TCP/IP Traffic with Efficient Bluetooth Technology

Shafqat Hameed[1], Umar F.Khan[2], *Muhammad Saleem[3]

[1,3]National University of Sciences and Technology (NUST), Pakistan
[2]University of Bradford, Bradford, UK,
Shafqat.hameed@ceme.nust.edu.pk; hameed.shafqat@yahoo.com

Abstract- As new communication technologies are emerging they are introducing new form of short range wireless networks. Bluetooth is one of them as well, which allows information exchange over a short range. Bluetooth is a low cost, short range and low power radio technology which was originally developed to connect the devices such as mobile phone handsets, portable computers, headsets without having cable. Bluetooth was started in about 1994 by Ericson mobile communications but version 1.0 of Bluetooth came out in 1999. Bluetooth is a fastest growing technology. Its applications are increasing as the research goes on. By using Bluetooth we can make connection for traffic between sender and receiver. It can be either synchronous traffic such as voice or asynchronous traffic such as traffic over the internet protocol. In this paper we shall discuss that how efficiently Bluetooth can carry the TCP/IP traffic and as well as we shall analyse that how retransmission and delays are handled when there is an error in a packet of data. In addition we shall discuss the Bluetooth layer model and how it works and make the comparison between OSI reference model and Bluetooth layer model.

[Shafqat Hameed, Umar F.Khan, Muhammad Saleem. **TCP/IP Traffic with Efficient Bluetooth Technology.** Journal of American Science 2011;7(10):58-62]. (ISSN: 1545-1003). http://www.americanscience.org.

Keywords: TCP/IP, Bluetooth, OSI reference model

1. Introduction

Bluetooth is an ad-hoc wireless network concept that was presented in mid nineties.

Bluetooth can connect the mobile terminals within the range of each other and can make an ad-hoc connection between them. Bluetooth is designed for both types of data synchronous and asynchronous data. The example of synchronous data is voice and example of a synchronous data is IP.

In any network especially ad-hoc network reliability should be the key consideration for transmission of data for different applications. For this reason data packets in Bluetooth are protected by an ARQ in the link layer, but it is also important to have congestion control. To get the solution of this problem we use TCP (Transmission control protocol). TCP guarantees the reliable delivery of data packets.

TCP was originally used for the wireless networks that has low packet error rate. So the problem is that wireless networks usually has high data losses so losses of data packets may still trigger the congestion control in TCP even though there is no congestion. In our discussion we shall consider performance of TCP over Bluetooth and we shall discuss TCP/IP over Bluetooth in great detail. We shall discuss the throughput and delays of packets for TCP/IP under different conditions. We find that Bluetooth is powerful tool to carry TCP/IP and we can achieve low delays and high throughput.

2. Bluetooth

Bluetooth is a short range wireless link which originally developed to avoid the cables between electronic or portable devices. Now Bluetooth is an ad-hoc network which is used for both synchronous and asynchronous data. Bluetooth consists of number of protocols which are all residing on physical and data link layer of the OSI reference model. Detailed description will be discussed later on in the paper.

2.1 *Bluetooth system architecture*

In the architecture we shall discuss how Bluetooth is developed from its connections point of view with the devices and what type of topology does it use. Bluetooth operates in an environment where there is high level of interference between the networks. To make strong links Bluetooth uses acknowledgement which is fast and it also uses the frequency hoping scheme.

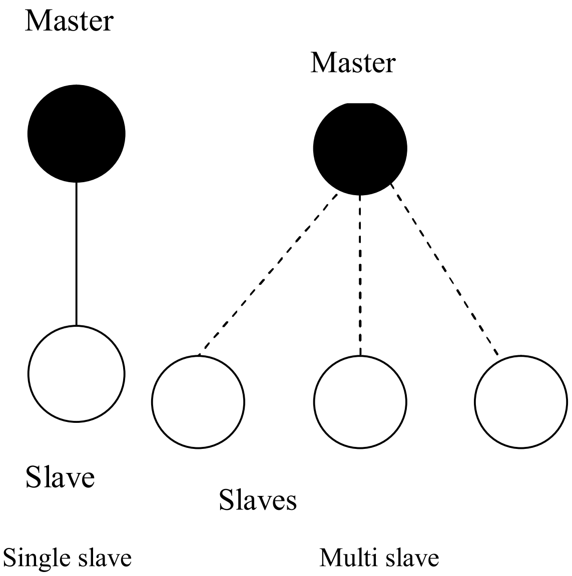


Fig.1.Bluetooth topology

As you can see from the above figures that how the topology of Bluetooth will look like (Figure 1).

2.45 GHZ band.

Bluetooth uses point to point (PPP) or point to multipoint connections. Several connections can be established and linked together in an ad-hoc fashion. Bluetooth actually uses a master slave approach between two connections (Figure 2).

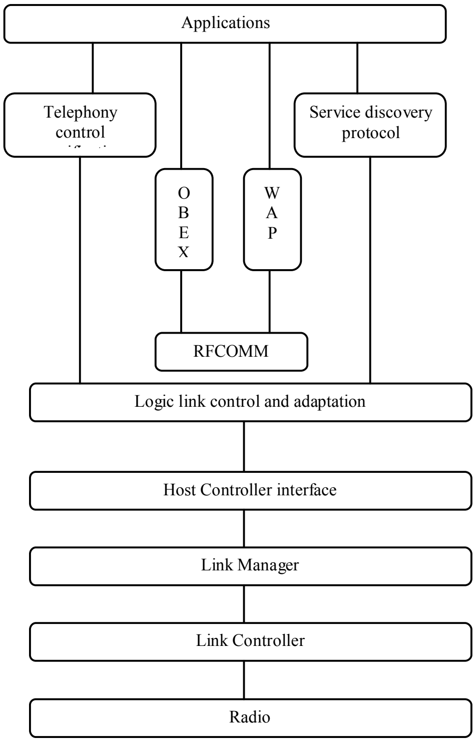

Applications
Telephony control
Service discovery protocol
O B E X
W A P
RFCOMM
Logic link control and adaptation
Host Controller interface
Link Manager
Link Controller
Radio

Fig.2.Bluetooth protocol stack

2.2 Bluetooth protocol Stack

Bluetooth protocol stack given below will give you an idea that how the Bluetooth protocol architecture will look like actually.

The figure below shows the Bluetooth protocol stack. Now we shall discuss the functionality of the above parts of the protocol stack. In the Applications Bluetooth profiles actually give guidelines on how applications should use the Bluetooth protocol stack.

Telephony Control Service (TCS) provide telephony services and Service discovery protocol (SDP) lets Bluetooth devices discover that what are other services supported by Bluetooth devices.

WAP and OBEX provide the interface to other communication protocols. RFCOMM provides serial interface. Logic Link Control (LLC) and adaptation multiplexes data from upper layers and convert that data into different packets. Host Controller Interface (HCI) handles communications between a separate host and a Bluetooth device. Link manager controls and configures links to other devices.

Link controller controls physical links via radio assembles packets and controls frequency hopping.

The job of the Radio is to do the modulation of data for transmission and reception.

As it is discussed earlier that Bluetooth uses a master slave approach. In Bluetooth all units are peer to peer with identical interfaces. When two or more units share a same channel they from a piconet. In piconet one unit acts as master and other becomes slave. There is no condition that which one should be master and which one should be slave. Any of them can be master or slave. For full duplex transmissions Bluetooth uses time division duplex (TDD) scheme. It is divided into slots and mostly slots are 0.62ms long.

2.3 *Relationship with OSI reference Model*

Actually Bluetooth does exactly match with OSI reference model, but still it will be useful to relate the Bluetooth with OSI reference model so that we can analyze the difference and similarities between them. Figure below will show you the both models. As you can see both these models are not same but it is useful to relate various parts of OSI reference model and Bluetooth model. This comparison will highlight the responsibilities of different part of the Bluetooth. The physical layer is responsible for electrical interfaces and channel coding so it covers the radio and some part of the baseband.

Link layer is responsible for transmission, framing and error control so it overlaps the link control tasks and the control end of the baseband which includes error checking and error correction as well.

Network layer is responsible for data transfer across the network so this consists of the higher end of the link controller; it also sets up and maintains multiple links as well. It covers most of the link manager tasks as well.

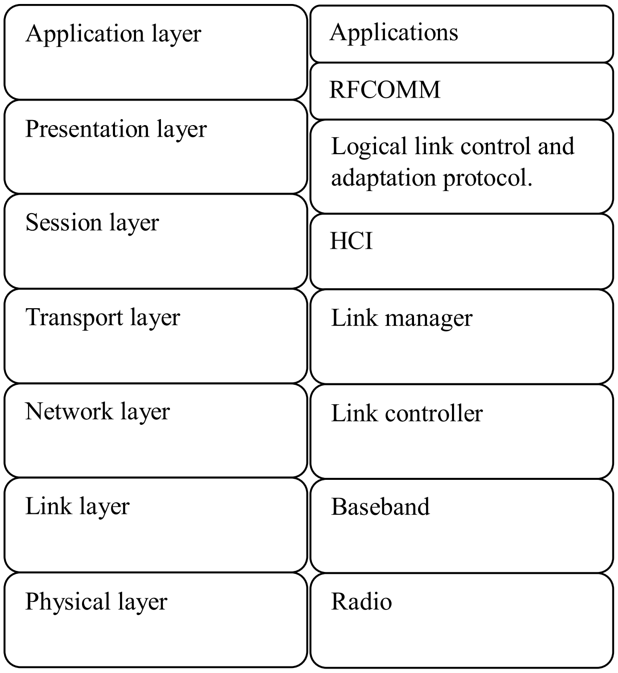
Transport layer is responsible for reliability and multiplexing of data transfer across the network to the level which provided by the application layer. It overlaps at the high end of the link manager and Host Controller Interface (HCI). It actually provides the data transport mechanism.

The session layer provides the data flow and the management services. These are covered by RFCOMM and logical control. Presentation layer actually provides a common representation for application layer data by adding the service structure to the units of data which is the main task of RFCOMM and sequence discovery protocol (SDP).

The application layer is finally responsible for managing communication between host applications.

3. **Transmission Control Protocol (TCP)**

In this section we shall discuss about the TCP its congestion control scheme and TCP is one of the core protocols of the internet protocol suite. Using TCP applications on the networked host create connections with one another, over which they exchange data packets or information. TCP guarantees reliable and orderly correct exchange of information between the transmitter and the receiver. When a connection is established between the two nodes, TCP transmits the information with the same order and sequence as well. Before transmitting this should be clear that data is divided into segments and these segments have equal size and segmentation is done before the TCP layer. To make sure that no packet is lost TCP gives a sequence number to each data packet; this sequence number also makes sure that data is given in the correct order at the other end (Figure 3).



OSI Model Bluetooth model

Fig.3. OSI Vs Bluetooth

3.1 *Time calculation for complete Round and Congestion Control*

The time when a segment is transmitted and the acknowledgement is received is called the Round Trip Time for one segment. It is very important that control scheme has very good estimation of time so that it will not create any chances of overlapping and corruption of data. It is also equally important for congestion control. The congestion control is a sliding window flow control. The flow control scheme is self clocking. When it gets the ACK signal from the receiver it actually submits the next segment for the transmission. Transmitter actually adjusts the congestion window depending upon the state. When there is the beginning of the transmission, which means there is no acknowledgement to receive the scheme is in the slow start mode. When the transmission starts running and it is running smoothly as well then the state changes to congestion avoidance state. If there is any packet lost then the state goes to retransmit and fast recovery or the state slow mode.

Each time state window is updated the scheme then checks whether it is possible to transmit segments or not. This is determined by the value of the counter which calculates the estimated rate and actual rate. The expected rate is the estimated transmission rate for a non congested connection and the actual rate is the estimated current transmission time. Both these rates are estimated every time the segment is transmitted. As all the segments have equal size so it is calculated straightaway.

There are actually three states which determine the status of transmission.

a) *Slow start*

The purpose of the slow start is to increase the transmission rate. As there is no acknowledgement signal coming from the receiver, so it works following way. Firstly it sets the current congestion value to 1 segment. After that is increased by doubling its value at every segment transmission.

b) Congestion avoidance state

The purpose of the congestion avoidance state is to get maximum throughput without causing congestion by finding a window size.

c) Fast retransmit and fast recovery

This scheme is used to retransmit the data or to recover the lost data. It does the functionality the by adjusting the counter's value which is called"sstresh". "sstresh" is initially set ½ of the congestion window. The segment which was lost is retransmitted and congestion window is wet to "sstresh+3".

4. Transmission Procedure

The transmission procedure containing TCP/IP and Bluetooth is described by using the piconet topology with sender and receiver. You may consider it as a PC and the Server attached with each other.

To understand the transmission process you will have to consider the protocol stack of the Bluetooth. Now we shall discuss that how segments are transmitted and received with the help of the figure below.

Initially the sender's application layer generates segments of data. These segments then are transmitted to the TCP layer. Now on the TCP layer a TCP header is added with the segments. From the TCP layer the segment is transmitted to the IP layer. IP keeps on storing the segments and then transmitting them to the Logical control and adaptation protocol layer (L2CAP) at regular intervals. Now form L2CAP segments are divided into Bluetooth data packets and then they are transmitted to the Bluetooth baseband layer. Now Baseband transmits the packets.

Packets may be lost during transmission because of BER (Bit Error Rate). The errors are modelled in a table called constant Packet Error Probability (PEP).

When the receiver receives a packet of data at its Baseband layer, it sends the packet to L2CAP layer of the stack. Now the job of L2CAP layer is to reassemble the packet into the TCP/IP segment and send it to Network or IP layer. IP layer receives the segment and then transfer it to the TCP layer. At this point TCP sends an acknowledgement signal for the segment and then the segment is shifted to the Application layer.

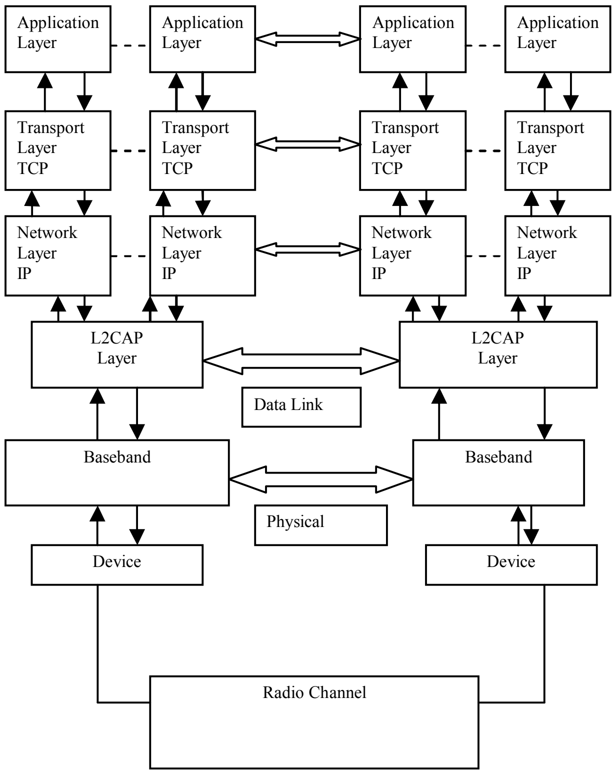


Fig.4. Transmissions of packets

4.1 *Data Packet Format*

A data packet which is received by a Bluetooth layer consists of the following three parts

1. TCP header
2. IP header
3. Payload

Now L2CAP adds 4 bytes which are for channel identification and the length of packet.

The size of TCP header is normally 32 Bytes, IP header is 20 bytes and the payload is of 1429 Bytes. Total packet size is 32+20+1429+4 = 1485 Bytes.

There are certain delays which need to take in account as well. e.g. when a packet of data is transmitted to TCP is delayed by an amount of time which allows the TCP to add a TCP header. Similarly IP header is added, which takes fraction of second. So every time data is passed from one layer to another, data is stored in the buffer for a certain amount of time.

Table1 parameters and values

Parameter	Value
Data segment size	1429 Bytes
Maximum receiver window	12 segments
Buffer size on Bluetooth	15 kbytes
Delay on TCP layer	1 micro seconds
Delay on IP layer	1 micro seconds
Delay on L2CAP layer	1 milli seconds
Delay on Baseband Layer	1 milli seconds

5. The Arrival process

How the application layer delivers the data to the TCP layer is determined by arrival process. There are two cases for that. One is when the TCP has always the data to send. So it means that we want to check the maximum throughput. So the arrival data of the data is high enough to fill the queue of the TCP layer.

The second case is when we assume more bursty application process. The process is modelled by an Interrupted Bernoulli Process (IBR). Arrival occurs according to the Bernoulli process this is called as an active state. This period is followed by an idle state during which there is no arrival. When we are in active state the process will stay in the active state with the probability l-p or it will go the idle state with the probability p. Now if the process is in the idle state it will stay in the idle state for the probability l-q or it will go the active state with the probability q. when there is an active state a slot contains a packet with probability l. The time slots for the traffic generator are aligned with the time slots of the piconet which has been modelled.

6. Security Architecture

The Bluetooth security architecture is defined on the basis of modes. There are actually different modes of security.

- Security mode 1 is no secure it means that device in this mode will never initiate any security procedure.
- Security mode 2 gives service level enforced security services using L2CAP decide whether or not security is required.

- Device in this mode initiates security procedure before it establishes a connection.

As TCP guarantees sequential and reliable transmissions so TCP services mostly operate at mode 3. The security architecture of the Bluetooth is given in the following figure below. Host controller interface (HCI) queries to find out whether to apply authentication to the connection or not the user interface is queried by the security manager to get the PIN and to authorise the new services as well. Protocol layers like RFCOMM, L2CAP etc. query the security manager with access request. The device data base holds the information on whether devices are authenticated and authorised as well. The service data base holds the information on whether the authorisation, authentication and encryption are required for access to the services.

It is the service that decides the level of security to be enforced. Security is enforced only when access is required to protocol or a service which required security. The protocol or service requests access from the security manager. The security manager looks up the service or protocol in the service database to see what level of security to be imposed. Then it looks up the connecting device in the device database to see whether it meets the requirements of the service. If necessary the security manager enforces authentication and/or encryption, and sends necessary queries to PIN or authorisation to the user interface. Access is then granted or refused, and if access is granted then service can be used, and if not then service cannot be used.

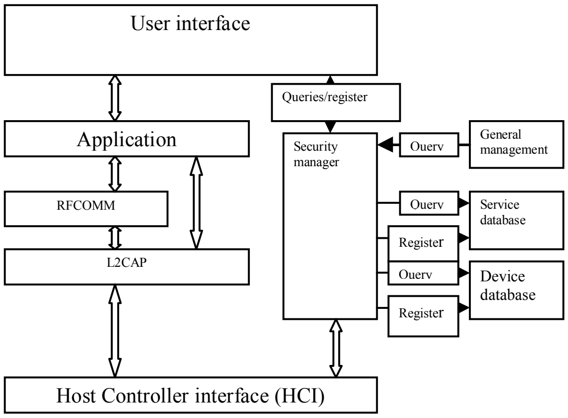


Fig.5. Security architecture

It is possible that some services may use the connections without encryption, and then another service will begin using the service which requires encryption. Encryption will be set up for the service which requires it.

Other than the link management messages required to configure security, there is no impact on the bandwidth. Same no of bits are sent for encrypted link as are sent on a UN encrypted link.

7. Conclusion

Carrying TCP over wireless networks causes the degradation in the throughput and increased delays as flow control mechanism in TCP reacts on delays which are introduced by retransmitting the packets having errors is the network was congested.

We have looked into the Bluetooth protocol stack and compared the Bluetooth model with OSI model to have a clear understanding of the layered architecture.

We have tried to show that how Bluetooth handles the TCP/IP traffic and we also tried to show that Bluetooth wireless ad-hoc network handles the traffic very well. Throughput is kept at high level and end to end delays caused at different layers during buffering and retransmission is also at acceptable level.

REFERENCES

Gehrmann, Christian, Person, Joakim, Smeets,
Ben. ”Bluetooth Security” Boston, Mass, London 2004.

Insam and Edwards” TCP/IP embedded internet
applications” Oxford, newness, 2003.

Jennifer Bray and Charles F Sturman,”Bluetooth 1.1: Connect without cables” 2nd Ed, Prentice Hall PTR 2001, 2002.

Muller and J. Nathan “Bluetooth Demystified/Nathan
J.Muller” McGraw Hill, 2001 New York; London.

Niklas Johnson, Maria Kihl and Ulf Korner “TCP/IP over
the Bluetooth Wireless ad-hoc network” Lund University
Sweden.

Sidnie Feit and Jay Renade, “TCP/IP Architecture,
Protocols and Implementation” McGraw-Hill Series on
computer communications.

7/24/2011