Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 45798, 13 pages
doi:10.1155/2007/45798

Research Article

Fuzzy Logic Control of Adaptive ARQ for Video Distribution over a Bluetooth Wireless Link

R. Razavi, M. Fleury, and M. Ghanbari

Department of Electronic Systems Engineering, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK

Received 27 April 2007; Accepted 23 July 2007

Recommended by Tasos Dagiuklas

Bluetooth's default automatic repeat request (ARQ) scheme is not suited to video distribution resulting in missed display and decoded deadlines. Adaptive ARQ with active discard of expired packets from the send buffer is an alternative approach. However, even with the addition of cross-layer adaptation to picture-type packet importance, ARQ is not ideal in conditions of a deteriorating RF channel. The paper presents fuzzy logic control of ARQ, based on send buffer fullness and the head-of-line packet's deadline. The advantage of the fuzzy logic approach, which also scales its output according to picture type importance, is that the impact of delay can be directly introduced to the model, causing retransmissions to be reduced compared to all other schemes. The scheme considers both the delay constraints of the video stream and at the same time avoids send buffer overflow. Tests explore a variety of Bluetooth send buffer sizes and channel conditions. For adverse channel conditions and buffer size, the tests show an improvement of at least 4 dB in video quality compared to nonfuzzy schemes. The scheme can be applied to any codec with I-, P-, and (possibly) B-slices by inspection of packet headers without the need for encoder intervention.

1. INTRODUCTION

The enhanced data rate (EDR) of IEEE 802.15.1, Bluetooth [1] version 2.0 [2] now has a peak user payload of 2.2 Mb/s, which is the same average rate offered by some implementations of IP-TV. Therefore, a bottleneck free way exists of distribution encoded video clips from a server across an IP network to a Bluetooth master node, and, thence, over a Bluetooth wireless interconnect. Moreover, many cellular phones are also equipped with a Bluetooth transceiver and larger resolution screens of CIF (352×288) and QCIF (176×144) pixel size. Compared to IEEE 802.11 (Wi-Fi)'s typical current usage of 100–350 mA [3], Bluetooth's consumption is 1–35 mA, implying that for mobile multimedia applications Bluetooth is preferable. Nokia's proprietary Wibree technology, with a similar design to Bluetooth, uses lower-power button-cell batteries but its throughput is apparently restricted to a gross air rate of 1 Mbps. IEEE 802.15.4 (ZigBee) also has similarities to Bluetooth but as it is intended for sensor applications, its capacity is limited to 250 kbps. Bluetooth availability as a low-cost transceiver (<$5 US) makes it an attractive proposition for bespoke mobile video streaming applications, such as cordless TV within a variety of vehicles [4] or augmented reality for wearable computers [5].

However for video transmission, 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. Unfortunately, the default Bluetooth infinite retransmission limit for stop-and-wait automatic repeat request (ARQ) is unsuitable for delay-sensitive video streaming. This is a significant weakness, because, in general, ARQ has proved more effective than forward error correction (FEC) [6] in ensuring statistically guaranteed quality of service (QoS) over wireless networks. 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.

Real-time delivery of video is delay-sensitive, as a frame cannot be displayed if its data arrive after their decoded deadline. A further deadline exists for reference picture types if their presence contributes to decoding of future frames [7]. 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 extreme cases arriving packets may find the send buffer full. ARQ adds to delay and, therefore, the number of retransmissions should be minimized. A side effect of reducing the number of retransmissions is that power usage is reduced, which is especially important when there is an imbalance of activity in Bluetooth's centralized packet scheduling scheme between a master node and its slaves.

As an alternative to the default ARQ, research reported in [8] appears to have first introduced to Bluetooth priority-based retransmission for video picture packets, though that paper went no further than a static scheme favouring Intra-coded pictures (I-pictures) at the link layer, based on similar application layer techniques to those in [9]. The need to adjust adaptively Bluetooth's default retransmission timeout for multimedia applications was established in [10], with adaptation by relatively conventional means. In our work, ARQ adaptation allows picture importance, channel conditions, and buffer fullness to be accounted for in retransmission decisions. However, 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 queuing delay to other packets and possibly buffer overflow. Therefore, an active discard policy for deadline-expired packets is required as an addition to adaptive ARQ. In our case, 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 or real-time transfer protocol (RTP) headers, whereas accounting for picture content rather than picture importance may require intervention at the source encoder.

This paper introduces fuzzy logic control (FLC) to adaptive ARQ over Bluetooth. To the best of our knowledge, FLC has not been used for this purpose for Bluetooth and, in general, has not been applied in this way to video distribution over a wireless network. The aim of the current work is to retransmit a packet as many times as needed to ensure error-free reception but without delaying that packet beyond its deadline and without leading to send buffer overflow. The main reason for introducing FLC is that we found that its performance in terms of delivered video quality is simply better than a conventional scheme, as the tests in Section 4 illustrate. The fuzzy scheme also reduces the average number of retransmissions, its key advantage being that it can directly adapt to delay conditions, rather than simply by indirect means through active discard of expired packets. In general, a fuzzy scheme is more easily tuned by adjustment of its membership functions. By introducing two control inputs, a fuzzy scheme can trim its response. The two inputs in our scheme were buffer fullness *and* the deadline margin of the packet at the head of the Bluetooth send queue (the direct delay input). 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.

The remainder of this paper is organized as follows. Section 2 surveys related work on implementing video quality of service (QoS) through ARQ and FLC for wireless networks. Section 3 gives details of Bluetooth ARQ, FLC ARQ, and the evaluation methodology. Section 4 contains the results of the evaluation, while Section 5 draws some conclusions.

2. RELATED WORK

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 to start-up delay. On a mobile device, memory contributes significantly to the power budget [11], resulting in relatively small buffers. For example, the experiments in [12] assumed a send buffer size of fifty packets. The complexity of an adaptive rather than fixed-size play-out buffer [13], which can subsequently vary its size according to network conditions, may also deter mobile device implementation. Video smoothing was transferred to wireless networks in [14], given that the available bandwidth is even more subject to fluctuations than a fixed network. In [14] 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 nonlayered 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 [9], simple packet type discrimination is proposed as a means of implementing differentiated services QoS on the fixed Internet.

As IEEE 802.11 has no built-in QoS mechanism, there has been interest in closed-loop error control through ARQ. Though the 802.11 point coordination function (PCF) access protocol is centralized to limit delay, weaknesses in its specification [15, 16] have meant little attention has been paid to it, unlike Bluetooth's centralized control. An exception is the work in [17], where centralized control is considered most appropriate to multimedia applications and there is brief consideration (among other QoS techniques) to MAC-level adaptation of the IEEE 802.11a retry limit. Despite the title of the paper, it is not the retry limit but the ARQ mode in IEEE 802.11e's distributed coordination function (DCF) that is adapted according to channel conditions in [15]. The available modes are No ACK, stop-and-wait ARQ (as in Bluetooth), and block ARQ, whereby a number of successfully received packets are collectively acknowledged. The IEEE 802.11e variant of IEEE 802.11 also includes a buffer discard threshold, though this is not adaptive. The hybrid coordination function in IEEE 802.11e is another way of reintroducing centralized control whereby during contention-free periods, the access station assigns mobile stations transmit times. In IEEE 802.11 in general, larger frames are sent by a request-to-send (RTS)/clear-to-send (CTS) handshake [16], which reduces the impact of contention. With or without RTS/CTS, it is possible to alter the maximum number of retransmissions and in IEEE 802.11e, it is also possible to set a maximum limit to the time spent in the transmitter buffer [15]. In [12] for IEEE 802.11b, 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 [12], whereby packets dependent on lost packets are removed from queues. The problem with purging, as opposed to deadline-aware active discard, is that it appears only actionable when I-picture packets have been lost. The scheme 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 assumed in our work.

Mention should also be made of hybrid ARQ [18] in which the ARQ also contains a notification of uncorrectable errors. Further redundancy is then added to a packet before retransmission. A related technique [19], which was also made deadline aware, employs selective-repeat ARQ to control the bit-rate at the encoder. Both hybrid ARQ and ARQ bit-rate control appear not to be suitable for pre-encoded video and for the latter, the close proximity of the encoder also is needed to avoid delay in bit-rate adjustments. In [19], the error propagation impact of packets is found at the encoder and a retry limit with active discard is set for IEEE 802.11 DCF retries. This would appear to not be a general solution, as it requires collusion between the encoder and the link layer transmitter. Turning to Bluetooth, apart from [8, 10] mentioned in Section 1, adaptive ARQ seems to have been little explored except in previous research by us [20], which introduced a conventional adaptive ARQ scheme according to wireless channel conditions. Channel conditions are inferred from send buffer fullness and priority is given according to B- and P- and I-picture types and not just to I-picture packets. In Section 4, the adaptive ARQ scheme in [20] is compared with that based on FLC.

In [21], FLC was applied to Bluetooth packet scheduling on a piconet, in which multiple Bluetooth slaves are present. The work in [22] applied FLC to Bluetooth rate control through tandem controllers in an open loop system. Outside Bluetooth, FLC has found applications [23] in wireless networks with access control of a time division multiple access system. In [24], FLC is used in a random early drop (RED) router again as a form of access control. In TCP, any packet losses cause the TCP source to reduce its sending rate. Of course, deliberate and random packet losses are unsuitable for encoded video and the unbounded delay introduced by TCP's reliability mechanism also makes it unsuitable for video display. Nonetheless, in [25], an interesting application of FLC to tandem network (wired and wireless links) controlled the retransmission rate and the RED rate.

3. METHODOLOGY

Bluetooth employs variable-sized packets up to a maximum of five frequency-hopping time-slots of 625 microseconds 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 the case of 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 causing 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 [26], this is only effective if interference is not across all or most of the 2.402 to 2.480 GHz unlicensed band. IEEE 802.11b operating in direct sequence spread spectrum mode may occupy a 22 MHz subchannel (with 30 dB energy attenuation over the central frequency at ±11 MHz) within the 2.4 GHz band. IEEE 802.11g employs orthogonal frequency division multiplexing to reduce intersymbol interference but generates similar interference to 802.11b. 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.

3.1. Bluetooth ARQ

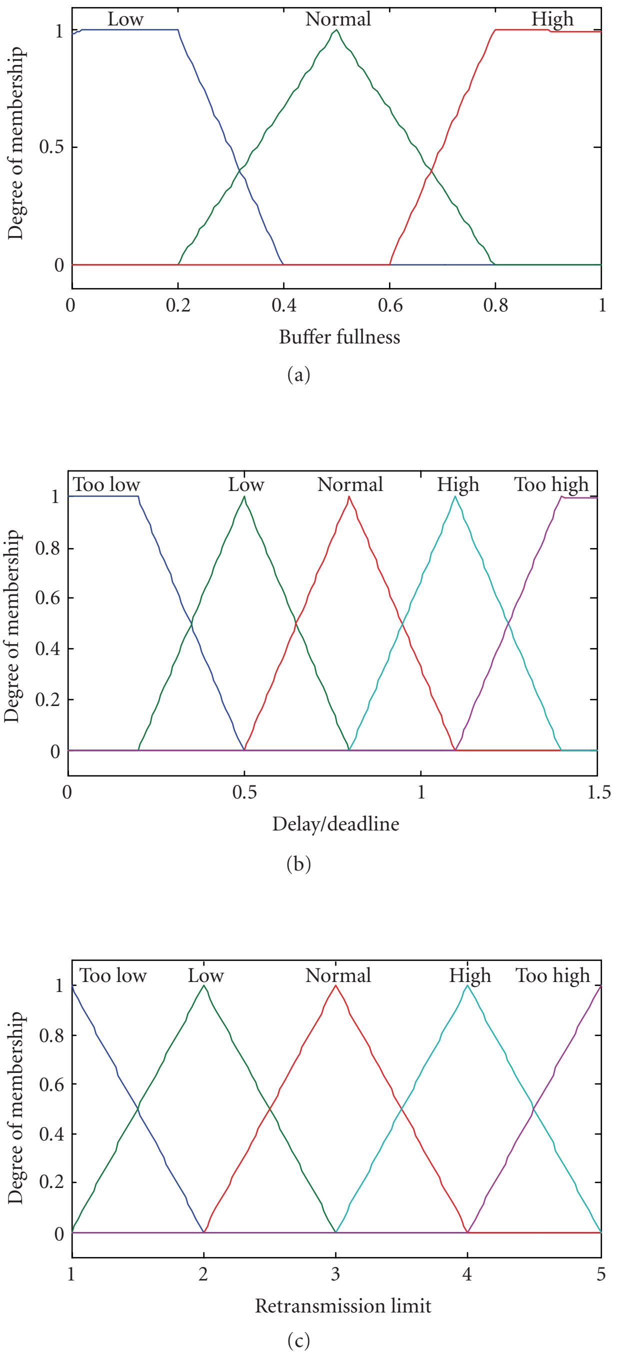
For Bluetooth, an ARQ may occur in the following circumstances [27]: (a) failure to synchronize on the access header code; (b) header corruption detected by a triple redundancy code; (c) payload corruption detected by CRC; (d) failure to synchronize with the return packet header; (e) header corruption of the return packet. Notice that a faulty ARQ packet can itself cause retransmission. The main cause of packet error [27], however, is (c) payload corruption. As mentioned in Section 1, the default value of the ARQ retransmission timeout in most Bluetooth chipsets [10] is set to infinity, resulting in unlimited retries. In [10], a fixed retransmission timeout and an adaptive retransmission timeout were considered. The disadvantage of a fixed retransmission timeout is that it is difficult to arrive at a value that avoids either excessive delay or excessive packet drops in *all* circumstances. The adaptive retransmission timeout, which was upper and lower bounded, was based, in [10], on a smoothed round-trip time. The retransmission timeout was adapted downwards or upwards if the new smoothed round trip time, respectively, is less than or more than the previous smoothed round trip time.

3.2. *Fuzzy logic control of ARQ*

This section briefly introduces FLC before introducing FLC ARQ. 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. In a fuzzy subset of (say) "buffer fullness," the possibility that a buffer with a given fullness taken from the set S of fullness may be called high 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 in fuzzy "if then" rules to make inferences such as if x is high and y is low, then z is normal, in which high, low, and normal 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, which results in a crisp (nonfuzzy) value.

For the adaptive ARQ FLC, 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 [28]. 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. Retransmissions avoid the effect of noise and interference but also cause the master's send buffer queue to grow, with the possibility of packet loss from send buffer overflow.

The retransmission timeout of the packet at the head of the Bluetooth send queue will affect the delay of packets still to be transmitted. Therefore, the second FLC input moderates the buffer fullness input. The assigned membership functions, which were arrived at heuristically, are shown In Figures 1(a) and 1(b), and once found were fixed. The buffer fullness range in Figure 1(a) is $[0, 1]$ corresponding to a percentage fullness. In Figure 1(b), the horizontal axis represents the delay time of the packet at the head of the queue divided by the display deadline. In Figure 1(b), unit delay/deadline 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 (Section 3.3). However, this takes place after the fuzzy evaluation of the desired ARQ retransmission timeout. In practice, the inputs to the FLC were sampled versions of buffer fullness and packet delay/deadline to avoid excessive ARQ retransmission timeout oscillations over time. The sampling interval was every 20 packets. Table 1 shows the "if $\cdots$ then" rules that allow input fuzzy subsets to be combined to form an output. Notice that more than one rule may apply because of the fuzzy nature of subset membership.



(a)

(b)

(c)

Figure 1: Fuzzy membership functions: (a) input buffer fullness (b) input delay/deadline (c) output retransmission limit.

The inputs were combined according to the well-known Mamdani model [29] to produce a single output value. The standard center of gravity method was employed to resolve to a crisp output value according to the output membership functions shown in Figure 1(c). Notice that the output in Figure 1(c) corresponds to the full range of possibilities, whereas if a deadline-aware-buffer (Section 3.3) is incorporated, then discard of expired packets will mean that the higher end of the output range will not occur. In Figure 1(c),

Table 1: FLC If · · · then rules used to identify output fuzzy subsets from inputs.

		Delay/Deadline				
		Too low	Low	Normal	High	Too high
Buffer Fullness	High	Normal	Normal	Low	Too low	Too low
	Normal	High	High	Normal	Low	Too low
	Low	Too high	High	Normal	Low	Too low

the retransmission limits correspond to the retransmission timeout of the current packet. Clearly, a packet can only be retransmitted an integer number of times but the 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 more than the fractional part of the crisp output value, then that value was rounded down to the nearest integer, otherwise it was rounded up. The advantage of this procedure over simple quantization is that, in the long term, the resolution of the number of transmissions will be higher and the mean value will converge to a desired output level. The output value was subsequently scaled according to the priority of the packet's picture type.

For reasons of error resilience, encoded video is transmitted as a repeating sequence of GOP [30], with the start of each GOP formed by an I-picture. An I-picture is the basis for prediction of all other pictures in the GOP (usually 12 to 15 pictures in all) and, hence, its loss has drastic consequences for all other pictures. P-pictures also form the basis for predictions but are not essential for the reconstruction of other pictures within the GOP (as other I- or P- anchor pictures retained in a decoded buffer can be applied). Lastly, the third type of picture, the bipredictive B-picture, has no predictive value.

A simple scaling of 5 : 3 : 1 was applied, respectively, for I-, P-, B-pictures, given a choice of five maximum retransmissions. Normalizing this scaling to a factor of one for I-picture packets, results in a ratio of 1 : 0.6 : 0.2, giving for maximum retransmissions five, just $1 \times 5 = 5$ retransmissions for I-picture packets but $0.6 \times 5 = 3$ maximum retransmissions for a P-picture packet, and $0.2 \times 5 = 1$ maximum retransmissions for a B-picture packets. In practice, the scaling is applied to the crisp value output after defuzzification. For example, if the crisp output value was 2.3, and a P-picture packet was involved, then the value after scaling is $2.3 \times 0.6 = 1.38$. Then, the random-number-based resolution results in two retransmissions if the random number is less than or equal to 0.38 and one retransmission otherwise. It should be mentioned that a maximum value of five retransmissions was also adopted in the priority queueing tests in [12], albeit for an IEEE 802.11 wireless network.

The fuzzy control surface is represented in Figure 2, as derived from the Matlab fuzzy toolbox v. 2.2.4. A scaled version of this output surface is applied with scaling dependent on picture type. As mentioned in Section 1, by means of an LUT derived from the surface, a simple implementation becomes possible.

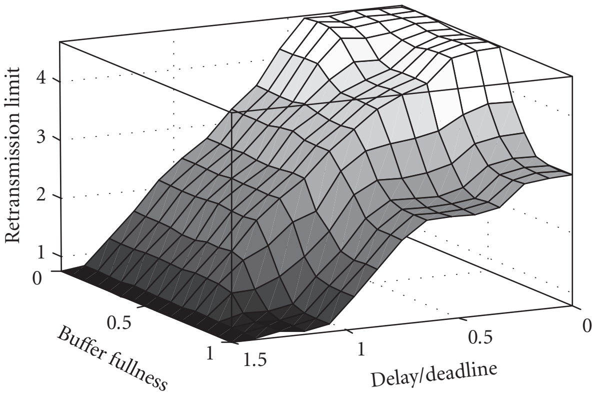


Figure 2: Control surface resulting for FLC ARQ.

3.3. 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 play-out buffer expressed as a time beyond which buffer underflow will occur. In a conservative policy, in which there is no need for play-out buffer fullness updates, the deadline is set as the maximum time that the play-out buffer can delay the need for a packet. Play-out buffers are normally present to smooth out jitter across a network path (if the Bluetooth master was also an access point) and in this paper the size is assumed to be constant. In the 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 decoded 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 frame GOP, this is the time to display 11 frames, that is, 0.44 second at 25 frame/s. For P-picture packets, the time will vary depending on the number of frames to the end of the GOP. For B-pictures the decoded deadline is set to zero.

The decoded 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

$\text{IBBP}_1\text{BBP}_2\text{BBIBBP}_3\cdots$ $\text{IP}_1\text{BBP}_2\text{BBIBBP}_3\text{BB}\cdots$

(a) (b)

Figure 3: I-, B-, P-picture reorderings: (a) display order, (b) send buffer output order.

implementation performs one subtraction to find each decoded deadline. Account has been taken of I- B- P-picture reordering at encode and send buffer output, Figure 3, 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.

In analytical terms, consider a buffer filled with packets from just one type of picture, with total deadline time D second. (For I- and B-picture packets D is fixed but for P-pictures a mean total deadline time might be substituted.) Assume that the Bluetooth frame (outgoing packet together with incoming single time-slot acknowledgement packet) handling and propagation time is t so that total time before packet expiration is $(D-t)$. As dynamic packetization is used in the simulations in Section 4, the Bluetooth frame-size, S, can be taken as constant. Unfortunately, for VBR streams, although the display rate is constant, the output data rate varies by time. Nevertheless, the per packet consumption time is S/R if only one transmission is necessary and $(S \times N)/R$ if a maximum of N retries are necessary. Additionally, the time before packet expiration is further reduced by the need to wait for $(N-1)$ prior transmissions without consumption, that is, reduced to $(D-t-(N-1)t) = D-Nt$. Dividing the time before expiration by the packet consumption time gives the sustainable send queue length, Q, before packet discard becomes necessary:

$$Q = \left\lfloor \frac{((D-Nt) \times R)}{(N \times S)} \right\rfloor. \qquad (1)$$

With suitable adjustments to take account of different packet types, (1) might serve to regulate the flow from a compliant encoder (or transcoder) but, in case of a VBR video stream, adaptive ARQ is a convenient way to increase the available queue length by varying N in (1).

3.4. Channel model

A number of studies [31, 32] have established that the validity of employing a first-order Markov chain is a good approximation in modeling the packet-level error process in a fading channel. A Gilbert-Elliott [33, 34] two-state discrete-time, ergodic Markov chain modeled the wireless channel error characteristics between a Bluetooth master and slave node. By adopting this model it was possible to simulate burst errors of the kind that cause problems to an ARQ mechanism. The Gilbert-Elliott model was also employed for modeling the channel in a study of go-back-n and selective ARQ [35] in a CDMA spread-spectrum system and in [36] was 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 microseconds (the Bluetooth time slot duration), $T_g = 3200$ and $T_b = 400$, which implies from

$$T_g = \frac{1}{1-Pgg}, \qquad T_b = \frac{1}{1-Pbb} \qquad (2)$$

that, given the current state is good (g), Pgg the probability that the next state is also g is 0.9996875 and Pbb, 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 BER, are approximately similar to those in [37], but the mean state durations are adapted to Bluetooth. At 3.0 Mb/s, the bit error rate (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.

3.5. Simulation setup

This research employed the University of Cincinatti Bluetooth (UCBT) extension (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. All links were set at the maximum EDR 3.0 Mbps gross air rate. Simulation runs were each repeated ten times and the results averaged to produce summary statistics.

The simulations were carried out 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." Peak signal-to-noise ratio (PSNR) was found by reconstructing with a reference MPEG-2 decoder. The display rate was 25 frame/s resulting in all in 750 frames in each run. The source video was common intermediate format (CIF)-sized (366 × 288 pixel) with a GOP structure of $N = 12$, and $M = 3$ (M is the number of pictures from the I-picture to the first P-picture, i.e., including two B-pictures). In [38], fully filled Bluetooth packets were formed using maximal bandwidth five time-slot packets, regardless of slice boundaries. While this results in some loss in error resilience, as each MPEG-2 slice contains a decoder synchronization marker, in [38], it is shown that the overall video performance is superior to the choice of smaller packet sizes.

Figure 4 summarizes the testing environment. The source videos are encoded to act as traces, which are introduced into the NS-2 with UCBT extension simulator. The simulator is configured to output a trace from which packet statistics are extracted. The trace determines which of the video-bearing packets were lost. Together with the input video bitstream and its packetization details, this serves to recreate an encoded video bitstream as it would have been received according to the simulations conditions. The bitstream is then decoded and the resulting video is compared with the original to find the delivered video quality.

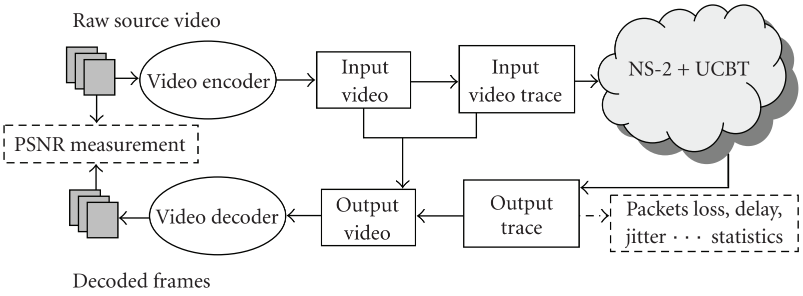


Figure 4: The testing environment for the experiments of Section 4.

4. SIMULATION RESULTS

This section examines FLC ARQ's performance in three different directions: (1) when the wireless channel conditions are varied; (2) when the send buffer size is altered; and (3) when different types of video are transmitted in addition to the "News" sequence described in Section 3.5.

As a point of comparison, FLC ARQ is compared with an adaptive ARQ scheme [20]. In this scheme, the ARQ retransmission timeout can be adaptively selected in terms of number of retransmissions allowed, to avoid further delay after the packet enters the tail of the send buffer. A threshold is set, that is, the maximum number of retransmissions allowed when the buffer is empty. The maximum number of retransmissions is subsequently changed by a factor depending on the buffer fullness reported by the Bluetooth HCI. The formula employed is summarized as

$$N = \frac{m(c - f)}{c}, \tag{3}$$

where N is the maximum number of retransmissions allowed—the retransmission timeout, m is the maximum integer-valued number of retransmissions allowed when the buffer is empty, f is the number of packets buffered in the send buffer (buffer fullness), and c is the buffer capacity. Notice that N is real valued, but is further adjusted in the same manner as the FLC output (Section 3.2), that is, by generating a random number and rounding up or down according to a comparison with the fractional part. When the buffer is empty, $f = 0$, then the maximum number of retransmissions occurs, whereas when the buffer approaches full occupation, then no retransmissions may occur. The smaller the value of m becomes the sooner this latter event occurs. In the comparative tests, the value of m varies according to the picture type of the packet, allowing a form of priority-based adaptive ARQ. The same maximum retransmission weighting was applied as for the FLC scheme (Section 3.2), namely, $m = 5, 3, 1$. respectively, for I-, P-, B-picture packets.

To examine the response to changing channel conditions, the BERs for the good and bad states of Section 3.4 were scaled by an integer-valued factor a, while Pgg and Pbb retained the values set in Section 3.4. In this way, the effect of differing (deteriorating) channel conditions could be assessed. Figure 5 plots number of transmissions needed to achieve a successful transmission according to factor a, with a buffer size of 150. The superioriority of the FLC with DAB scheme is confirmed in Figure 5, as it is in terms of mean packet delay in Figure 6. A feature of this plot is that for the worst channel conditions ($a = 5$), the average delay actually extends beyond the display deadline when the infinite ARQ with DAB scheme is employed. This is explained by the weighting given to the average by delayed I- and P-picture packets that attract an extra decode deadline, given that many B-picture packets are delayed up to their display deadline.

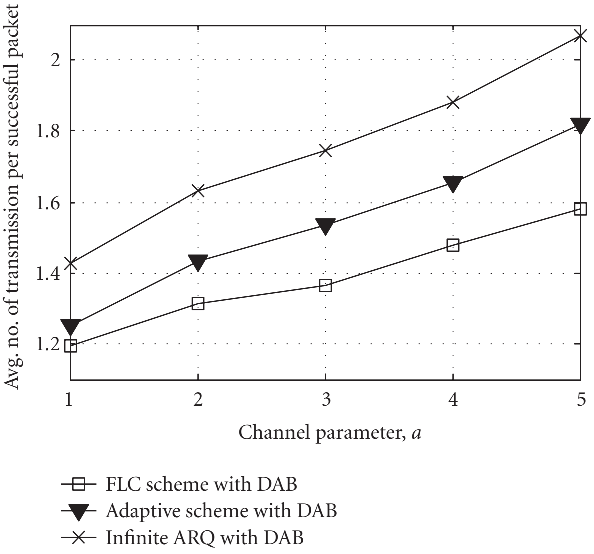


Figure 5: Mean number of transmissions by ARQ with DAB scheme, according to channel conditions.

Table 2: Various ARQ schemes applied in Figure 9.

Index	ARQ control scheme
1	FLC with DAB
2	Adaptive ARQ with DAB
3	Adaptive ARQ without DAB
4	Infinite ARQ with DAB
5	Infinite ARQ without DAB
6	No ARQ

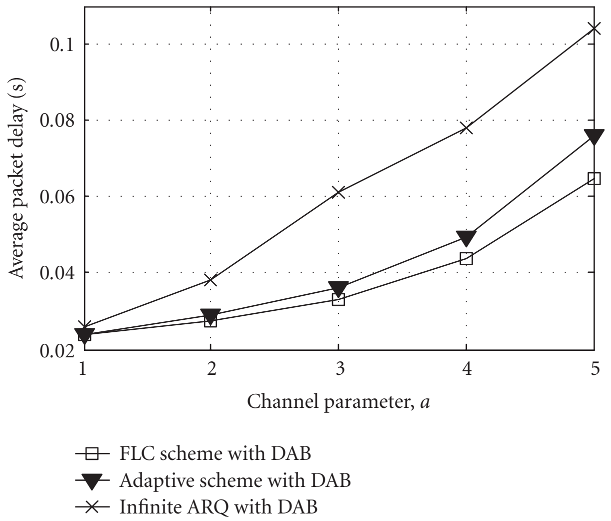


Figure 6: Mean packet delay ARQ with DAB scheme, according to channel conditions.

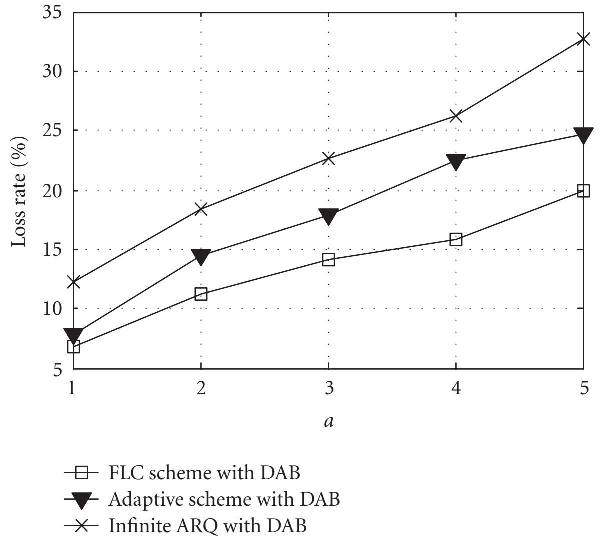


Figure 7: Mean packet loss rates for three DAB-based schemes, with changing channel conditions.

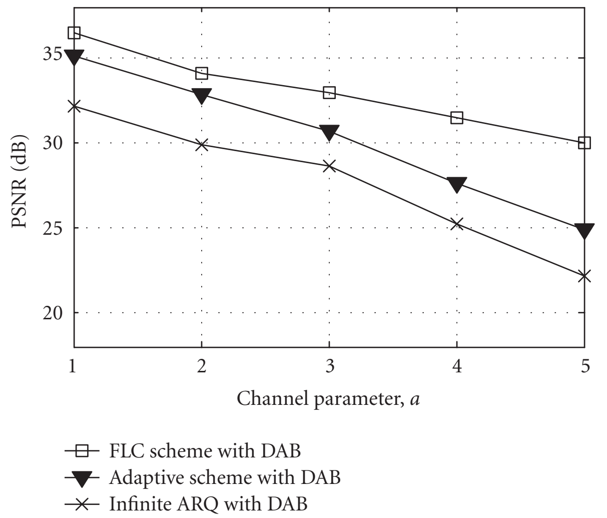


Figure 8: Mean PSNR for three DAB-based schemes, with changing buffer size.

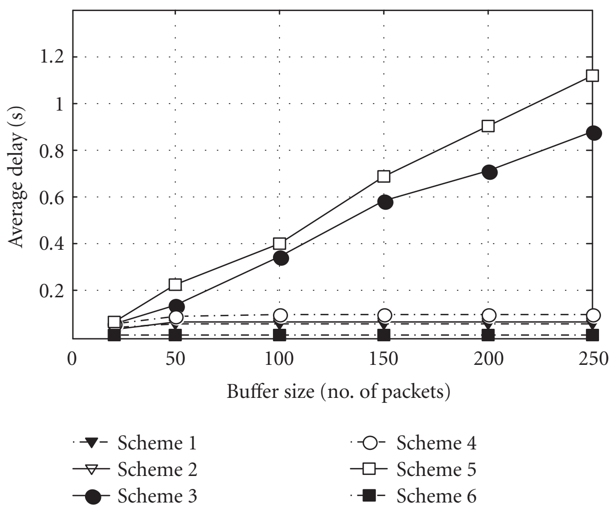


Figure 9: The impact of buffer size on the various ARQ control schemes in Table 2.

The superiority of the FLC with DAB scheme according to channel conditions is confirmed by Figure 7, again with a large number of simulation runs (fifty) to approach converged rates. As one might expect, mean PSNR follows a similar trend to packet loss rate, as illustrated by Figure 8. For our purposes, PSNR suffices as a measure of received video quality, as it certainly indicates an improvement (or little to no improvement) from applying a technique. PSNR is, of course, a relative technique and only applies to comparisons for the same video sequence.

There is no guarantee that the Bluetooth send buffer size will be set favourably, given the need for other types of traffic to utilize the link. In general, as the send buffer size is increased then more packets accumulate during a bad state, leading to an increase in the number of packets retained up to their deadline. In Figure 9, the delay as a consequence of six ARQ control schemes is compared with the schemes listed in Table 2. In the buffer-size experiments, factor *a* was set to 3. Whether adaptive ARQ or infinite ARQ is employed, if there is no DAB, then, as buffer size increases, mean delay also increases. Other schemes in the mean do not differ greatly, and in Figure 9, the plots for schemes 1 and 2 partially overlap. The time to recover from a bad state to subdeadline levels of delay is significant, as the larger the buffer the slower the recovery, as Figure 10 shows for adaptive ARQ without DAB. Therefore, one can conclude that inclusion of a DAB is clearly vital to any scheme transporting video.

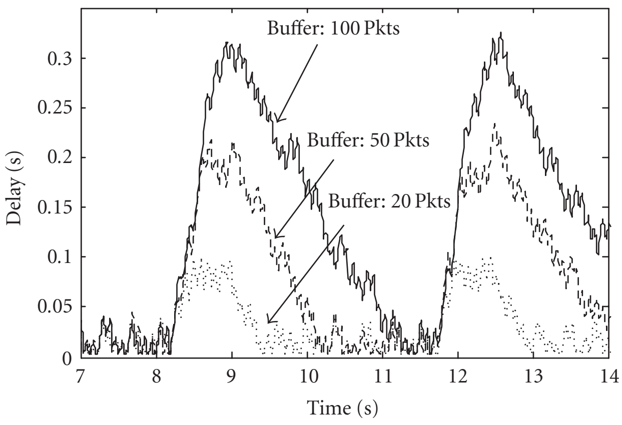


Figure 10: Example of delay recovery behavior for scheme 3 in Table 2.

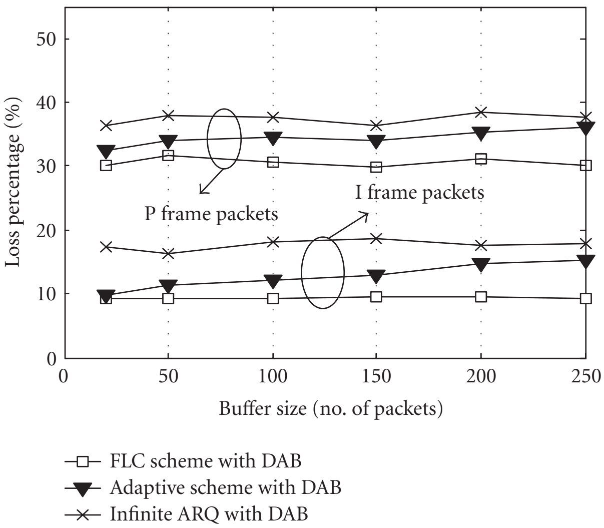


Figure 11: Packet loss rates for three DAB-based schemes, with changing buffer size.

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

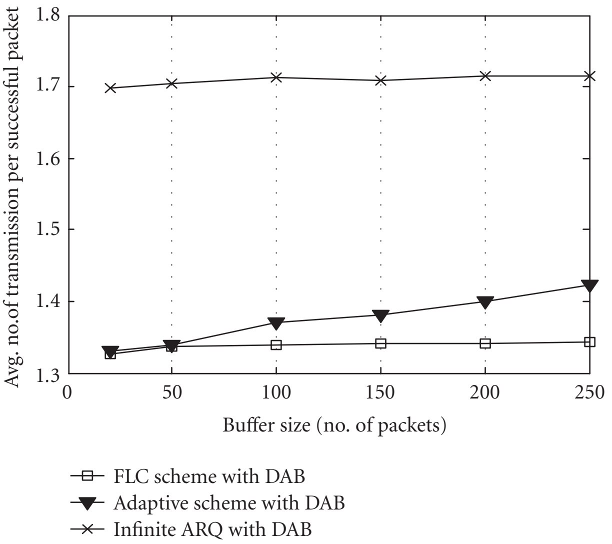


Figure 12: Mean number of transmission for three DAB-based schemes with changing buffer size.

Turning from delay to received video quality for DAB-enabled schemes, in Figure 11 the loss rates are analyzed by packet picture type. The input video is the same as in Section 3.5. The original distribution of packets is shown in Table 3. For infinite ARQ with DAB, no distinction is made in terms of the ARQ policy between different picture types, and, therefore, the discard rate reflects the ratio of picture types recorded in Table 3. As the distribution of bad states in the two-state channel model is erratic, it is only when fifty independent runs were simulated that the loss rate distribution for the infinite ARQ with DAB converged, as in Figure 11. From Figure 11, clearly the performance of the FLC with DAB scheme is significantly improved upon adaptive ARQ with DAB, and becomes more so as the buffer size is increased. In the adaptive ARQ with DAB scheme, the sole control upon retransmission is the buffer fullness ratio. As the buffer size increases, the DAB policy keeps the number of buffered packets to previous levels, and hence the buffer fullness factor reduces. Therefore, packets can be retransmitted a greater number of times and if these are B-picture packets then retained I- and P-picture packets are more likely to pass their expiration deadline. This understanding is confirmed by Figure 12, which shows that, as buffer size increases, the mean number of transmissions increases under the adaptive ARQ scheme but remains stable under FLC ARQ. The resulting impact on PSNR is recorded in Figure 13. While, the default Bluetooth scheme, even with a DAB, results in poor-quality received video, the adaptive ARQ with DAB scheme's performance is buffer-size-dependent. The video quality under the FLC with DAB scheme is reasonable and fairly constant in the mean, despite the particularly poor channel conditions that occur in bad states.

Figure 14 is a timewise comparison of PSNR of infinite ARQ, adaptive ARQ, and FLC ARQ, all with a DAB in place. The buffer size was 50 and factor *a* was set to 3 in the results of Figure 14. From Figure 14(a) it is clear that infinite ARQ even with a DAB in place, represents a much poorer experience for the viewer, especially from frame 300 to 700. The level of adaptive ARQ with DAB video quality is generally closer to 30 dB rather than 40 dB, whereas under FLC ARQ approaches a level of 40 dB, though there are some drops in quality.

BER is related to packet error rate (PER) according to packet payload length *L* (assuming that payload corruption

Table 4: Summary statistics for various ARQ control schemes, all with DAB, giving mean PSNR in dB, mean packet loss rate as a percentage, mean delay in seconds (s), with buffer size B in numbers of Bluetooth packets, and channel condition factor a being higher for worse wireless channel conditions. The results are for three different video sequences.

	ARQ Scheme	News			Friends			Football		
		PSNR	Loss	Delay	PSNR	Loss	Delay	PSNR	Loss	Delay
$B = 50$ $a = 2$	**FLC**	**34.11**	**11.45**	**0.027**	**33.56**	**12.28**	**0.028**	**32.97**	**12.41**	**0.028**
	Adaptive	33.08	13.85	0.029	32.90	15.02	0.029	32.14	15.65	0.029
	Infinite	29.92	18.05	0.038	28.79	19.89	0.040	27.80	20.31	0.041
$B = 150$ $a = 2$	Fuzzy	34.09	11.29	0.027	33.62	12.17	0.028	33.09	12.38	0.028
	Adaptive	32.87	14.59	0.029	32.55	15.83	0.029	32.03	16.31	0.029
	Infinite	29.89	18.41	0.038	28.82	19.81	0.040	27.86	20.30	0.041
$B = 50$ $a = 3$	**FLC**	**32.91**	**14.17**	**0.033**	**32.14**	**16.24**	**0.035**	**31.44**	**17.10**	**0.036**
	Adaptive	31.19	16.90	0.035	31.02	18.76	0.037	30.67	19.60	0.038
	Infinite	28.66	22.57	0.061	26.87	24.30	0.066	26.22	25.14	0.068
$B = 150$ $a = 3$	Fuzzy	32.97	14.05	0.033	32.20	16.19	0.035	31.45	17.07	0.036
	Adaptive	30.68	17.84	0.036	30.76	19.11	0.039	30.17	20.11	0.040
	Infinite	28.67	22.58	0.062	26.91	24.27	0.066	26.23	25.10	0.069
$B = 50$ $a = 5$	**FLC**	**29.89**	**20.12**	**0.065**	**28.76**	**21.90**	**0.068**	**27.63**	**22.30**	**0.072**
	Adaptive	25.32	22.25	0.073	24.11	23.57	0.076	23.00	25.41	0.081
	Infinite	22.04	33.14	0.104	20.80	34.97	0.110	19.46	35.10	0.112
$B = 150$ $a = 5$	Fuzzy	30.01	19.83	0.065	28.80	21.85	0.068	27.63	22.28	0.073
	Adaptive	24.89	24.70	0.076	23.71	24.12	0.079	22.67	26.08	0.085
	Infinite	22.11	32.97	0.105	20.89	34.70	0.111	19.51	25.33	0.112

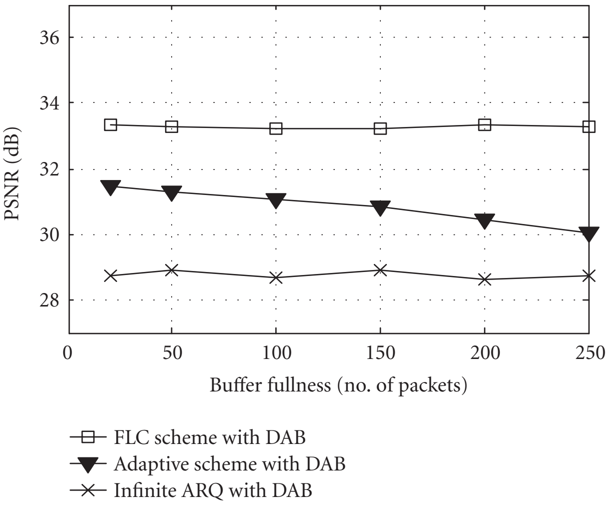


Figure 13: Mean PSNR for three DAB-based schemes with changing buffer size.

is the dominant form of PER). Set BER = p then for m retransmissions in addition to the first attempted transmission, the PER is calculated as

$$\text{PER} = \left(1 - (1 - p)^L\right)^{m+1}. \tag{4}$$

Plotting PER against BER for various retry counts (m), with fixed L appropriate to 5-time-slot Bluetooth packets in 3.0 Mbit/s EDR mode (Section 3), Figure 15 shows in general terms that there are variations in the range of BERs for which adaptive ARQ is appropriate. As the BER becomes low, ARQ becomes less appropriate as all of the plotted retry limits result in very low PERs. For high retry BERs, all retry limits result in impossible PER levels.

Table 4 provides summary statistics (mean of 50 runs) for the different schemes, with three input video sequences: (1) "News" as in previous experiments in this section, (2) "Friends" from the well-known American situational comedy, with more "action" than in "News," and (3) "Football" with rapid movement. The additional clips had the same GOP structure as the "News" sequence and similarly were CIF-sized at 25 frames/s. A DAB was employed for all the schemes, resulting in similar mean packet delay times for the two priority-based ARQ schemes (adaptive and FLC), according to channel condition. However, it is important to realize that delay is only recorded for delivered packets, whereas the presence of the DAB results in packet discard before delay is recorded. Obviously, delay increases as the wireless channel BERs increase through the application of scaling factor a to the good and bad state BERs of Section 3.4. Across the different video clips, packet loss rates increase and PSNR decreases approximately according to the degree of motion in each video, ranked in order "News," "Friends," and "Foot-

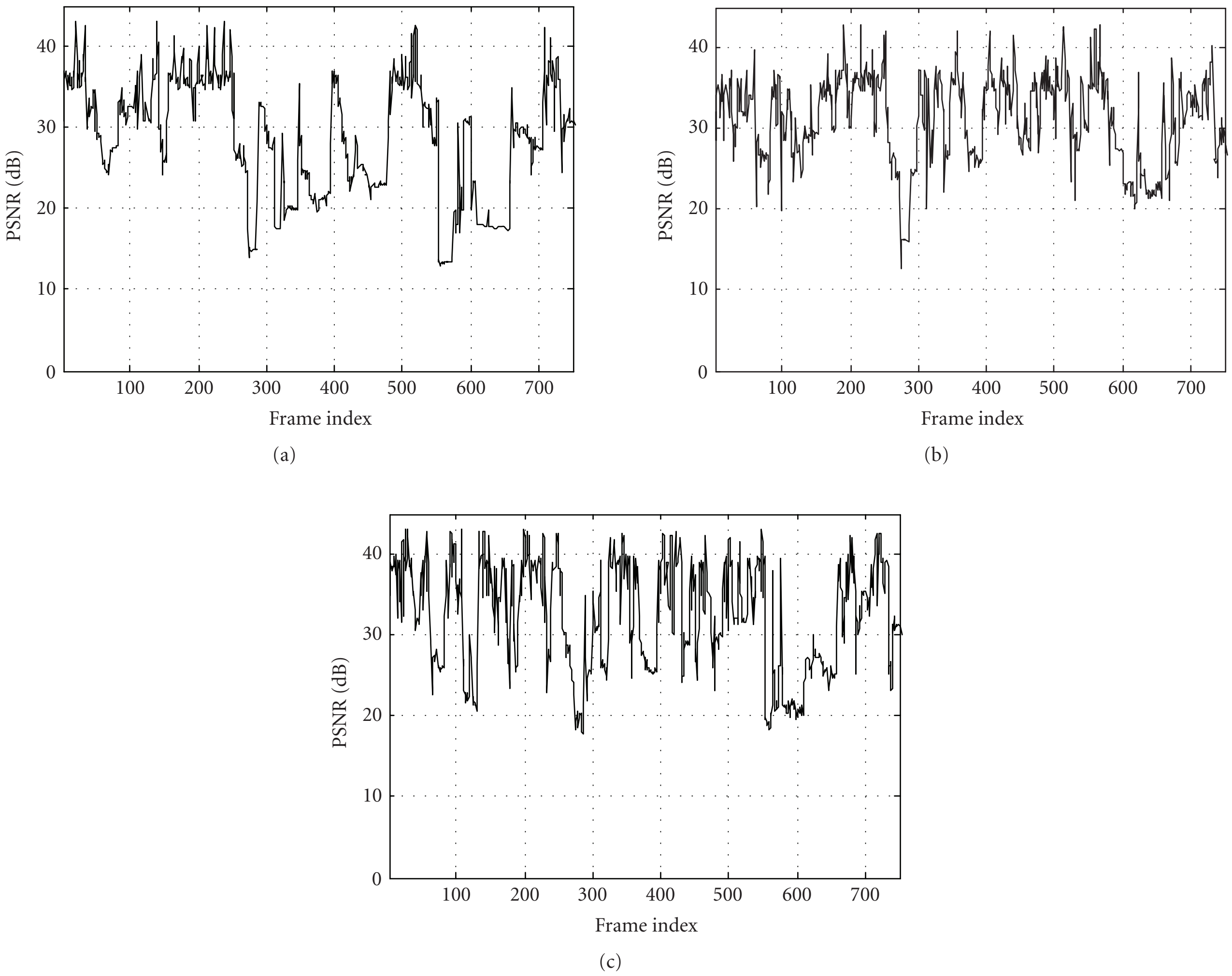


Figure 14: PSNR for the test video under (a) infinite ARQ with DAB; (b) adaptive ARQ with DAB; (c) FLC ARQ with DAB.

ball.” Increasing the buffer size results in a negative impact on packet loss rates in respect to the adaptive ARQ scheme, in the sense that the video quality for all three clips deteriorates as the buffer size is increased (owing to the adverse effect on buffer queue waiting times when the ARQ scheme is regulated by a buffer fullness factor). The default Bluetooth ARQ scheme never results in a mean PSNR above 30 dB. Lastly, the FLC with DAB results are emboldened as, in all cases, the received video quality is superior compared to the other schemes, whatever the channel conditions, buffer size, or type of input video clip.

5. CONCLUSIONS

This paper compared various data-link layer schemes for control of ARQ timeouts and found that fuzzy logic control results in a quite considerable improvement in received video quality over a traditional scheme. Though adaptive control of ARQ at the link layer is known in the literature, mainly for other than Bluetooth, the identification of a near optimal scheme is not. Fuzzy logic control can readily be tuned but once the operating parameters are established, no further modifications are required. Though the detailed experiments in this paper are specific to Bluetooth, there is no reason why the same approach should not be applied to other wireless technologies that employ ARQ as a form of error control. Equally, though the scheme was tested with the widely deployed MPEG-2 codec, I and P slices are present in the more recent H.264 and B slices occur in all but H.264’s baseline profile.

Summary results found that in poor channel conditions fuzzy logic control of adaptive ARQ resulted in at least 4 dB improvement in video quality. A secondary finding of the paper was that by the addition of a deadline-aware buffer, delivered packet delay is reduced, though this is only significant if the number of discarded packets through deadline expiration is not high. The delivered video quality of the fuzzy logic controlled scheme is relatively immune to change in send buffer size, whereas adaptive ARQ using buffer fullness to judge the number of retransmissions is buffer-size-dependent, with larger buffer sizes having a negative effect on received video quality. Fuzzy logic control of ARQ in this paper adjust the number of retransmissions in a way that time

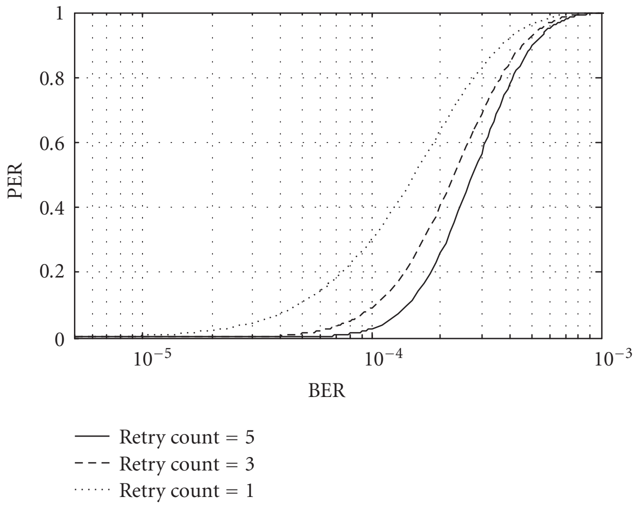


Figure 15: PER against BER for differing ARQ retry limits.

delay constraints are honored and buffer overflow is sharply reduced. As fuzzy logic control reduces the number of retransmissions it also reduces power consumption. The possibility of regulating power consumption in mobile devices by an additional power-control factor is open to a fuzzy logic scheme, whereas such an enhancement is less obvious in a tradional ARQ scheme.

ACKNOWLEDGMENT

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

REFERENCES

[1] J. C. Haartsen, "The Bluetooth radio system," *IEEE Personal Communications*, vol. 7, no. 1, pp. 28–36, 2000.

[2] "Specification of the Bluetooth System—2.0 + EDR," 2004, http://www.bluetooth.com.

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

[4] A. K. Arumugam, S. M. D. Armour, M. F. Tariq, and A. R. Nix, "Proposed evolution technologies for Bluetooth," in *Proceedings of the IEEE 54th Vehicular Technology Conference (VTC '01)*, vol. 4, pp. 2523–2527, Atlantic City, NJ, USA, October 2001.

[5] R. Razavi, M. Fleury, and M. Ghanbari, "Low-delay video control in a personal area network for augmented reality," in *Proceedings of the 4th Visual Information Engineering (VIE '07)*, pp. 1245–1300, London, UK, July 2007.

[6] 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. 123–134, 2005.

[7] 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. 2, pp. 661–664, Barcelona, Spain, September 2003.

[8] R. Kapoor, M. Cesana, and M. Gerla, "Link layer support for streaming MPEG video over wireless links," in *Proceedings of the 12th International Conference on Computer Communications and Networks (ICCCN '03)*, pp. 477–482, Dallas, Tex, USA, October 2003.

[9] 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*, Kyongju, Korea, May 2001.

[10] 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.

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

[12] 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.

[13] N. Laoutaris and I. Stavrakakis, "Adaptive playout strategies for packet video receivers with finite buffer capacity," in *Proceedings of the International Conference on Communications (ICC '01)*, vol. 3, pp. 969–973, Helsinki, Finland, June 2001.

[14] Z. Jiang and L. Kleinrock, "A packet selection algorithm for adaptive transmission of smoothed video over a wireless channel," *Journal of Parallel and Distributed Computing*, vol. 60, no. 4, pp. 494–509, 2000.

[15] 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, December 2004.

[16] H. Zhu, M. Li, I. Chlamtac, and B. Prabhakaran, "A survey of quality of service in IEEE 802.11 networks," *IEEE Wireless Communications*, vol. 11, no. 4, pp. 6–14, 2004.

[17] M. van der Schaar, S. Krishnamachari, S. Choi, and X. Xu, "Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLANs," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 10, pp. 1752–1763, 2003.

[18] 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.

[19] C.-M. Chen, C.-W. Lin, and Y.-C. Chen, "Packet scheduling for video streaming over wireless with content-aware packet retry limit," in *Proceedings of the IEEE 8th Workshop on Multimedia Signal Processing (MMSP '06)*, pp. 409–414, Victoria, BC, Canada, October 2006.

[20] R. Razavi, M. Fleury, and M. Ghanbari, "Adaptive timeout for video delivery over a Bluetooth wireless network," in *Proceedings of the International Conference on Applied Computing (IADIS '07)*, pp. 245–254, Salamanca, Spain, February 2007.

[21] J. Bandara, X. Shen, and Z. Nurmohamed, "A fuzzy resource controller for non-real-time traffic in wireless networks," in *Proceedings of the IEEE International Conference on Communications (ICC '00)*, vol. 1, pp. 75–79, New Orleans, La, USA, June 2000.

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

[23] Y. Xiao, P. Chen, and Y. Wang, "Optimal admission control

for multi-class of wireless adaptive multimedia services," *IEICE Transactions on Communications*, vol. E84-B, no. 4, pp. 795–804, 2001.

[24] L. Rossides, C. Chrysostomou, A. Pitsillides, and Y. A. Sekercioglu, "Overview of fuzzy-RED in diffServ networks," in *Proceedings of the 1st International Conference on Computing in an Imperfect World*, vol. 2311 of *Lecture Notes In Computer Science*, pp. 1–13, Belfast, Ireland, April 2002.

[25] C. Luo and C. Ran, "An adaptive retransmission and active drop mechanism based on fuzzy logic," in *Proceedings of Asia-Pacific Radio Science Conference (APRASC '04)*, pp. 162–165, Qingdao, China, August 2004.

[26] 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.

[27] 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, Kowloon, China, March 2002.

[28] R. Razavi, M. Fleury, and M. Ghanbari, "Detecting congestion within a Bluetooth piconet: video streaming response," in *London Communications Symposium*, pp. 181–184, London, UK, September 2006.

[29] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice Hall, Upper Saddle River, NJ, USA, 1996.

[30] M. Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding*, Institution Electrical Engineers, Stevenage, UK, 2003.

[31] M. Khansari, A. Jalali, E. Dubois, and P. Mermelstein, "Low bit-rate video transmission over fading channels for wireless microcellular systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 1, pp. 1–11, 1996.

[32] M. Zorzi, R. R. Rao, and L. B. Milstein, "ARQ error control for fading mobile radio channels," *IEEE Transactions on Vehicular Technology*, vol. 46, no. 2, pp. 445–455, 1997.

[33] E. N. Gilbert, "Capacity of burst-noise channel," *The Bell System Technical Journal*, vol. 39, no. 8, pp. 1253–1265, 1960.

[34] E. O. Elliott, "Estimates of error rates for codes on burst noise channels," *The Bell System Technical Journal*, vol. 42, no. 9, pp. 1977–1997, 1963.

[35] C.-Y. Hsu, A. Ortega, and M. Khansari, "Rate control for robust video transmission over burst-error wireless channels," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 5, pp. 756–773, 1999.

[36] L.-J. Chen, T. Sun, and Y.-C. Chen, "Improving Bluetooth EDR data throughput using FEC and interleaving," in *Proceedings of the 2nd International Conference on Mobile Ad-Hoc and Sensor Networks (MSN '06)*, vol. 4325 of *Lecture Notes In Computer Science*, pp. 725–736, Hong Kong, December 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] R. Razavi, M. Fleury, E. Jammeh, and M. Ghanbari, "Efficient packetisation scheme for Bluetooth video transmission," *Electronics Letters*, vol. 42, no. 20, pp. 1143–1145, 2006.