

Calculation of probabilistic worst-case response-time for message in bluetooth

V. Umadevi Chezhian¹, K. Ramar² and S. Geetha³

¹ Dept of Computer Science, College of Business and Economics, Asmara, State of Eritrea, East-Africa

² Dept of Computer Science Engineering, National Engineering College, Kovilpatty, India

³ Dept of MCA, Anna University, Trichrapalli, India

yazh_1999@yahoo.co.in; kramar@rediffmail.com; kasagee1971@yahoo.co.in

Abstract

The response time is one of the main constraints in bluetooth synchronization and message transfer. This paper provides a novel approach for calculating a probabilistic worst-case response-time for messages in the bluetooth. We are using frequency hopping mechanism in order to retrieve the useful distribution of pockets instead of the worst-case scenario in bluetooth communication. Since the bluetooth systems are not synchronized, the number of bluetooth hops which it overlaps will depend not just on the 802.11 packet's length but also on the time offset between the packet and the hops. The best option to avoid the overlap problem is the time period. This analysis was concerned only with the timing relationships between an 802.11 packet and the bluetooth hops.

Keywords: Bluetooth synchronization, worst-case, response-time, Overlap, bluetooth hops.

Introduction

The technology allows users to form wireless connections between various communication devices in order to transmit real-time voice and data. The bluetooth radio is built into a small microchip and operates in the 2.4 GHz band, a globally available frequency band ensuring communication compatibility worldwide. It uses frequency hopping spread spectrum, which changes its signal 1600 times per sec and helps to avoid interception by unauthorized persons.

We investigate the use of adaptive frequency hopping (AFH) technologies aimed at modifying the bluetooth frequency hop sequence. We examined the condition to improve the performance and time delay of the bluetooth message. The AFH brings the delay down to the same level than when no interference is present. This leads to overhead. The main difficulty for AFH is to dynamically communicate the changes to all slaves in the piconet in order to keep the synchronization (Golmie *et al.*, 2003).

The adaptive part of AFH is constrained by the channel estimation and how often the devices in the piconet need to synchronize. That determines the response time and the performance. Asymptotically, the bluetooth frequency hop sequence is uniformly distributed. Bluetooth uses a slow hop frequency hopping spread spectrum scheme with a 79 frequency slots scheme in the 2.4 FHz band, each with 1 MHz (Brian S. Peterson *et al.*, 2003).

Some modifications for the standard bluetooth protocol and chip-set required for real-time embedded systems are discussed in this paper. To improve performance, a technique known as adaptive frequency hopping has been introduced by the bluetooth special interest group (SIG) to diminish the impact of interference which the IEEE 802.15.2 coexistence task group and the

bluetooth special interest group (SIG) are looking at techniques for alleviating the impact of interference between IEEE 802.11b and bluetooth devices. Bluetooth specification being adopted in 2003 includes adaptive frequency hopping, a technique proven to be an effective remedy to the problem of interference in WLAN and similar environments (Brian S. Peterson *et al.*, 2003).

Reliability and timing analysis of distributed systems are well known. Now the real-time scheduling theory has targeted hard real-time systems. Typically, these systems can be found as small-embedded applications, operating in a controlled environment. Since these systems are not allowed to have any missed deadlines, we need precise timing analysis taking all possible scenarios of execution environment and execution times into consideration.

We analyzed the paper "Bluetooth adaptive frequency hopping and scheduling" by Golmie *et al.* (2003) which investigate the use of adaptive frequency hopping (AFH) techniques aimed at modifying the bluetooth frequency hopping sequence in the presence of WLAN direct sequence spread spectrum devices. We examined the conditions such as the applications, topologies and scenarios under which AFH techniques improve performance that is measured in terms of packet loss, TCP delay and channel efficiency. We also compare the results obtained with AFH to others obtained using a scheduling technique that consist in delaying the transmission of a bluetooth packet until the medium is 'idle'. The results show that an obvious performance improvement with AFH is in terms of delay and throughput. AFH brings the delay down to the same level than when no interference is present. On the other hand, AFH is rather slow in responding to changes in the environment and the packet loss is more significant than with the scheduling. This is probably due to the limitations

Fig. 1. Transmission frequency range

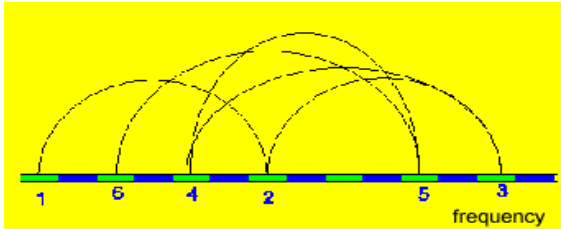
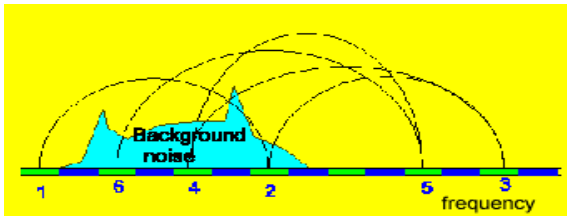


Fig. 2. Transmission with noise frequency range



imposed by the communication overhead. The difficulty for AFH is to dynamically communicate the changes to all slaves in the piconet to keep the synchronization.

The objective of our study includes: Analyzing and introducing the maximum response time in real-time embedded system; Reduce the time delay and loss less reception of the messages in real-time embedded system; To test the system with realistic estimates and constraints; To comply with available regulatory requirements in the conduct of Bluetooth and Simple performance and schedulability analysis to demonstrate that a bluetooth based system successfully meets real-time constraints.

The frequency-hopping schemes

Bluetooth has been designed to operate in noisy radio frequency environments, and uses a fast acknowledgement and frequency-hopping scheme to make the link robust, communication-wise. Bluetooth radio modules avoid interference from other signals by hopping to a new frequency after transmitting or receiving a packet.

Compared with other systems operating in the same frequency band, the bluetooth radio typically hops faster and uses shorter packets. This is because, short packages and fast hopping limit the impact of microwave ovens and other sources of disturbances. Use of forward error correction (FEC) limits the impact of random noise on long-distance links (Thomas *et al.*, 2003).

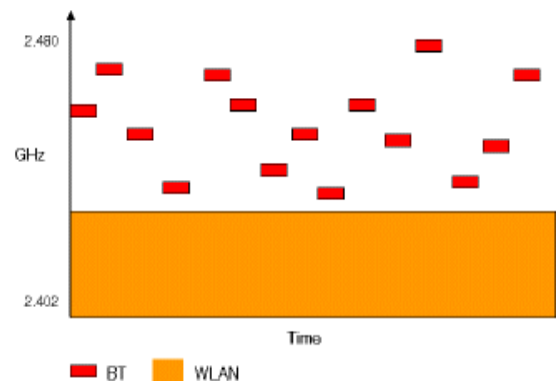
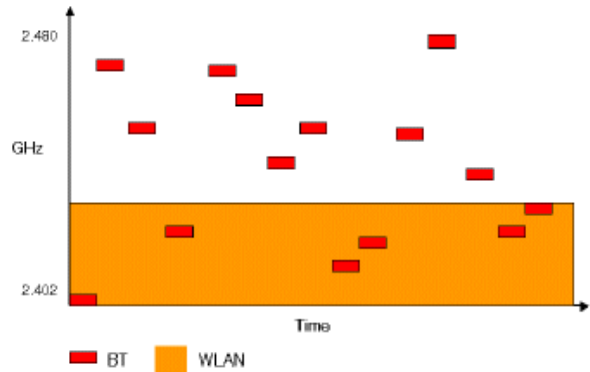
Hopping out and into a continuous range of frequencies that are subject to noise gives the communication link a better chance to remedy transmission errors when out of the disturbed frequencies, than would have been the case if the transmission had stayed for several timeslots within the noisy frequency range (Core Version 1.1. Bluetooth SIG). This argument does not hold true if the noise is spread over the whole frequency range in a random fashion, but

sources of electro-magnetic noise usually give the pattern illustrated in Fig 1 and 2.

Whenever a connection (a 'piconet') is first established between 2 (or more) units, the master unit establishes a frequency hopping scheme which is communicated to the other units. This frequency selection scheme consists of two parts: 1) Selecting a sequence and 2) Mapping this sequence onto the hop frequencies. The IEEE 802.11 standard was approved in 1997. Ten types of frequency hopping sequences are defined- 5 for the 79-hop and 5 for the 23-hop system. These sequences are:

- A page hopping sequence with 32 (16) unique wake-up frequencies distributed equally over the 79 (23) MHz, with a period length of 32 (16).
- A page response sequence covering 32 (16) unique response frequencies that are all in a one-to-one correspondence to the current hopping sequence. Master and slave use different rules to obtain the same sequence.
- An inquiry sequence with 32 (16) unique wake-up frequencies distributed equally over the 79 (23) MHz with a period length of 32 (16).
- An inquiry response sequence covering 32 (16) unique response frequencies that are all in a

Fig. 3. Collisions resulting from random frequency hopping adapting to the environment



one-to-one correspondence to the current inquiry hopping sequence.

- A channel hopping sequence with very long period, which does not show repetitive patterns over a short time interval, but which distributes the hop frequencies equally.

How AFH works?

Bluetooth products developed prior to the emergence of working AFH solutions employ another form of frequency hopping which is random by design. These first generation bluetooth devices use 79 of the 83.5 available channels in the 2.4 GHz band hopping across these channels in a random fashion and at a rate of 1600 times per sec. As soon as another wireless device is introduced into the environment this type of hopping results in occasional collisions. There is no AFH bluetooth lacks the ability to avoid these conflicts and thus to adapt to its environment. The result is illustrated in Fig. 3 showing an environment where both bluetooth (BT) and wireless LAN (WLAN) are in operation.

Adaptive frequency hopping allows bluetooth to adapt to the environment by identifying fixed sources of interference and excluding them from the list of available channels. This process of re-mapping also involves reducing the number of channels to be used by bluetooth. The bluetooth specification requires a minimum set of at least twenty channels. Fig. 4 shows the same environment as illustrated above but now with adaptive frequency hopping activated.

Worst case response time

We focus on the bluetooth wireless network analyzing its ability to support the quality of service (QoS) requirements defined by QoS sensitive applications. First we establish a bound on the worst case response time of any sporadic message using a worst case analysis. We then compare the computed bounds with the worst case response times obtained by simulation for different traffic distributions and show in which configurations our bounds are tight. We finally compare those different scheduling and show how to extend our results on worst case response times taking into account message segmentation (Antoine Mercier *et al.*, 2006a,b). In this paper we have analyzed the minimum and maximum number of hops overlapped. The actual number of hops that the packet overlaps depends upon the relative timing of the start of the packet and the hop.

The problem

Similar to 802.11b, bluetooth devices operate within the 2.4 GHz band. The difference is that that bluetooth uses frequency hopping (at 1,600 hops per sec) to hop over the entire 2.4 GHz band. 802.11b, on the other hand, uses direct sequence and only occupies approximately one third of the 2.4 GHz band. As a result, bluetooth hops all over 802.11b transmissions.

An 802.11 station (client or access point) is polite and

first listens to the medium before transmitting. If the 802.11 station doesn't sense RF energy above a certain threshold (meaning that the medium is idle), the 802.11 station can transmit a frame. While the 802.11 station is sending the frame, other 802.11 stations will hold off their transmissions by following the same protocol. This provides a fairly good method of sharing a common RF channel among devices complying with the 802.11 standard.

A critical problem is that bluetooth and 802.11b neither understand each other nor follow the same rules. A bluetooth radio may haphazardly begin transmitting data while an 802.11 station is sending a frame. This results in a collision, which forces the 802.11 station to retransmit the frame when it realizes that the receiving station is not going to send back an acknowledgement. This lack of coordination is the basis for RF interference between bluetooth and 802.11.

Calculation

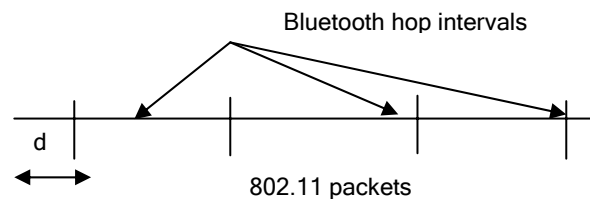
Since the bluetooth systems are not synchronized, the number of bluetooth hops which it overlaps will depend not just on the 802.11 packet's length but also on the time offset between the packet and the hops. Let H be the duration of a bluetooth hop (typically 625 microsec) and consider an 802.11 packet of duration L. Then the minimum number of hops which the 802.11 packet overlaps is

$$\lceil L/H \rceil \dots (1)$$

and the maximum number of hops overlapped is

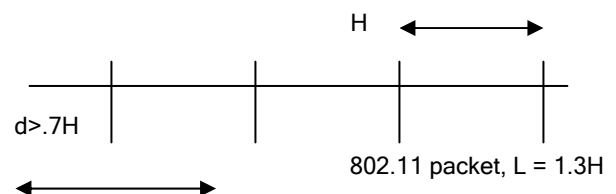
$$\lceil L/H \rceil + 1 \dots (2)$$

Let d be the "delta" between the last Bluetooth hop and the start of the packet as indicated in the following timing figure:



Note that $0 < d \leq H$. If d is zero, then the 802.11 overlaps with its minimum number of bluetooth dwell periods, namely $\lceil L/H \rceil$. If d is greater than $\lceil L/H \rceil * H - L$ then the 802.11 packet overlaps with $\lceil L/H \rceil + 1$ Bluetooth hops. This can be readily seen by considering an example: say $L = 1.3 H$. Then $\lceil L/H \rceil$ is 2, so $\lceil L/H \rceil * H - L = 2H - L = .7H \dots (3)$

And if $d > .7H$ we have the following figure showing that the packet overlaps 3 Bluetooth dwell periods.



Consequently, an 802.11 packet will overlap $\lceil L/H \rceil$ bluetooth dwell periods whenever

$$0 < d \leq \lceil L/H \rceil * H - L \quad \dots (4)$$

And will overlap $\lceil L/H \rceil + 1$ dwell periods whenever

$$\lceil L/H \rceil * H - L < d \leq H \quad \dots (5)$$

This translates into probabilities by expressing these intervals as fractions of the interval $[0, H]$, yielding the following: The probability that an 802.11 packet of duration L will overlap with $\lceil L/H \rceil$ bluetooth dwell periods of duration H is

$$\lceil L/H \rceil - L/H \quad \dots (6)$$

The probability that it overlaps with $\lceil L/H \rceil + 1$ dwell periods is

$$1 - \lceil L/H \rceil + L/H \quad \dots (7)$$

So the other option to avoid the overlap problem is the time period. To get better time response you select high stability crystal oscillator based on the above calculation.

Factoring in the bluetooth frequency channels

The above analysis was concerned only with the timing relationships between an 802.11 packet and the bluetooth hops. Now we add to this analysis the actual frequency hopping characteristics of bluetooth and the specific frequency channelization of the 802.11 direct sequence standards.

We can now derive a formula that expresses the probability of a bluetooth-induced collision for a given 802.11 DS packet of duration L on a given wideband DS channel. When this packet's timing relationship with bluetooth is such that it overlaps with N bluetooth dwell periods, the probability that a bluetooth transmitter is never on a narrowband channel within the wideband DS channel for those N dwell periods is $(2/3)^N$. From our earlier calculation of probabilities for the number of bluetooth dwell periods overlapped by a packet we get the following:

The probability that an 802.11 packet of duration L experiences no bluetooth collisions is

$$(2/3)^{\lceil L/H \rceil} (\lceil L/H \rceil - L/H) + (2/3)^{(\lceil L/H \rceil + 1)} (1 - \lceil L/H \rceil + L/H) \dots (8)$$

The corresponding probability of a bluetooth induced collision is thus

$$1 - (2/3)^{\lceil L/H \rceil} (\lceil L/H \rceil - L/H) + (2/3)^{(\lceil L/H \rceil + 1)} (1 - \lceil L/H \rceil + L/H) \dots (9)$$

These formulas assume a fully-utilized bluetooth picocell.

This can be understood as follows: the percentage of packets that overlap with $\lceil L/H \rceil$ dwell periods is $\lceil L/H \rceil - L/H$, and the probability of no collision for such packets is $(2/3)^{\lceil L/H \rceil}$. Similarly, the percentage of packets that overlap with $\lceil L/H \rceil + 1$ dwell periods is $1 - \lceil L/H \rceil + L/H$, and the probability of no collision for such packets is $(2/3)^{\lceil L/H \rceil + 1}$. For all packets, the probability of no collision is therefore given by the weighted sum as expressed in the above boxed formula.

The effect of partial utilization within a bluetooth picocell can be easily accommodated by simply multiplying the above collision probability by the picocell's percentage utilization.

Conclusion

The number of bluetooth hops which it overlaps will depend not just on the 802.11 packet's length but also on the time offset between the packet and the hops. The actual number of hops that the packet overlaps depends upon the relative timing of the start of the packet and the hop. The time period is the option to avoid the overlap problem. The first calculation analysis in this paper was concerned only with the timing relationships between an 802.11 packet and the bluetooth hops. So we find out another calculation that factorizing in the bluetooth frequency channels, and find out the effect of partial utilization within a bluetooth picocell. It can be easily accommodated by simply multiplying the above collision probability by the picocell's percentage utilization. We can get the loss less reception of messages in bluetooth.

References

1. Antoine Mercier, Pascale Minet and Laurent George (2006a) Worst case response times in a bluetooth piconet: scheduling combining 1-RR or E-RR with FIFO or EDF. ICN/ICONS/MCL 2006, 135.
2. Antoine Mercier, Pascale Minet and Laurent George (2006b) Worst case response times for 1-RR/FIFO and 1-RR/EDF in a Bluetooth piconet. International Conference on Networking, *Intl. Conf. on Systems and Intl. Conf. on Mobile Commun. and Learning Technol.* (ICNICONSMCL'06).232.
3. Brian S. Peterson, Rusty O. Baldwin, Jeffrey P. Kharoufeh, and Richard A. Raines (2003) Refinements to the packet error rate upper bound for bluetooth networks. *IEEE Commun. Lett.* 7 (8), 382-384.
4. Charles Hodgdon (2003) *Adaptive Frequency Hopping for Reduced Interference between Bluetooth® and Wireless LAN*. Ericsson Technology Licensing.
5. Golmie N, Rebala O and Chevrollier N (2003) Bluetooth adaptive frequency hopping and scheduling. In: *Proc. of the IEEE Military Commun. Conf.* (MILCOM), Oct.13-16. 2, 1138-1142.
6. Thomas Nolte, Hans Hansson and Christer Norström (2003) Probabilistic worst-case response-time analysis for the controller area network. *Proc. of the 9th IEEE Real-Time and Embedded Technology and Applications Symp.* (RTAS'03). IEEE Computer Soc., Washington, DC (relocated from Toronto), USA. May, pp: 200-207.