Energy Efficient Video Streaming Over Bluetooth Using Rateless Coding

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

This Letter proposes energy-efficient error control for IEEE 802.15.1 (Bluetooth) video communication. The scheme is based on block-oriented incremental redundancy provided by rateless coding and receiver feedback. Results are presented for time-sensitive video streaming applications under a Markovian channel model. When the proposed algorithm is compared to variations of Bluetooth Forward Error Control, there are improvements of around 3 dB in received video quality and over 10% in energy efficiency.

Introduction: Energy efficiency is an important issue for mobile IEEE 802.15.1 (Bluetooth) devices. As data transmission typically accounts for more than a third of the total energy consumption on a mobile device [1], energy saving can be effectively achieved if unnecessary re-transmissions are avoided. Bluetooth communication is based on a simple Time Division Multiplexing scheme in which for each packet transmission, at least one time slot is reserved for the receiver's data transmission. Therefore, no systematic overhead will be introduced if a per packet acknowledgment scheme is employed to request added redundancy. In the proposed algorithm, each Bluetooth data packet is divided into multiple small blocks and a Luby Transform (LT) [2] or equivalent rateless code is applied. Consider K original blocks. With LT coding, decoding will succeed with small probability of failure if any of *K(1+ε)* blocks are received where ε refers to the coding overhead. In the event of a packet failure, additional coded blocks are generated and piggy-backed onto the next outgoing packets until the original packet can be successfully decoded. Detection of packet failure is by an adaptation of Bluetooth's user payload Cyclic Redundancy Check, without the need to pass up a packet with possible errors to the video decoder. The Letter determines in tests that compared to native Bluetooth Forward Error Control (FEC) schemes, rateless codes both improves delivered video quality in terms of Peak-Signal-to-Noise Ratio (PSNR) and energy efficiency.

Methodology: Bluetooth employs variable-sized packets up to a maximum of five frequency-hopping time-slots of 625 μs in duration. Bluetooth v. 2.1 Enhanced Data Rate (EDR) supports gross air rates of 2 Mbps and 3 Mbps, though in this Letter in tests the higher rate is selected. FEC-bearing Data Medium (DM) packets for EDR were proposed in [4]. As a point of comparison with rateless codes, both fixed and adaptive 3DM packets are employed in our simulations. Bluetooth's default FEC scheme, expurgated (15, 10) Hamming code, can cope with burst sizes of two, depending on decoder. Since employing FEC comes with a fixed overhead, an adaptive FEC strategy for Bluetooth is proposed in [5], when the FEC is enabled only under poor channel conditions.

In the proposed scheme, each packet is divided in to small blocks (15 bits herein) and LT coding is applied. LT coding can generate a continuous stream of output blocks, but *K(1+ε)* blocks are transmitted in the first try, with *ε set to 5%* in these tests. If all *K(1+ε)* blocks are received correctly , then decoding can be successfully performed with probability $1\text{-}2^{-K.\varepsilon}$ and with decoding complexity of order $K \log_e K$ (and linear complexity for some forms of rateless coding).

In the case of a block failure, signaled by a negative acknowledgement, depending on the amount of space available in the next outgoing packet, additional coded blocks are generated and piggy-backed onto the next packets until a positive acknowledgement is received or the original packet expires. The maximum number of retries allowed for each packet i.e. retransmission depth (set to 10 retransmission herein) depends on the display deadline of the packet, which is determined through the receiver playout buffer by the video startup delay. Notice that each packet may contain redundant data blocks from more than one previously failed packet. In which case, the amount of available space for redundant data is divided between different packets in proportion to their deadline margin i.e. more space will be allocated to packets closer to their expiration. Finally, a simple acknowledgment of the differing importance of frame types is made by altering the allocation for I-, P-, and B-frame packets, in this Letter for simplicity in the ratio 3:2:1 respectively.

In a previous correspondence [3], it was shown that the Bluetooth throughput is dependent on packet size. Although by re-packetizing the video stream data into fully-filled Bluetooth packets (herein using five slots) the slice synchronization markers may be lost, the gain from the throughput improvement by fully utilizing the channel capacity is superior. However, when, depending on the wireless channel status, some redundant data may need to be piggy-backed upon each packet, an algorithm should adapt the space in the packet allocated for redundant blocks. If the packets are nearly full with video data, there might not be enough space for the redundant blocks needed to recover previously failed packets, which consequently results in a high packet loss rate. Therefore, in the tests, the space for a minimum and maximum number of redundant blocks was set to 5 and 50 blocks, with a setting of 5 piggy-backed blocks once transmission was under way.

The space was then adapted according to the channel conditions, allowing more room for piggy-backed blocks in poor channel conditions. Here, the allocated space for piggy-backed blocks was reduced by one block upon reception of every *n* consecutive successful packets (100 herein) and was multiplicative factor (1.5 herein) once a packet recovery fails.

A Gilbert-Elliott two-state, discrete-time Markov chain modeled realistic 'bursty' channel error characteristics between a Bluetooth master and slave node. The mean duration of a good state, T_g, was set at 2 s and in a bad state, Tb, was set to $a \times T_g$ where *a*, the channel parameter, refers to the ratio of bad and good state durations. In units of 625 μs (the time-slot duration), T_g= 3200.

The Bit Error Rate (BER) during a good state was set to 10^{-5} and during a bad state to 10^{-3}. The BER values are approximately similar to those in [6], but the mean state durations are adapted to Bluetooth. As input, a European-formatted SIF-sized MPEG-2 video clip at 25 frame/s, with Group-of-Picture structure of N=12, M=3 for duration of 40 s, was encoded at an average rate of 1.5 Mb/s. The clip contained moderate motion and, hence, a moderate bitrate for the given quality and size.

This research employed the University of Cincinatti Bluetooth (UCBT) extension to the well-known ns-2 network simulator (v. 2.29 used).

Results: Fig.1 shows the packet loss performance of different FEC strategies under varying channel conditions. The packet loss rate is minimal in the proposed scheme while the no FEC policy results in the highest packet loss. The performance of the fixed and adaptive FEC schemes starts to converge when the duration of the bad channel states increases. Alternatively, Fig.2 compares the quality of the received video under the three FEC-enabled schemes while the 'no FEC' policy results in unacceptable video quality and is not presented. Again the proposed algorithm provides the highest video quality. The fixed FEC scheme is superior to the adaptive one but this comes at the cost of extra overhead and energy consumption. Energy efficiency is defined by us as the ratio of successfully received bits divided by the total number of transmitted bits. This metric is calculated and illustrated in Fig.3 for different schemes under varying channel conditions, showing the dominance of the proposed algorithm compared to other schemes.

Conclusion: Block-based coding as introduced in this Letter allows additional redundant blocks rather than packets to be generated, scaling the overhead that occurs. In the Bluetooth example in which acknowledgement latency is conveniently low, energy efficiency was increased by over 10% by rateless coding compared to other low-complexity FEC-bearing schemes, while video quality is raised by about 3 dB.

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

References

1 Chen, J-C, Sivalingam, K. M., Agrawal, P., and Kishore, S..: ‘A comparison of MAC protocols for wireless local networks based on battery power consumption’, IEEE INFOCOM, vol. 1, pp. 150-157, March 1998

2 Luby, M.: ‘LT codes‘, 43rd Ann. IEEE Symposium on Foundations of Computer Science, pp. 271-282, November 2002

3 Razavi, R., Fleury, M., Jammeh, E. and Ghanbari, M.: ‘An efficient packetization scheme for Bluetooth video transmission‘, Electronic Letters, vol. 42, no. 20, pp. 1143-1145, 2006

4 Chen, L.-J., Sun, T. and Chen, Y.-C.: ‘Improving Bluetooth EDR data throughput using FEC and interleaving‘, Mobile Sensor Networks, LNCS 4325, pp. 724-735, Hong-Kong, 2006

5 Das, A., Gose, A., Gupta, V., Razdan, A., Saran, H., and Shorey, R.: ‘Adaptive Link-level Error Recovery Mechanisms in Bluetooth’ IEEE International Conference on Personal Wireless Communications, pp.85 -89, Hyderabad, December, 2000

6 Fantacci, R. and Scardi, M.: ‘Performance evaluation of preemptive polling schemes and ARQ techniques for indoor wireless networks’, IEEE Trans. on Vehicular Tech., vol. 45, no. 2, pp. 248-257, 1996.

Authors’ affiliations:

R. Razavi, M. Fleury and M. Ghanbari (Department of Computing and Electronic Systems, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, United Kingdom)

Corresponding author:

M. Fleury, fleum@essex.ac.uk

Figure captions:

Fig. 1 Packet loss performance of different error recovery schemes under varying channel conditions with a maximum of ten retransmissions

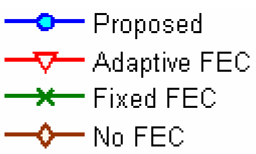


Fig. 2 Comparison of the received video quality for various FEC-bearing schemes under different channel conditions with a maximum of ten retransmissions

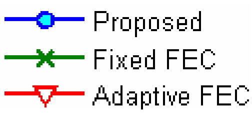


Fig. 3 Energy efficiency comparison of different error recovery schemes with a maximum of ten retransmissions

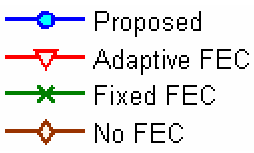
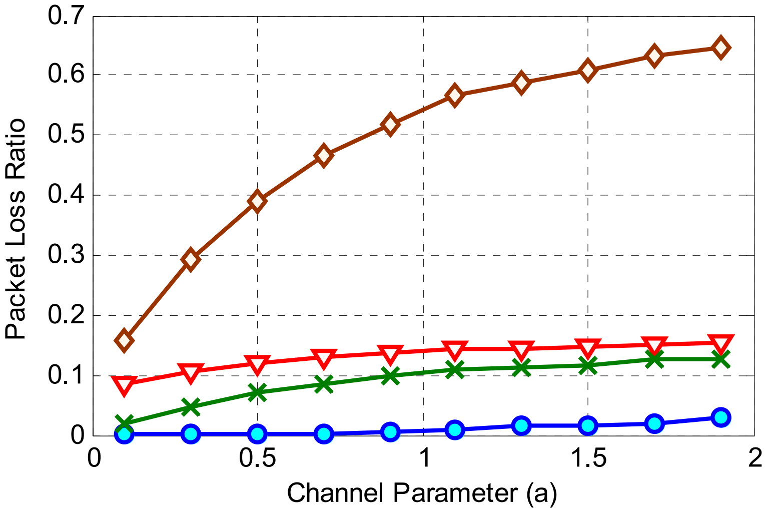


Figure 1



0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Packet Loss Ratio
0
0.5
1
1.5
2
Channel Parameter (a)

Figure 2

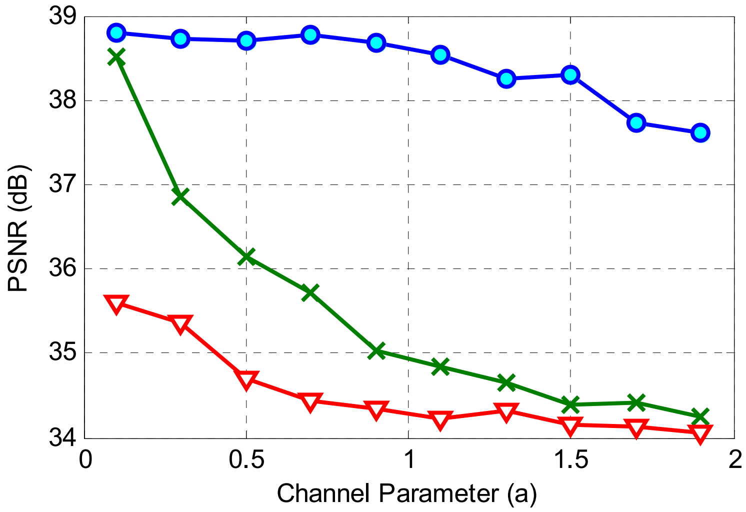
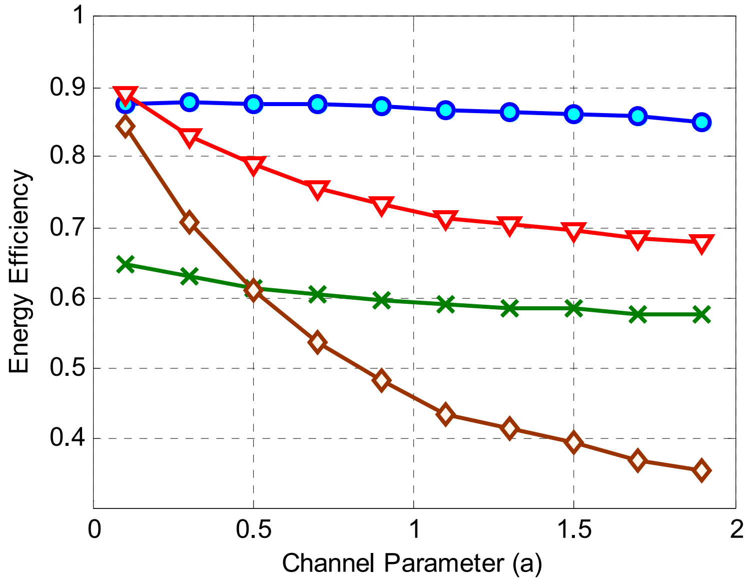


Figure 3



1
0.9
0.8
0.7
0.6
0.5
0.4
Energy Efficiency
0
0.5
1
1.5
2
Channel Parameter (a)