Analysis of UDP Performance over Bluetooth

Martin Connolly, Cormac J. Sreenan
University College Cork
Department of Computer Science
Email: cjs@cs.ucc.ie

Abstract

The Bluetooth protocol is one of the better-known technologies in the field of pervasive computing and is now beginning to emerge in the consumer electronics marketplace in devices such as digital cameras, mobile phones and laptops. This paper examines the performance of the Bluetooth protocol using a simulator developed as part of the University of Berkeley's Network Simulator (*ns*) [1]. In particular, the performance of the UDP transport protocol when Bluetooth is used as the bearer carrier is examined.

1 Introduction

The main aim of the Bluetooth Protocol is to eliminate hard connections between devices. Instead, devices are connected to each other via a universal radio link in the Industrial Scientific and Medical (ISM) Band. Bluetooth is a single low-power wireless communication chip that, when embedded within a device, allows that device to communicate with other Bluetooth-enabled devices without the need for cables. The protocol is a Radio Frequency (RF) specification that facilitates short-range point-to-multipoint voice and data transfer.

The Bluetooth protocol can be used to connect almost any device to another device. A typical example is the wireless linking of a laptop to a printer. Once two devices have established a connection with each other the resulting ad-hoc network is referred to as a *piconet*. There are two distinct roles in a piconet - the Bluetooth device that initiates the connection between the two devices is referred to as the master device while the device with which it establishes the connection is referred to as the slave device. A slave in one piconet can act as the master or slave in another piconet. This concept of overlapping piconets is termed a *scatternet*.

The lowest layer in the Bluetooth protocol stack is the radio system which is responsible for moving information between two devices connected by a single physical link. The Baseband layer defines the process by which devices search for each other and establish connections for data transfer. The Baseband layer also defines the master and slave roles for devices, the device initiating the connection being the master while the device responding to the connection request becomes the slave. Communication security and bandwidth negotiation is a critical feature of any data transport protocol. The Link Manager Protocol (LMP) manages this functionality. The LMP also reserves a certain Quality of Service (QoS) for data traffic. This traffic is routed through the Logical Link Control and Adaptation Protocol (L2CAP), which manages data traffic through segmentation and reassembly functionality and protocol multiplexing to allow several different protocols and applications to share the air-interface.

The above is a brief introduction to the components of the Bluetooth protocol stack. [2] provides the details of the Bluetooth protocol specification in depth while [3] describes its profiles and use cases.

One of the key areas of interest for a bearer protocol such as Bluetooth is the behaviour of transport layer protocols that use it. In particular, the distinguishing features of the bearer protocol may influence the behaviour of the transport layer. This paper investigates how the performance of the UDP transport protocol is influenced by the use of Bluetooth as its bearer. Any anomalies in performance found may be significant given the role UDP plays in several of the Bluetooth use cases and profiles, for example the LAN Access Profile.

Section 2 of this paper will now discuss research that has been carried out for the Bluetooth protocol. Section 3 will briefly describe the simulator and experiment methodology used. Section 4 will then discuss UDP performance, in particular the effect IP packet size and Baseband inquiry interval has upon the behaviour of the transport protocol. Section 5 concludes the paper.

2 Related Research

The most basic area of interest for any protocol is relates to its performance. [4] shows that link throughput is significantly degraded by interference if a very large number of interfering piconets exist. While this interference varies depending on the packet type – DM1 and DH1 being the most robust – degradation is present for all packet types. However, as shown in [5], if piconets are spaced in the correct way up to 20 equally spaced piconets can be supported in a medium to large size room.

Aside from interference the other major area of interest in assessing the performance of the Bluetooth protocol is to examine the throughput of various transport and application layer protocols using Bluetooth as a bearer protocol. Of course, this is one of the purposes of this paper but for now it is worthwhile to refer to [6] which examines the performance of TCP/IP over Bluetooth. The Bluetooth protocol utilises an ARQ system, which means that the Bluetooth Link Layer automatically retransmits packets that have been lost as the result of a bad channel. There is a substantial delay in the transmission of data packets as the error sequences tend to come in large bursts as error correlation increases. As a result, the TCP/IP layer does not receive an ACK before its transmission timeout, which of course results in the reduction of the TCP window size and a corresponding drop in throughput.

The main perceived competitor with the Bluetooth protocol is undoubtedly 802.11b. Indeed, there are some question marks over the technology's survival as attempts to turn it into a LAN technology will face tough competition with 802.11b and 802.11a [7]. However, one must consider the rationale behind the Bluetooth protocol as that of a short-distance wireless technology that is optimised for low cost and low power consumption.

If one examines the strengths of the Bluetooth protocol as outlined previously it becomes clear that there are certain applications for which Bluetooth is the more suitable technology. For example, given that it uses much less electrical power than 802.11b it can be incorporated into pervasive devices such as mobile phones, digital cameras and PDAs. 802.11b, on the other hand, will never be incorporated into such devices as it uses too much electrical power to be feasible.

It is clear then that Bluetooth, though superficially similar to 802.11b, is intended for very different applications. It is ideally suited for pervasive computing as its low cost and low power features make its integration possible for any type of device. 802.11b, however, is superior to Bluetooth in the enabling of Wireless LANs and this will probably be the case for the foreseeable future.

3 Simulation of the Bluetooth Protocol

The simulator used for carrying out the performance assessments presented in this paper use the University of Berkeley's Network Simulator (ns) as its base platform. The rationale behind this decision is the amount of functionality provided by ns such as TCP agents, schedulers and packet structures. The simulator was implemented using C++ and the TCL/TK programming language and runs on any LINUX or UNIX operating system.

Given that the principal reason for the implementation of the simulator is to assess the performance of data traffic using Bluetooth as the bearer carrier only those components of the protocol stack germane to data transmission were implemented i.e. the Baseband, the LMP and the L2CAP. The design of the simulator module roughly correlated to a C++ class for each layer of the protocol stack that is modelled. Of course, there are also a number of ancillary classes to provide functionality such as transmission timers and clocks, as detailed in the specification.

A number of TCL/TK scripts were also implemented. One of the principal values of TCL/TK for ns is its interface to C or C++ programs. Using this mechanism a TCL class corresponding to a Bluetooth device consisting of a Baseband, LMP and L2CAP layer can be constructed. Scripts are also used to construct ad-hoc networks consisting of 2 or more Bluetooth devices. These scripts are configurable so that parameters such as the number of devices and the transport and application layer protocols used can be specified.

4 UDP Performance

We will now examine the performance of UDP (User Datagram Protocol) over the Bluetooth protocol. Specifically we will look at:

- The impact that packet size has on throughput performance.
- The impact that frequency of the periodic inquiry has on throughput performance.

UDP is a transport layer protocol and along with TCP is one of the main protocols used to transfer information over the Internet using IP (Internet Protocol) as the network-layer protocol. The key characteristics of UDP is that it is more lightweight than TCP but is unreliable as no attempt is made to retransmit packets which have been dropped.

A number of simulations are run which typically vary between 60 and 200 seconds. The number of slaves in the piconet and the periodic inquiry interval are also varied. In the section analysing packet size, that particular parameter is varied between 50 and 1000 bytes. The data transmission is always from a master to one of its slaves and is unicast rather than multicast. The duplex rate and time is 1 Mb and 3 ms respectively and the exponential traffic model is used to generate traffic.

4.1 IP Packet Size

The experiments used for analysing the effect of IP Packet size on the performance of UDP over the Bluetooth protocol consist of a number of tests, which have a packet size varying between 50 and 1000 bytes. In the case of a single slave piconet the simulation time is varied between 60 and 140 seconds while the 7-slave piconet scenario has simulation times between 120 and 160 seconds so as to allow all slaves to join the piconet. The piconet suspension interval to facilitate inquiry operations is scheduled to take place every 15 seconds until the maximum number of slaves has joined the piconet.

The first scenario examined is that of a single slave piconet. It is clear from Figure 1 that throughput does not vary greatly regardless of the size of the packets being sent, the mean throughput being 24.091 kbs (Kilobits per second) with a standard deviation of 0.228 kbs. This is a surprising result as one would expect that the throughput would be larger, the larger the packet size. However two factors in the single slave piconet explain this behaviour. The first consideration is the segmentation, which takes place in the L2CAP layer. For example, a 1000 byte packet is subdivided into 4 DM5 packets of 226 bytes each with the remainder being transmitted in a single 123-byte DM3 packet. This means that the size of the packet being transmitted through the Bluetooth protocol does not vary significantly. The other issue to bear in mind is that there is little or no impact on throughput in a single slave piconet as though the piconet is periodically suspended to facilitate inquiries no other slave device responds so the impact of processing an additional inquiry does not occur.

Figure 2 illustrates the system throughput for a 7-slave piconet. Unlike the single-slave case the system throughput varies widely from approximately 60 kbs to 140 kbs. The mean system throughput is 100.577 kbs while the standard deviation of 37.646 kbs denotes a wide disparity in the system throughput depending on the packet size. There thus appears to be a significant disparity in the throughput vs. packet size case for single slave and multiple slave piconets.

Figure 2 also shows that this variation takes place for individual slaves in the piconet – in these examples, the first, third and sixth slaves to join the piconet have mean throughputs of 14.292 kbs, 14.305 kbs and 14.999 kbs respectively. The system throughput is equitably divided among the piconet slaves but the standard deviations of 4.819, 5.042 and 5.549 emphasise the point that the packet size has an impact on the throughput for a multi-slave piconet. In addition to the inconsistency with the single-slave piconet a further curious aspect of the impact that packet size has on UDP throughput is the non-linear increase in throughput as packet size increases. In fact in certain cases throughput declines – for example, though system throughput increases by 23% if the packet size is increased from 500 to 750 bytes, throughput declines by 4.5% if the packet size is increased from 750 to 1000 bytes.

Again, segmentation, which occurs at the L2CAP layer, plays a role in the throughput performance. The reason why throughput does not differ significantly as packet size increases in certain circumstances is due to the wrapping of a UDP packet at the L2CAP layer in the smallest Bluetooth Data Packet available in terms of size. For example, a 50-byte and a 100-byte UDP packet are both wrapped in a 123-byte DM3 packet – the smallest packet size available to both. Segmentation also explains why system throughput is greater for a 750-byte packet compared to a 1000-byte packet. A 750-byte packet consists of 4 Bluetooth Data packets. The payload in the first 3 226 DM5 packets is completely occupied while the payload of the last packet consists of only 72 bytes and is thus transported in a 123-byte DM3 packet. Similar segmentation takes place for a 1000-byte packet. However in this case 4 DM5 packets and 1 DM3 packet are required to send 1 UDP packet from the Master to one of its Slaves. This increase in the volume of packet traffic sent over the Bluetooth network to transmit one UDP packet and indeed the cost of segmenting and reassembling the packet offsets the increase in the packet size and therefore leads to an overall decline in system throughput. The corollary of this situation occurs in the case of a 500-byte packet where despite requiring only 2 DM5 and 1 DM3 packets to transport a single UDP packet the decrease in size compared to a 750-byte packet results in less throughput than in the case of the larger packet size.

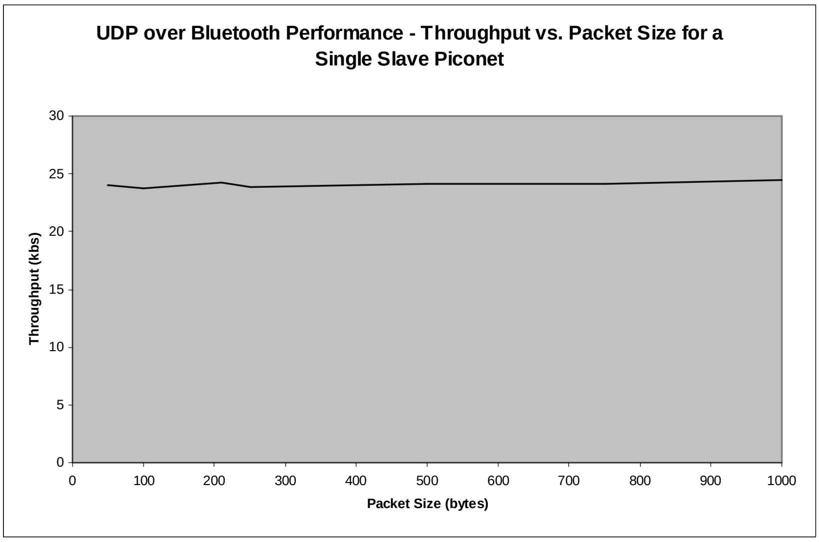

Figure 1: UDP over Bluetooth Performance - Throughput vs. Packet Size for a Single Slave Piconet

The other question that needs to be addressed is the large deviation between a single-slave and a multiple slave piconet. Looking at the physical number of packets passed through a Bluetooth Piconet over 140 seconds given a periodic inquiry operation of 15 seconds gives a mean percentage decrease in packet traffic of 45.618% with a very large standard deviation of 50.041%. If the figures in Table 1 are examined more closely we see that the percentage decrease in packet traffic increases as the packets become smaller. It is contended that this occurs because of the sheer volume of data traffic that can be sent as the packet size becomes smaller. The disparity in traffic volume becomes more acute as fair bandwidth allocation becomes a significant factor to ensure that each slave in a 7-member piconet receives a reasonable amount of data traffic.

4.2 Piconet Inquiry Frequency

A key metric that may affect the performance of UDP over the Bluetooth protocol is how often piconet operations may be suspended. The Bluetooth specification does not denote a range for this interval possibly because of the diverse range of Bluetooth applications. For example, while an interval of 20 seconds may be appropriate for a personal computer this interval may be much too large for another application such as a Bluetooth enabled PDA. A number of simulations of between 80 and 160 seconds are also run here. Both the number of slaves and the frequency of the inquiry interval are varied here. The packet size is fixed at 210 bytes.

Figure 3 displays the throughput performance of a single slave piconet when the periodic inquiry period is varied from 20 seconds down to 0.01 seconds. As one would expect, the less frequent the periodic inquiry operation the higher the overall throughput. This lack of consistency is emphasised by a large standard deviation of 28.815 kbs given a mean of 15.268 kbs. What is surprising though is that there is some throughput for periodic inquiry intervals of less than one second. This can reasonably be explained for a single slave piconet as though the piconet inquiry interval may be quite small there is no other slave responding to the inquiry, which lessens the impact of the operation.

Figure 3 also illustrates the 3 and 6 slave piconet scenarios respectively and shows that the system throughput for a multi-slave piconet behaves in a similar fashion to that of the single-slave piconet. Again there are large standard deviations of 18.75 kbs and 22.165 kbs for the 3-slave and 6-slave scenarios respectively given means of 27.286 kbs and 32.086 kbs. Again there is some throughput for piconet inquiry intervals of less than one second. While this may be explained by the fact that though a number of operations whereby a slave will join the piconet will takes place in both piconet scenarios a number of inquiries that will not be replied to will take place, it must also be noted that when the piconet inquiry interval is set at a very small frequency the difference in system throughput ostensibly appears to decline dramatically. However while the difference in system throughput for a 3-Slave and 6-Slave piconet where the piconet inquiry interval is 0.01 appears to be small (3.260 kbs vs. 2.783 kbs respectively) the percentage decline is in fact 17.139% which does not differ significantly from the percentage difference in system throughput for a 20 second piconet inquiry interval.

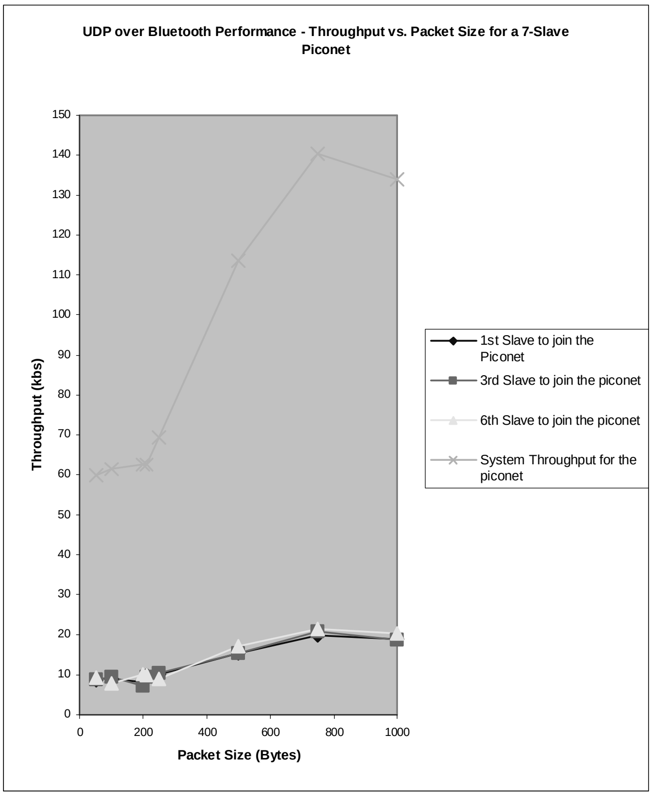

Figure 2: UDP over Bluetooth Performance - Throughput vs. Packet Size for a 7-Slave Piconet

Table 1: Comparison of Packet Size vs. Packet Volume for Single-Slave and 7-Slave Piconets

Packet Size (Bytes)	**No. Packets – Slave 1 (Single Slave Piconet)**	**No. Packets – Slave 2 (7-Slave Piconet)**	**% Decrease in Packet Volume**
1000	424	332	21.698
750	564	469	16.844
500	848	545	35.731
250	1671	696	58.348
210	2020	827	59.059
100	4163	1552	62.719
50	8471	2971	64.927

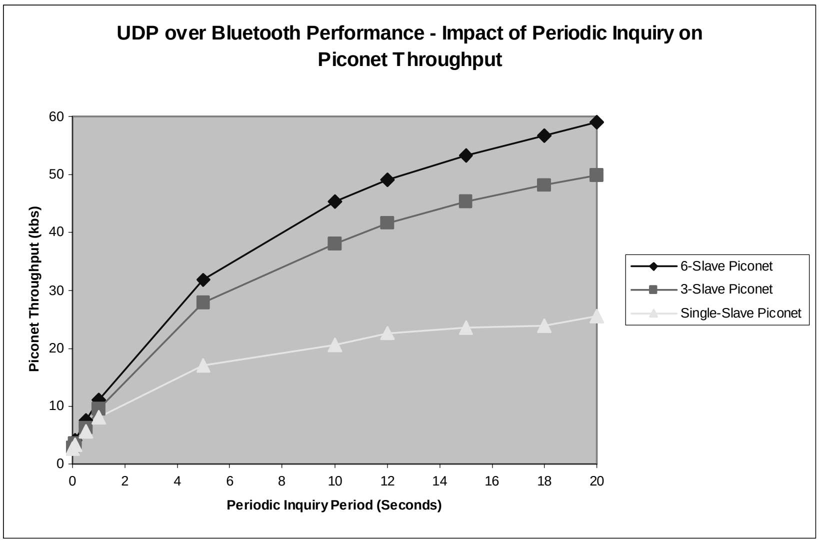

Figure 3: UDP over Bluetooth Performance - Impact of Periodic Inquiry on Piconet Throughput

5 Conclusions and Future Research

The Bluetooth technology is a key player in the fields of ad-hoc networking and pervasive computing. Though quite similar to 802.11 on a superficial level the technology has quite distinctive applications for devices such as PDAs, mobile phones and digital cameras. To date, the majority of research relating to the protocol has addressed its impact in terms of interference with 802.11. This paper attempts to partially redress the lack of performance-based analyses of the protocol.

This paper presents an assessment of the performance of UDP over the Bluetooth protocol. The most significant findings here perhaps are that UDP packet size can affect system performance in certain circumstances while in general the higher the frequency of the inquiry period for a UDP piconet the worse the per-second throughput performance. Further research in the area would include TCP performance over Bluetooth. Given the complexity of TCP compared to its UDP counterpart one would expect more unexpected results. In particular, one would expect that the requirement for a TCP acknowledgement and the resultants decrease in window size when TCP-senders do not send ACKs would lead to erratic results in Bluetooth piconets given the presence of a periodic inquiry interval which results in the suspension of piconet traffic.

References

[1] The Network Simulator, University of Berkeley, California, http://www.isi.edu/nsnam/ns

[2] Sundry Contributors, Specification of the Bluetooth System Version 1.0B, Volume 1 Core, December 1st 1999, http://www.bluetooth.com

[3] Sundry Contributors, Specification of the Bluetooth System Version 1.0B, Volume 2 Profiles, December 1st 1999 http://www.bluetooth.com

[4] Zurbes, Stefan. "Considerations on Link and System Throughput of Bluetooth Networks", *The 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2000. Volume: 2, Page(s): 1315 – 1319*

[5] Barton, M, Galli, S, Wong, K.D. and Koshy, B.J., "Bluetooth Technology: Link Performance and Networking Issues", *European Wireless Conference 2000 EW '00*

[6] Primak, Serguei and Snow, Chris, "Performance Evaluation of TCP/IP in Bluetooth Based Systems", *IEEE Vehicular Technology Conference (VTC), Spring 2002*

[7] Bhagwat, Pravin, "Bluetooth Vs. 802.11", *Hot Interconnects, Aug 24, 2001*