Performance of a Bluetooth Based Structural Health Monitoring Telemetry Network

Vilas Uchil
Kurt Kosbar Faculty Advisor
Department of Electrical and Computer Engineering
University of Missouri – Rolla
Rolla, MO 65409-0040

ABSTRACT

The Bluetooth standard is intended to provide short-range (10-100 meter) wireless connectivity between mobile and desktop devices. It was developed as a replacement for short cables, and has the ability to form ad-hoc networks. A large inter-connection of piconets can be arranged to form a scatternet for data collection in a Bluetooth based structural health monitoring Telemetry network. The Bluetooth protocol architecture supports the formation of a daisy chain network. However Bluetooth technology was not intended for long daisy chain networks. In this work, we propose to evaluate the throughput and latency for data transmission in a long daisy chained Bluetooth based telemetry network.

INTRODUCTION

There is an increasing interest in implementing wireless technology for telemetry applications. For applications such as measuring vibrations at different points along a highway bridge, the sensors are typically connected with cables. These cables increase cost and difficulty of installing and maintaining the system. Several wireless protocols such as Bluetooth, Wireless LAN (WLAN) are available for commercial purpose. The Bluetooth standard [1,2] has enabled the design of small, low power, low cost, embedded radios that can replace cables, heading towards ubiquitous connectivity. A Bluetooth system has the ability to form ad-hoc networks. A group of Bluetooth modules inter-acting with each other forms what is known as piconet. Sets of piconets can be arranged to communicate with each other and is known as scatternet. A daisy chain network formed from a scatternet can be used for data collection in a Bluetooth based structural health monitoring telemetry network [3]. Being a short-range technology, Bluetooth was not intended for long daisy chain networks. Our work analyses the throughput and latency for data transmission in a long Bluetooth based daisy chained network and its effect on sampling rate of the sensors in a structural health monitoring telemetry system.

This paper provides an introduction to the Bluetooth standard, its networking arrangements and different feasible configurations of a Bluetooth based telemetry network. We then analyze a simple Bluetooth based telemetry network and evaluate its throughput and latency with respect to the sensor-sampling rate. Excellent reviews on some of the performance evaluation of a Bluetooth based network are provided by references [4-5].

Bluetooth Standard

Bluetooth based systems operate in the unlicensed Industrial Scientific Medicine (ISM) band at 2.4 GHz. [1-2]. The nominal range of a single Bluetooth link is 10 m at 0 dBm and can be increased with an external power amplifier to extend up to 100 m at +20 dBm. It uses Frequency-Hopped Code Division Multiple Access (FH-CDMA) with short data packets to make Bluetooth link robust in a noisy radio environment. Bluetooth system uses a set of 79 hop frequencies with 1 MHz spacing. The unit that controls the FH-CDMA channel is called the master. All other units participating on the same hopping channel are called slaves. Any unit can become a master or a slave. An ad-hoc network formed by a master slave arrangement is called a piconet. In a piconet, as the master hops in the frequency band all the slaves listens to the master at that frequency. Figure 1 illustrates a piconet arrangement.

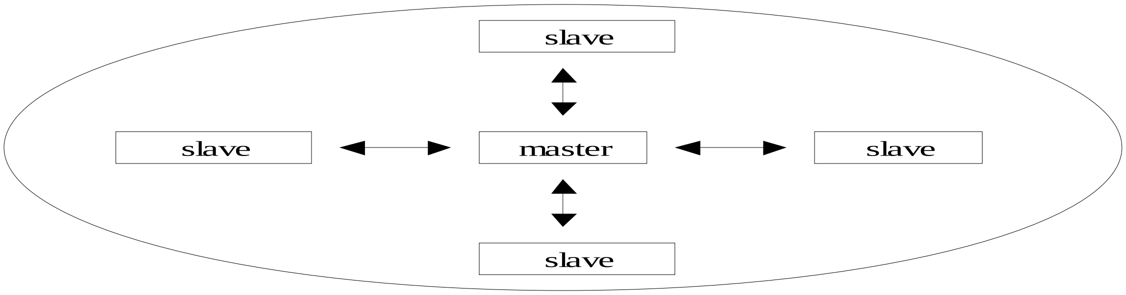

Figure 1. A Master-Slave Arrangement in a Piconet

Each piconet maintains its own frequency-hopping channel. All the members associated with the respective piconet remains synchronized to that hopping channel. Two or more piconets can interact with each other to form a scatternet. A long chain of interconnecting Bluetooth modules can be arranged for data collection. As the number of piconet increases, there is a considerable decrease in performance. Two factors that affect performance are amount of bridging and number of established links in each piconet [6]. Figure 2 illustrates a scatternet.

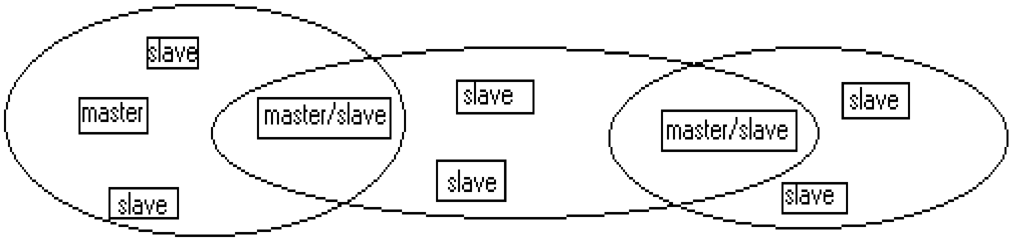

Figure 2 A Scatternet

Bluetooth uses Time-Division Duplexing (TDD) [7,8], where the master and slave can operate alternately. In a normal connection mode, the master transmits in even numbered slots and the slave transmits in odd numbered slots. Each time slot corresponds to one hop frequency. So a master-slave communication takes place at different carrier frequencies. The hop frequency remains the same for the duration of the packet. The Bluetooth standard supports three types of packet format, occupying one slot, three slots or five slots. Due to the packet type covering more then one slot, the master transmission may continue in odd numbered time slots followed by slave in an even numbered time slot. Figure 3 illustrates master slave TDD.

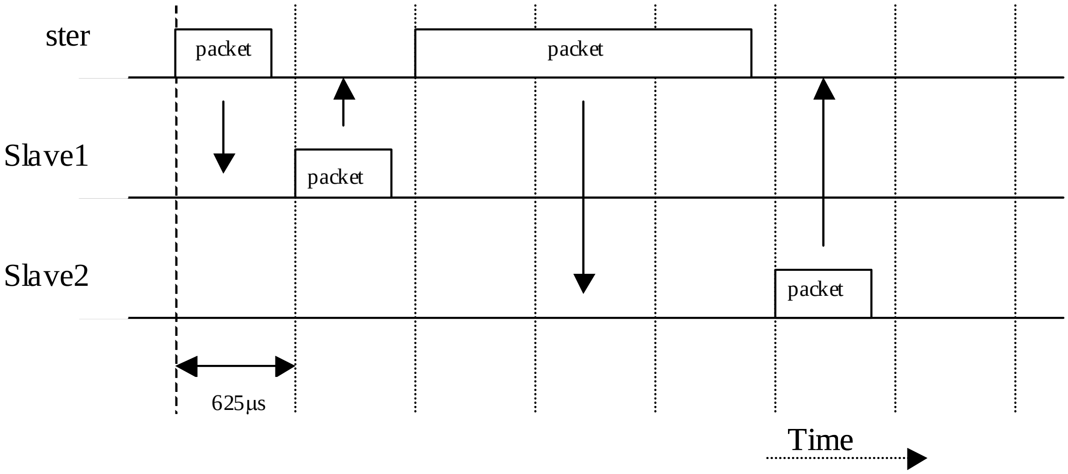

Figure 3. TDD in Bluetooth System

Bluetooth network

Paging and inquiry procedures are used for establishing new connections in a Bluetooth system [1,2]. The inquiry procedure enables a unit to discover those units that are in the range of the piconet along with their device addresses and clock information. The paging procedure then helps to establish the actual connection. A unit that establishes the connection will carry out a page procedure and will automatically be the master of the connection.

To discover new units in the piconet range, a potential master will enter into the inquiry state and the potential slave will enter the inquiry scan state. The master will listen for responses as it sends short packets containing its ID and the Inquiry Access Code (IAC). All slave units that are in the inquiry scan state will listen for IAC packets. On receiving the IAC, the respective slave will respond with its Bluetooth device address (BD_ADDR), its clock, its frequency parameters and details about page scan time interval [1,2]. On receiving a response during the inquiry state, the master goes into the page scan state. The master generates a Device Access Code (DAC) for the corresponding slave using its slave device address. Since the master has the slave's page scan time interval, it sends a slave DAC at that particular interval to the slave. The slave remains in the page scan state and listens for its DAC. As the slave hops slowly to cover the possible frequency hops from the paging master, the slave waits for one particular frequency to match. At that particular frequency, the slave transmits a page response to the master. The mater responds by sending a packet containing information about its BD_ADDR, its clock value, and the Active Member Address (AM_ADDR) for the slave. The slave also notes down this timeslot as master-transmit timeslot in the piconet. The master and slave then exchange a poll and null packet to mark the establishment of master-slave connection and thus enter into the connection state. Once in the connection state, the master and slave can transmit data depending on the different master-slave engagements.

There are four different modes of master-slave engagement; Active, Sniff, Hold and Park [9].

- **Active** In this mode, the master and slave actively participate on the channel. The Active slave listens on the master-slave time slots for packets. During a mater transmission, if an active slave is not addressed, it may sleep until the new master transmission begins. All Active slaves maintain an active member address.
- **Sniff** In this mode the slave listens to the master less frequently than on the active state. The slave in the Sniff mode is still an active member of the piconet and maintains its active member address.
- **Hold** The slave and master agree on a time duration, for which the slave is released from the connection. Other than an internal timer, all other operations are suspended. When the timer expires, the slave wakes up, synchronizes with the hopping sequence and waits for further instructions from the master. This brings about a considerable power saving. This mode is more helpful in forming daisy-chained scatternet for telemetry applications.
- **Park** This is a low power mode with very little activity in the slave. The slave remains synchronized to the network, but does not take part in any data transmission. The slave forfeits its active member address and receives a park member address when entering this mode. The master module wakes up the associated slave using the parked member address for data transmission.

An infrastructure monitoring telemetry system can be formed using a piconet. With the master forming the central hub, for all the slaves that pass on the health information of the system to the master. For distances spanning more then 100 meters, a daisy-chained scatternet can be formed [3].

Network Structure

In telemetry applications, such as measuring vibrations at different points along a highway bridge, different types of Bluetooth based network structures are feasible. The simplest structure feasible would be the linear daisy chain scatternet. Figure 4 illustrates a linear daisy-chained scatternet.

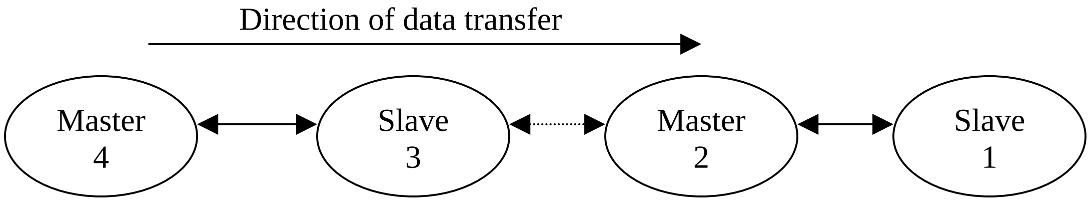

Figure 4. Linear Daisy-Chained Scatternet

As shown in figure 4, the data is transmitted from the source unit (Master 4) to the destination unit (slave 1) in a linear pattern. All the sensors connected to the Bluetooth unit are generating data at a fixed data rate. The network is arranged in a chain of alternating master and slave units. Initially all the masters establish an active connection link with the adjacent slave in the direction of data transfer. After the data transfer from master to slave, the existing slave goes into a hold state for that master and activates its connection with another master down the link. The slave remains in hold mode for the previous master till all the data packets are transferred to the adjacent master down the link. This process repeats and data gets transferred down the chain in this pattern. A major disadvantage for a linear daisy-chained scatternet is when one of the nodes fails, resulting in network breakdown. This can be prevented by providing an additional slave in park mode as redundant unit. In cases of failure, this unit can be brought into active state and thus the network connection can still be maintained. Figure 5 illustrates a linear daisy-chained scatternet with backup slaves in parked mode.

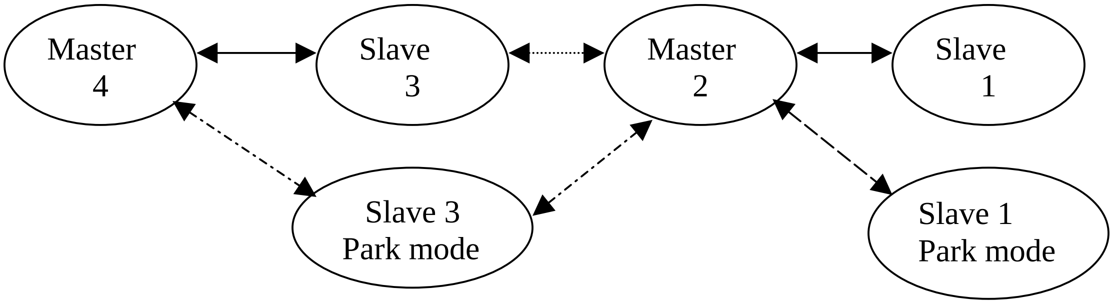

Figure 5. Linear Daisy-Chained Network with Redundant slaves in Parked Mode

The above structure again has the same problem of network failure when one of the master units breaks down. This can be resolved by arranging a more complex layout of master slave arrangement. One way would be providing a lattice structure so that incase of failure the nodes can still be able to interconnect with each other to pass on the data from one end to another. Figure 6 illustrates lattice structure of Bluetooth network.

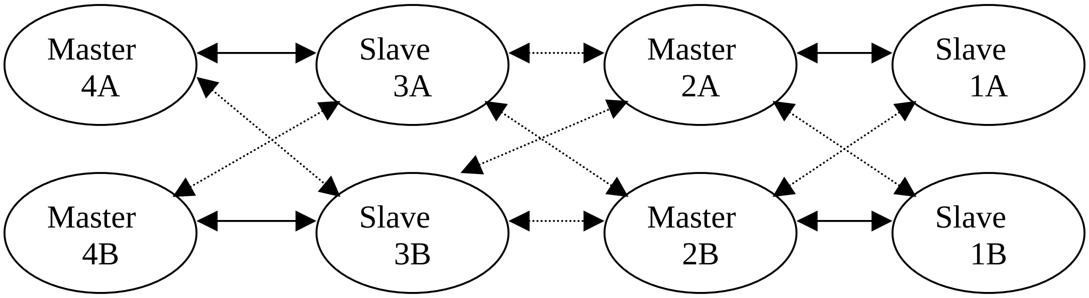

Figure 6. Lattice Structure Network Layout

The lattice network layout is a parallel arrangement of two linear daisy-chained networks, with master-slave connection possible between the two linear chains of units. This layout protects against single point failures, however it needs a more complex algorithm for data transfer. An addition of nodes into a piconet also degrades the network performance.

Network Analysis

For simplicity we will consider a simple linear daisy-chained scatternet layout as shown in Figure 4. During the initial stage, all the masters transmit data into the adjacent slaves. Master 4 connects to Slave 3, and Master 2 connects to Slave 1. The destination node (Slave 1) is in turn connected to a data collection node, which can be a central data collection hub. We assume that all the sensors connected to the Bluetooth unit are generating data at the same rate. During the start of the transmission all the nodes has a single packet to be transmitted. We calculated the latency of the network and its effect on the sensor-sampling rate. We denote notations M4P1 as Master 4 packet one, S3P1 as Slave 3 packet 1 and so on. Figure 7 shows the packet transfer for each of the slots. E refers to even slot and O refers to odd slot.

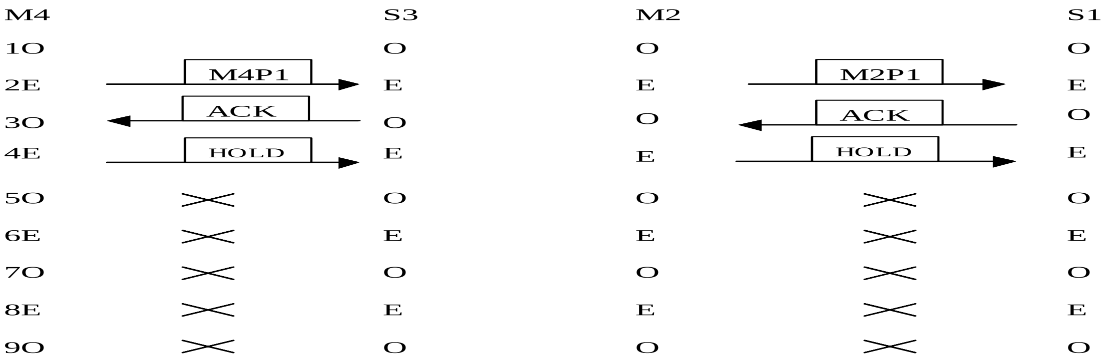

Figure 7. Data Transfer Between Master and Adjacent Slaves

As we see from the figure initially all the masters pass on their first packet to the adjacent slave. After successful data transfer, the master puts the adjacent slave into hold mode, so that the slave can start initiating a new connection with the next master down the line. A Bluetooth system takes six time slots of overhead for passing into the hold mode. We observe that the initial master to slave data transfer for the first packet takes eight time slots. Figure 8 shows transfer of M4P1 from Slave 3 to Master 2. Interaction between Slave 3 and Master 2 starts with two-time slot of synchronizing process, which is required to adjust the clock offset and other signaling parameters. Master 2 then polls Slave 3 and the connection is established. Slave 3 then transfers both the packets M4P1 and S3P1. Master 2 passes Slave 3 into the hold mode. It takes 13 time slots for two packets to be transmitted from a slave to a master. Slave 3 goes into hold mode and Master 2 Initiates connection with the next adjacent Slave 1.

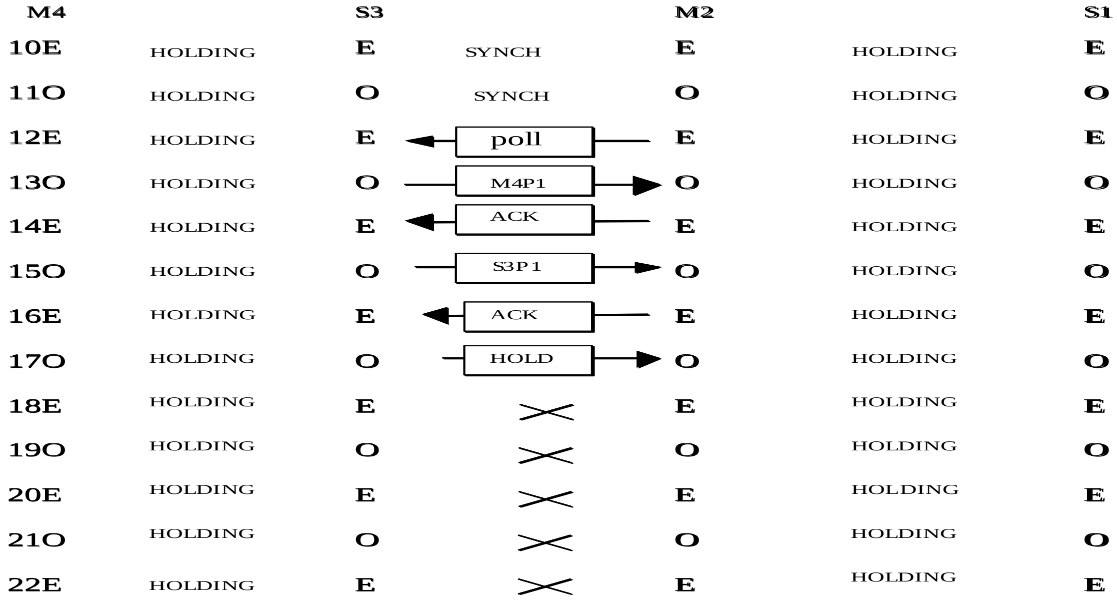

Figure 8. Slave to Master data transfer

Figure 9 shows the final data transfer between Master 2 and Slave1.

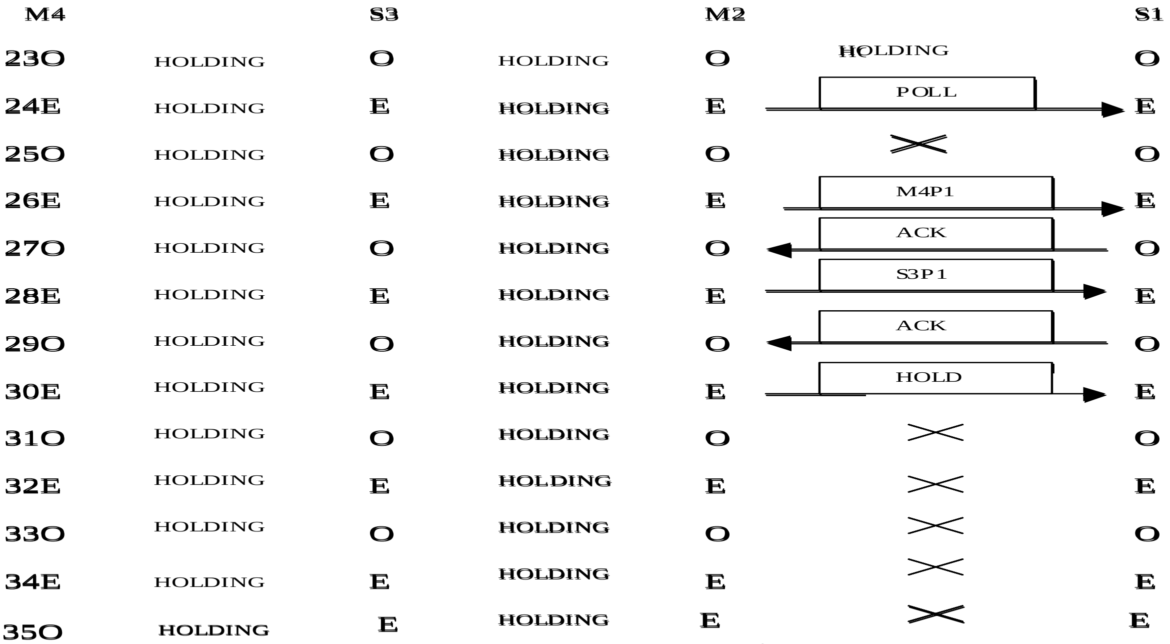

Figure 9. Master 2 to Slave 1 Data Transfer

From figure 9 we see that it takes 13 time slots for the data to be transferred from Master 2 to Slave 1. For all the packets from four nodes to be transferred from the first master to the destination unit requires up to 35 timeslots. Each master or slave uses 13 slots to accommodate polling, synchronization, data transfer and acknowledgement. The bandwidth for each hop is calculated as follows

Bandwidth for single hop = Bluetooth maximum bandwidth / master (slave) hop timeslot.
= 1MHz / 13 timeslots
= 76.9 kbps

So the throughput is 7.7% of the maximum data rate supported by Bluetooth system between two devices. The degradation of bandwidth is due to the scheduling activity needed for master and slave to communicate in a scatternet.

The propagation delay is the time taken for a data packet to traverse from the source to the destination node. In our application each of the node is generating data at a particular sensor-sampling rate. We need to calculate the propagation delay between the two farthest nodes in the telemetry chain. For analysis, let N be the number of nodes in the chain. So the propagation delay is calculated as

$$T_{prop} = 9 \text{ [slots]} + (N-2)*13 \text{ [slots]}.$$

Now each timeslot is equal to 625 µsec. So we can calculate the total time delay in seconds. For N = 4, we have

$$\begin{aligned} T_{propagation} &= 35 \text{ slots} \\ &= 21.8 \text{ msec} \end{aligned}$$

Now in this application, we assume that new packets are generated after all the packets have been transferred to the last node. The sensor sampling time would be inversely proportional to the propagation delay.

Sensor Sampling Rate = 1/Tprop

As the length of the telemetry network increases, the propagation delay increases and hence the sensor-sampling rate decreases. This method would be more convenient if the length of the telemetry network is of nominal range. One method for increasing the sampling rate would be to add buffer memory in each of the nodes to store the additional data that is been generated and transfer them down the link. However as the length of the telemetry network chain increases, the node buffer size would increase proportionally. For applications such as measuring vibrations along a highway bridge, where we do not expect the sampling rate to be too high, we can use Bluetooth based telemetry nodes for data collection.

Conclusion

This paper supports the usage of Bluetooth networking for telemetry applications. The Bluetooth networking feature of scatternet can be used to form a long daisy chain network. Thus the analysis concludes that we can use Bluetooth networking for data transfers for applications requiring slower data rate. The propagation delay increases for applications requiring longer length of the chain, correspondingly the sampling rate decreases proportionally. The sampling rate can be improved by addition of buffer memory in each of the Bluetooth nodes, however this leads to additional cost and brings about more complexity in data management. These limitations motivate research to increase the robustness of Bluetooth networking protocols and thus provide more efficient packet transfer.

References

[1] "Specification of the Bluetooth System", Core, V1.1, Dec. 2000.

[2] "Specification of the Bluetooth System", Profiles, V1.1, Dec. 2000.

[3] "Feasibility of a Bluetooth based structural health monitoring system", Vilas Uchil and Kurt Kosbar, proceedings of International Telemetry Conference, pp. 32-39, Las Vegas, Nevada, Oct 23-25, 2001

[4] Tae-jin Lee, Kyunghun Jang, Hyunsook Kang and Jonghun Park, "Model and Performance Evaluation of a Piconet for point-to-multipoint communications in Bluetooth", Vehicular Technology Conference, 2001. VTC 2001 spring. IEEE VTS 53rd Volume: 2, 2001

[5] Stefan Zurbes, "Considerations on link and System Throughput of Bluetooth Networks", Personal, Indoor and Mobile Radio Communications, 2000. PIMRC 2000. The 11th IEEE International Symposium on, Volume: 2, 2000

[6] Miklos G, Racz A, Turanyi Z, Valko A, Johansson P, "Performance Aspects of Bluetooth Scatternet Formation", Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC 2000 First Annual Workshop on, 2000

[7] Haartsen J.C, “The Bluetooth Radio System”, IEEE Personal Communications, Volume: 7 Issue: 1, Feb. 2000

[8] Zhang Pei, Li Weidong, Wang Jing, Wang Youzhen, “Bluetooth-The Fastest Developing Wireless Technology”, Communication Technology Proceedings, 2000. WCC - ICCT 2000 International Conference on, Volume: 2, 2000

[9] Kalia Manish, Garg Sumit, Shorey Rajeev, “Efficient Policies for Increasing Capacity in Bluetooth: An Indoor Pico-Cellular Wireless System”, Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st, Volume: 2, 2000