Fuzzy Logic Control of Power-aware Video Streaming over a Bluetooth Interconnect

Rouzbeh Razavi, Martin Fleury and Mohammed Ghanbari
ESE Dept., University of Essex, Colchester, United Kingdom

***Abstract*—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, FLC is shown to outperform the default Bluetooth scheme in terms of packet delay and video quality. A deadline aware buffer with discard of expired packets is introduced to improve efficiency. The FLC ARQ control also accounts for packet picture type importance.**

I. 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 reduce the total data transmitted, as there is approximately a linear relationship [3][4] between bitrate and energy consumption. A goodly number of works, for example [4][5][6][7][8], have investigated ways to manage power in a wireless network when streaming video. Though the Enhanced Data Rate (EDR) of Bluetooth version 2.0 [9] now has a peak user payload of 2.2 Mb/s (gross air rate 3.0 Mb/s), which is the same average rate offered by some implementations of IP-TV, it still must compete with lower power alternatives, such as Wibree from Nokia, intended for button-cell batteries, with a gross air rate of 1.0 Mb/s. 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 $\times$288) and QCIF (176 $\times$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 typically 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 Mb/s 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 multi-faceted control problem, as account must also be taken of wireless channel conditions, and 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. 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. In general, a fuzzy scheme is easily tuned by adjustment of its membership functions. A fuzzy scheme is also well-suited to 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 [13] 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 [14]. In practice, a play-out buffer exists on a mobile device to account for start-up delay and also absorbs delay jitter (variation of delay). Therefore, the maximum delay permissible corresponds to the start-up 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 II is a survey of related work, with a concentration on power-aware video streaming. Section III contributes background material on Bluetooth and explains the FLC in detail. The research methodology is also detailed. Section IV contains our simulated results, while Section V summarizes and draws some general conclusions.

II. RELATED WORK

In [15] 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, [16] 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 bitrate 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 [17], 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 pre-encoded 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 piece-wise linear model of playback power consumption. In [18], an energy constraint is introduced into a rate-distortion encoding model. In [19] 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 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 play-out 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 play-out buffers (at up to several seconds of start-up delay) may be applied in video-on-demand applications, Web-based video clip distribution with click-level interactivity is less tolerant of start-up delay. On a mobile device, memory contributes significantly to the power budget [20], resulting in relatively small buffers. For example, the experiments in [21] assumed a send buffer size of fifty packets, as also assumed in our experiments. In [21] also, selected packets are given priority transmission, rather than enforce rate changes at the encoder, which discriminates against pre-encoded video. However, layered encoding is assumed, while much content exists in non-layered format.

For single-layer video, the packet type is a simple way of applying either a delay- or a loss-priority packet transmission. Packet type indicates content importance without the need for content awareness at the link layer. In [22], 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 [23]. 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 non-scaleable 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.

III. METHODOLOGY

A. Bluetooth background

Bluetooth is a short-range (less than 10 m for class 2 devices), radio frequency interconnect. Bluetooth's short range, robust Frequency Hopping Spread Spectrum (FHSS) and centralized medium access control through time division multiple access and TDD means it is less prone to interference from other Bluetooth networks. Bluetooth employs variable-sized packets up to a maximum of five frequency-hopping time-slots of 625 μs 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 multi-path echoes, as error bursts occur. Another source of error bursts is co-channel 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 [25], 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 sub-channel (with 30 dB energy attenuation over the central frequency at ± 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 hot-spots.

For Bluetooth, an ARQ may occur in the following circumstances [26]: 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; e) header corruption of the return packet. Notice that a faulty ARQ packet can itself cause retransmission. The main cause of packet error [26], however, is c) payload corruption, which is the simplified assumption in this paper.

B. 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

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

where L is the bit length of a packet. Furthermore, the expected number of retransmissions, N, under the default ARQ scheme is:

$$E[N] = 0 \times P_s + 1 \times P_S \times (1 - P_S) + 2 \times P_S \times (1 - P_S)^2 + \ldots \qquad (2)$$

$$E[N] = \frac{1 - P_S}{P_s} \qquad (3)$$

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

$$E[T] = E[N] + 1 = \frac{1}{P_S}. \qquad (4)$$

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

$$E[N] = P_S \times \sum_{n=1}^{M-1} n \times (1 - P_S)^n + M \times (1 - [P_S \times \sum_{n=1}^{M-1} (1 - P_S)^n]) \qquad (5)$$

$$E[N] = \frac{(1 - P_S)(1 - (1 - P_S)^M)}{P_S} \qquad (6)$$

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

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

$$S = \frac{1}{(n+1) \times 625\,\mu s \times E[T]}, \qquad (7)$$

where n is the number of slots occupied by a Bluetooth packet. Assume that packets are fully-filled (refer to Section III.F) 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 (7) based on the packet bit length and in Fig. 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 Fig. 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 $P_e = 10^{-4}$.

C. Fuzzy logic control of ARQ

Fig. 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 [27]. 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.

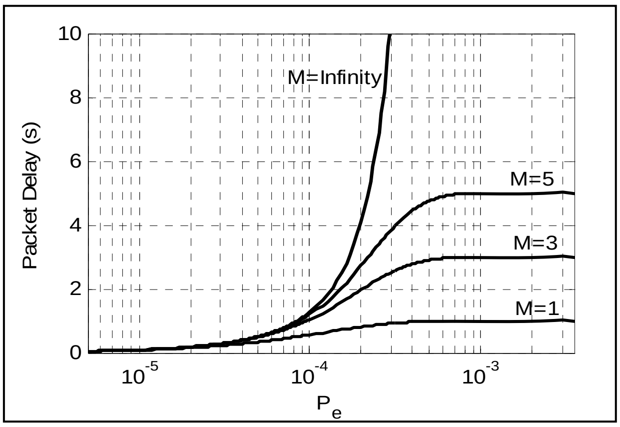

Fig. 1. Packet delay against P_e (logarthmic horizontal scale).

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 to also 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 pre-determined 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 arrived at heuristically, are shown In Fig. 3 a) and b), and once found remain fixed. The buffer fullness range in Fig. 3 a) is [0,1] corresponding to a percentage fullness. In Fig. 3 b), the horizontal axis represents the delay time of the packet at the head of the queue divided by the display deadline. In Fig. 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 non-scaled transmission count are in Table 2.

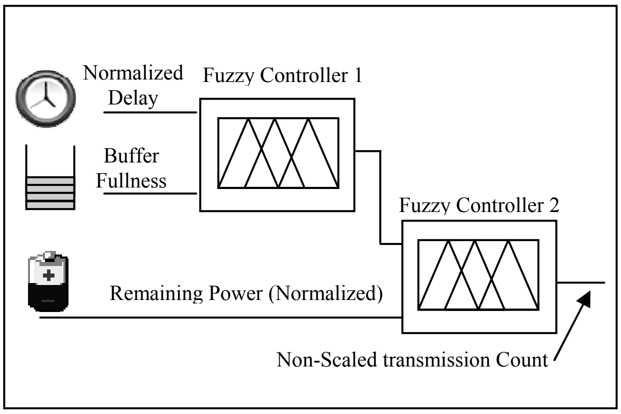

Fig. 2. Overview of the FLC of ARQ system.

The inputs were combined according to the well-known Mamdani model [28] 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 Fig. 3 c) and 3 e). The fuzzy control surfaces are represented in Fig. 4, as derived from the Matlab Fuzzy Toolbox v. 2.2.4. As mentioned in Section I, 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 Fig. 5.

Table 1. FLC stage 1 `If..then` rules used to identify output fuzzy subsets from inputs.

		Delay/Deadline				
Buffer Fullness		*TooLow*	*Low*	*Normal*	*High*	*TooHigh*
	High	Normal	Normal	Low	TooLow	TooLow
	Normal	TooHigh	High	Normal	Low	TooLow
	Low	TooHigh	TooHigh	High	Low	TooLow

Table 2 FLC stage 2 `If..then` rules used to identify output fuzzy subsets from inputs.

Remaining Power	Output1				
	TooLow	*Low*	*Normal*	*High*	*TooHigh*
High	TooLow	Low	High	TooHigh	TooHigh
Normal	TooLow	Low	Normal	High	High
Low	TooLow	TooLow	Low	Low	Normal

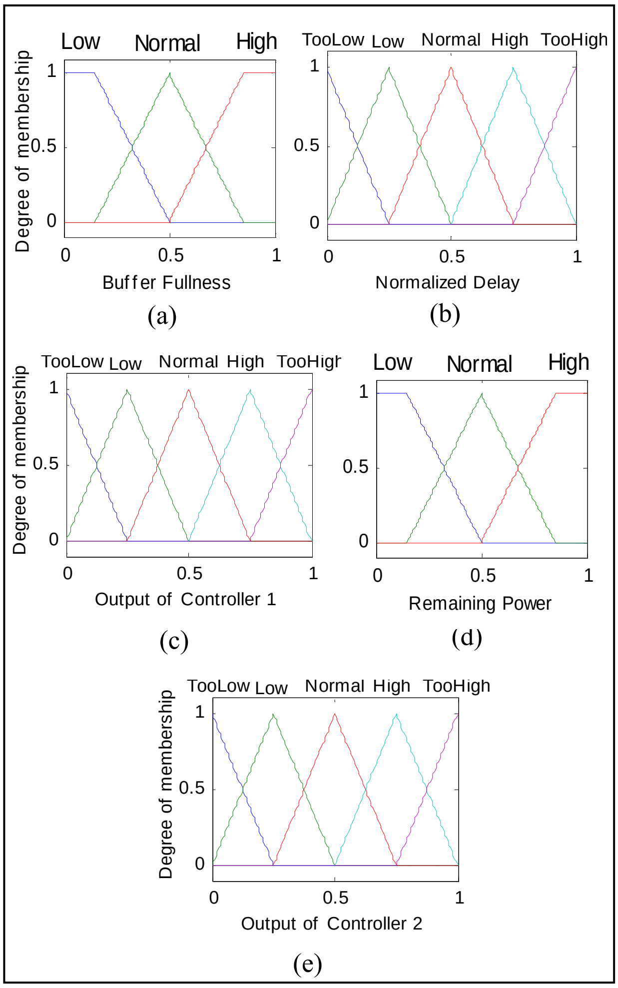

Fig. 3. Fuzzy membership functions a) Stage 1 input buffer fullness b) Stage 1 input delay deadline c) Output of stage 1 controller d) Stage 2 input remaining power e) Stage 2 output transmission count.

A simple scaling of 5:3:2 was applied respectively for I-, P-, B-pictures, given up to five maximum transmissions. 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. It should be mentioned that a maximum value of five retransmissions was also adopted in the priority queuing tests in [21], albeit for an IEEE 802.11 wireless network.

D. 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 simulations of Section IV, the display deadline was set to 0.10 s.

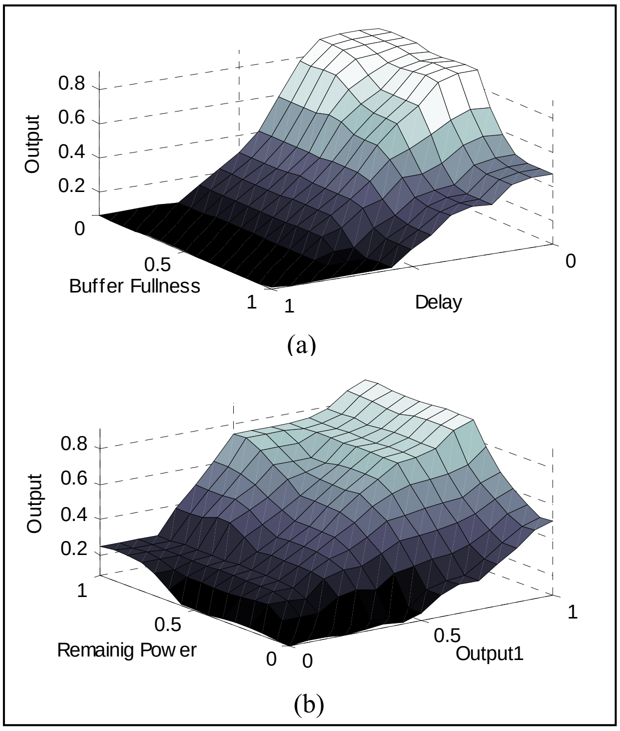

Fig. 4. a) Stage 1 FLC control surface resulting from FLC ARQ b) Stage two control surface giving the transmission count output (before subsequent scaling).

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, i.e. 0.44 s 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 is assumed (optimistically) to be constant. In all experiments, the buffer queue discipline is assumed to be First-In-First-Out.

E. Channel model

A Gilbert-Elliott [29][30] two state discrete-time, ergodic Markov chain models 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 [31], applied to the same version of Bluetooth as herein.

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

$$T_g = \frac{1}{1-Pgg}, T_b = \frac{1}{1-Pbb} \tag{8}$$

that, given the current state is good (*g*), *Pgg*, the probability that the next state is also *g* , is 0.9996875 and *Pbb*, and given the current state is bad (*b*), the probability that the next state is also b, is 0.9975. The transition probabilities, *Pgg* and *Pbb*, as well as the Bit Error Rate (BER) are approximately similar to those in [36], but the mean state durations are adapted to Bluetooth. At 3.0 Mb/s, the BER during a good state was set to $a \times 10^{-5}$ and during a bad state to $a \times 10^{-4}$, where *a* is a scaling factor and is subsequently referred to as the channel parameter.

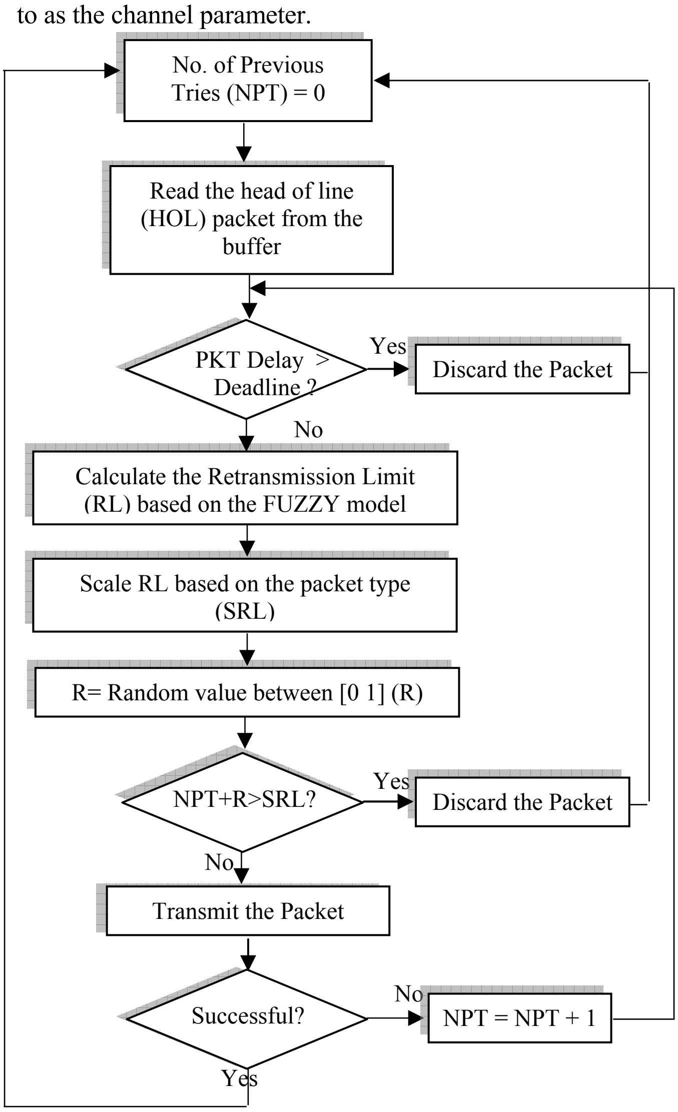

Fig. 5. FLC algorithm for processing a packet.

F. Simulation setup

This research employed the University of Cincinatti Bluetooth (UCBT) extension[1] 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 otcl script. All links were set at the maximum EDR 3.0 Mbps gross air rate. Simulation runs were each repeated twenty times and the results 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 s video clip with moderate motion, showing a newsreader and changing backdrop, which we designate 'News'. (Other video inputs are summarized in Section IV.) PSNR was found by reconstructing with a reference MPEG-2 decoder. The display rate was 25 frame/s, resulting in 750 frames in each run. The source video was Common Intermediate Format (CIF)-sized (366×288 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 Fig. 6. In [32], 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 [32] it is shown that the overall video performance is superior to choice of smaller packet sizes.

IV. RESULTS

A. Fuzzy logic model response

Fig. 7 shows the output of stage 1 of the FLC as the 'News' video clip of Section III.E 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 III.D).

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 by Fig. 8, again with channel parameter set to two.

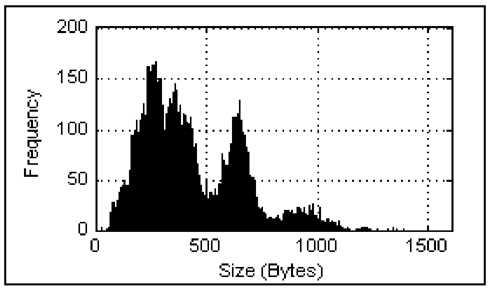

Fig. 6. Distribution of slice sizes for the encoded video clip.

[1] (download is available from `http://www.ececs.uc.edu/~cdmc/UCBT`)

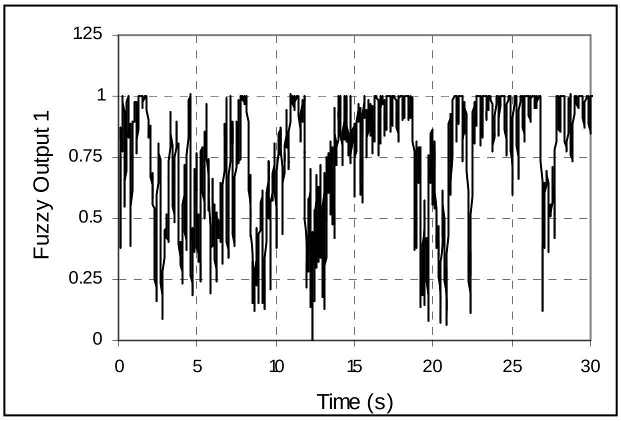

Fig. 7. Output from stage one of the FLC, with $a = 2$.

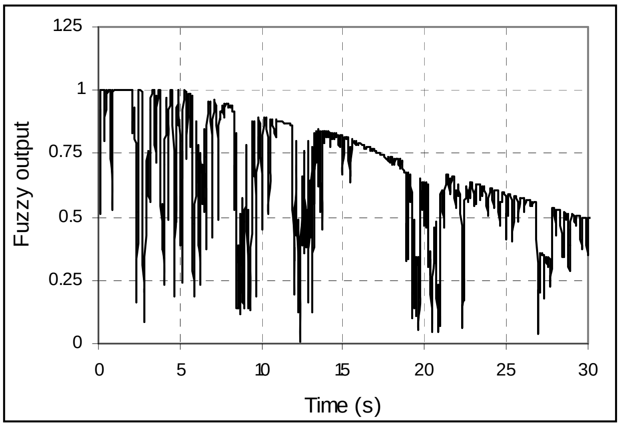

Fig. 8. Output from stage two of the FLC, with $a = 2$.

After the removal of deadline expired packets, through operation of the deadline aware buffer (DAB) described in Section III.D, the buffer fullness input to stage one of the send buffer oscillates around a level well-below the 50 packet maximum, Fig. 9. Head-of-line packet delay, Fig. 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, Fig. 11, delay is frequently over the 0.10 s display deadline and, therefore, B-picture packets face the possibility of being dropped without transmission if they have already spent longer that that time in the send buffer, while I- and P-picture packets have the grace arising from their extra decode deadline time.

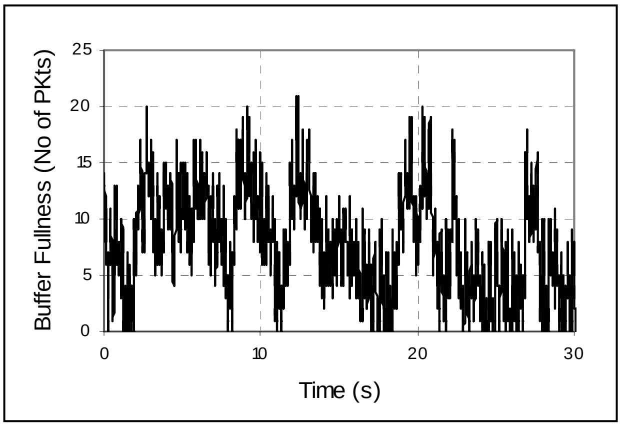

Fig. 9. Buffer fullness input to stage one of the FLC, with $a = 2$.

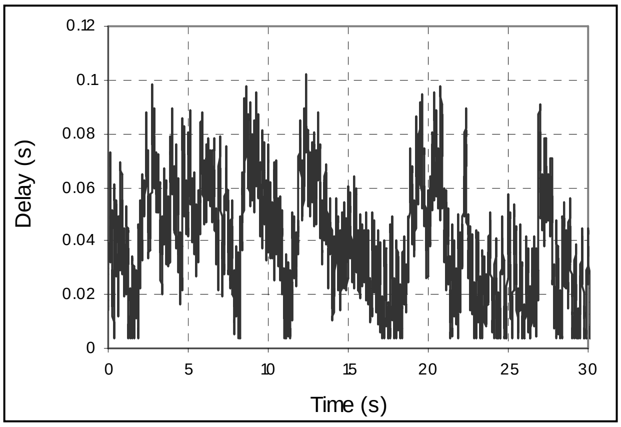

Fig. 10. Delay input to stage one of the FLC, with $a = 2$.

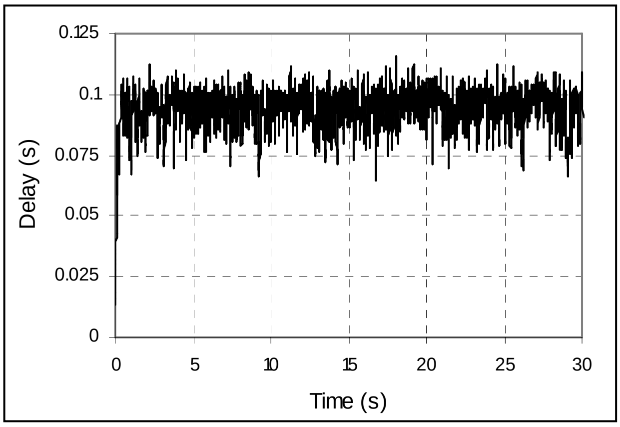

Fig. 11. Delay in default ARQ with DAB, with $a = 2$.

B. Response of FLC and default ARQ

A comparison was made between the default scheme with infinite ARQ and no power control with the FLC scheme with power control. That is the default scheme was allocated an infinite power budget. For fairness both schemes were compared with a DAB in place, though, 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. As an aid to comparison, the FLC scheme was also tested with an infinite power budget. Unfortunately, in respect to Bluetooth, we are not aware of other power-aware schemes which use adaptive ARQ and would form a direct point of comparison. Fig. 12 compares the ratio of packets lost to total packets arriving in the send buffer. Even though the FLC scheme is compensating for a diminishing power budget, it shows a clear superiority. The effect is more pronounced with worsening channel conditions. The average delay of successfully transmitted packets was also considerably reduced under the FLC scheme, Fig. 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 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).

Crucially, the FLC is able to save power over the non-power-aware default ARQ scheme, Fig. 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, Fig. 16, as less B-picture packets and more reference picture packets are lost under default ARQ, Fig. 15.

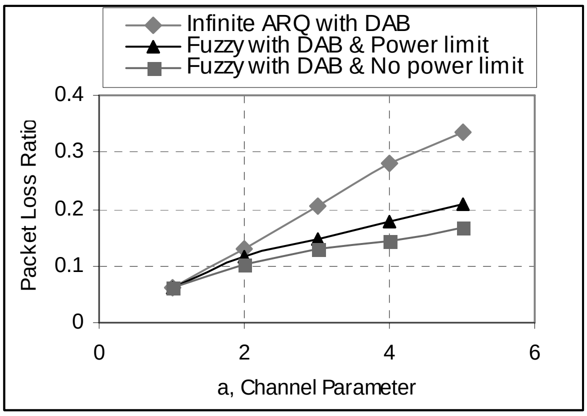

Fig. 12. Packet loss during transmission of the News video clip, with the default scheme and the FLC power-aware scheme.

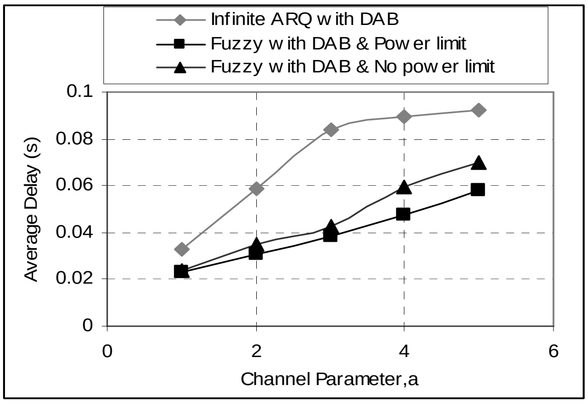

Fig. 13. Average packet delay during transmission of the News video clip, with the default scheme and the FLC power-aware scheme.

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. 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.

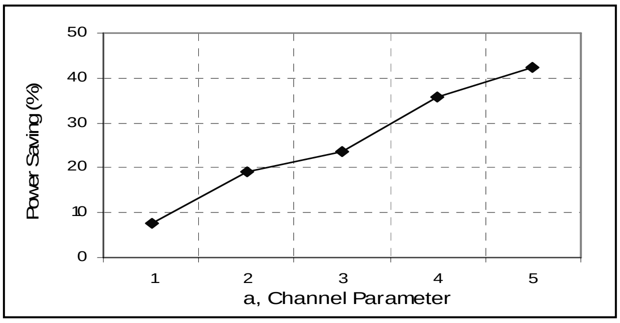

Fig. 14. Relative power saving of the FLC power-aware ARQ scheme compared to that of the default ARQ scheme.

Considering the packet loss statistics of Fig. 12 and the distribution of those packet losses between packet picture types from Figs. 15 and 16, it is not surprising, Fig. 17, that the mean PSNR of FLC ARQ is better than that of the default scheme 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 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 [33] that for the very worst channel conditions shown in Fig. 17, i.e. $a = 5$, 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, Fig. 14, in the same conditions.

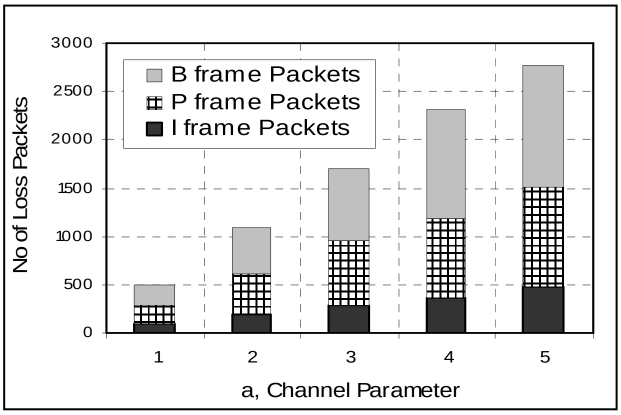

Fig. 15. Contribution of I-, P- and B- frame packet losses to total packet loss in the infinite ARQ with DAB scheme.

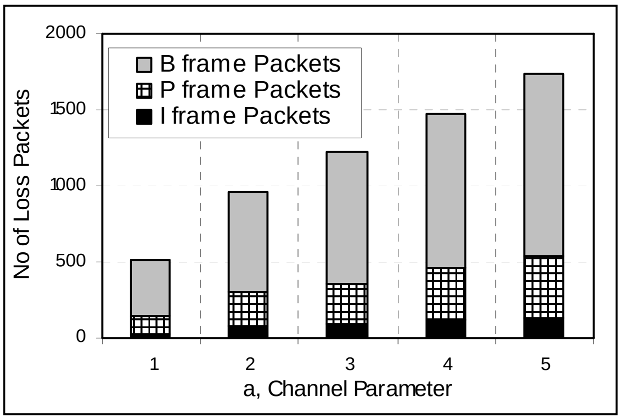

Fig. 16. Contribution of I-, P- and B- frame packet losses to total packet loss in the FLC with DAB and power-aware scheme.

Table 3. Distribution of packets by picture type in the test video.

Picture type	*Percentage packets*
I	17.97
P	37.93
B	44.10

C. Detailed comparison of video quality

For a comparison of video quality over time, for the purposes of this test only, the power budget was artificially set to 60×10^6 bits for both the Bluetooth default ARQ scheme and power-aware FLC ARQ. Again, a DAB was in place for both tests. Fig. 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, Fig. 19, not only is the video quality lower but there is an abrupt end to transmission as the bit budget runs out at about 652 frames.

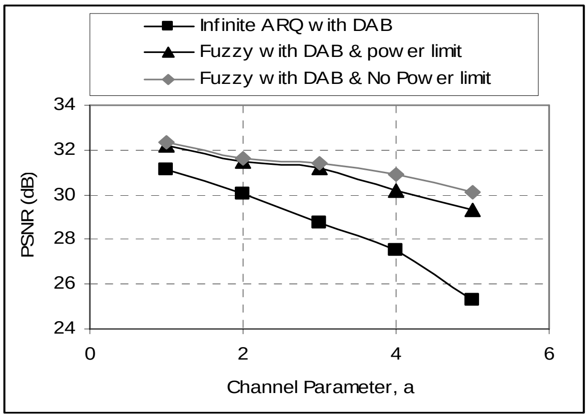

Fig. 17. Comparison of PSNR for the News video clip between the Bluetooth default and FLC ARQ schemes with DAB.

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.

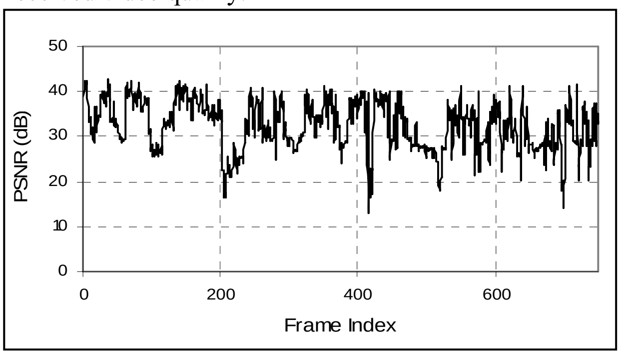

Fig. 18. PSNR over the 750 frames, showing a falling trend in PSNR over time, for the power-aware FLC ARQ with DAB

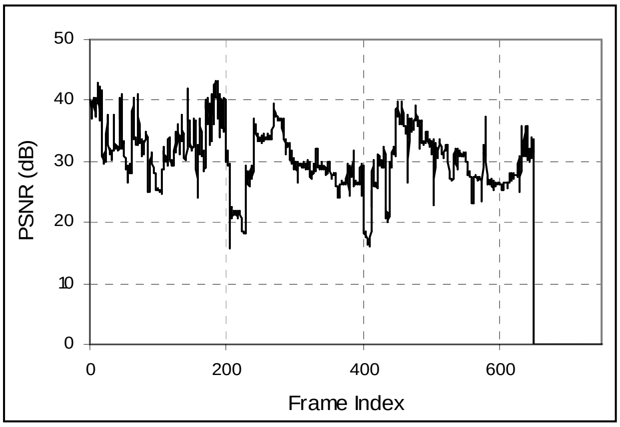

Fig. 19. PSNR over the 750 frames of the input video, under a fixed power budget with the default ARQ with DAB.

V. 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 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 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. However, FLC, which varies its transmission policy with packet picture type, still outperforms the default fully-reliable ARQ scheme, resulting, in the end analysis, in superior delivered video quality and reduced delay. 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 Comms.*, vol. 7, no. 1, pp. 28-26, 2000..

[2] S. Mattisson, "Low-power considerations in the design of Bluetooth", *Int. Symposium on Low-power Electronics and Design*, pp. 151-154, 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", *14th Int. Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, 2004.

[4] P. Agrawal, J.-C. Chen, S. Kishore, P. Ramanathan, and K. Sivalingam, "Battery Power Sensitive Video Processing in Wireless Networks", *IEEE Int. Symposium on Personal, Indoor, and Mobile Radio Comms.*, vol. 1, pp. 116-120, Sept. 1998.

[5] K. Choi, K. Kim, and M. Pedram, "Energy-aware MPEG-4 FGS streaming", *40th Design Automation Conference*, pp. 912-915, 2003.

[6] A. Iranli, K. Choi, and M. Pedram, "Energy-aware wireless video streaming", *Embedded Systems For Real-time Multimedia (ESTimedia)*, pp. 48-55, 2005.

[7] H-S. Kim, D. T. Duong, J.-J. Jeong, B.-K. Dan, and S.-J. Ko, "Power-aware rate control for mobile multimedia communications", *ADHOC-NOW 2006*, LNCS #4104, pp. 458-471, 2006.

[8] R. Cornea, A. Nicolau, and N. Dutt, "Content-aware power optimizations for multimedia streaming over wireless networks", University of California, Technical report CECS #06-13, 2006.

[9] Specification of the Bluetooth System -- 2.0 + EDR, 2004, Available online at http://www.bluetooth.com.

[10] E. Ferro and F. Potorì, "Bluetooth and Wi-Fi wireless protocols: A survey and a comparison", *IEEE Wireless Communications*, vol.12, no. 1, pp. 12-26, Feb. 2005.

[11] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "End-to-End QoS for Video Delivery Over Wireless Internet", *Proc. of the IEEE*, vol. 93, no. 1, pp. 123-134, 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 Trans. On Consumer Electronics*, vol. 50, no. 1, pp. 158-167, 2004.

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

[14] M. Kalman, P. Ramanathan, and B. Girod, "Rate-Distortion Optimized Video Streaming with Multiple Deadlines", *Int. Conf. on Image Processing*, vol. 3, pp. 662-664, Sept. 2003.

[15] 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", *IEEE INFOCOM*, vol. 1, pp. 150-157, March 1998.

[16] J.-C. Cano, J.-M. Cano, E. Gonzálex, C. Calafate, and P. Manzoni, "Power characterization of a Bluetooth-based wireless node for ubiquitous computing", *Int. Conf. on Wireless and Mobile Comms.*, pp. 13-18, July 2006.

[17] Q. Zhang, Z. Ji, W. Zhu, and Y. Q. Zhang, "Power minimized bit allocation for video communications over wireless channels", *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 6, no. 2, pp. 394-410, 2002.

[18] Y. Liang and I.. Ahmad, "Power and content aware video encoding for video communication over wireless networks", *Signal Processing Systems (SPIS 2004)*, pp. 269-274, Oct. 2004.

[19] M. Tamai, T. Sun, K. Yasumoto, N. Shibata, and M. Ito, "Energy-aware QoS adaptation for streaming video based on MPEG-7", *IEEE Int. Conf. on Multimedia and Expo.*, vol. 1, pp. 189-192, June 2004.

[20] M. Yokotsuka, "Memory motivates cell-phone growth", *Wireless Systems Design*, vol. 9, no. 3, pp. 27-30, April, 2004.

[21] Q. Li, M. van der Schaar, "Providing 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.

[22] W. Tan and A. Zakhor, "Packet Classification Schemes for Streaming MPEG Video over Delay and Loss Differentiated Networks", *PacketVideo Workshop*, May, 2001.

[23] J. Wall and J. Y. Khan, "An Adaptive ARQ Enhancement to Support Multimedia Traffic using 802.11 Wireless LANs", *IEEE GLOBECOM*, pp. 3037-3041, Nov. 2004.

[24] Q. Li, M. van der Schaar, "Providing QoS to Layered Video Over Wireless Local Area Networks Through Real-Time Retry Limit Adaptation", IEEE Trans. on Multimedia, vol. 6, no. 2, pp. 278-290, 2004.

[25] N. Golmie, N. Chevrolier, and O. Rebala, "Bluetooth and WLAN Coexistence: Challenges and Solutions", *IEEE Wireless Communications*, vol. 6, pp. 22-29, 2003.

[26] M. C. Valenti, M. Robert, and J. H. Reed, "On the Throughput of Bluetooth Data Transmissions", *IEEE Wireless Commun. and Networking Conf.*, pp. 119-123. Mar. 2002.

[27] R. Razavi, M. Fleury, and M. Ghanbari, "Detecting Congestion within a Bluetooth Piconet: Video Streaming Response", *London Comms. Symposium*, pp. 181-184, Sept. 2006.

[28] J. S. Jang, C. T. Sun, and E. Mizutani, *Neuro-fuzzy and Soft Computing*. Prentice Hall, Upper Saddle River, NJ, 1997.

[29] E. N. Gilbert, "Capacity of Burst-Noise Channel", *Bell System Technical J.*, vol. 39, pp. 1253-1265, 1960.

[30] E. O. Elliott, "Estimates of Error Rates for Codes on Burst Noise Channels", *Bell System Technical J.*, vol. 42, pp. 1977-1997, 1963.

[31] L.-J. Chen, T. Sun, and Y.-C. Chen, "Improving Bluetooth EDR Data Throughput Using FEC and Interleaving", *Mobile Sensor Networks*, Hong Kong, 2006.

[32] 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.

[33] R. Razavi, M. Fleury, and M. Ghanbari, "Fuzzy control of adaptive timeout for video streaming over a Bluetooth interconnect", 2nd mediaWin Workshop at IEEE 12th Int. Symposium on Computers and Communications, 2007.

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