Bluetooth: Carrying Voice over ACL Links

Rohit Kapoor, Ling-Jyh Chen, Yeng-Zhong Lee, Mario Gerla

3803 H, Boelter Hall, University of California, Los Angeles

Abstract - **The Bluetooth technology will enable users to connect a wide range of electronic devices such as laptops, headsets, cellular phones etc. Bluetooth devices can connect to form a piconet, which consists of a master and upto 7 slaves. The master controls the medium access in the piconet using a polling scheme. Two types of connections can be established in a piconet: the Synchronous Connection-Oriented (SCO) link, and the Asynchronous Connectionless (ACL) link. SCO links provide a circuit-oriented service with constant bandwidth based on a fixed and periodic allocation of slots. They require a pair of slots once every two, four or six slots, depending upon the SCO packet used. ACL connections, on the other hand, provide a packet-oriented service and span over 1, 3 or 5 slots. The master of the piconet uses a polling mechanism to divide the piconet bandwidth among the ACL links. Since SCO links require a periodic allocation of a pair of slots, they leave very little of the piconet bandwidth available to ACL links. Moreover, the controlled access of Bluetooth ensures that no ACL link gets starved. Under such an access mechanism, ACL links may be sufficient to carry high-quality voice and SCO links may not be needed. Our simulation and hardware experiments show that though the voice quality is affected slightly by using ACL instead of SCO links for voice, TCP connections perform much better if SCO links are not used. This paper, thus, makes a case for using ACL in place of SCO links for carrying voice. This renders SCO links redundant.**

1. INTRODUCTION

Bluetooth [1] is a universal radio interface in the 2.4 GHz ISM frequency band, which will enable users to connect a wide range of small electronic devices such as notebook computers, cellular phones and other portable handheld devices easily and quickly, without the need for cables. The key distinguishing features of Bluetooth are its minimal hardware dimensions, low complexity, low price and low power consumption [2].

Bluetooth is based on a centralized connection-oriented approach. Bluetooth devices sharing a wireless channel form a *piconet*. One device in a piconet has the role of the master and controls access to the channel, while the others are slaves. There may be up to 7 slaves in a piconet. Bluetooth uses a Time-Division Duplex (TDD) scheme to divide the channel into 625us time slots. Master and slave units transmit alternately. Each piconet is characterized by a particular fast frequency-hopping pattern; the frequency is uniquely determined by the master's address and is followed by all the devices participating in the piconet.

There are two types of connections that can be established between a master and a slave: the Synchronous Connection-Oriented (SCO) link, and the Asynchronous Connectionless (ACL) link. SCO links provide a circuit-oriented service with constant bandwidth based on a fixed and periodic allocation of slots. SCO links require a pair of slots once every two, four or six slots, depending upon the SCO packet chosen for the link. ACL connections provide a packet-oriented service and span over 1, 3 or 5 slots. For ACL links, Bluetooth uses a fast acknowledgment and retransmission scheme to ensure reliable transfer of data. The master controls the traffic on ACL links by employing a polling scheme to divide the piconet bandwidth among the slaves. A slave is only allowed to transmit after the master has polled it.

SCO links have been designed to support time-bounded information like voice. Since these links require a periodic allocation of a pair of slots once every 2, 4 or 6 slots, they leave very little of the piconet bandwidth available to ACL links. Consider a situation in which two TCP connections and two voice connections coexist in a piconet and the voice connections are carried over SCO links. It is easy to see that even in the best case (if the SCO links require a pair of slots every 6 slots), the TCP connections may easily get starved.

Another point to note is that Bluetooth has a very controlled channel access. Each node in a piconet is given a chance to transmit by the master: the presence of a polling mechanism to divide the piconet bandwidth among the slaves ensures that no ACL link gets starved. Under such an access mechanism, ACL links may be sufficient to carry high-quality voice (voice over IP) and SCO links, may thus be rendered redundant. This would also allow the master to spare more bandwidth for other TCP connections in the piconet. It is expected that the quality of voice, if transmitted over ACL links, would degrade as compared to SCO links. It remains to be seen, though, how much the quality of voice suffers, how large are the voice delays, are these delays acceptable etc. Moreover, one would also like to see how much better TCP connections might be supported if voice was carried over ACL links.

In this paper, we try to answer some of the questions regarding supporting voice over ACL links. We perform experiments with Ericsson Bluetooth hardware kits considering SCO and ACL links under various configurations (varying the number of ACL and SCO links, varying the packet types etc). Section 2 gives a discussion of SCO and ACL links. In Section 3, we

0-7803-7605-6/02/$17.00 ©2002 IEEE.

describe the experimental testbed, i.e., the Bluetooth hardware. In Section 4, we present simulation and experimental results of voice delays and TCP performance when voice is carried over ACL and SCO links. In Section 5, we draw conclusions on carrying voice over ACL links.

2. SCO AND ACL LINKS

Bluetooth supports both synchronous services such as voice traffic and asynchronous services such as bursty data traffic. The specifications define two different physical link types:

SCO: *Synchronous Connection-Oriented*
ACL: *Asynchronous Connectionless*

2.1 SCO Link

The SCO link is a symmetric, point-to-point link between the master and a single slave in the piconet. The SCO link involves reservation of slots and can therefore be considered as a circuit-switched connection between the master and the slave. The master maintains the link by using reserved timeslots at regular intervals. The SCO link is typically used to support time-bounded information like voice. The master can support up to three SCO links to the same slave or to different slaves.

The master sends SCO packets at regular intervals. The specific SCO slave is allowed to respond in the following slot unless the master addressed a different slave in the previous slot. On the SCO links, the packets used include a CRC and are never retransmitted. The SCO packets have been designed to support 64 Kb/s speech. The specifications define three pure SCO packets and one hybrid SCO packet, which carries an asynchronous data field in addition to a synchronous voice field.

HV1 packet: HV stands for high-quality voice. The HV1 packet carries 10 information bytes, which are protected with a rate 1/3 FEC. This packet has to be sent every two time slots and can support 1.25ms of speech at a 64Kb/s rate.

HV2 packet: This packet carries 20 information bytes, which are protected with a rate 2/3 FEC. This packet has to be sent every four time slots.

HV3 packet: This packet carries 30 information bytes, which are not protected. This packet has to be sent every six time slots.

DV packet: The DV packet is a combined data-voice packet. The payload is divided into a voice field of 80 bits and a data field of up to 150 bits.

2.2 ACL Link

The ACL link provides a packet-switched connection between the master and all active slaves in the piconet. A slave can send an ACL packet if it has been addressed by the master in the previous slot. To ensure data integrity, ACL packets are retransmitted. Only a single ACL link can exist between a master and a slave. The master schedules ACL packets in the slots not reserved for the SCO links. The specifications define 7 kinds of ACL packets, three DM (data-medium rate) packets, three DH (data-high rate) packets and one AUX packet.

DM: These packets are coded with a rate 2/3 FEC; they contain a 16-bit CRC code and are retransmitted if no acknowledgement is received. Three DM packets have been defined, DM1, DM3 and DM5, which cover 1, 3 and 5 time-slots respectively.

DH: These packets are similar to the DM packets, except that the information in the payload is not FEC encoded.

Similar to the DM packets, three DH packets (DH1, DH3 and DH5) have also been defined.

AUX: This packet is like a DH1 packet, but has no CRC code and is not retransmitted.

The experiments in this paper concentrate on the pure HV SCO packets and the DM and DH ACL packets.

3. EXPERIMENTAL TESTBED

The experimental test bed used in the paper consists of Ericsson Bluetooth Application Toolkits, the ROK 101 007 modules [3]. This module consists of 3 major parts: a baseband controller, a flash memory and a radio that operates in the 2.4 GHz ISM band. The module is compliant with Bluetooth version 1.1. It is a Class 2 Bluetooth module (0 dBm) and is type-approved. The modules run the Ericsson Bluetooth Host Stack, which implements the Bluetooth functionality in the module.

The module supports both ACL and SCO links. Up to 3 SCO links and 7 ACL links can be supported in a piconet. When creating an SCO link, an ACL link must be up and running between the two devices; an SCO link can be thereafter added. The module supports point-to-multipoint functionality.

The module can be used in two fundamentally different solutions [4]:

The One-Processor Architecture: The application resides together with the Bluetooth protocols (RFCOMM, SDP, L2CAP, LM, BB and RF) in the module.

The Two-Processor Architecture: The application resides together with the higher-level Bluetooth protocols in one piece of hardware (the PC). The lower level protocols reside in the module.

We use the module in the first architecture.

4. EXPERIMENTS AND RESULTS

4.1 Simulation Experiments

We first present simulation results that show voice delays when ACL links are used to carry voice. The simulation environment is NS-2 [5], enhanced with the Bluetooth simulator. Our Bluetooth module implements the Baseband, LMP and L2CAP layers of Bluetooth. We present simulation results for voice carried over ACL links. The Bluetooth slave polling strategy used by the master is the one given in [6]. It tries to assign slots to slaves based on their traffic history and activity.

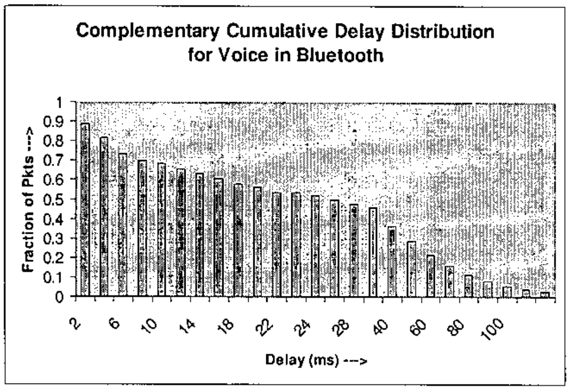

Fig 1: Voice delay distribution when voice is carried over ACL links

The topology consists of a single piconet, having 7 slaves. We consider TCP and voice connections, with each connection having a slave as source and another slave as destination. The source and destination points are randomly chosen. The number of TCP and voice connections is in the ratio 1:1. The TCP data connections are active large file backlogs, with 500-byte packets. The voice connections are modeled according to the Brady model [7]. The voice connections are "on-off" sources; the on and off times are exponentially distributed, with mean 1 s and 1.35 s respectively. The voice-coding rate is 8 Kb/s and the packetisation period is 20 ms, which gives a payload size of 20 bytes. Header compression is assumed for voice packets in Bluetooth and the total packet size is 30 bytes. Voice packets are sent using RTP over UDP. Each experiment lasts 32 seconds of simulation time. The connections start at 0.5s and run till the end of the simulation.

Fig 1 shows the voice delay distribution when the total number of connections in the piconet is varied from 5 to 15. Note that the figure shows the delay distribution for all voice connections over all experiments. It is easy to see that even for large number of connections in a piconet, voice delays do not exceed 100 ms or so, which is quite acceptable. From the figure, we note that a packet loss ratio of less than 5% can be obtained for a play-out buffer of about 80 ms in the case of Bluetooth. Note that all these results are for connections internal to a piconet; (we do not consider the case of a scatternet here).

4.2 Hardware Experiments

Our simulations results indicated that ACL links might be sufficient for carrying voice. In this section, we show results when using Bluetooth hardware modules. We implemented a TCP (Tahoe) application and a voice application on the hardware modules. Voice was implemented as 8Kbps coded speech. The speech standard recommends voice packets of size 20 bytes with the coding time as 20msec. But, the hardware modules were unable to process packets every 20msec. We, thus, modeled voice packets of size 100 bytes with the coding time as 100msec. The voice packet delays are thus, larger than they would be if the recommendations of the speech-coding standard were used.

DH1 packets were used when carrying voice over ACL links. For TCP, the packet size was 520 bytes including 20 bytes of header. The packet type used for the TCP connections was also DH1. Each experiment was run for 2 minutes. We had only 3 Bluetooth modules and the topology for the experiments consisted of a piconet with a maximum of 2 slaves.

We first ran a single voice connection between a slave and the master in a piconet consisting of 1 slave. This experiment was run over either an SCO or an ACL link. The SCO packet type was HV1, HV2 or HV3.

Table 1 shows the average packet delay for voice for all the cases. The delay when using the ACL link is higher than that when HV1 packets are used but is almost the same as the HV2 delay. Note that all the delays are larger than what they may be expected to be for voice packets. This is due to the large packet size of 100 bytes used for voice packets, as explained earlier.

	SCO (HV1)	SCO (HV2)	SCO (HV3)	ACL
Avg. Packet Delay for Voice (msec)	7.094	15.061	20.061	15.385

Table 1: Average delay for voice packets

We then considered a piconet with two slaves, each having a voice connection with the master. One slave ran voice over an SCO and the other over an ACL link. The SCO packet type was either HV2 or HV3. Note that we could not run HV1 in this case since it is not possible to have 2 HV1 voice connections in a piconet (an HV1 packet requires 2 slots every 2 slots; thus, no bandwidth is remaining for other connections). In fact, the hardware modules do not allow the setting-up of two voice connections using HV1 packets.

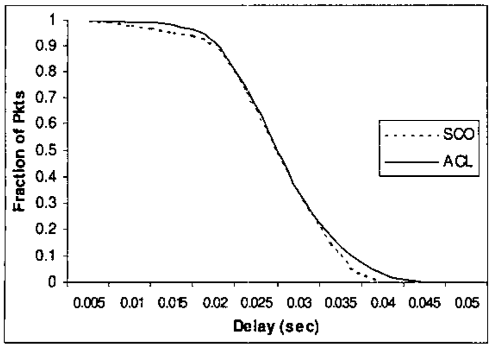

Fig 2: Delay distributions for voice packets for ACL and SCO with HV2 packets

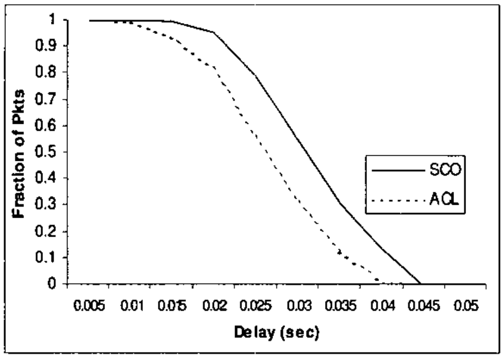

Fig 3: Delay distributions for voice packets for ACL and SCO with HV3 packets

Figs 2 and 3 show the complementary cumulative delay distribution for voice (both for ACL and SCO links) when the SCO packet type is HV2 and HV3 respectively. The delay takes into account the delay of both the voice connections. The ACL delay is slightly larger than the delay with HV2 packets, but better than when HV3 packets are used.

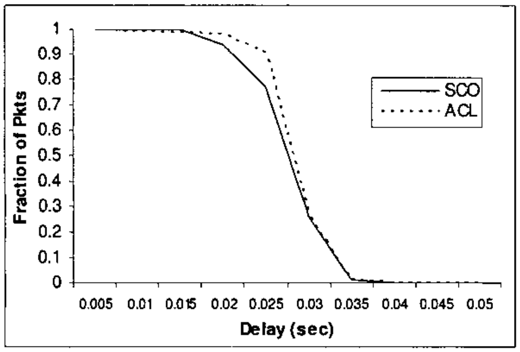

Fig 4: Delay distributions for voice packets for ACL and SCO with HV2 packets

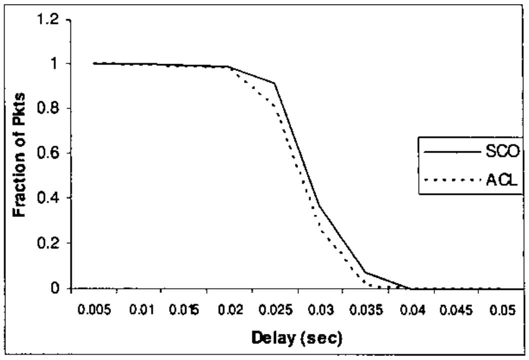

Fig 5: Delay distributions for voice packets for ACL and SCO with HV2 packets

We then considered a piconet with two slaves, one having a TCP connection and the other a voice connection with the master. Voice was carried over either an SCO or an ACL link and the SCO packet type was either HV2 or HV3. Figs 4 and 5 show the complementary cumulative delay distribution for voice (both for ACL and SCO links) when the SCO packet type is HV2 and HV3 respectively. The delays show similar trends as in Figs 2 and 3, i.e., the ACL voice delay is worse than HV2 but better than HV3.

The throughput for the TCP connection showed a dependence on the type of voice connection. Table 2 shows the TCP throughput when the voice connection is SCO (HV2), SCO (HV3) or ACL. TCP shows much better performance if ACL links are used to carry voice. Thus, this experiment shows that voice delays are almost as good when ACL links instead of SCO links are used to carry voice, and the TCP performance is much better. This clearly points to the use of ACL links for carrying voice.

	SCO (HV2)	SCO (HV3)	ACL
TCP Throughput (Kbps)	41.6	52.901	76.197

Table 2: TCP Throughput for different types of voice connections

Note that the lower than expected throughput is due to the use of DH1 packets for the TCP connections. The theoretical throughput achievable using a DH1 packet is about 179 Kbps. Since we also have two slaves in the piconet, the throughput is further reduced.

Though our experiments were restricted to the use of 2 slaves, they indicated that ACL links are a good alternative for carrying voice. In fact, we expect that as the number of slaves (and the number of connections) increases, the performance of TCP when voice is carried over ACL links will show greater improvement over that when voice is carried over SCO links. The voice packet delays when using ACL links are also expected to grow bigger as compared to the delays when using SCO links. But, our experiments showed that voice over ACL links shows almost as good delays as voice over SCO links; in fact, the delays are better for ACL links when HV3 packets are used for SCO links. Thus, unless there is a need for extremely high-quality voice, ACL links are a good choice for carrying voice.

5. CONCLUSIONS

In this paper, we discussed the support of voice connections using ACL links. Using simulations and experiments with Bluetooth hardware, we compared voice packet delays when voice is carried over ACL or SCO links. Our experiments show that though the delays of voice connections are increased slightly if they are sent over ACL links as opposed to SCO links, these delays are still quite acceptable. On the other hand, when voice is carried over ACL links, TCP connections show a marked improvement compared to voice over SCO links. This clearly points to the use of ACL connections for carrying voice unless there is a need for voice to be of extremely high quality.

In fact, using ACL links also has another advantage in that avoiding the use of SCO links makes scheduling algorithms much simpler. This paper does not explore this issue, but in reality, the presence of SCO links would make scheduling both in a piconet and in a scatternet, extremely difficult. This is another hidden advantage of using ACL links for carrying voice.

REFERENCES

[1] *"Specifications of the Bluetooth System"*, Core vol.1 v1.1, www.bluetooth.com

[2] J. Haartsen, *"BLUETOOTH - the universal radio interface for ad hoc wireless connectivity"*, Ericsson Review, n.3, 1998, pp. 110-117.

[3] *"Ericsson Bluetooth Application Toolkits"*, http://www.ericsson.com/microe/products/bluetooth_solutions/pdf/rok101007r1a_ds.pdf

[4] *"Ericsson Bluetooth Host Stack"*

[5] *"Network Simulator (NS-2)"*, www-mash.cs.berkeley.edu/ns/

[6] A. Capone, R. Kapoor and M. Gerla: *"Efficient Polling Schemes for Bluetooth Picocells"*, ICC 2001.

[7] P.T. Brady, *"A model for generating on-off speech patterns in two-way conversation"*, Bell System Technical Journal, Sept. 1969, pp. 2445-2471.