Routing in Bluetooth Personal Area Networks

J. Yang & J.K. Pollard

Department of Electronic and Electrical Engineering University College London

Abstract: It is proposed that a new role: Router Node is introduced to the Bluetooth Personal Area Network profile to enable multiple hops between piconets. A proposed structure that allows transparent piconet connection is illustrated. The resulting increase in data throughput is particularly important for scalable PAN communications where multiple hops between piconets are needed.

A software system has been built as a "proof-of-concept" of the Router Node construction. A prototype Bluetooth router program has been developed in the C program language. This will enable rapid realisation of a working Router Node in hardware and software.

1. Introduction

Bluetooth is a cable replacement wireless technology that operates in the unlicensed 2.4GHz Industrial, Scientific, and Medical (ISM) band. It uses Frequency Hopping Spread Spectrum (FHSS) to improve reliability. The maximum raw data speed is 1Mbps and typical communication range is within 10 metres.

Currently, most popular Bluetooth applications communicate on a point-to-point basis. However, several *piconets* of Bluetooth enabled devices can be connected to form *scatternets* in an *ad hoc* Personal Area Network (PAN). Research has been done on areas such as Bluetooth ad hoc network formation and routing algorithms [1].

This project is to design a Router Node (RN) in PAN in order to improve the data throughput. In this paper, section 2 describes a proposed PAN profile extension and the original Bluetooth Network Encapsulation Protocol (BNEP) specification. In section 3, the design of the RN is outlined. Finally, section 4 illustrates the on-going work in the laboratory.

2. PAN Profile and BNEP

The Bluetooth PAN profile describes how two or more Bluetooth devices communicate with each other. The PAN specification [2] defines three roles: Network Access Point (NAP), Group Ad-hoc Network (GN), and Personal Area Network User (PANU). The NAP is a device acting as a bridge to connect a wireless piconet to a wired IP network. It forwards packets between the two networks. A GN connects PAN Users (PANU), forwarding packets between them. A new Router Node replaces this functionality. A PANU is the data source and sink for all communications. It is either a Master or Slave at the Baseband view. It can either use the NAP (or the GN under the original PAN profile) as an intermediary and it can also communicate directly with another PANU. However, communication between NAP and GN was originally not allowed and this limitation requires modification.

All the data exchanged between these Bluetooth devices are encapsulated into BNEP frames [3] and are sent as Logical Link Control and Adaptation Layer Protocol (L2CAP) messages: When an Ethernet packet is encapsulated into a BNEP packet, the Ethernet Header is replaced with the BNEP Header and this is encapsulated by L2CAP messages.

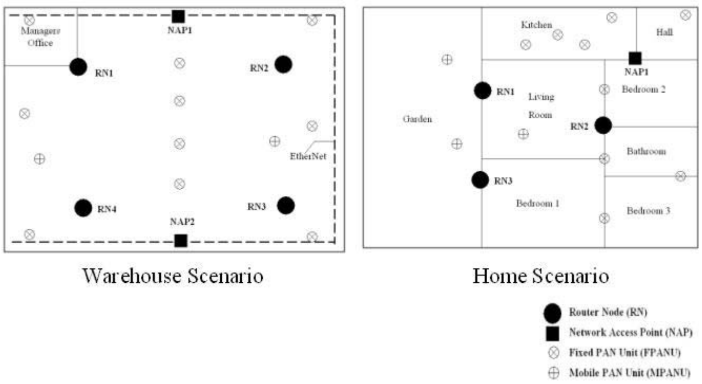

Figure 1 Warehouse and home scenarios

Proposed extension to PAN profile

Warehouse and home scenario reference models are shown in Figure 1. In these scenarios, about 50% of Bluetooth devices such as NAP, printer and fixed computers do not change position on a regular basis. These devices can be used to form a backbone network. Mobile devices can then use this quasi-static backbone network for communication. A new profile role: a Router Node (RN) is proposed to enable device communication over multiple hops between piconets [4]. The NAP(s) communicate(s) with the Router Nodes and these communicate with the PANUs.

3. Router Node structure

Router Nodes connect piconets into scatternets and the structure is shown in Figure 2:

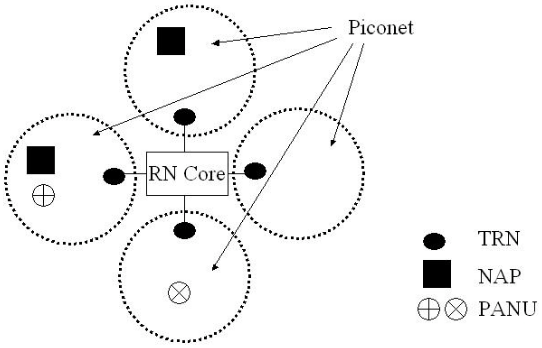

Figure 2 Structure of Router Node

A microprocessor unit is a part of the RN Core, controlling shared memory and an address lookup table. The RN contains one Bluetooth *Tentacle* Router Node (TRN) in each piconet.

The advantage of this proposed shared memory link mechanism between piconets is that it is not necessary to implement time multiplexing, nor the complex Master/Slave role switch (with the attendant need to Park other Slaves) [5]. In addition, wire connection between TRN and RN Core can greatly improve reliability and bandwidth inside the Router Node.

It is envisaged that each TRN collects addresses of other Bluetooth devices in the piconet to which belongs. It forwards BNEP frames and advertises Service Discovery Profile (SDP) service records to other RNs.

The RN Core is the central control unit of RN, which maintains a routing table. It receives data packets from TRNs. It checks the packet headers to identify BNEP packet types and destinations and then routes them to the appropriate TRN.

4. Router Node "proof-of-concept"

4.1 Analysis

In practice, wire connections can be achieved through standard interfaces such as Universal Asynchronous Receiver/Transmitter (UART), Synchronous Serial Interface (SPI) or through having the same data bus shared by TRNs and RN Core.

The Bluetooth specification defines three Host Controller Interface (HCI) transport layers: USB, UART and RS232 [6]. UART and USB are two most common physical transports in HCI, which are used to connect a host to several Bluetooth devices [7]. The number of TRNs in a Router Node may be chosen according to how many piconets a particular RN are joined. For example: A RN with only two TRNs actually works as a bridge. It simply forwards data between two piconets. The diagram of the protocol stack for a Router Node with two TRNs can be seen in Figure 3.

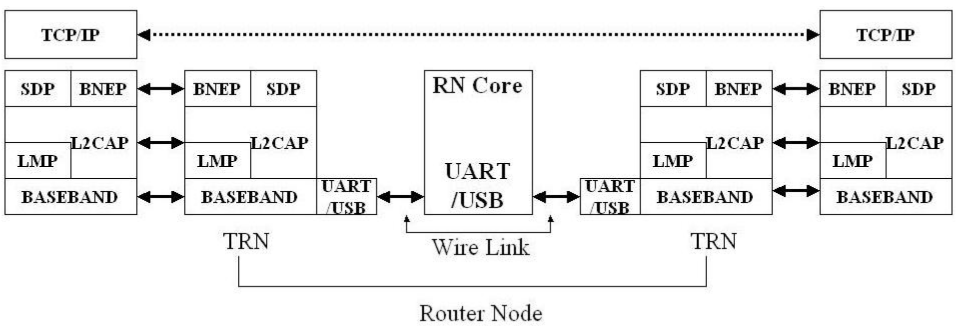

Figure 3 Bluetooth Protocol Stack in RN

The stack of TRN on the left is connected to the RN Core (a Personal Computer, PC) and this communicates with the stack of the TRN on the right.

The Maximum Transmission Unit (MTU) of a BNEP frame is 1691 bytes (maximum Ethernet packet payload 1504 bytes + BNEP header 15 bytes + possible extension header). If UART is used at a transmission speed = 115,200 bps, the transmission time from TRN to RN Core is 1691*8/115,200 = 117ms.

A USB could be used to reduce the delay – a transmission speed of 1.5 Mbps results in a delay of 9ms.

4.2 Emulation Environment descriptions

A Router Node is emulated as illustrated in Figure 4.

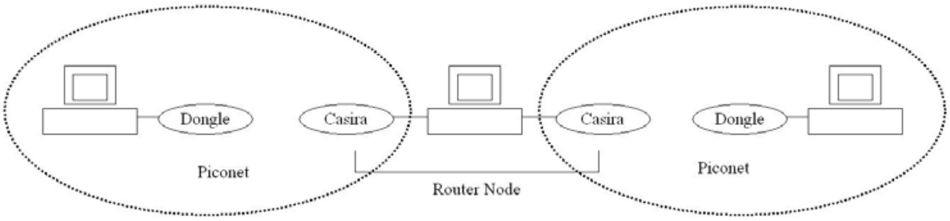

Figure 4 Router Node emulation

One Personal Computer (PC) acts as the Router Core and two programmable Bluetooth development kits (Casiras) [7] act as TRNs that are attached to the serial ports. Two Bluetooth adaptors (dongles) provide Bluetooth interfaces to two external computers. These external computers now can communicate with each other indirectly through the RN.

A static routing table that can be modified manually is currently used in the PC to route all types of BNEP packets between the TRN serial port and the RN Core. Once the RN Core finds an entire packet available in the buffer, it reads the header and decides where to forward the packet.

4.3 Structure of the RN Core

A prototype Bluetooth router program has been developed in the C program language to run on the RN Core. Figure 5 shows the structure in the PC that emulates an RN Core.

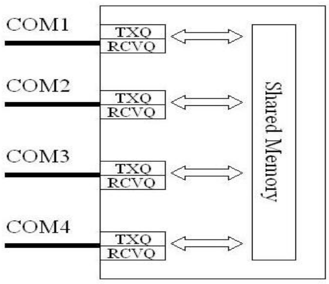

Figure 5 Structure of the RN Core

Every serial port has a Receiver Queue (RCVQ) and a Transmitter Queue (TXQ) and a shared memory is used to exchange packets between different serial ports. The program sequentially polls every serial port buffer queue. If there is any data available in the RCVQ, it checks whether it is a whole BNEP packet. Once it finds a BNEP packet, it allocates a particular size of shared memory according to the size of the packet. Then the whole packet is read into the memory.

A simplified program flow chart is shown in Figure 6.

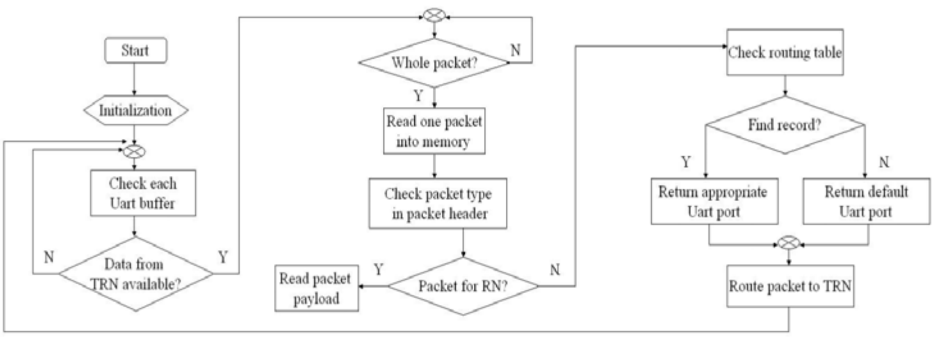

Figure 6 Program flow chart of RN Core

The type of the BNEP packet is identified by reading the first byte of the header. From the packet type, the RN Core finds the destination Bluetooth address. Routing information is determined from a routing table and the packet is flushed to the appropriate TXQ. If no routing record is found, the packet is routed to a default TXQ. Finally, the shared memory is released for other uses. The parameters (baud rate, handshaking) of each serial port can be set separately. A shared memory is adopted to exchange packets between different ports to improve efficiency. The routing table can be modified manually. All the information including parameters of serial ports and routing table are saved when the program exits.

4.4 Flow Control

A flow control mechanism is necessary: The TRNs are asynchronous and have equal priority to transmit and receive so it is possible that there could be a time conflict in the PC. In addition, if packets flood from more than one port, it could cause congestion.

The programme can enable both software and hardware flow control for each serial port. In software handshaking, XON and XOFF characters are used to suspend and resume communications, wheras hardware handshaking uses Request to Send (RTS) and Clear to Send (CTS) lines to suspend and resume communications.

As an example: Receiving and transmitting buffer queues for each serial port are set to be 4096 bytes. *Thresholds* are set as three-quarters of the size of these buffers (3072 bytes) and this determines when the receiver will assert the corresponding handshaking signals.

5. Summary

A proposed Personal Area Network profile extension: a Router Node (RN) allows efficient communication between multiple piconets.

This paper presents a RN structure in which several Bluetooth slave devices, each in a separate piconet, communicate using a shared memory. This has been implemented in a PC and demonstrates the feasibility of reading a BNEP packet header and routing packets according to an internal lookup table. The use of this RN reduces latency and operational complexity of links between piconets. The resulting increase in data throughput is particularly important for scalable PAN communications where multiple hops between piconets are needed.

6. References

[1] N.J. Muller, “Bluetooth Demystified”, McGraw-Hill, (2000).

[2] Bluetooth SIG, “Personal Area Networking Profile Version 1.0” (2003).

[3] Bluetooth SIG, “Bluetooth Network Encapsulation Protocol (BNEP) Specification”, (2002).

[4] J.K. Pollard, “Bluetooth Personal Area Networks”, University College London, (Aug. 2003).

[5] L. Huang, *et al*., “Impact of Topology on Bluetooth Scatternets”, J. Pervasive Computing & Comms, 1(2), (May, 2005), pp 123-134.

[6] R. Morrow, “Bluetooth Operation and Use”, McGraw-Hill (2002)

[7] D. Kammer, G. McNutt, B. Senese & J. Bray, “Bluetooth Application Developer’s Guide”, Syngress, (2002).