

Block-Based Rateless Coding for Energy-Efficient Video Streaming over Bluetooth

Rouzbeh Razavi, Martin Fleury and Mohammed Ghanbari

CES Dept., University of Essex, Colchester, United Kingdom

Abstract— This paper introduces block-based rateless coding for video streaming over a wireless interconnect, in which after a packet erasure additional coded blocks are generated and piggy-backed onto outgoing packets. The advantage is shown in comparison to default Bluetooth FEC schemes, as, through the block-based rateless scheme, transmission energy consumption is reduced by a factor of up to 1.8, depending on Rayleigh channel bad state durations. In poorer channel conditions, the rateless scheme improves delivered video quality by as much as 10 dB relative to a Bluetooth v. 2.1 EDR mode. Decode complexity for rateless Raptor codes is linear in block size.

I. INTRODUCTION

In a Bluetooth (IEEE 802.15.1) [1] interconnect, error control is critical, owing to the presence of radio frequency noise and many causes of interference such as multi-path and fast fading. However, preservation of battery charge is also an essential feature of mobile devices, to reduce the frequency of recharges. As error control involves the transmission of redundant data then minimization of this overhead will contribute to energy-efficient video streaming. This paper demonstrates that Fountain or rateless codes [2] better reduce transmission energy consumption than other Forward Error Control (FEC)-based methods. Furthermore, with a block-based method of rateless coding the redundancy is reduced because the unit of coding is not a packet but a block within a packet.

Rateless coding is ideally suited [2] to a binary erasure channel in which either the error-correcting code works or the decoder fails and reports that it has failed. In erasure coding all is not lost as flawed packets may be reconstructed from a set of successfully received packets (if sufficient of these packets are received). In all applications of rateless coding that we are aware of additional redundant packets are generated upon a feedback message request. However, in this paper a Bluetooth packet is split into blocks of 15 bits each and additional redundant blocks not packets are generated. By piggybacking redundant blocks onto newly transmitted packets, redundancy is incrementally achieved until either a prior video-bearing packets received in error are reconstructed or the display deadline of the frame of which that packet is a part expires.

Unlike fixed-rate erasure coding (typically Reed-Solomon (RS)), rateless coding relies on feedback.

Bluetooth is very suitable for application of rateless coding as there is automatic feedback to the sender and because of its short range (typically for Class 2 devices less than 10 m) that feedback is of low latency. In Bluetooth, feedback comes for free by virtue of Time Division Duplex (TDD) polling, which is necessary for transmit/receive recovery, allowing a single-chip implementation. On the other hand, application of rateless coding to the Internet, e.g. [3], is not always ideal because the feedback channel is slow. Bluetooth packets also automatically contain a Cyclic Redundancy Check (CRC) for the user payload, allowing detection of a failed decode. It is assumed in this paper that the CRC is applied to the payload after rateless decoding.

In rateless coding, each packet certainly contains $k(1 + \varepsilon)$ blocks, where ε is a small fractional overhead, typically 5% [3], to ensure with high probability that all k blocks are decodable if received without error (rateless codes are constructed in probabilistic fashion). Raptor codes [4] have constant time encode and linear decode computational complexity, though additional pre-coding is performed prior to formation of the rateless code.

Through the Enhanced Data Rate (EDR), Bluetooth version 2.1 [5] now has a peak user payload of 2.2 Mbps (gross air rate 3.0 Mbps), which is the same average rate offered by some implementations of IP-TV. Though Bluetooth devices have hold, park, and sniff low-activity modes, and the transceiver is designed to minimize power [6], it is still important that an application reduce the total data transmitted, as there is approximately a linear relationship [7] between bitrate and energy consumption. In [8] it was shown that transmission accounts for more than a third of the total energy consumption in communication on a mobile device. In [9], 78% of energy consumption is attributed to transmission and playback at the receiver. Bluetooth must co-exist with related low-power technologies, such as ZigBee (IEEE 802.15.4) and Wibree from Nokia, intended for button-cell batteries, with a gross air rate of 1.0 Mb/s.

The rest of this paper is organized as follows. Section II discusses essential background, including the complexity and overhead offered by a rateless code. Section III details the block-based rateless coding algorithm and generally the experimental methodology. Section IV presents a number of results showing that

energy is conserved but delivered video quality is improved. Finally, Section V draws some conclusions.

II. BACKGROUND

A. Rateless codes

An (n, k) RS erasure code over an alphabet $q=2^L$ (where L is the number of bits in a block) has the property that if *any* k out of the n blocks transmitted are received successfully then the original k blocks can be decoded. However, in practice not only must n, k and q be small but also the computational complexity of the decoder is of order $n(n - k) \log_2 n$.

The class of Fountain codes [2] allows a continual stream of additional blocks to be generated in the event that the original blocks could not be decoded. It is the ability to easily generate new blocks that makes Fountain codes rateless. Decoding will succeed with small probability of failure if any of $k(1 + \varepsilon)$ blocks are received. In its simplest form, the blocks are combined in an exclusive OR (XOR) operation according to the order specified by a random low-density generator matrix and in this case, the probability of decoder failure is $f = 2^{-k\varepsilon}$, which for large k approaches the Shannon limit. The random sequence must be known to the receiver but this is easily achieved through knowledge of the sequence seed. Luby Transform (LT) codes [10] reduce the complexity of decoding a simple Fountain code (which is of order k^3) by means of an iterative decoding procedure, provided the column entries of the generator matrix are selected from a robust Soliton distribution. In the LT generator matrix case, the expected number of degree one combinations (no XORing of packets) is $S = c \log_e(k/f) \sqrt{k}$, for small constant c . Then setting $\varepsilon = 2 \log_e(S/f)S$ ensures that by sending $k(1 + \varepsilon)$ blocks these are decoded with probability $(1 - f)$ and decoding complexity of order $k \log_e k$.

Furthermore, if the blocks are pre-encoded with an erasure code then a weakened LT transform can be applied to the blocks and their parity blocks. The advantage of this Raptor code [4] is a decoding complexity that is linear in k . Notice that an essential difference between Fountain erasure codes and RS erasure codes is that Fountain codes in general are not systematic¹ and that even if there were no channel errors there is a very small probability, assuming correct design, that the decoding will fail. In compensation, they are completely flexible, have linear decode computational complexity, and generally their overhead is considerably reduced compared to fixed erasure codes.

B. Related work

Fountain codes are now attracting applications in video streaming applications. In video streaming for cognitive radio [11], rateless error coding compensates an opportunistic secondary source from interference by the primary occupant of the wireless channel. In essence, this is the same network coding technique as applied in [12], because it allows a set of sub-channels distributed across the available wireless spectrum to stream scalable video without coordination between the sources. Because the symbols (packets or blocks) of a Fountain code are generated from a sparse distribution, any uncoordinated sources are unlikely to construct the same two symbols. However, this is *not* how the present paper proposes to employ rateless coding, as in [12] there are multiple uncoordinated channels, whereas herein there is a single channel that is coordinated with the receiver. Therefore, BlueTorrent, concerning which [13] mentions in passing network coding for Bluetooth, is not related to the current paper.

In [14], it was observed that classic error control methods work poorly in terms of energy conservation, in line with similar comments in Section I. It was proposed in [14] that the channel should be probed to find the error conditions, whereupon the level of ARQ retransmissions is adjusted. However, the volatility of the wireless channel may make measurements unreliable. The work in [15] proposed a scheme of error control which varied according to the channel conditions and to the relative energy budget for RS coding and Selective Repeat ARQ. As in our paper, a two-state type model allowed (Rayleigh) fading conditions to be modeled, in way that is independent of packet size. It was found that there was a threshold, beyond which FEC was necessary, despite the increase in energy budget. In [16], packet-level FEC (not block-level as in our paper) and power allocation are jointly optimized across cellular radio. The work combines layered video coding with FEC, with the degree of protection varying according the priority of the layer. The layers actually transmitted depend on the power resources of the sender.

In [17] rateless coding is selected for reasons of reduced decode computational complexity in an energy reduction scheme for wireless mesh networks. This scheme is compared to network coding and similar schemes for data broadcast. Others have noticed the advantage of rateless coding for energy conservation, for example in [18], rateless coding is applied in a sensor network context but for data not video and from the reduced decode complexity point-of-view and not necessarily because of reduced transmission overhead.

C. Bluetooth packetisation

Bluetooth employs variable-sized packets up to a maximum of five frequency-hopping time-slots of 625 μ s in duration. In TDD, every Bluetooth frame consists of a packet transmitted from a transmitter 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

¹ In the 3GPP standard, a systematic Raptor code is arrived at [4] by first applying the inverse of the inner LT to the first k symbols before the outer pre-coding step.

even number of slots. Bluetooth v. 2.1 EDR supports both a gross air rate of 2.0 Mbps (mgup of 1.4485 Mbps) and 3.0 Mbps, through respectively $\pi/4$ -DQPSK or 8DPSK modulation. In EDR, the symbol rate (1 Msps) remains the same as in the basic rate.

Though Bluetooth EDR makes video streaming feasible, the FEC-bearing Data Medium (DM) packets of the basic rate were not included in EDR. However, in [19] these are introduced and in Section IV, we compare rateless coding with these FEC-bearing modes. Bluetooth's default FEC scheme, expurgated (15, 10) Hamming code, which is applied to 15-bit blocks [19], can cope with burst sizes of two, depending on decoder. Table I summarizes the additional EDR Asynchronous Connection-Less (ACL) mode packet types currently available (according to the specification), as well as EDR DM-type packets in the event that symbol-level FEC were to be added to EDR. Because packets contain header overhead, there are throughput quantization effects imposed by the packetization structure of Table I. When rateless coding blocks are added to packets, as described in Section I, then it is important to determine how many redundant blocks to add in the event of error(s) in prior packets, because there is a trade-off between achievable throughput and the probability of error correction by means of the additional redundant blocks. By way of illustration for an error-free channel, in Fig. 1 these quantization effects are shown by the stepped changes in throughput according to packet size. The horizontal access refers to the size of arriving IP packets prior to Bluetooth packetization, which in this case is subsequently performed with one or more of the Data High (DH) packet types from Table I. Section III.C returns to this trade-off but for a bursty channel.

III. METHODOLOGY

A. Channel model

A Gilbert-Elliott two-state discrete-time, ergodic Markov chain models the wireless channel error characteristics between a Bluetooth master and slave node. The Gilbert-Elliott model was, in [19], applied to the same version of Bluetooth as herein. However, notice that 'bursty' channels may more thoroughly be represented by a multi-state model.

The mean duration of a good state, T_g , was set at 2 s and in a bad state, T_b was set to $a \times T_g$, where a is a parameter which is varied to alter the duration of bad states. In units of 625 μ s (the Bluetooth time slot duration), $T_g = 3200$ which implies from:

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

that, given the current state is good (g), P_{gg} , the probability that the next state is also g , is 0.9996875. Both good and bad state is modeled with a Rayleigh channel with the mean SNR being 35 ± 1 dB and 25 ± 1 dB in the g and b states respectively.

TABLE I. EDR PACKET TYPES IN BLUETOOTH ACL MODE

Packet type	User payload in bytes	Asymmetric max. rate (kbps)
2-DM1	0-36	230.4
2-DM3	0-245	782.9
2-DM5	0-453	965.7
2-DH1	0-54	345.6
2-DH3	0-367	1174.4
2-DH5	0-679	1448.5
3-DM1	0-55	354.1
3-DM3	0-368	1184.3
3-DM5	0-681	1452.0
3-DH1	0-83	531.2
3-DH3	0-552	1776.4
3-DH5	0-1021	2178.1

Length and master to slave bitrates, for a single ACL master-slave logical link, with DM = Data Medium rate (FEC added) and DH = Data High rate (no FEC). 2-DH3 is 2.0 Mbps modulation three time-slot packet.

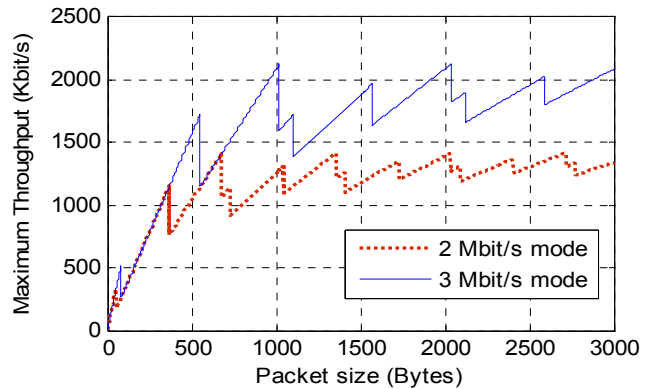


Fig. 1. Throughput quantization effects for Bluetooth EDR mode

B. Simulation setup

This research employed the University of Cincinnati Bluetooth (UCBT) extension² 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 100 times and the results averaged to produce summary statistics.

The simulations were principally carried out with input from an MPEG-2 encoded bitstream at a mean rate of 1.5 Mbps for a 30 s video clip with moderate motion, showing a newsreader and changing backdrop, which we designate 'News'. (Other similar 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 (352×288 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). In [20] it

² (a download is available from <http://www.ececs.uc.edu/~cdmc/UCBT>)

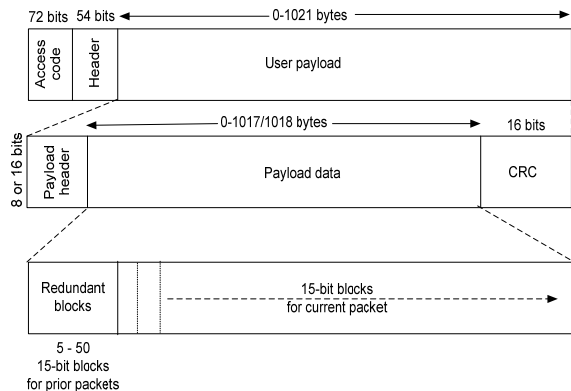


Fig.2. Bluetooth packetization structure showing the incorporation of redundant blocks into the payload.

was demonstrated that forming fully-filled Bluetooth packets outweighed the need to preserve MPEG-2 slice boundaries, which over the fixed Internet are preserved for error-resilience purposes

C. Redundant block transmission algorithm

Fig. 2 shows the partition of a video-bearing Bluetooth packet payload into three parts: 1) a variable-sized redundant block portion, with the blocks within this portion generated by the rateless algorithm from prior packets; 2) the data of the next packet divided into blocks with an additional ε blocks generated by the rateless algorithm, as $k(1 + \varepsilon)$ blocks are required for reconstruction with high probability of the original k blocks; 3) a CRC which is a default part of a Bluetooth packet but which we assume is applied to the decoded k blocks of the current packet. Upon failure of the CRC, additional blocks are requested from the sender and these are sent in the first part of the next packet together with any other blocks from yet to be reconstructed packets. No doubt because the 64-bit Bluetooth access code is protected by a 34-bit BCH block code and the header fields are sent at the basic rate and protected at a rate of 1/3, it is reported in [21] that payload errors are most likely, and consequently access code and header errors are not included in the simulations in Section IV.

Assume for simplicity that just one prior packet is in error then the redundant blocks now add to the original $k(1 + \varepsilon)$ blocks to increase the probability of a successful decode³ and after decode the CRC of that prior packet is applied to establish whether there has been an erasure. If there is an erasure additional blocks are requested through Bluetooth feedback, unless the duration of block retransmissions already exceeds the display deadline of the frame of which that packet's data forms a part. By sending in the feedback a bitmap indicating which previous packets are in error, the Bluetooth sender can decide from which previous packets (including the just transmitted packet) redundant blocks still need to be sent.

It is important to realise that, in rateless coding, upon detection after decoding of a payload in error through the CRC, subsequently sending further redundant

blocks allows the same CRC to act as a check upon successful decoding. There is no need to isolate which block is in error and, indeed, this would not be possible anyway in versions of rateless coding other than Raptor in [4] because the data is not separated from the redundancy. Subsequent piggy-backed blocks together with the existing transmitted blocks are decoded and then checked by the CRC that was sent with the corresponding original $k(1 + \varepsilon)$ blocks.

Critical to the operation of rateless error correction is the number of blocks contained in part 1 of a Bluetooth packet payload. If redundant blocks are to be sent then a minimum and a maximum number of 15-bit blocks is defined, being 5 and 50 respectively in the simulations of Section IV. Leaving aside initialisation packets, the starting number of redundant blocks was the minimum number (five blocks) in our simulations. Upon receipt of a consecutive sequence of errorless packets, 100 in the simulations, then the limit is reduced by one. Upon a failure then the number of blocks included is increased by a factor of 1.5. This conservative policy for a volatile channel results in a rapid increase in redundancy when un-correctable errors first occur.

If more than one prior packet of the same frame type has errors then the redundant block allowance is split equally between the prior faulty packets, allowing for some irregularity due to the need to apportion an integer number of blocks. A simple acknowledgment of the differing importance of frame types was made by altering the allocation in the ratio 3:2:1 for I-, P-, and B-frame packets respectively.

IV. RESULTS

Experiments were conducted streaming the video of Section III.B. A varying number of redundant blocks were included in the packet payload if one or more prior packets were found to be in error. In these and subsequent experiments, the 3DH packets were selected from Table I. For any one packet in error, retransmissions continued until the number of retransmissions, d , exceeded ten. This limit is taken to be the display limit. In practice, the display deadline is controlled by the size of a playout buffer at the receiver but, in the experiments, the retransmission limit served as a gauge of the display deadline. After d is exceeded then the packet is declared as lost. Fig. 3 shows how there is a sharp reduction in the packet loss ratio (the number of lost packets to the number of packets transmitted) at a given average SNR (the good and bad state SNRs are not fixed in these experiments) for a relatively small investment in redundant blocks. In a different context and application, similar sharp changes in performance with increase in redundancy have been noticed elsewhere [22]. Our experiments governed the settings for the minimum and maximum redundant block numbers described in Section III.C. The system was also tuned to find when the packet loss ratio begins to level off for varying durations of the bad state, controlled by variable a in Section III.A. From Fig. 4, it will be seen that a value of $d = 10$ is a reasonable.

The video of Section III.B was streamed under various FEC-bearing schemes as a way of judging the packet loss ratio versus the overhead from including FEC. A more efficient scheme will reduce packet loss

³ See [4] for analysis of the error probability of Raptor codes.

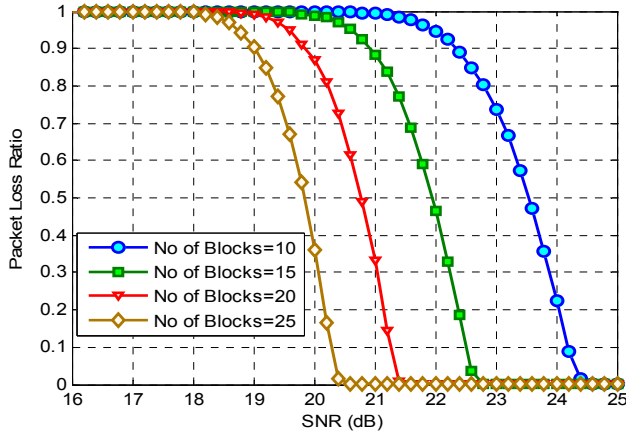


Fig. 3. Packet loss ratio according to the number of redundant blocks in a Rayleigh channel with varying SNR.

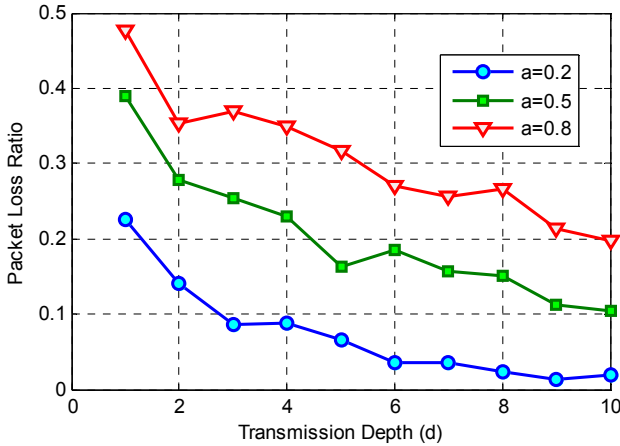


Fig. 4. Packet loss ratio according to the transmission depth, for varying duration (indexed by a) of bad state in a two-state Rayleigh channel.

and reduce the transmission overhead, which in Section I is identified as a principal contributor to energy consumption. In Fig. 5 for packet loss or unrecoverable packets, the FEC-bearing schemes use the 3DM packets of Table I. The adaptive FEC-bearing scheme assumes perfect channel knowledge, as FEC packets are only selected when the channel enters a bad state. This scheme is introduced as it has the ability to save energy by reducing the overhead when channel conditions ease. Again, the duration of the bad state is controlled by parameter a . Because the native Bluetooth scheme already has a rate of 1/3, i.e. considerable overhead, automatic ARQ is turned off to avoid increasing the FEC overhead. (ARQ is effectively turned off [5] by setting the flush timeout to a minimal value.) From Fig. 5, it is apparent that the proposed rateless scheme outperforms the native schemes and increasingly so as the bad state durations increase.

The various schemes were compared in terms of energy consumption as judged by the data successfully transmitted while streaming the video clip. In Fig. 6 it is important to note that all schemes are relative to the proposed rateless scheme. Furthermore, the energy saving is taken relative to the ratio of successfully transmitted bits. This explains why, though the 'no FEC' plot has no overhead from FEC, it still has a poor

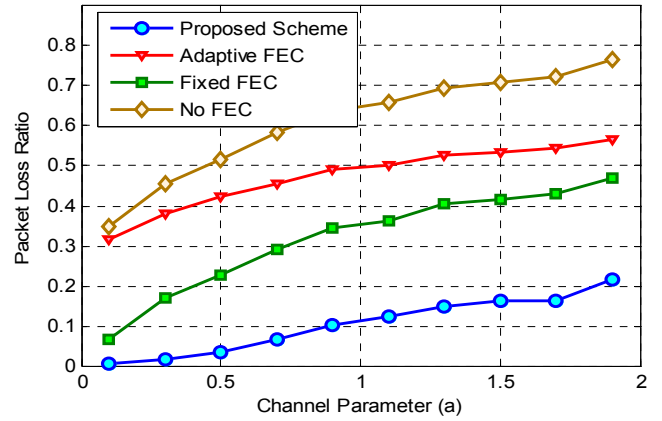


Fig. 5. Packet loss ratio comparison between various FEC bearing schemes, for varying duration (indexed by a) of bad state in a two-state Rayleigh channel.

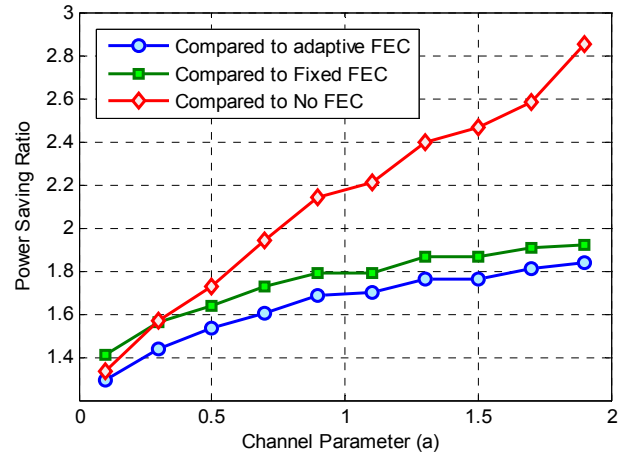


Fig. 6. Energy saving for successfully transmitted bits relative to the proposed rateless scheme of varying Bluetooth schemes when transmitting a video clip across a two-state Rayleigh channel.

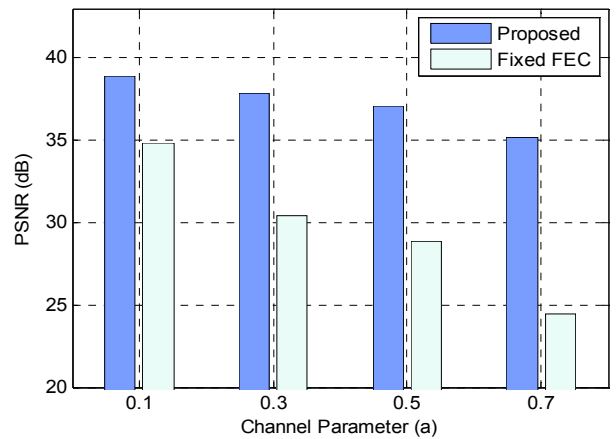


Fig. 7. Video quality comparison for transmission at 3.0 Mbps gross air rate using the rateless scheme and the Bluetooth v.2.1 scheme. energy saving ratio compared to the proposed scheme.

The FEC-bearing schemes use more energy in transmission by a factor approximately between 1.3 and 1.9, depending on bad state durations. Adaptive FEC is relatively better at energy reduction than fixed FEC but, of course, the number of unrecoverable packets is greater from Fig. 5. For the plots of Fig. 5 (and 6), a comparison between the delivered video quality for selected bad state durations for which PSNR is of a

TABLE II. VIDEO QUALITY COMPARISON BETWEEN NATIVE BLUETOOTH EDR TRANSMISSION MODE AND RATELESS CODING

	Football Clip PSNR		Friends Clip PSNR	
	Proposed (dB)	Fixed FEC (dB)	Proposed (dB)	Fixed FEC (dB)
$a=0.1$	36.11	35.21	37.17	34.87
$a=0.3$	35.82	34.07	36.43	34.02
$a=0.5$	34.79	33.22	35.20	32.67
$a=0.7$	33.46	31.83	34.65	31.10

reasonable level, Fig. 7 shows that the rateless scheme also outperforms transmission under the existing EDR mode without FEC. (Recall from Section II.C that FEC mode packets are not actually part of the Bluetooth 2.1 standard.) The relative improvement increases with worsening bad state duration.

To confirm the advantage of the proposed rateless scheme in terms of delivered video quality, two other video clips were transmitted over a Bluetooth interconnect at 3.0 Mbps, with the same configuration of ‘News’ in Section III.B. ‘Football’ contains considerable movement and includes scene cuts and as a result its average PSNR is reduced compared to ‘News’. The ‘Friends’ clip from the well-known situational comedy has less motion than ‘Football’ but more motion than ‘News’. In all comparisons in Table II, the proposed scheme is superior.

V. CONCLUSIONS

Energy usage becomes an important factor when the receiver is a mobile device. The proliferation of such devices implies that any reduction in the recharge frequency is a welcome development. Transmission of higher quality video over a Bluetooth interconnect has been long sought. However, it is important to factor in energy usage and not simply regard a wireless channel as a fixed channel with the addition of errors, to caricature one view. Rateless coding in its Raptor form has a constant encode complexity and a linear decode complexity. Compared to RS erasure codes, not only is the complexity sharply reduced, which is important for a low-cost device operating at video rates, but the overhead in normal operation is 5% (if decode is to succeed with high probability). This paper has further demonstrated that when incremental coding is required the overhead is still significantly reduced compared to native Bluetooth schemes. While on the fixed Internet, rateless coding has mostly, perhaps exclusively, been used for packet erasure coding, this paper introduces block erasure coding. Block-based coding allows additional redundant blocks rather than packets to be generated, scaling the overhead that occurs. Thus, block rather than packet-based rateless erasure coding also contributes to the energy reductions that occur.

REFERENCES

[1] J. Haartsen, “The Bluetooth radio system”, *IEEE Personal Comms.*, vol. 7, no. 1, pp. 28-26, 2000.

[2] D. J. C. MacKay, “Fountain codes” *IEE Proceedings in Communication*, vol. 152, no. 6, pp. 1062-1068, 2005.

[3] S. Ahmad, R. Hamzaoui, M. Al-Akaidi, “Robust live unicast video streaming with rateless codes”, *Int. PacketVideo Workshop*, Nov. 2007.

[4] A. Shokorallahi, “Raptor codes”, *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551-2567, 2006.

[5] Specification of the Bluetooth System -- 2.1 + EDR, Nov. 2007, Available online at <http://www.bluetooth.com>.

[6] S. Mattisson, “Low-power considerations in the design of Bluetooth”, *Int. Symposium on Low-power Electronics and Design*, pp. 151-154, July 2000.

[7] 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.

[8] 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.

[9] 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.

[10] M. Luby, “LT codes”, *43rd Ann. IEEE Symposium on Foundations of Computer Science*, pp. 271-282, November 2002

[11] H. Kushwaha, Y. Xing, R. Chandramouli, and H. Heffer, “Reliable multimedia transmission over cognitive radio networks using fountain codes”, *IEEE Proceedings*, vol. 96, no. 1, pp. 155-165, 2008.

[12] J. Wagner, J. Chakareski, and P. Frossard, “Streaming of scalable video from multiple servers using rateless codes”, *IEEE Int. Conf. on Multimedia and Expo*, Jul. 2006.

[13] S. Jung, U. Lee, A. Chang, D.-K. Cho, and M. Gerla, “BlueTorrent: Cooperative Content Sharing for Bluetooth Users”, *IEEE Pervasive Computing and Comms.*, pp. 47-56, March 2007.

[14] M. Zorzi, and R. R. Rao, “Error control and energy consumption in communications for nomadic computing”, *IEEE Trans. On Computers*, vol. 46, no. 3, pp. 279-289, March 1997.

[15] P. Letteri, C. Fragouli, and M. B. Srivastava, “Low power error control for wireless links”, *ACM/IEEE MobiCom*, pp. 139-150, 1997.

[16] S. Zhao, Z. Xiong, and X. Wang, “Joint error control and power allocation for video transmission over CDMA networks with multiuser detection”, *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 425-437, June 2002.

[17] N. Rahnavard, B. N. Vellambi, and F. Fekri, “Efficient broadcasting via rateless coding in multihop wireless networks with local information”, *Int. Wireless Comms. and Mobile Computing Conf.*, pp. 85-90, 2007.

[18] A. W. Eckford, J. P. K. Chu, and R. S. Adve, “Low-complexity cooperative coding for sensor networks using rateless and LDGM Codes”, *IEEE Int. Conf. on Comms.*, June 2006.

[19] L.-J. Chen, T. Sun, and Y.-C. Chen, “Improving Bluetooth EDR data throughput using FEC and interleaving”, *Mobile Sensor Networks*, Hong Kong, 2006.

[20] R. Razavi, M. Fleury, E. Jammeh and M. Ghanbari, “An efficient packetization scheme for Bluetooth video transmission”, *Electronic Letters*, vol. 42, no. 20, pp. 1143-1145, 2006

[21] M. Valenti, M. Robert, and J. H. Reed, “On the throughput of Bluetooth data transmissions”, *IEEE Wireless Comms. and Networking*, pp. 119-123, March 2002.

[22] M. N. Krohn, M. J. Freedman, and M. Mazières, “On-the-fly verification of rateless erasure codes for efficient content distribution”, *IEEE Symp. on Security and Privacy*, May 2004.