Push-to-Talk over Bluetooth

Valter Rönnholm
Communications Laboratory, Helsinki University of Technology
P.O. Box 3000, FIN-02015 TKK, Finland
valter.ronnholm@hut.fi, valter.ronnholm@iki.fi

Abstract— **Push-to-Talk over Cellular (PoC) is an emerging service enabling a walkie-talkie-like service over GPRS. An open standard for PoC has been specified by the Open Mobile Alliance (OMA) [1]. OMA PoC is based on an IP/UDP/RTP protocol stack and a client-server based architecture. Group management, floor control[1] etc are administered by PoC servers. The systems exploits the SIP signalling capabilities of the IP Multimedia Subsystem (IMS) [2].**

The objective of this paper is to propose an outline for a Push-to-Talk (PTT) system that utilizes a Bluetooth scatternet and the Personal Area Networking (PAN) profile for data communications [3], [4]. The proposed system is thus fully independent of cellular networks. A reasonable range of several hundred meters can be obtained with multihop communications between Bluetooth class 1 devices, whose range is up to 100 m. The proposed outline for Push-to-Talk over Bluetooth (PoB) comprises e.g. methods for group formation, network formation, communication, and floor control. The network formation method, which can be utilized in other applications as well, is based on creating a scatternet among a predefined set of devices and on avoiding loops. This approach enables usage of a simple broadcasting based communication method, in which the devices bridging the piconets into a scatternet act as repeaters.

I. Introduction

The PoC-service has gained significant interest in the field of mobile cellular communications. Within mobile wireless telecommunications, the method in which one person's speech is conveyed to a group of people in a half duplex fashion is ancient. However, PoC rejuvenates this capability by integrating it into e.g. GSM (Global System for Mobile Communications) terminals, which traditionally provide dial-up speech communications. In addition to combining these two capabilities traditionally associated with separate devices and technologies, PoC could some day provide a virtually global walkie-talkie type communications service. OMA has specified an open PoC standard with the aim to ensure interoperability over a variety of mobile access networks [1], [5]. If PoC becomes a success and the OMA PoC standard becomes the prevailing standard worldwide, millions of users will be accustomed to the user interface and user experience of OMA PoC. Therefore, this paper assumes the requirements document of the OMA PoC standard as a baseline for user experience.

In the somewhat different field of short-range wireless communications, Bluetooth is becoming the preferred local connectivity technology in mobile phones, laptops and other devices. One of the profiles offered by the Bluetooth specification is the PAN profile, which enables IP-based applications to exploit Bluetooth connectivity without remarkable adaptations or modifications. From the perspective of superior protocol layers and applications, the PAN profile gives Bluetooth connections the appearance of an Ethernet-network. Among the public, Bluetooth connectivity is currently associated with a range of approx. 10 meters. The reason for this is that the majority of commercially available end-user devices use the middlemost power class of the three classes defined by the Bluetooth core specification. However, Bluetooth devices and modules with power class 1, which provides a range of approx. 100 meters, are currently commercially available as well.

The above developments lead to the notion of providing a free-of-charge, short range PTT-functionality over Bluetooth, which is the subject of this paper.

A. Related Work

Kargl et al [6] discuss conveying point-to-point voice over Bluetooth. Particularly, a routing protocol called Bluetooth Scatternet Routing (BSR) is proposed for routing voice data over multiple hops on Synchronous Connection Oriented (SCO) links. The proposed protocol is similar to Ad-Hoc On-Demand Distance Vector Routing (AODV) [7] and Dynamic Source Routing (DSR) [8] and sets up a scatternet on demand by flooding Path Request messages over Asynchronous Connection-less (ACL) links to devices in the vicinity. If a request message reaches the desired recipient, it sends a Path Found message over the discovered path to the originator. In contrast to AODV and DSR, BSR does not necessarily find the shortest path but settles for the first discovered path, as it probably is one of the shortest. As soon as the first path is found, the flooding algorithm is aborted with Path-End messages. The reason for this approach is that the delay induced by the setup of the ACL connections, increases the overall delay for finding the path. The algorithm is thus expedited by reducing the number of ACL links needed. Once a path is found, SCO links may be set up for voice traffic. In the evaluation of BSR, the end-to-end connection setup times are found to be too long. This is bound to produce problems in the case of a broken link whereby an alternative path is needed. Some optimizations are outlined for remedying this problem. It is also stated that the end-to-end connection setup times become quite large when the number of nodes is increased. As an example, the average delay for finding a path in an environment with 20 or 55 nodes is 5 s and 10 s

[1]In the OMA PoC specifications, the term "talk burst control" is used alongside "floor control" in the identical meaning.

respectively. Kargl et al also state, that a number of changes to the Bluetooth 1.1 specifications are needed to make BSR efficient. Even though the Bluetooth 1.2 specification halves both the Inquiry and Paging times [9], the delay for finding a path with BSR seems to be quite long, particularly in an environment with a dense penetration of devices supporting BSR. Even though it might be acceptable to have a rather long delay for the initial connection setup, the procedure for replacing broken links with alternative paths is unacceptably slow.

B. Research Problem

When an enabling technology gains popularity and reaches significant market penetration, an obvious consequence is the aspiration to develop innovative features and services based on the technology in question. In many cases, it is more lucrative to enhance existing and proven features and services than to develop and nurture totally new ones. Enhancing PTT by providing it for free over Bluetooth seems to be a promising alternative for exploiting the opportunity emerging by the rising popularity of Bluetooth. The range of Bluetooth obviously constrains the PTT functionality to a significantly smaller area than e.g. PoC. However, taking into account that most oral communication takes place in local context, this may not be an as drastic disadvantage as first perceived. Two or more people located at a distance of say 50 to 100 meters from each other may easily communicate without any aids, if the environment is noiseless and no bystanders need to be taken into account. However, if the environment is noisy or crowded, discussing at this distance may not be very convenient. For example, if there is background noise such as loud discussions, music, traffic noise or wind, discussing at these distances may be difficult or even impossible. On the other hand, a loud ”long-distance” discussion in a somewhat quieter environment, such as a department store, crowded outdoor area or an open-plan office, may be considered as indiscrete or rude, or may attract unpleasant attention.

The general research problem of this paper is:

- How can mobile phone users be provided with a free-of-charge PTT-feature with PoC-like user experience by means of Bluetooth technology?

The objective of this paper is thus to specify how Bluetooth can be used to enable a PoC-like feature free of charge. The fact that it is not possible to achieve all the functionalities and capabilities of PoC in a Bluetooth-based solution is appreciated. Therefore some of the PoC functionalities are curtailed and some drawbacks are accepted. The PAN profile of Bluetooth is chosen as a platform for the specification and possible implementation. The reason for this selection is that PoC is designed for an IP network, which seems to imply that adapting the PoC protocol stack onto the IP-enabling PAN profile should be relatively straightforward compared to other alternatives.

Security issues are important whenever telecommunication systems are designed and specified. However, the level of security is always a question of policy. Traditionally, basic walkie-talkies have been associated with no security whatsoever. The extreme alternative would thus be to accept the possibility of eavesdropping and to communicate this flaw to the user. Outlining a security policy for PoB is therefore outside the scope of this paper.

Speech coding plays an important role since its characteristics, such as capacity requirements, error tolerance and processing requirements, set requirements on the underlying protocols and thus affect the system design significantly. In order to limit the scope of this paper, the codec selection issue is not discussed in detail.

II. Bluetooth

Bluetooth employs a master-slave approach to communication between devices and a master can have up to seven active slaves. Such a network is called a piconet. A device can be the master of only one piconet but can simultaneously be a slave in a number of other piconets. Such a network consisting of several interconnected piconets is called a scatternet. Inquiry and Page procedures are used for discovering nearby devices and connecting to them. The Inquiry procedure provides a list of nearby devices that are receptive to Paging. The inquiring device sends inquiry requests and nearby devices respond with messages including their Bluetooth device address (BD_ADDR). The Paging procedure establishes a connection between the initiating device, which becomes the master, and the responding device, which becomes a slave. The recipient of a Page message is indicated by the BD_ADDR. The Bluetooth core provides the profiles e.g. with asynchronous links using the Logical Link Control and Adaptation Protocol (L2CAP). The baseband level in turn provides L2CAP with an asynchronous connection-less (ACL) link.

The Host Controller Interface (HCI) is an optional but commonly used interface e.g. between a Bluetooth chip (i.e. controller) and a processor controlled device (i.e. host) exploiting the Bluetooth capabilities of the Bluetooth chip. HCI provides commands e.g. for setting up connections, sending data and monitoring link quality. When a connection is set up, the controller assigns a 12 bit Connection Handle, which is used for identifying the recipients and originators of packets. The host can also set the Flush Timeout for baseband flow control of ACL packets i.e. define how long the controller attempts to transmit a message. When the timeout occurs, a Flush Occurred event is given to notify the host that transmission has failed. [3]

A. Personal Area Networking

The PAN profile [4] specifies how IP networking capabilities can be implemented in Bluetooth devices with the Bluetooth Network Encapsulation Protocol (BNEP) [10]. Three types of PAN units are defined: Network access points (NAP), units providing group ad-hoc network (GN) service, and PAN users (PANU). A unit providing GN service is a piconet master with 1-7 PANU slaves and possibly additional non-active PANU slaves in park mode. The GN master is able to forward data from one PANU to another with BNEP. PANUs are units that are able to use the NAP and GN services.

The PAN profile uses Ethernet packets and NAPs or GN service nodes act as Ethernet bridges.The GN acts as a learning bridge with BNEP connections to the PANUs. Multicast and broadcast addresses may also be used. Methods for allocating and resolving IP addresses for devices within a piconet are also defined. The BD_ADDR is regarded as the Medium Access Control (MAC) address. A host may be allocated several link-local IP addresses [11] if it has interfaces to several Bluetooth piconets employing the PAN profile. This is referred to as multihoming. Multihoming causes the problem of scoped addresses, which can be solved by exposing the address-interface associations to the application level and letting the application handle the problem. Automatic network formation, general ad-hoc networking with multiple piconets (e.g. scatternets), and QoS mechanisms are not defined by the current specification.

III. Push-to-Talk over Cellular

As stated in the charter of the PoC working group of OMA: ”PoC service is a half-duplex form of communications that allows users to engage in immediate communication with one or more receivers, similar to Walkie Talkie type operation, simply by pushing a button on their handsets.” [5] The system is client-server based and uses Session Initiation Protocol (SIP) signalling enabled by IMS. PoC servers handle PoC and SIP sessions, floor control etc.

The OMA PoC requirements document defines floor control as a ”mechanism for the arbitration of the sequence of PoC participants to speak”. To put it simply, the PoC server acts like a chairman that gives the floor to the participants of a session on the basis of their requests to speak. Basically, each PoC session participant is able to request the floor upon which the PoC server either grants a permission to speak or, if the floor is already occupied by another user, denies the request. The term ”talk burst” is used to denote a speech message conveyed from one user to others. [12], [13]

The OMA PoC transport plane protocol stack is based on the Internet Protocol (IP), User Datagram Protocol (UDP), Real-time Transport Protocol (RTP), and RTP Control Protocol (RTCP). The default codec is Adaptive Multi-Rate (AMR). [14] A Talk Burst Control Protocol (TBCP) is defined for floor control [15].

A. Performance Requirements

The OMA PoC Requirements document specifies certain performance requirements in order to ensure a satisfactory quality of experience (QoE) for the user.

- QoE1: Right-to-Speak Response Time
  The right-to-speak response time when establishing a PoC session. That is, the time from the instant the user initiates a session to the instant he/she receives a right-to-speak indication. This time should typically be less than 2.0 seconds.
- QoE2: Start-to-Speak Response Time
  The start-to-speak response time in an active PoC session. That is, the time from the instant the user request the right to speak to the instant he/she receives either a start-to-speak indication, queuing indication or reject indication. This time should typically be less than 1.6 seconds.
- QoE3: End-to-End Channel Delay
  The end-to-end channel delay, i.e. the time from the instant one PoC session participant starts to speak until the instant the recipient(s) starts to hear the speech in question. This time should typically be less than 1.6 seconds.
- QoE4: Mean Opinion Score
  The mean opinion score (MOS) is a commonly used measure for evaluating voice quality. The requirement is that MOS should be at least 3 at a bit error rate (BER) of 2 per cent or less.

One additional requirement that is dependent on user behavior, is also defined, namely the turnaround time. The turnaround time is defined as the time from the instant a user releases the floor after he/she has stopped speaking until the instant the same user hears the beginning of a reply from another user. That is, the time from e.g. releasing the PTT button until hearing the following user’s voice. The turnaround time should typically be less than 4 seconds. [12]

IV. Push-to-Talk over Bluetooth

In this section, a proposal for deploying a PoC-like PTT feature over the Bluetooth PAN profile is outlined. The problem is approached from a PoC point of view, i.e. whenever possible the features and characteristics of OMA PoC are adhered to and modifications are made only when required by the limitations of Bluetooth. In this way, the user experience of PoB resembles that of PoC as much as possible. Some of the functionalities provided by the PoC servers are curtailed in PoB.

A. Transport Plane Protocols and Voice Coding

Since the Bluetooth PAN profile provides a platform for IP communications, there are no obstacles for employing the IP/UDP/RTP protocol stack providing that the throughput is sufficient. The AMR codec may thus also be used without complications.

The throughput requirement of the above protocol stack depends on the AMR interface format (IF) and mode and on how many AMR frames are conveyed per RTP packet. The throughput requirements for AMR IF2 with different AMR modes and frames per packet values are presented in Figure 1. The throughput calculations are presented in the Appendix. The highest requirement with these variables is 36.8 kbps, which occurs for AMR mode 12.2 with only one AMR frame per RTP packet. The lowest requirement on the contrary is 6.08 kbps, which corresponds to AMR mode 4.75 and 50 AMR frames per RTP packet, which implies a buffering delay of 1 s at the sender.

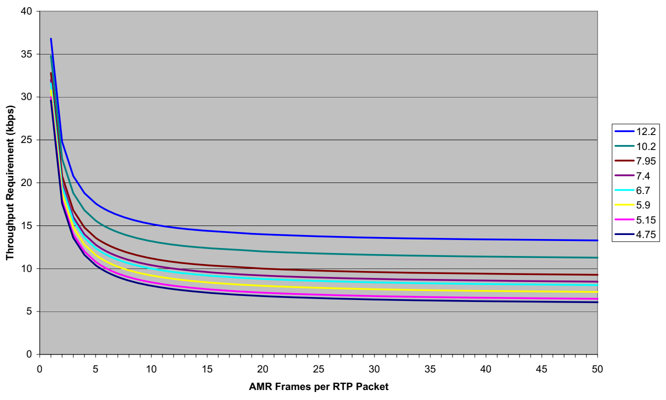

Fig. 1. Throughput Requirements for Different AMR Modes with Different Amounts of AMR Frames per RTP Packet

According to the PAN profile, broadcasting is implemented by multiple unicasting[2] on the baseband level. The throughput requirement for broadcast is thus multiplied by a factor corresponding to the number of slaves, if the master is the originator of the data. If the originator is one of the slaves the corresponding factor is $s-1$, where s is the number of slaves in the piconet, since the data is not sent back to the originator.

One additional factor affecting the performance requirements in a scatternet is the bridge scheduling algorithm. In a scatternet, some devices act as bridges between two piconets and they must thus balance their presence in the piconets since they cannot be active in several piconets simultaneously. Scheduling refers to the method according to which a bridging device allocates its time between the two piconets. Since scheduling is not defined in the Bluetooth specifications, a number of scheduling algorithms have been proposed. An overview of these is given in a report by Misic et al [16]. Scheduling algorithms can be divided into algorithms with uninterrupted and interrupted transmission. Uninterrupted transmission means that once a need for transmission is detected in one piconet, a bridge remains active in this piconet until the need is fulfilled, i.e. all incoming data is accepted by the bridge. On the contrary, interrupted transmission means that a bridge may be inactive in any of the two piconets for a maximum period only. If this period is exceeded, the bridge must interrupt the transmission of data and become active in the other piconet. Since a bridge is active only in one piconet at a time, access delays occur. A sender must wait for the bridge to become active in the piconet in question, if the bridge is currently active in another piconet. In order to take access delays into account, the estimated throughput requirement for bridges is doubled compared to non-bridge devices. This estimate is very rough and is based on the assumption that a bridge spends half the time in each piconet and the time for transmitting data is thus halved. Since a device can be both a master and a bridge, the worst case throughput requirement can be roughly approximated by multiplying the throughput requirements of Figure 1 by a factor of $s+1$.

The theoretical maximum throughput of a Bluetooth ACL link is dependent on the ACL packet type. For DM1 (Data - Medium Rate 1) packets the maximum rate is 108.8 kbps and for DH5 packets (Data - High Rate 5) 723.2 kbps. Zürbes [17] simulated the throughputs for ACL links with different packet types in a scenario with a number of piconets in a single room of 10 m $\times$ 20 m. In the simulations, the piconets consisted of one master with one slave, which communicate unidirectionally (master to slave) at maximum rate (i.e. each time-slot is occupied either by data sent by the master or an acknowledgement sent by the slave). According to Zürbes, the values are valid for the case of multislave piconets as well, because only one device at a time transmits in a piconet. The results of Zürbes suggest that e.g. the worst case requirement of 67.2 kbps[3] corresponding to AMR mode 5.15 and 10 AMR frames per RTP packet, would be achievable with DM1 packets even if 50 interfering piconets exist in the vicinity.

A model for calculating the throughput of a similar scenario is presented by Zanella et al [18]. The model suggests clearly lower throughput results than Zürbes simulations. In their analysis of the model Zanella et al found, however, that the model becomes less accurate as the number of interfering devices increases.

The above mentioned studies seem to suggest that the throughput requirements stated earlier are achievable if the AMR mode, number of AMR frames per RTP packet, and baseband packet are suitably chosen. However, simulations and/or tests are needed to ensure sufficient throughput.

Chen et al [19] propose an adaptive Automatic Repeat Request (ARQ) scheme for audio streaming over Bluetooth. The scheme seems to improve throughput and delay considerably in noisy environments and could thus be beneficial. The idea is basically to measure the Round Trip Time (RTT) of e.g. RTP packets and adapt the HCI Flush Timeout depending on the RTT. The disadvantage of this scheme is that it increases the complexity and buffering requirements somewhat.

In addition to the throughput requirements, delay issues must be taken into account in order to ensure that the QoE requirements are met. However, the delays occurring in a Bluetooth scatternet are relatively small compared to the GPRS environment. The end-to-end requirement of 1.6 s should thus give plenty of room for e.g. access and transmission delays (assuming that the amount of non-PoB traffic in the scatternet is marginal, and the access delays thus small). As an example, the tests of Kargl et al [6] produced an average delay of 28 ms, when transferring data between a master and one slave in an interference free environment.

B. Control Plane and Signalling

Since the OMA PoC architecture relies on client-server based signalling, which is not available as such in the Bluetooth domain, an alternative way to govern communications

[2]In other words, broadcasting is implemented by sending data to each recipient separately.

[3]Assuming that the device is acting as both a bridge between two piconets and as a master for seven slaves: 8×8.4 kbps $= 67.2$ kbps

must be used. In an architectural sense, a simple approach is to avoid the client-server model altogether and utilize a peer-to-peer based signalling scheme. Instead of SIP signalling, an application specific signalling method is used on the application level. To simplify this signalling scheme as much as possible, the group creation, scatternet formation, and communication methods are designed with the objective of reducing the need for signalling traffic. This implies avoiding complexity in general. The advantage of this approach is that the diverse problems associated with maintaining centralized or cluster based control in a dynamic ad-hoc network are avoided. Schemes for cluster based control are discussed e.g. by Steenstrup [20], who states that the deployment of these methods has been limited to experimental setups. This technology is thus far from mature which implies a reliability risk. The disadvantage of the peer-to-peer alternative, on the other hand, is that a centralized point of control cannot be used to handle e.g. floor control. This problem is discussed in Section IV-F.

C. Group Formation and Addressing

In OMA PoC, groups are administered by PoC servers, which are accessible by the clients for group management. Obviously, maintaining such a server is not possible in a peer-to-peer solution. Therefore, a new method for creating and maintaining groups is needed.

PoB Groups are defined by the BD_ADDRes of the group members. A group is thus defined by a GROUP list containing BD_ADDRes. The creation of such a list can be performed by sending an invitation message over e.g. Bluetooth, SMS, or an Infrared Data Association (IrDA) link to the desired group members. Upon reception of an invitation, the PoB application queries the user whether the group should be joined or not. If the user's reply is affirmative, a reply including the BD_ADDR is sent as a response to the invitor. When all invitees have replied, the group is defined as a GROUP list containing the BD_ADDRes. To complete the group formation, the list of BD_ADDRes is sent from the invitor to all the other group members. Groups may also be modified in a corresponding manner later to add or remove members.

The group members could also exchange e.g. a randomly generated encryption key in the group formation phase. This could be done securely over SMS or IrDA. Since the security issues are outside the scope of this paper, these issues are not discussed further.

D. Network Formation and Maintenance

A wide range of methods for Bluetooth scatternet formation have been proposed in recent years. An overview and bibliography of these is given in Chapter 3 of [21] and additional methods are presented in e.g. [22], [23]. This area of research has received a considerable amount of attention lately but a clearly superior solution, balancing complexity and reliability of connectivity, has not been presented. Therefore, the network formation problem is approached with an aim for simplicity in this paper.

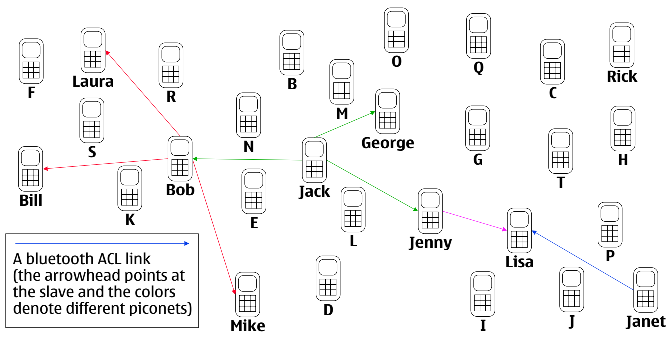
F
Laura
R
B
O
Q
C
Rick
M
George
S
N
G
H
T
Bill
K
Bob
Jack
E
L
Jenny
Lisa
P
A bluetooth ACL link (the arrowhead points at the slave and the colors denote different piconets)
Mike
D
I
J
Janet

Fig. 2. An example of a Scatternet According to the Presented Method

Since each device knows which devices it desires to communicate with, connections are established with these devices only. The scatternet is thus formed by including only devices with BD_ADDRes that are on the GROUP list. However, in order to enable easy broadcasting, loops are not allowed in the network. An example of such a network is presented in Figure 2. In the figure, the names denote group members' devices, the single letters outsiders'[4] devices, the arrows Bluetooth ACL links, the arrowheads the slaves, and the arrow color different piconets. As can be seen, all group members but one ("Rick" in the upper right corner) are connected to the Scatternet. A method for forming such a network is presented in the flowchart of Figure 3 and explained below. In order to form and maintain the desired network, each PoB device runs this method.

Each device needs to keep track of which devices are reachable via the scatternet and which are not. This can be done by maintaining CONNECTED and UNCONNECTED lists, which include the BD_ADDRes of the devices that are and are not connected to the scatternet respectively. Each entry in the CONNECTED list is associated with two additional BD_ADDRes which indicate via which device the device in question is reachable and which device initiated the connection that connects the device in question to the network. An example of a CONNECTED list is presented in Table I, which corresponds to the perspective of Jack's device in Figure 2. The left hand column indicates which devices are reachable and the middle column indicates via which link the device in question is reachable. The right hand column is related to loop avoidance which is discussed in more detail later on in this section. As an example, the last entry of Table I indicates that the device with the BD_ADDR "00001000..." (i.e. Janet) is reachable via the device with the BD_ADDR "00000011..." (i.e. Jenny). It should be emphasized that the device maintaining the CONNECTED list of Table I (i.e. Jack's device) has a *direct* ACL link to the device with the BD_ADDR "00000011..." (i.e. Jenny) and all other devices in the middle column. However, the devices in the left hand column are not necessarily *directly* connected

[4]The term "outsider" refers to people whose devices are not members of a particular PoB group.

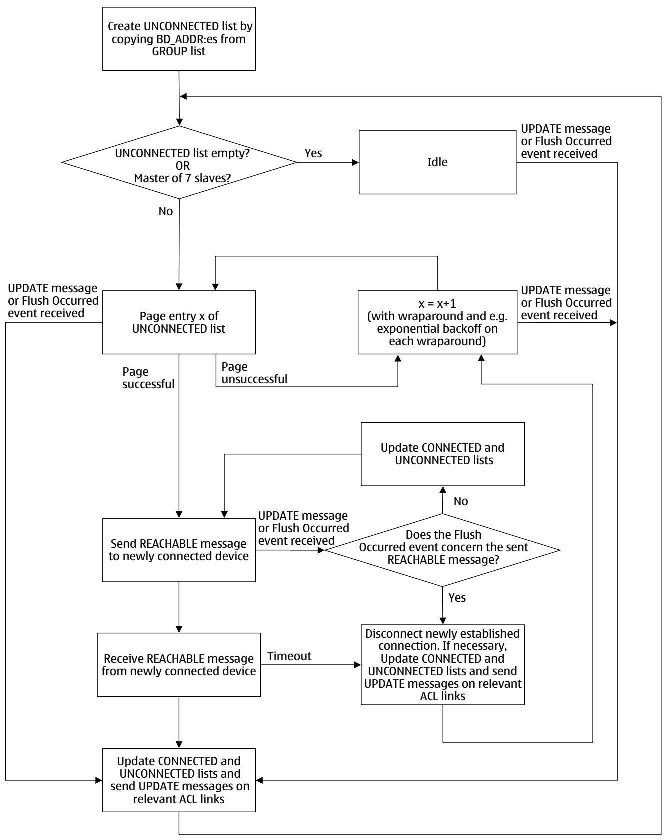
Create UNCONNECTED list by copying BD_ADDR:es from GROUP list
UNCONNECTED list empty? OR Master of 7 slaves?
Yes
Idle
UPDATE message or Flush Occurred event received
No
UPDATE message or Flush Occurred event received
Page entry x of UNCONNECTED list
x = x+1 (with wraparound and e.g. exponential backoff on each wraparound)
UPDATE message or Flush Occurred event received
Page successful
Page unsuccessful
Update CONNECTED and UNCONNECTED lists
No
Send REACHABLE message to newly connected device
UPDATE message or Flush Occurred event received
Does the Flush Occurred event concern the sent REACHABLE message?
Yes
Receive REACHABLE message from newly connected device
Timeout
Disconnect newly established connection. If necessary, Update CONNECTED and UNCONNECTED lists and send UPDATE messages on relevant ACL links
Update CONNECTED and UNCONNECTED lists and send UPDATE messages on relevant ACL links

Fig. 3. Network Formation and Maintenance Method

with device ”00000000. . . ” (i.e. Jack).

A device initiates the method by creating an UNCONNECTED list by copying the GROUP list. Then the device attempts to connect to one of the devices on the UNCONNECTED list by initiating a Bluetooth Page process.[5] If the attempt is unsuccessful, another device from the UNCONNECTED list is Paged. This cyclic Paging procedure continues until the UNCONNECTED list is empty or the device is a master of seven slaves[6]. To reduce power consumption and avoid causing interference, the frequency of Page attempts can be gradually reduced e.g. by exponential backoff (i.e. by waiting for a certain period of time between Page procedures and doubling this period every time the list wraps around). The user may, however, be allowed to reset this period in order to ”refresh” the network.

Upon a successful Page process and the creation of an ACL connection, the devices exchange REACHABLE messages to inform each other about their connectivity information. The REACHABLE messages include a list of BD_ADDRes of the devices that are reachable via the device sending the message and the corresponding Connection Initiator information. In other words, a REACHABLE message includes the left hand column and right hand column of the senders CONNECTED list. After the exchange of REACHABLE messages, the devices can thus update their CONNECTED and UNCONNECTED lists by adding and removing the appropriate BD_ADDRes respectively. The REACHABLE messages of two newly connected devices should never include overlapping BD_ADDRes because this would imply that a loop has been formed. Loop avoidance is discussed in more detail later on.

[5]It may be beneficial to perform Inquiry prior to Paging in order to avoid unnecessary Pages.

[6]It may be beneficial to limit the number of slaves to less than seven, in order to improve performance.

TABLE I
AN EXAMPLE OF A CONNECTED LIST (THE BLUETOOTH DEVICE ADDRESSES ARE SIMPLIFIED FOR CLARITY)

Connected Device (BD_ADDR)	**Reachable Via (BD_ADDR)**	**Connection Initiator (BD_ADDR)**
00000000. . . (Jack)	00000000. . . (Jack)	
00000001. . . (George)	00000001. . . (George)	00000000. . . (Jack)
00000010. . . (Bob)	00000010. . . (Bob)	00000000. . . (Jack)
00000011. . . (Jenny)	00000011. . . (Jenny)	00000000. . . (Jack)
00000100. . . (Mike)	00000010. . . (Bob)	00000010. . . (Bob)
00000101. . . (Laura)	00000010. . . (Bob)	00000010. . . (Bob)
00000110. . . (Bill)	00000010. . . (Bob)	00000010. . . (Bob)
00000111. . . (Lisa)	00000011. . . (Jenny)	00000011. . . (Jenny)
00001000. . . (Janet)	00000011. . . (Jenny)	00001000. . . (Janet)

When a connection has been established or broken, an UPDATE message is sent to all directly connected devices (i.e. the PoB devices to which a direct ACL link exists) in order to disseminate information about the change in the network. The UPDATE message notifies other devices of changes in the CONNECTED list. If a new connection has been established, the UPDATE message contains the BD_ADDR of the newly connected device, the list of BD_ADDRes of the REACHABLE message received from it and the corresponding Connection Initiator information. If a connection is broken, the UPDATE message correspondingly contains the BD_ADDR of the lost device and the possible BD_ADDRes of the devices which used to be reachable via the lost device. On the basis of the UPDATE message, the recipients can update their own CONNECTED an UNCONNECTED lists and thus keep track of which devices are reachable. An UPDATE message includes a list of BD_ADDRes with indications whether they should be added or removed from the recipients CONNECTED list and, in the case of adding, the Connection Initiator information. UPDATE messages are also sent on relevant links when the CONNECTED list is updated on the basis of a received UPDATE message. Information regarding changes in the network are thus conveyed to all the devices connected to the network.

As mentioned in Section II a Flush Occurred event is given on the HCI when a message has been discarded due to unsuccessful transmission. In order to maintain an sufficiently accurate picture of the network i.e. keep the CONNECTED lists up to date, breakage of links must be detected rather quickly. To keep track of the link statuses, some traffic thus needs to be sent periodically over all links. The link maintenance messages cause Flush Occurred events on the HCI when a device is no longer reachable. When this event occurs, the CONNECTED and UNCONNECTED lists are updated and UPDATE messages are sent. However, since a device may be a member in several piconets, it is possible that a device is not reachable because it is currently sending or receiving data in another piconet [24]. To avoid unnecessary disconnections, the period between the maintenance messages should thus be optimized. It is possible that in order to ensure stable functioning, a link cannot be regarded as broken until a number (e.g. three or five) of sequential maintenance messages have been flushed (in Figure 3 this alternative is not used).

In most cases, the CONNECTED and UNCONNECTED lists are updated and the UPDATE messages sent right away when an UPDATE message or Flush Occurred event is received. Any other tasks are thus interrupted as presented in Figure 3.

However, if the device has just connected to another device and the REACHABLE message has not yet been successfully sent, the procedure differs somewhat. If a Flush Occurred event is received in this situation, the device should first check whether the event concerns the REACHABLE message that is currently being sent (e.g. by comparing the Connection Handle parameter of the event with the Connection Handle associated with the BD_ADDR of the device to which the REACHABLE message is being sent). If this is the case, the newly established connection should be disconnected and the device should return to the cyclic Page procedure. If, however, the interrupt is caused by a received UPDATE message, the device should flush the REACHABLE message it is currently sending, update its CONNECTED and UNCONNECTED lists, and then send a new REACHABLE message according to the updated CONNECTED list. In this way some unnecessary traffic can be avoided. Similarly, if the device has sent a REACHABLE message to a newly connected device but a REACHABLE message is not received as a response within a certain period of time, the connection is disconnected. In this case the CONNECTED and UNCONNECTED lists may need to be updated and UPDATE messages sent, if an UPDATE message was received while awaiting a REACHABLE message from the newly connected device.

1) Loop Avoidance: In order to avoid and eliminate loops, the PoB devices should adhere to the following rules.

- A device should not Page devices that are on its CONNECTED list.
- A device should ignore Page requests from devices that are on its CONNECTED list.
- If a device is in the process of establishing a connection with another device (i.e. it has received or sent a Page request but has not yet received e.g. a HCI Connection Complete event and exchanged REACHABLE messages) the device ignores all incoming Page requests.
- If a device receives an UPDATE message including an entry that is in conflict with an entry in its current CONNECTED list, the conflict must be resolved e.g. by prioritizing the entry with the larger BD_ADDR value in the Connection Initiator field. For example, if a device receives an UPDATE message indicating that a BD_ADDR should be added to the CONNECTED list but the identical BD_ADDR is already listed on the CONNECTED list, the Connection Initiator values of the CONNECTED list and the UPDATE message must be compared. If the Connection Initiator value of the UPDATE message is smaller, the UPDATE message is considered incorrect and its originator is sent a correct UPDATE message, which includes the correct data from the CONNECTED list. Correspondingly, if the Connection Initiator value of the UPDATE message is larger than that of the CONNECTED list, the UPDATE message is considered correct. In this case the CONNECTED list needs to be corrected and corresponding UPDATE messages must be sent over the relevant connections. [7]

2) Network Formation Delay: One issue related to network formation is when the network should be formed. One alternative is to form the network as soon as the PoB application is initiated. The problem with this alternative is that maintaining the network consumes some power since some periodical traffic is needed in order to keep the network updated. The other extreme alternative is to form the network on demand i.e. when the PTT button is pressed. Even though this alternative would be optimal from a power consumption point of view, it incurs the network formation delay issue. According to the calculations of Busboom et al [25], the average Paging delay without prior Inquiry is 1.28 s for Bluetooth version 1.1, if transmission is error free and the devices do not have any SCO links. Kargl et al [6] reached a corresponding value of 1.1 s in their tests. Even though the Paging delay is approximately halved in Bluetooth version 1.2 [9], the delay is unacceptably long for the alternative of creating the network on demand. Therefore, a solution balancing between the two extremes of on-demand and immediate network formation is selected.

The proposal is that the network is formed on-demand when the user either selects e.g. a ”Connect” command or presses the PTT button. When the network has been formed, the user can be given e.g. a ”Connected to PoB Network”-notification. The established network may then be maintained until a period of e.g. 15 or 30 minutes has elapsed without any incoming or outgoing PoB traffic. This period may be user definable. When the period has elapsed the user can be given e.g. a ”Disconnection from PoB network in 30 seconds! Continue? Yes/No”-indication. The user can thus be given the chance to maintain the network for another period by selecting the ”Yes”-option.

The advantage of the method described above is its simplicity compared to alternative solutions. The disadvantage is that due to implementation restrictions, it is possible that a device is left outside the scatternet even though it is within Bluetooth

[7]Despite the other rules, loops may form e.g. in case two separate networks are merged by two separate paging procedures at different locations in the networks. This rule is therefore needed to eliminate such loops. Since it takes some time to eliminate these loops, a timeout may be needed after each reception of an UPDATE message in order to ensure proper functioning.

range of a device in the scatternet.

3) Extending the scatternet: The range of PoB could be extended by using outsiders' devices for reaching more distant group members (such as Rick in Figure 2). First a scatternet would be formed with any available devices belonging to the group members. If unreachable devices still remain in the UNCONNECTED list, e.g. an AODV-, DSR- or BSR-based algorithm would be used to locate them via outsiders' devices. Paths including outsiders' devices would then be maintained, in order to be able to communicate with these more distant group members. The advantage of this approach would be that the range perceived by the user would be extended further. Exploiting outsiders' devices increases their power consumption and incurs thus the question of whether outsiders would allow such exploitation. A reciprocity policy (allowing the own device to be exploited by other devices that in turn allow others to exploit themselves) might serve a solution to this problem. However, it is assumed that PoB group members will tend to stay within range with of each other in order to maintain PoB service. Outsiders, on the other hand, do not share this interest and are thus more likely to move farther away from the PoB group members than the group members. This would cause more link breakages and thus more signalling traffic. In the worst case an outsider is merely passing by and providing a very temporary link to a group member farther away. In this case, adding the outsider's device to the scatternet only causes extra signalling traffic without bringing any real benefit. The implications of this must be evaluated by simulations and/or testing in order to find out whether including outsiders' devices in the scatternet is beneficial or not.

E. Communication

Since the network formation method of Section IV-D creates a network that contains all the reachable group members without allowing loops, the communication scheme itself is quite simple. Since the formed Scatternet is loop-free and consists of group members only, broadcasting can be used to convey PoB data. The broadcasting capability of the PAN profile [4] is thus used to enable IP broadcasting [26] to all devices in a particular piconet. Since the PAN profile does not support Scatternets, the inter-piconet communication must be handled on a higher layer e.g. by the PoB application. A straightforward way to solve this problem is to let every device that is a member in several piconets act as a repeater between the piconets. In other words, any PoB messages (both speech data and control) that are received from one piconet are broadcasted on the other piconets. In this way, all messages are broadcasted over the entire scatternet. The obvious advantage of this solution is its simplicity.

F. Floor Control by Request to Speak

In PoC, floor control is one of the services provided by the PoC server. In the PoB environment, a similar method is needed for controlling who is allowed to talk and when in order to avoid talk burst collisions.

One way to solve this problem is to broadcast a time stamped Request-to-Send (RTS) message, when the user pushes the PTT button. When such as message is received, the recipients await the incoming talk burst and refrain from sending RTSs themselves. After sending an RTS message, the originator waits until a timeout expires in order to ensure that incoming RTSs were not underway when the RTS message was sent. If an RTS message is received from another device within this timeout period, the time stamps are compared and the sender of the earlier message takes the floor.

The time stamping procedure naturally requires a way of synchronizing the time stamping clocks or counters of the PoB devices. This can be done by sending synchronization data periodically (possibly piggybacked in link maintenance or other messages). If two users press the PTT buttons of their devices within an interval of say 10 ms, the users perceive the presses as more or less concurrent. Therefore, the accuracy requirement for the time stamp clocks is rather lenient. A simple method for synchronizing distributed clocks with each other is to let each device periodically broadcast the reading of its own clock. The recipients then compare the received value with their own clock reading. If the received value is larger than the reading of the own clock, the own clock is set to the received value. Otherwise, the synchronization message is ignored. In this way, all clocks remain within a reasonable tolerance of each other, assuming that they are originally synchronized at some accuracy. To achieve this initial synchronization, a single PoB device adopts the clock value of the device it connects to when joining a scatternet. This value may be included in a REACHABLE message. In some cases, e.g. when two single devices connect or two scatternets merge, the clock values may differ significantly. In order to avoid unstable situations, a special procedure may be used in these cases if the clock values differ e.g. by more than a quarter of the clock register space. (That is, if e.g. an eight bit counter is used as a clock, the special procedure is used if the clocks differ by a value of 64 or more.) In this case a simple rule may be used for selecting the dominant counter value. When exchanging the REACHABLE messages, the counter value of the device with the larger BD_ADDRes value may e.g. be ruled as dominant. In this case the overridden device broadcasts a synchronization message, with an indication that the counters must be set to the sent value, on its original scatternet.

V. Conclusions

A. Evaluation

The outline of PoB proposed in this paper provides a baseline for developing this notion further and possibly implementing it. The proposed outline exploits existing conventions and protocols rather extensively and the new methods were designed with an aim for simplicity. Implementing PoB and developing PoB products on the basis of these outlines may therefore be relatively uncomplicated and cost effective. In addition, this may enable integrating PoB and PoC into one unitary application.

On the other hand, the greatest flaw of the proposal is that there is no factual evidence that the throughput provided by Bluetooth in all or most real-life situations will be sufficient for satisfactory PoB operation. Therefore, the proposal does not provide a solid platform for further development but requires further research to verify its feasibility.

The network formation method of Section IV-D may also be used for various other purposes besides PoB. For example, applications enabling multiplayer gaming, chat or shared folders or calendars may be based on communication over a scatternet formed according to the method. Furthermore, broadcasting according to Section IV-E may also be used in such applications.

B. Significance

Throughout modern history, technology has affected the behavior of people and vice versa. Some technology driven products and services have become commodities of everyday life whereas others have flopped in spite of great expectations. The typical examples of these are the Short Message Service (SMS) and Wireless Applications Protocol (WAP) respectively. PoB could instigate new behavior models within local communication. Currently, telephony is usually associated with more or less long range communication because it does not make sense to pay for a phone call and go through the trouble of calling, if the called party is nearby. Since PoB is free of charge and the message can be conveyed without an actual calling procedure, the threshold for communicating is lower for PoB than for conventional phone calls or PoC.

C. Further Work

As indicated in the above discussion, one obvious issue requiring further research is the performance characteristics of scatternets and the PAN profile in various scenarios and with various settings. Tests and/or simulations are needed in order to acquire reasonably reliable estimates of these. The outcome of such tests or simulations will guide further work. If the requirements of the PoB proposal could be met, the following step would obviously be to develop PoB further on the basis of the proposed outline. On the other hand, if the results would suggest the contrary, PoB could possibly be redesigned to confine within the estimated throughput and delay characteristics. Alternatively, features with lower requirements such as text-based chat, turn-based gaming, and shared folders or calendars could be pursued.

The loop avoidance algorithm may construct networks with sub-optimal topologies and thus cause unnecessary delays in the network. If links are broken as users move in relation to each other, the method of Section IV-D actively creates new connections in order to maintain the PoB service. Link breakage may nevertheless interrupt talk bursts and thus affect the usability of PoB adversely. The frequency and severity of these problems shall also be studied with tests and/or simulations. These issues are naturally highly dependent on the behavior of the users, and the remedies to these problems should thus be accommodated to typical usage scenarios.

The outline presented in this paper merely set a framework for implementing PoB. Many details such as timeout values, intervals for periodical transmissions, and the time stamping counter characteristics remain to be specified and optimized. Other open issues are the Bluetooth baseband packet selection, specifying message formats, AMR mode selection, and the selection of a suitable AMR-frame per RTP-packet ratio.

As stated in the introduction, the security issues are delimited outside the scope of this paper. This issue must nevertheless be considered, if PoB is to be provided to customers. The alternatives range from no security whatsoever, as in many traditional walkie-talkies, to providing a security level equivalent to the conventions of mobile cellular communications.

An additional issue related to implementation is the possibility that other applications exploiting Bluetooth may be run concurrently with the PoB application. Preferably, the PoB application should not interfere with other applications and vice versa. However, if other applications besides PoB create and use Bluetooth connections, a method for administering the connections and the media access is needed.

VI. Acknowledgments

The author would like to express his gratitude to professor Sven-Gustav Häggman, Matias Järnefelt, Nokia, and Iqua.

References

[1] OMA Enabler Release Definition for Push-to-Talk over Cellular, Candidate Version 1.0 – 29 April 2005, OMA-ERELD-PoC-V1_0-20050429-C, 21p.

[2] Poikselkä, M. et al, "The IMS, IP Multimedia Concepts and Services in the Mobile Domain", Chichester, 2004, John Wiley & Sons Ltd, 419p.

[3] Specification of the Bluetooth System, Wireless connections made easy, Volumes 0–4, Version 2.0, 2004, 1230p.

[4] Bluetooth Personal Area Networking Profile, Version 1.0, Bluetooth Special Interest Group, 2003, 65p.

[5] Push to Talk Over Cellular (PoC) Charter, OMA Push to talk over Cellular Working Group, 2003, 3p.

[6] Kargl, F. et al, "Bluetooth-based Ad-Hoc Networks for Voice Transmission", Proceedings of Hawaii International Conference on System Sciences 36, Hawaii, USA, January 2003, 9p.

[7] Perkins, C. et al, "Ad-Hoc On-Demand Distance Vector (AODV) Routing", IETF RFC 3561, 2003, 37p.

[8] Johnson, D. et al, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)", Internet Draft, IETF MANET Working Group, 2004, 112p.

[9] Linsky, J et al, "Bluetooth 1.2: Version 1.2 of Bluetooth has now been launched, but what does it bring?", Wireless Business and Technology, Volume 4, Issue 1, 2004, pp.48–50

[10] Bluetooth Network Encapsulation Protocol (BNEP) Specification, Version 1.0, Bluetooth Special Interest Group, 2003, 55p.

[11] Cheshire, S. et al, "Dynamic Configuration of Link-Local IPv4 Addresses", IETF RFC 3927, 2005, 33p.

[12] OMA Push to Talk over Cellular Requirements, Version 1.0 – 29 March 2005, OMA-RD-PoC-V1_0-20050329-C, 77p.

[13] OMA Push to talk over Cellular (PoC) – Architecture, Candidate Version 1.0 – 28 April 2005, OMA-AD_PoC-V1_0-20050428-C, 156p.

[14] OMA PoC User Plane Version, Candidate Version 1.0 – 28 April 2005, OMA-TS_PoC-UserPlane-V1_0-20050428-C, 161p.

[15] OMA PoC Control Plane, Candidate Version 1.0 – 28 April 2005, OMA-TS-POC-ControlPlane-V1_0-20050428-C, 274p.

[16] Mišić, V. et al, "Polling and Bridge Scheduling Algorithms in Bluetooth", Technical Report TR 03/04, Department of Computer Science, University of Manitoba, September 2003, 17p.

[17] Zürbes, S., "Considerations on link and system throughput of Bluetooth networks", The 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC, Volume 2, September 2000, pp.1315–1319.
[18] Zanella, A. et al, "On the impact of fading and inter-piconet interference on Bluetooth performance", The 5th International Symposium on Wireless Personal Multimedia Communications, Volume 1 , October 2002, pp.218–222.
[19] Chen, L.-J. et al, "Audio streaming over bluetooth:an adaptive arq timeout approach" Proceedings of 24th International Conference on Distributed Computing Systems Workshops, 23–24 Mar. 2004, pp.196–201.
[20] Steenstrup, M. "Cluster-Based Networks" In:Perkins, C. ed., "Ad Hoc Networking", Boston 2000, Addison-Wesley, pp.75–138.
[21] Daptardar, A., "Meshes and Cubes: Distributed Scatternet Formations for Bluetooth Personal Area Networks", Master's thesis, May 2004, 112p.
[22] Persson, K., "Bluetooth scatternet formation: criteria, models and classification", First IEEE Consumer Communications and Networking Conference, 5–8 January 2004, pp.59–64.
[23] Li, X.-Y. et al, "Partial Delaunay triangulation and degree limited localized Bluetooth scatternet formation" IEEE Transactions on Parallel and Distributed Systems, Volume 15, Issue 4, April 2004, pp.350–361.
[24] Mišić, J. et al, "Bridges of Bluetooth County: Topologies, Scheduling, and Performance", IEEE Journal on Selected Areas in Communications, Volume 21, Issue 2, February 2003, pp.240–258.
[25] Busboom, A. et al, "Unambiguous Device Identification and Fast Connection Setup in Bluetooth", European Wireless 2002 Next Generation Wireless Networks: Technologies, Protocols, Services and Applications, Florence, Italy, February 25–28, 2002, 5p.
[26] Mogul, J., "Broadcasting Internet Datagrams", IETF RFC 0919, 1984, 8p.
[27] Schulzrinne, H., "RTP: A Transport Protocol for Real-Time Applications", RFC IETF 3550, 2003, 104p.
[28] Postel, J., "User Datagram Protocol", RFC 0768, 1980, 3p.
[29] Postel, J., "Internet Protocol" RFC 0760, 1981, 45p.
[30] 3GPP TS 26.101 V6.0.0 (2004–09);3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Mandatory speech codec speech processing functions; Adaptive Multi-Rate (AMR) speech codec frame structure (Release 6), 20p.
[31] Sjoberg, L. et al, "Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs", RFC 3267, 2002, 49p.

APPENDIX: THROUGHPUT REQUIREMENT CALCULATIONS

This appendix explains how the throughput requirements presented in Figure 1 have been calculated. The calculations correspond to the HCI level and the throughput requirement values thus represent the throughput requirement imposed on the Bluetooth Controller. The overhead caused by the L2CAP is thus included. The overhead caused by the L2CAP/BNEP/IP/UDP/RTP protocol stack is presented in Table A.I.

Usage of AMR in RTP involves the following variables, which influence the throughput requirement: interface format (IF1 or IF2), AMR Mode, and the number of AMR frames per RTP packet. In the calculations below, usage of AMR IF2 is assumed, because its capacity requirement is higher than that of IF1. The requirement of IF2 can thus also be used as an upper bound for approximating the requirement of IF1. If a more accurate requirement is needed for IF1, it may be calculated in a corresponding manner as that of IF2. The frame sizes of different AMR modes, when one frame is included per RTP packet, are presented in Table A.II. For IF1, additional overhead is caused by the fact that the IP frame length is a multiple of eight. This overhead is presented in the table in parenthesis.

TABLE A.I
PROTOCOL STACK OVERHEAD (ABOVE HCI LEVEL)

Protocol	Overhead (bits)
RTP [27]	96
UDP [28]	64
IP [29]	160
BNEP [10] (uncompressed)	120
L2CAP [3]	32
TOTAL	**472**

TABLE A.II
AMR FRAME SIZES FOR IF1 AND IF2 WHEN A SINGLE AMR FRAME IS SENT PER RTP PACKET. [30], [31]

AMR Mode	IF 1 (bits)	IF 2 (bits)
4,75	105 (+7)	120
5,15	113 (+7)	128
5,90	128 (+0)	144
6,70	144 (+0)	160
7,40	158 (+2)	168
7,95	169 (+7)	184
10,2	214 (+2)	224
12,2	254 (+2)	264

Since only one occurrence of the CMR field is included per RTP packet, the RTP payload size S_P is

$$S_P = S_F + (x-1)(S_F - 8) \tag{1}$$

bits per RTP packet when x AMR IF2 frames are included per RTP packet. The value of S_F corresponds to the AMR IF2 frame size depending on the AMR mode. The valid values of S_F are presented in the IF2-column of Table A.II.

Since the frame rate of AMR is 50Hz, the number of RTP packets sent per second is

$$f = \frac{50}{x} \tag{2}$$

and the required throughput capacity C is thus

$$C = f \cdot (472 + S_P) = \frac{50 \cdot (472 + S_F + (x-1)(S_F - 8))}{x} \tag{3}$$

bits per second.