BLUETOOTH - AD-HOC NETWORKING IN AN UNCOORDINATED ENVIRONMENT

Jaap C. Haartsen
Ericsson Research
Nieuw Amsterdamsestraat 40, 7801 CA Emmen, The Netherlands

ABSTRACT

Recently, a new universal radio interface called Bluetooth™ was developed enabling electronic devices to connect and communicate via short-range radio links. The technology allows the design of low-power, small-sized, and low-cost radios that can be embedded in a wide range of future products.

The Bluetooth system operates in the unlicensed Industrial-Scientific-Medical (ISM) band at 2.4 GHz which is globally available. Bluetooth radios use frequency hopping to spread their signals and to provide resistance against interference from other Bluetooth hoppers and other radio transmitters in the band.

This paper addresses the challenges to provide ad-hoc network functions in the Bluetooth system. Both the needs for frequency hopping and the lack of a central controller in ad-hoc radio networks have placed special requirements on the design of the air interface. The paper further describes the co-existence of and bridging between independently hopping piconets, the concept of scatternets, and discusses hop synchronization in general.

1. INTRODUCTION

The enormous progress in microelectronics and VLSI technology seen in the last decades has led to a widespread use of (portable) computing and communication devices like laptops, PDAs, organizers and mobile phones. Information transfer between these devices has been cumbersome mainly relying on cables applying various connectors and various protocols. Recently, a new universal radio interface has been developed to provide wireless device-to-device connectivity via short-range ad-hoc radio connections. The Bluetooth™ technology – which has gained the support from leading equipment manufacturers like 3Com, Ericsson, IBM, Intel, Lucent, Microsoft, Motorola, Nokia, Toshiba, and many others – eliminates the need for wires, cables and the corresponding connectors between mobile phones, modems, headsets, PDAs, computers, printers, projectors, and so on. The technology enables the design of low-power, small-sized, and low-cost radios that can be embedded in existing devices. In analogy with *embedded computing* resulting from the proliferation of low-cost microprocessors placed in every device ranging from washing machines to car engines, the Bluetooth technology will result in the *embedded connectivity* concept providing wireless access to any device equipped with a low-cost radio. Eventually, these embedded radios will truly connect everything to everything. Radio technology will allow this connectivity to occur without any explicit user interaction enabling a whole new area of applications.

2. AD-HOC NETWORKS

Most of the radio systems in use today are based on a hierarchical architecture. Base stations placed at fixed positions provide local cell coverage, and are interconnected via a wired backbone infrastructure; mobile terminals can move freely within the cells while keeping a connection to the fixed network through the base stations. The infrastructure (consisting of base stations, controllers, switching centers, etc.) contains the intelligence in the wireless system and provides critical functions like channel selection, registration, call setup, and so on. Control channels are vital and form the navel string with which the terminals are locked to the mobile network.

In contrast, in ad-hoc systems, there is no hierarchical architecture. There are no distinctive base stations or terminals. Ad-hoc connectivity is based on *peer communications*. There is no wired infrastructure to support the connectivity between mobile units; there is no central controller (or other intelligence) for the units to rely on for making interconnections nor is there support for coordination of communications. Challenges for systems based on ad-hoc radio connectivity are in 1) finding mobile units to communicate with, 2) selecting an interference-free channel to communicate over, 3) providing quality of service, and 4) enabling low-power idle modes. These functions are normally provided by the control channels broadcasted by the base stations.

An extra complication for ad-hoc radio systems is the choice of radio spectrum. Since there are no operators involved and the application typically targets the consumer market, the spectrum must be license-free. In addition, since there is no geographically fixed infrastructure, ad-hoc radios can operate anywhere in the world as long as they are within range. Therefore, the radio spectrum deployed must be unlicensed and available worldwide. At this moment, only the Industrial Scientific Medical (ISM) band at 2.45 GHz meets these requirements.

Ad-hoc networks come in two types. In the conventional ad-hoc network, a group of peer units in range share a single network. All traffic on the channel is received by all radio units. In a scatter ad-hoc network, several groups of units co-exist in the same area each with their own channel. The scatter ad-hoc environment poses extra challenges with respect to interference. The two types of ad-hoc networks are visualized in Figure 1.

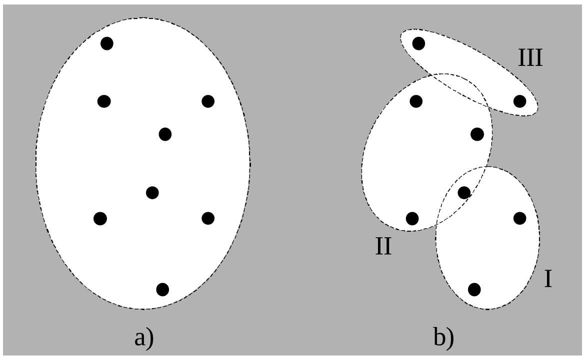

Fig. 1. Topology for a) conventional ad-hoc networks, and b) scatter ad-hoc networks.

The ad-hoc environment is chaotic and highly dynamic. Due to the lack of coordination, communication will be exposed to interference. Due to movement of the mobile units, units enter and leave the ad-hoc channels. Due to different connection scenarios at one point in time a unit is part of one ad-hoc channel whereas at another point in time it is part of another channel.

3. SPREAD SPECTRUM TECHNOLOGY

Radio systems operating in the unlicensed 2.45 GHz ISM band are required to spread their power in frequency [1]. This can either be accomplished by Direct-Sequence spread spectrum (DSSS) or Frequency Hop spread spectrum (FH). Due to the inherent processing gain provided by the spreading techniques, several channels can share the same band without the need for coordination. As mentioned in the previous section, this is of particular importance for scatter ad-hoc scenarios as shown in Figure 1b.

Yet, FHSS and DSSS behave quite different under interference conditions. While the processing gain in DSSS results from interference suppression, in FHSS the processing gain results from interference avoidance. The success of suppression depends on the signal power received from both the intended transmitter and the jammer. If the jammer is much closer to the receiver than the intended transmitter, the DS processing gain will not be able to overcome the jamming power. This is called the near-far problem. For CDMA systems applying DSSS, power control is used to minimize the power differences. However, in a (scatter) ad-hoc network, firstly relative distances differ greatly, and secondly there is no mechanism to provide coordinated power control. In contrast, FHSS is much less sensitive to the near-far problem. Instantaneously, a small (hop) channel is filtered out while adjacent channels are suppressed. Relatively high jammer power levels are permitted on these adjacent channels. Therefore, in an uncoordinated scatter ad-hoc networks, FHSS is the preferred method to define independent communication channels. Bluetooth deploys the FH technique using 79 hop channels and a nominal hop rate of 1600 hops/s [2].

4. PICONETS AND SCATTERNETS

A Bluetooth channel is defined as a hop sequence which is a pseudo-random hop pattern over the 79 hop carriers. The phase in the sequence determines the instantaneous hop carrier to use. Each channel has a different hop sequence and a different phase. In the example of Figure 1b, the channels used by groups I, II and III use different FH sequences. Yet, they all use the same pool of 79 hop carriers. Since there is no coordination between the groups, once in a while two channels may select the same hop carrier. Depending on the relative distances (carrier-to-interference ratios), such an collision may result in distortion of the transmitted signals. Because of the fast hopping rate of 1600 hops/s, the collision will be of short duration.

Bluetooth units that share the same (FH) channel form a so-called piconet. This piconet is a local ad-hoc network with a limited number of active units. Each piconet uses a different FH sequence. Since no coordination regarding the selection of FH sequences can occur, the channel selection is coupled to addresses. Each Bluetooth radio transceiver has a unique 48b Bluetooth device address (BD_ADDR) included during the manufacturing. The FH sequence of the piconet is derived from the BD_ADDR of one of the units participating in the piconet. By definition, this unit is called the master of the piconet whereas the remaining units are slaves. The phase in the FH sequence is derived from the system clock of the master unit. During connection setup, both the master address and the master clock are conveyed to all the slaves so that they can all synchronize to the master.

Co-located piconets form a scatternet. The three piconets I, II, and III in Figure 1b are all controlled by a different master and therefore use different FH channels. Since each channel can only support a data rate of 1 Mb/s (the maximum data rate on a Bluetooth connection), the throughput per user is higher in the scatter ad-hoc network of Figure 1b where three 1 Mb/s channels are shared (ignoring collisions), than in the conventional ad-hoc network of Figure 1a where a single, 1 Mb/s channel is shared. Basically, a radio unit belongs to a single piconet. However, in a scatternet, a unit can share its time between different piconets as is discussed in section 6.

In addition to the FH synchronization support, the master provides traffic control. To guarantee quality of service in the piconet, a contention-free channel is realized. The master deploys a polling scheme to divide the 1 Mb/s bandwidth between the slaves. The channel is divided into time slots of nominally 625µs length. Each time slot is allocated a different FH carrier according to the FH sequence. The master and slave transmissions alternate, resulting in a Time Division Duplex (TDD) scheme. In the master slot, the master can choose to transmit a packet to any slave in the piconet; in the slave slot, only the slave that is addressed by the master in the preceding master slot is permitted to return a packet to the master. This scheme is visualized in Figure 2. If the master has information to send to the slave (even if it is only an acknowledgement message), there is no poll overhead. In case there is no information from the master to send, the master has to periodically send a poll packet to the slave to enable it to send information from the slave to the master. The polling interval depends on the allowed latency in the application.

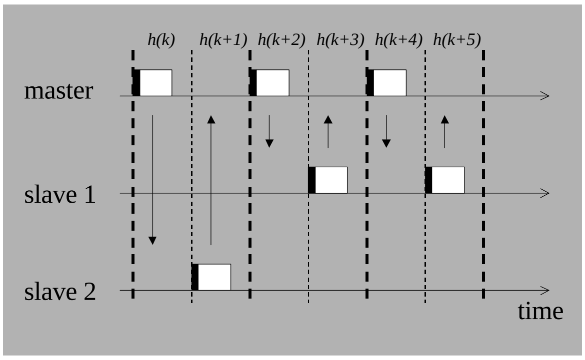

Fig. 2. Master-slave communications via polling concept.

Connections can be established by the page and inquiry procedures. When there is no connection, the Bluetooth unit is in *standby* and periodically listens for a page or inquiry message on a few specified hop carriers. No control channels or other beacon messages are broadcasted. If two units A and B come into range, unit A can page unit B and establish a connection. Since unit A does not know in which hop carrier and at what moment in time unit B will listen, there is a frequency-time uncertainty to solve. The paging unit A repetitively transmits a page message on different hop carriers. The page message sent by unit A is derived from the BD_ADDR of unit B. The information which units are in range and what their BD_ADDRs are can be obtained with the inquiry procedure [3]. The unit that sends the page message, automatically becomes the master of the piconet just established. The connection setup procedure depends on the sleep period of unit B, but also about the knowledge unit A has about unit B. If in addition to unit B's BD_ADDR, the system clock of unit B is known in unit A, the setup procedure is improved by a factor of two.

5. SYNCHRONIZATION

A key element in the Bluetooth piconet is synchronization. All units have to use the same hop carrier at the same time in order to communicate. Hop selection is carried out in a hop selection box which maps the master BD_ADDR to a particular FH sequence and the master clock to the momentary phase in this sequence, see Figure 3. The slaves use the master BD_ADDR to follow the FH sequence used by the piconet. The master clock is derived from a native (system) clock which is free running in every Bluetooth transceiver. A positive or negative offset Δ (representing the difference between the master native clock and the slave native clock) is added in order to feed the hop selection logic with a representation of the master clock. The native clocks run at a nominal clock speed of 1600 ticks/s.

Due to temperature drift, aging, and implementation difference, the clocks will never be completely synchronized (for a Bluetooth radio, a 20ppm accuracy is required for active mode, whereas for the low-power modes only 400ppm accuracy is required). Therefore, the offset values Δ are constantly adjusted to keep the phase input in the slaves' hop selection logic in line with the master phase. For re-

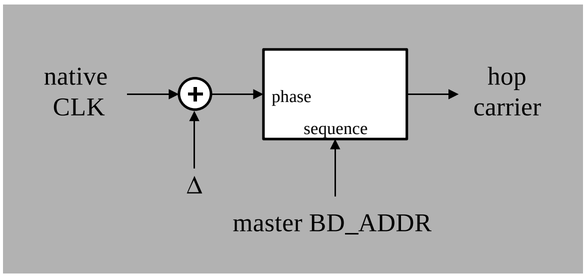

Fig. 3. Hop selection logic to keep FH synchrony.

synchronization, a slave can use any packet exchanged on the piconet channel. Each packet whether transmitted by the master or the slave, is preceded by an access code which can be used in the (re)synchronization process. The access code has pseudo-random properties (good auto- and cross-correlation properties) and is derived from the master BD_ADDR. The Bluetooth radio receivers contain a correlator which is matched to the access code, see Figure 4. The correlator output is used to detect the presence of a packet, but also to re-adjust the slave timing. Since each piconet has a different master, each piconet carries different access codes as well. Since the correlator is always matched to the own master, packets carried by co-existing piconets cannot derail each other's synchronization.

It will be understood that clock information (or rather the clock offset) is crucial in the Bluetooth system. At connection establishment, the master unit transmits its real-time clock to the slave unit. The real-time clock is carried in a special FH synchronization (FHS) packet. This FHS packet contains the BD_ADDR of the sender but also the value of the native clock valid during the first bit of the packet. This information is then used in the slave to determine the time offset between the native clock in the master and in the slave. Clock offsets are also stored (and can be regularly updated) for later use. An estimate of the clock of a unit to be paged will accelerate the paging procedure since it reduces the frequency-time uncertainty.

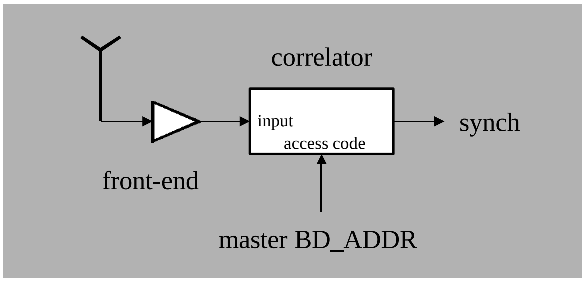

Fig. 4. Timing re-synchronization through the access code.

6. INTER-PICONET COMMUNICATIONS

The scatternet shown in Figure 1b consists of three independent piconet. Although the piconets' coverage areas overlap, each unit is only active (as master or slave) in one piconet. However, the FH synchronization concept allows units two switch in real time between different piconets. As a result, a unit can be virtually active in several piconets simultaneously, although at any one moment in time, it can be communicate on one piconet channel only. Since a channel is identified by a BD_ADDR and a time offset Δ as was shown in Figure 3, a unit can quickly switch to a different piconet by changing the BD_ADDR and the Δ value. By time division multiplexing (TDM) between different piconet channels, a unit can participate as slave in several piconets, or as master in one piconet and as slave in other piconets. This is called *inter-piconet communications*. Although the slots of different piconets cannot be time aligned, clock drifts are sufficiently small to obtain a semi-static situation such that some coordination of the TDM can be achieved. In Figure 5, an example is shown in the scatternet configuration of Figure 1b. The master units are indicated by squares whereas the slaves are represented by circles. In this particular example, unit X participates both in piconets I and II. Unit X thus forms a bridge between the two piconets.

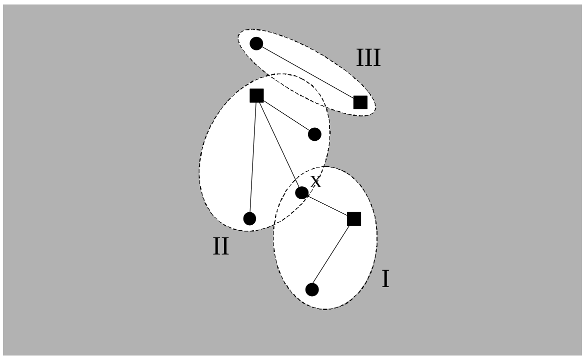

Fig. 5. Unit X provides inter-piconet communications between piconet I and II.

In Figure 6, the inter-piconet communications are explained in the time domain. Unit X first exchanges some packets with master I, using the FH sequence and phase used in piconet I. Then it switches to the piconet II of master II using the FH sequence and phase used in piconet II, and then returns again to master I. When unit X is not participating on the channel, the master can of course address other slaves in the piconet; this is not shown in Figure 6.

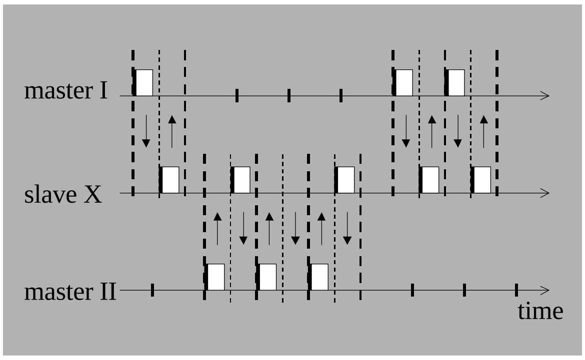

Fig. 6. Inter-piconet communications by TDM.

The inter-piconet communication concept can be extended to more than two piconets. All a unit has to have are the address/offset combinations of the piconet masters i.e. BD_ADDR_I/Δ_I, BD_ADDR_{II}/Δ_{II}, and $BD\_ADDR_{III}/\Delta_{III}$. The offset values Δ have to be updated regularly to compensate for clock drifts. This is achieved by listening to the traffic of the corresponding link and correlate against the access code.

Once a piconet has been established, the master and slave can enter a low-power mode called the *park* mode. In this mode, the master periodically transmits a beacon message (also frequency hopping) to which the slave can synchronize. Since the master and slave are completely synchronized, a traffic connection can quickly be established and only depends on the beacon interval. The *standby* and *park* modes in the Bluetooth system can be compared to the power-on and idle modes of a cellular system. In the power-on, the cell phone searches for a proper base station to lock to. Once found, the cell phone locks to the control channel and enters a low-power idle mode.

If the network topology is semi-stationary (that is the units remain in each other's range but relative distances may vary), the *park* mode can be exploited to speed up connection setups. A few units can act as masters to which all remain units (slaves) are parked. These masters are called anchor masters which control the anchor piconets. Address and clock-offset information can be distributed using the broadcast capabilities in the *park* mode or on other occasions when the anchor master has a connection with the slave. This information can then be exploited in the slaves to quickly setup connections between each other resulting in new traffic piconets with new master but on a temporary basis. While participating in the traffic piconets, the units remain parked in the anchor piconet using the inter-piconet communication concept. Traffic piconets are established and released depending on the traffic need between units. The anchor piconets are rather stationary and provide synchronization support to speed up the traffic piconet creation.

7. CONCLUSIONS

In this paper, the ad-hoc network capabilities of Bluetooth have been presented. The independently frequency hopping piconets form a challenge for general connectivity. Key elements in the FH synchronization are the master BD_ADDR and the time offset. If a unit has the address/offset information of various piconets, it can virtually participate in all these piconet through time division multiplexing. In semi-stationary environments, address/offset information can be distributed through anchor masters to which the slaves are parked.

REFERENCES

[1] US Federal Communications Commission, Part 15, paragraph 15.247.

[2] J.C. Haartsen, "The Bluetooth radio system," IEEE Personal Communications Magazine, Vol. 7, No. 1, pp. 28-36, 2000.

[3] J.C. Haartsen, "Bluetooth – the universal radio interface for ad hoc, wireless connectivity," Ericsson Review, Vol. 75, No. 3, pp. 110-117, 1998.