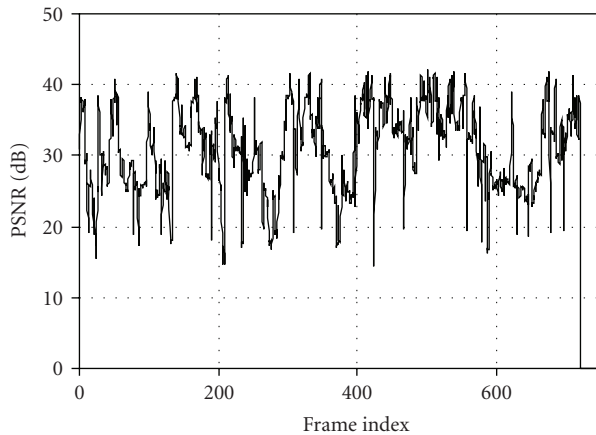


FIGURE 20: PSNR over the 750 frames of the input video, under a fixed power budget with an adaptive ARQ scheme [34].

prevents send buffer overflow and excessive waiting times for other queued packets. For fairness, both the default infinite ARQ scheme and the FLC scheme were compared with the addition of a DAB. An adaptive ARQ scheme specifically for Bluetooth from the literature was also compared, though the nature of the algorithm did not permit the use of a DAB. However, FLC, which varies its transmission policy with packet picture type, still outperforms both the default static ARQ scheme and the conventional, adaptive ARQ, resulting in the end analysis in superior delivered video quality. This is despite the need to adjust the transmission policy as available power diminishes, whereas infinite battery power is assumed for the default scheme. The FLC framework, being modular, allows for a future power model that takes into account not only energy loss from transmission but also models energy taken up at the encoder and/or the decoder.

## ACKNOWLEDGMENT

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

## REFERENCES

- [1] J. Haartsen, "The Bluetooth radio system," *IEEE Personal Communications*, vol. 7, no. 1, pp. 28–36, 2000.
- [2] S. Mattisson, "Low-power considerations in the design of Bluetooth," in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED '00)*, pp. 151–154, Portacino Coast, Italy, July 2000.
- [3] M. Tamai, T. Sun, K. Yasumoto, N. Shibata, and M. Ito, "Energy-aware video streaming with QoS control for portable computing devices," in *Proceedings of the 14th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '04)*, pp. 68–73, Cork, Ireland, June 2004.
- [4] P. Agrawal, J.-C. Chen, S. Kishore, P. Ramanathan, and K. Sivalingam, "Battery power sensitive video processing in wireless networks," in *Proceedings of the 9th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '98)*, vol. 1, pp. 116–120, Boston, Mass, USA, September 1998.
- [5] K. Choi, K. Kim, and M. Pedram, "Energy-aware MPEG-4 FGS streaming," in *Proceedings of the 40th Design Automation Conference*, pp. 912–915, Anaheim, Calif, USA, June 2003.
- [6] A. Iranli, K. Choi, and M. Pedram, "Energy-aware wireless video streaming," in *Proceedings of the 1st Workshop on Embedded Systems For Real-Time Multimedia (ESTImedia '03)*, pp. 48–55, Newport Beach, Calif, USA, October 2003.
- [7] H.-S. Kim, D. T. Duong, J.-Y. Jeong, B.-K. Dan, and S.-J. Ko, "Power-aware rate control for mobile multimedia communications," in *Proceedings of the 5th International Conference on Ad-Hoc, Mobile, and Wireless Networks (ADHOC-NOW '06)*, vol. 4104 of *Lecture Notes in Computer Science*, pp. 458–471, Ottawa, Canada, August 2006.
- [8] R. Cornea, A. Nicolau, and N. Dutt, "Content-aware power optimizations for multimedia streaming over wireless networks," Tech. Rep. CECS #06-13, University of California, Berkeley, Calif, USA, 2006.
- [9] Specification of the Bluetooth System—2.0 + EDR, 2004, <http://www.bluetooth.com>.
- [10] E. Ferro and F. Potorti, "Bluetooth and Wi-Fi wireless protocols: a survey and a comparison," *IEEE Wireless Communications*, vol. 12, no. 1, pp. 12–26, 2005.
- [11] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "End-to-end QoS for video delivery over wireless Internet," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 123–133, 2005.
- [12] M. Chen and G. Wei, "Multi-stages hybrid ARQ with conditional frame skipping and reference frame selecting scheme for real-time video transport over wireless LAN," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 158–167, 2004.
- [13] R. Razavi, M. Fleury, and M. Ghanbari, "Fuzzy logic control of adaptive ARQ for video distribution over a Bluetooth wireless link," *Advances in Multimedia*, vol. 2007, Article ID 45798, 13 pages, 2007.
- [14] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [15] H. Takagi, *Application of Neural Networks and Fuzzy Logic to Consumer Products*, vol. 1 of *IEEE Technology Updates Series: Fuzzy Logic Technology and Applications*, IEEE Press, New York, NY, USA, 1994.
- [16] A. Leone, A. Bellini, and R. Guerrieri, "An H.261-compatible fuzzy-controlled coder for videophone sequences," in *Proceedings of the 3rd IEEE Conference on Fuzzy Systems*, vol. 1, pp. 244–248, Orlando, Fla, USA, June 1994.
- [17] P. M. Grant, Y.-S. Saw, and J. M. Hannah, "Fuzzy rule based MPEG video rate prediction and control," in *Proceedings of EURASIP ECASP Conference*, pp. 211–214, Prague, Czech Republic, June 1997.
- [18] I. Baturone, A. Barriga, S. Sánchez-Solano, C. J. Jiménez-Fernández, and D. R. López, *Microelectronic Design of Fuzzy Logic-based Systems*, CRC Press, Boca Raton, Fla, USA, 2000.
- [19] M. Kalman, P. Ramanathan, and B. Girod, "Rate-distortion optimized video streaming with multiple deadlines," in *Proceedings of the International Conference on Image Processing (ICIP '03)*, vol. 3, pp. 661–664, Barcelona, Spain, September 2003.
- [20] J.-C. Chen, K. M. Sivalingam, P. Agrawal, and S. Kishore, "A comparison of MAC protocols for wireless local networks based on battery power consumption," in *Proceedings of the*

- 17th IEEE Annual Conference on Computer Communications (INFOCOM '98), vol. 1, pp. 150–157, San Francisco, Calif, USA, March–April 1998.
- [21] J.-C. Cano, J.-M. Cano, E. González, C. Calafate, and P. Manzoni, “Power characterization of a Bluetooth-based wireless node for ubiquitous computing,” in *Proceedings of the 2nd International Conference on Wireless and Mobile Communications (ICWMC '06)*, p. 13, Bucharest, Romania, July 2006.
- [22] Q. Zhang, Z. Ji, W. Zhu, and Y.-Q. Zhang, “Power-minimized bit allocation for video communication over wireless channels,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 398–410, 2002.
- [23] Y. Liang and I. Ahmad, “Power and content aware video encoding for video communication over wireless networks,” in *Proceedings of the IEEE Workshop on Signal Processing Systems (SIPS '04)*, pp. 269–274, Austin, Tex, USA, October 2004.
- [24] M. Tamai, T. Sun, K. Yasumoto, N. Shibata, and M. Ito, “Energy-aware QoS adaptation for streaming video based on MPEG-7,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '04)*, vol. 1, pp. 189–192, Taipei, Taiwan, June 2004.
- [25] M. Yokotsuka, “Memory motivates cell-phone growth,” *Wireless Systems Design*, vol. 9, no. 3, pp. 27–30, 2004.
- [26] Q. Li and M. van der Schaar, “Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation,” *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 278–290, 2004.
- [27] W. Tan and A. Zakhor, “Packet classification schemes for streaming MPEG video over delay and loss differentiated networks,” in *Proceedings of the 11th International Packet Video Workshop (PV '01)*, Kyongju, Korea, May 2001.
- [28] J. Wall and J. Y. Khan, “An adaptive ARQ enhancement to support multimedia traffic using 802.11 wireless LANs,” in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '04)*, vol. 5, pp. 3037–3041, Dallas, Tex, USA, November–December 2004.
- [29] N. Golmie, N. Chevrollier, and O. Rebala, “Bluetooth and WLAN coexistence: challenges and solutions,” *IEEE Wireless Communications*, vol. 10, no. 6, pp. 22–29, 2003.
- [30] M. C. Valenti, M. Robert, and J. H. Reed, “On the throughput of Bluetooth data transmissions,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '02)*, vol. 1, pp. 119–123, Orlando, Fla, USA, March 2002.
- [31] J. Proakis, *Digital Communications*, McGraw-Hill, New York, NY, USA, 2000.
- [32] R. Razavi, M. Fleury, and M. Ghanbari, “Detecting congestion within a Bluetooth piconet: video streaming response,” in *Proceedings of the London Communications Symposium*, pp. 181–184, London, UK, September 2006.
- [33] J. S. Jang, C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, Upper Saddle River, NJ, USA, 1997.
- [34] E. N. Gilbert, “Capacity of burst-noise channel,” *Bell System Technical Journal*, vol. 39, no. 9, pp. 1253–1265, 1960.
- [35] E. O. Elliott, “Estimates of error rates for codes on burst noise channels,” *Bell System Technical Journal*, vol. 42, no. 9, pp. 1977–1997, 1963.
- [36] L.-J. Chen, T. Sun, and Y.-C. Chen, “Improving Bluetooth EDR data throughput using FEC and interleaving,” in *Mobile Ad-Hoc and Sensor Networks*, vol. 4325 of *Lecture Notes in Computer Science*, pp. 724–735, Springer, Hong Kong, 2006.
- [37] R. Fantacci and M. Scardi, “Performance evaluation of preemptive polling schemes and ARQ techniques for indoor wireless networks,” *IEEE Transactions on Vehicular Technology*, vol. 45, no. 2, pp. 248–257, 1996.
- [38] L.-J. Chen, R. Kapoor, K. Lee, M. Y. Sanadidi, and M. Gerla, “Audio streaming over Bluetooth: an adaptive ARQ timeout approach,” in *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops (ICDCSW '04)*, pp. 196–201, Hachioji, Japan, March 2004.
- [39] R. Razavi, M. Fleury, and M. Ghanbari, “An efficient packetization scheme for Bluetooth video transmission,” *Electronic Letters*, vol. 42, no. 20, pp. 1143–1145, 2006.
- [40] R. Razavi, M. Fleury, and M. Ghanbari, “Fuzzy control of adaptive timeout for video streaming over a Bluetooth interconnect,” in *Proceedings of 12th IEEE Symposium on Computers and Communications (ISCC '07)*, pp. MW-27–MW-32, Santiago, Portugal, July 2007.