Detecting Congestion within a Bluetooth Piconet: Video Streaming Response

R.Razavi, M. Fleury, and M. Ghanbari
University of Essex

Abstract: In a Bluetooth piconet, congestion may arise as multiple flows pass via the master node across the shared wireless channel. While packet loss and delay are common measures of congestion, within a piconet the transmitter buffer fullness is a more suitable metric, as it more directly indicates congestion and the onset of congestion. For video traffic, simulation results clearly show the advantage of buffer fullness as the congestion indicator compared to other possible metrics, in terms of delivered video quality.

1 Introduction.

In general, before congestion control can be put in place, a way of detecting (or predicting) congestion should be available. A Bluetooth (B/T) piconet [1] master node employs centralized scheduling of packets onto a wireless channel shared between the master and up to seven slave nodes arranged in a star topology. Access to the channel is on a Time Division Duplex (TDD) basis with the central master polling slaves for packets. Though in B/T v. 2.0 Enhanced Data Rate (EDR) was introduced in late 2004, congestion control remains an issue. Self-congestion will occur if packets accumulate at a slave's transmit buffer without being serviced by the master. Cross-congestion will occur if a master receives packets destined for another slave, while the same buffer also acts as a transmit buffer for direct communication by the master to the same destination. Under EDR, the maximum user bandwidth capacity has risen to 2.178 Mb/s but for multimedia applications saturation can still easily occur. As scheduling is polling-based, when a new slave enters the piconet, available bandwidth to other slaves suddenly decreases, as these slaves are visited less frequently. For delay-sensitive applications, such as lower bitrate video, packet loss through buffer overflow reduces the delivered quality, which cannot easily be restored by re-transmission.

2. Bluetooth (B/T) congestion control.

In B/T, there is no direct slave-slave communication and, therefore, a master maintains separate queues for each master to slave link, Fig. 1. The standard does not specify the queue service discipline, and, along with B/T implementations, this paper assumes pure round-robin (1-limited) scheduling. The work in [2] showed that 1-limited servicing performed better under high load than an exhaustive queue discipline and in [3] it has been surprisingly demonstrated that, in symmetrical piconets with only uplink traffic, the mean waiting time is the same for both exhaustive and limited disciplines.

In [4], packet loss is the metric for congestion control in a B/T piconet. The authors of [4] suggest increasing the video sending rate (as their main traffic source) additively if the packet loss rate is less than 5% and reducing the rate if the loss rate exceeds 15%. (No action is taken when the loss rate is between 5 and 15%.) There are two main problems in regard to loss-based control. Firstly, there is no concept of congestion avoidance. Secondly, it is not always clear (especially in the case of wireless networks) whether the loss is due to congestion or not.

Delay-based congestion control is a general technique that has been widely adopted, *e.g.* [5], though apparently with no specific applications to B/T. Because there is a lag between the occurrence of congestion and the system response, delay-based control is also unable to prevent congestion altogether. For example, in a piconet consider the action of a transmitting slave when the available bandwidth is reduced. The transmitting slave is not notified of the change in the topology until after its buffer has already started filling up. Subsequently, packets that have suffered a long wait are transmitted. Then, based on feedback from the receiver, the transmitter is notified that there is something wrong and it starts reducing its rate. However, even reducing the rate at this time does not have an effect immediately, as there are already some packets in the buffer that have suffered long waits in the buffer queue. In fact, it is possible for rate oscillations to occur, as a buffer is successively drained and then fills up again.

Though both loss-based and delay-based congestion control clearly have their place in complex network graphs, in a B/T piconet with only single and double hops, inspection of transmitter buffer fullness has a more direct effect. Whereas, in the former two the receiver notifies the sender with a feedback message, buffer fullness is established at the sender. Therefore, the other two methods suffer from a delayed response from the receiver. Moreover, buffer-fullness indicates to the transmitter both self-congestion and cross-congestion. Finally, the 1-limited queue discipline with multiple queues sharply reduces the risk of buffer overflow at the master, as the master only collects one packet from each slave queue before servicing the next queue.

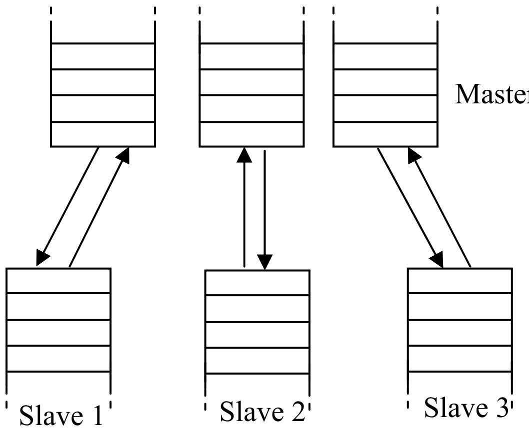

Figure 1: The queuing model for Bluetooth.

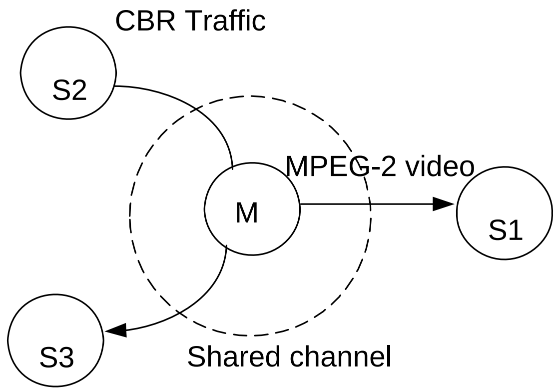

Figure 2: The simulation scenario for congestion control.

3. Experimental methodology.

This research employed the University of Cincinatti B/T (UCB/T) extension[1] to the well-known ns-2 network simulator (v. 2.28 used). The UCB/T extension supports B/T EDR but is also built on the air models of previous B/T extensions such as BlueHoc from IBM and Blueware. All links were set at the EDR 3.0 Mbps gross air rate.

In the simulations scenario, Fig. 2, a single master to slave asynchronous connection-less (ACL) link was established for transferring the main traffic source which was a 40 s video clip of a news bulletin, being European-formatted SIF-sized at 25 fps, MPEG-2 encoded (GOP structure n=12, m=3) at rate 1.77 Mbps. Six seconds after the start of this flow, another slave starts transmitting packets to another slave across ACL links. This cross traffic is a constant-bit rate (CBR) flow with a rate of 500 Kbps and a fixed packet size of 800 B. The cross traffic stops at 36s (4 seconds before the end of the video clip). When CBR traffic starts, the video traffic should reduce its rate to avoid congestion (packet loss). Only the video traffic is under congestion control. The video packets were the largest 3-DH5 size (payload 1021 B) and were fully-utilized through dynamic packing of MPEG-2 slices.

In the buffer fullness, delay, and loss-based control simulations, the change in the video rate all followed the Additive Increase Multiplicative Decrease (AIMD) rule given in (1). The AIMD controller was simply chosen as a convenient point of comparison; other controllers may well be superior.

$$\begin{aligned} I &: R \leftarrow R + \alpha \quad : \alpha > 0 \\ D &: R \leftarrow (1-\beta).R : \quad 0 < \beta < 1, \end{aligned} \tag{1}$$

where *I* refers to the increase in rate when the network is estimated to be under-loaded and *D* refers to the decrease in rate on detection of congestion. In the experiments, α was considered to be 0.05 (5% increase) and for β, 0.1 was considered, which is similar to the scheme in [4]. The transmitter buffer size for all scenarios was assumed to be 50 packets. Since this work concentrates on isolating the impact of congestion control metrics and not RF noise, the channel was considered to be error free.

---

[1] Download is available from `http://www.ececs.uc.edu/~cdmc/ucB/T/`.

In buffer-fullness-based control, a buffer was checked every 72 packets, corresponding to a congestion window of 72 packets. Two threshold values were defined: if the buffer contained less than 20 packets (40% of the total capacity) the rate increase procedure was executed. On the other hand, if the buffer had more than 40 packets, the rate decrease procedure was called. Delay-based control also used a 72 packet congestion window, this size reducing excessive rate oscillations that arise if a much smaller window size is chosen. If the delay was higher that 0.45 s the rate was decreased, while the rate was increased if the delay was less that 0.25 s. In loss-based control, the scheme already described in [4] was adopted, again with a congestion window of 72 packets. Finally, for comparison purposes, the encoded video source was transmitted with no change in rate.

The video sending rate was changed by means of a transcoder. The congestion controller applies a signal (between 0 and 1) to the transcoder, which then alters its re-quantization level. As a measure of delivered quality at the receiver, the Peak Signal-to-Noise Ratio (PSNR) was taken after passing the received video frames through a standard decoder. As start-up delay (and packet arrival jitter) requires increased buffering, which can result in power loss from increased memory usage on the mobile receiver device, this is also reported. Start-up delay above a few ms is noticeable.

4. Results.

Fig. 3 shows that without control the packet loss makes PSNR (measured in dB) is well below an acceptable level (34-38 dB) and would probably be unwatchable. Loss-based control also suffers from poor quality received video, once the cross traffic starts. There is also a series of peaks in the response, when the controller manages to briefly restore the rate (when the loss rate falls below the lower threshold) before more packet losses quickly cause the rate to be reduced again. In buffer fullness control, there is a slight lag before a somewhat reduced delivered PSNR is delivered. However, the response lag is especially marked for delay-based control, along with repeated drops in quality. Fluctuations in delivered quality are known to increase the subjective viewing experience due to temporal masking.

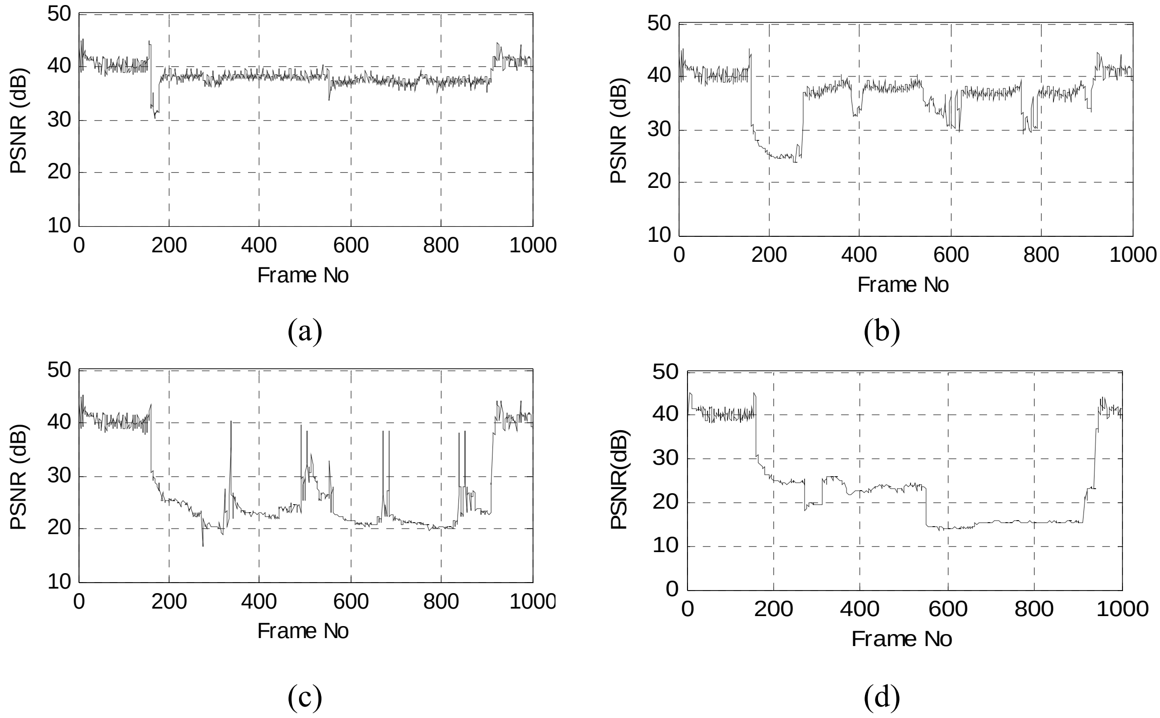

Figure 3: PSNR when the congestion indication metric is: a) transmitter's buffer fullness, b) packets' delay, c) packet loss rate and d) no control.

In Fig. 4, the packet delay response is reported. As loss-based control does not consider delay, its delay response is similar to that when no control is present. The delay-based controller produced the oscillatory response of Fig. 4 (b), as the rate alternates between the two threshold delay levels. The alternations of delay level may result in packets not meeting display deadlines, if adequate buffering is not in place. Buffer fullness-based control is more successful at reducing delay than the other schemes. Summary results, which are for the whole 1000 frames, are reported in Table 1, in which it can be seen that buffer-fullness response is superior to either loss-based or delay-based control.

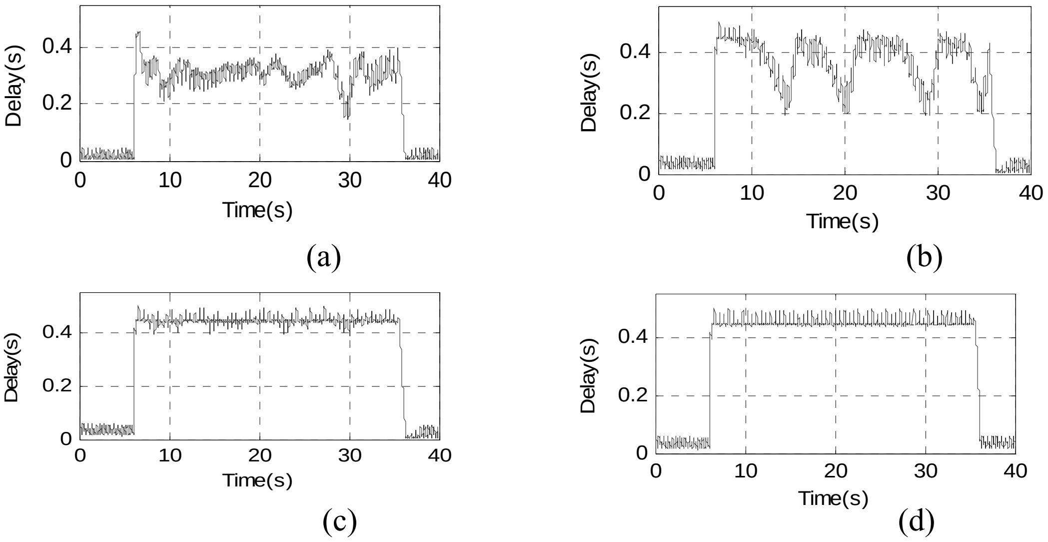

Figure 4: Packet delay when the congestion indication metric is: a) transmitter's buffer fullness, b) packets' delay, c) packet loss rate and d) no control

Congestion Indication	Mean PSNR (dB)	Mean Delay (s)	Mean No. of Buffered Packets	Packet Loss Rate	Mean Transcoding Rate
Transmitter's Buffer Fullness	38.45	0.1865	21.41	0.0026	0.582
Delay	36.43	0.2254	27.35	0.065	0.603
Loss	27.79	0.2665	33.05	0.256	0.710
No Control	24.33	0.2686	38.64	0.498	N/A

Table 1: Performance of thresholded AIMD controller with differing congestion indicators

7. Conclusion.

Though packet-loss-based congestion control has been advocated by others and delay-based control is widely used, it turns out that these forms of control are simply too indirect when simple B/T piconets are considered. By comparing the same AIMD congestion controller with different control metrics, it is clear that the buffer fullness metric is more direct, as it significantly increases delivered video quality and more generally reduces the packet loss rate.

References.

[1] J. Hartsen, "The Bluetooth radio system", IEEE Personal Comms., vol. 7, no. 1, pp. 28-36, 2000.

[2] Y.-Z. Lee, R. Kapoor, and M. Gerla, "An efficient and fair polling scheme for Bluetooth," in Proceedings MILCOM 2002, vol. 2, pp. 1062–1068, 2002.

[3] G. Zussman, A. Segall, and U. Yechiali, "Bluetooth Time Division Duplex - Analysis as a polling system", Proc. 1st IEEE Conf. on Sensor and Ad Hoc Comms. and Networks (SECON'04), Oct. 2004.

[4] R. Kapoor, M. Kazantzidis, M. Gerla, and P. Johansson, "Multimedia support over Bluetooth piconets", in Proc. of 1st Workshop on Wireless Mobile Internet, pp. 50 – 55, Rome, 2001.

[5] C. J. Wei, D.X. Low, S. H. Caltech, "The case for delay-based congestion control", in 18th IEEE Workshop on Computer Communications, pp. 99-104, 2003.