A Routing Vector Method (RVM) for Routing in Bluetooth Scatternets

Pravin Bhagwat
IBM, Thomas J. Watson Research Center
Yorktown Heights, NY
pravinb@us.ibm.com

Adrian Segall†
Dept. of Electrical Engineering
Technion, Israel
segall@ee.technion.ac.il

Abstract—**Bluetooth is a promising new technology for short range wireless connectivity between mobile devices. Initially, Bluetooth will be used as a replacement for point-to-(multi)point cables. However, in due course of time, solutions for forming multihop ad hoc networks over Bluetooth (referred to as *s*catternets) will be needed. In this paper, we explore the design space of routing protocols over Bluetooth scatternets. We argue that ad hoc routing protocols for Bluetooth must make a different set of design compromises compared to those being developed by the MANET working group. These differences result primarily from the specific characteristics of the Bluetooth physical and link layer.**

Our proposed routing method is based on the concept of route vector, which is an efficient method for encoding source route paths in Bluetooth scatternets. We describe the protocols for route discovery and packet forwarding. Our design illustrates three main design compromises, namely minimization of soft-state, protocol simplicity, and bandwidth conservation, all of which are crucial for efficient operation over small size Bluetooth scatternets.

I. Introduction

Bluetooth [1],[2] is a single chip, low-power, wireless communication module that will fit inside mobile phones, laptops, palm computers, digital cameras, and cordless headsets and allow users to interconnect these devices easily and quickly without the need for cables. Bluetooth utilizes a short-range radio link to exchange information, enabling effortless wireless connectivity between devices and other peripherals. The radio operates in the globally available 2.45GHz ISM band, allowing international travelers to use Bluetooth-enabled equipment worldwide. Since the radio link is based on frequency-hop spread spectrum, multiple channels (frequency hopping sequences) can co-exist in the same wide band without interfering with each other. Two or more units sharing the same channel form a piconet, where one unit acts as a master, controlling the communication in the piconet and the others act as slaves.

The MAC layer protocols for communication between the master and the slaves in a piconet are well defined by now [2]. Bluetooth channels use a frequency-hop/time-division-duplex (FH/TDD) scheme. The channel is divided into 625-msec intervals, called slots, where a different hop frequency is used for each slot. The master transmission starts in even-numbered slots, while the slave transmission starts in odd-numbered slots. Packets can be 1,3 or 5 slot long and are transmitted in consecutive slots. A slave is allowed to transmit in a given slot if the master has addressed it in the preceding slot. There is no direct MAC layer communication between slaves. One of the goals of the present work is to propose higher layer protocols for allowing inter-slave unicast and broadcast communication within a piconet.

In the Bluetooth architecture, multiple piconets with overlapping area of coverage can co-exist since their frequency hopping patterns are mutually orthogonal. A group of interconnected piconets is referred to as a scatternet. A node can participate in two or more piconets on a time sharing basis, i.e., at any instant the node can be active in only one piconet. Such a unit can receive packets from one piconet and relay them to the other piconet(s) it is connected to. Protocols for inter-piconet communication among units in a scatternet have not been defined yet. A plausible solution is to model a scatternet as an instance of an ad hoc network and use already known methods of routing [3], [4], [5], [6]. While this approach is certainly feasible, it may not be the most efficient solution. Bluetooth layer presents a unique set of features and limitations which known ad hoc routing protocols are not optimized for. For example, due to packet size limitation at the Bluetooth baseband layer, all MANET style solutions will require fragmentation and reassembly of packets at each relay node. This will lead to two problems; 1) increased buffering requirement at each node and 2) higher store and forward delay at each hop. Instead, if the forwarding is supported at the Bluetooth slot level, delay as well as buffering requirement will be reduced (at a marginal cost of link efficiency). In many ways, this tradeoff is similar to the hop-by-hop IP routing Vs. ATM switching tradeoff. There are conceptual similarities between the Bluetooth baseband layer and ATM cell transport layer and the overhead of running MANET style algorithms over Bluetooth is similar to the overhead of performing hop-by-hop IP routing over ATM networks.

The other distinguishing feature of scatternets is their topology of interconnection. Scatternets and wireless LANs, for example, differ in terms of their communication capabilities. Two bluetooth nodes cannot hear each other unless they form a master-slave pair. This is in contrast to the wireless LANs where any two nodes within proximity can hear each other's transmissions. It is also anticipated that scatternets will differ from classical ad hoc networks in terms of applications, traffic characteristics, mobility patterns, and scaling requirements. In most application instances, scatternets will be quasi-static, short lived, and small. Therefore, scalability and adaptivity features

†This work was performed when the author was visiting the IBM T.J.Watson Research Center

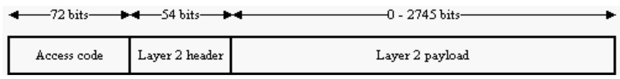

Fig. 1. Bluetooth packet format

of MANET style algorithms may not always be necessary.

The purpose of the present work is to explore the design space of scatternet routing protocols. Our proposed routing method is based on the concept of route vector, which is an efficient method for encoding source route paths in Bluetooth scatternets. We describe the protocols for route discovery and packet forwarding. Our design illustrates three main design compromises, namely minimization of soft-state, protocol simplicity, and bandwidth conservation, all of which are crucial for efficient operation over small size Bluetooth scatternets.

II. Model Elements

The units of our model are Bluetooth units [1]. Every Bluetooth unit has a globally unique 48 bit Bluetooth address. Bluetooth units are organized in piconets. Every piconet has a master and up to 7 slaves. A unit can be a slave in two or more piconets and a master in another piconet. A unit cannot be a master in more than one piconet. A unit that belongs to more than one piconet is referred to as a relay. Slaves in a piconet are assigned 3 bit MAC addresses, denoted by MacAddr. The 000 address is reserved for broadcast within the piconet. Between the master and a slave, two different link types can be established, a synchronous connection (SCO) and an Asynchronous connection (ACL). Our protocols in this paper use Layer 3 control packets transmitted over ACL channels. Bluetooth channels use a frequency-hop/time-division-duplex (FH/TDD) scheme. The channel is divided into 625-msec intervals, called slots. The master transmits in even-numbered slots, while the slaves transmit in odd-numbered slots. A slave is allowed to transmit in a given slot if it has been addressed by the master in the preceding slot. A packet can cover one slot, three slots or five slots. The packet has a 72-bit access code, a 54-bit header, referred to in this paper as the Layer 2 header, and a 0-2745-bit payload (see Fig. 1). One of the fields of the layer 2 header is the 3-bit MacAddr, MacAddr. The master of a piconet can send packets to a slave in its piconet by including the MacAddr of the slave in the MacAddr field. The master can broadcast a packet in its piconet by using the 000 address. Packets sent by a slave contain in the MacAddr field the MacAddr of the sending slave, allowing the master to identify the sender of the packet. Slaves cannot directly communicate with other slaves.

Packets can contain up to 340 payload bytes. The payload carries user information and control information for higher layers. Our protocols will use Layer 3 control information, which will be incorporated in the Layer 3 header (see Fig. 2). In the following we propose protocols for both intra- and inter-piconet communication.

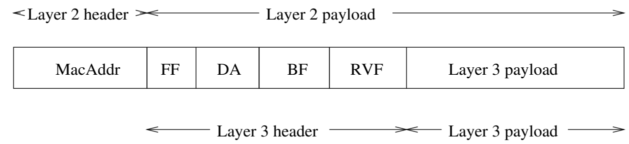

Fig. 2. Relevant fields in the Bluetooth packet

III. Intra-piconet Communication

A. Forwarding unicast packets by the master from one slave unit to another

Slaves within a piconet cannot communicate directly with each other. Such communication can be implemented only via the master. There is no destination address field in the Layer 2 header of a packet sent by a slave. Thus, if we want to make it possible for the master to forward packets, there is need for a Layer 3 header in the Layer 2 payload of the packet (see Fig. 2) . That header will contain a forwarding flag (FF) and a destination MacAddr (DA) field and other fields, as detailed later. The value FF=0 means that the payload of the packet is destined for the master and needs not be forwarded. If FF=1, then the payload of the packet is intended for another slave unit in the same piconet and DA contains the MacAddr of the destination unit. When the master receives a packet, it strips the layer 2 header and forwards the payload to layer 3. If FF=1, then the Layer 3 processor puts the payload in a new packet and sends it to the corresponding slave. For example the payload of a packet received by the master with FF=1,DA=011, MacAddr=101 will be forwarded by the master to the slave with MacAddr=011.

B. Forwarding broadcast packets received by the master from a slave device

If FF=1 and DA contains 000, then the contents of the packet are intended to be broadcast in the piconet. In this case, the master puts the payload in an intra-piconet broadcast packet, which is broadcast in the piconet. For example, the payload of a packet received by the master with FF=1, DA=000, MacAddr=011 will be sent by the master with MacAddr=000. The origin unit needs to use higher layer protocols to remember that it has already seen the packet. Since intra-piconet broadcast is not reliable, there is need to improve the reliability of broadcast of essential information. This can be achieved for example by repeating the broadcast several times. Again, it is the responsibility of higher layer protocols to detect repeated receipts of the same packet.

The master knows the Bluetooth address (48 bits) of every slave in its piconet and the corresponding MacAddr. A slave can learn about the 3-bit MacAddr of another slave unit by querying the master.

IV. Inter-piconet communication: design rationale

Routing of packets in a communication network can be based on tables held at the relays, like in IP or ATM or on source routing, namely information carried in the packets. Some proposals for table based routing protocols for wireless mobile networks are: the Lightweight Mobile Routing Protocol [3], the Destination-sequenced distance vector protocol [4], the Murthy-Garcia-Luna-Aceves protocol [5] and the temporally-ordered routing algorithm [7]. Examples of systems that have used source routing are PARIS [8], Ring Network Bridges [7] and more recently DSR [6]. Although the source routing vector method leads to a larger overhead, it has several advantages that make it preferable in scatternet systems. If connections are based on tables at the relays, we need protocols to construct the table entries and to delete those entries when the connection is completed or disrupted by failures or by mobility. No such action is necessary in the source routing method, because the routing information is carried in each user packet. Another advantage becomes evident in the propagation of broadcast packets. In broadcast protocols there is need to find out if a packet arriving at a node is a new packet or a copy of a packet that the node has already seen. Some sequence number scheme is necessary if the information is kept at nodes. On the other hand, if the list of nodes that the packet has visited so far is carried in the packet, the node can simply look at this list to find out if the packet has already visited the node. Some time-stamp or sequence number scheme is still necessary to determine if the information in the packet is more recent than the previously received information, if the order of the broadcast packets is important.

The commonly used method of representing routes in source routing headers is to include the list of the node ID's. This method would be very wasteful in Bluetooth scatternets. In order to reduce the overhead of carrying information in the packets, we propose for unicast and multicast traffic an efficient method of representing routes referred to as the routing vector method (RVM). In RVM, piconets are represented by Local Identification numbers (LocID) selected locally by relays and the sequence of LocIDs (as opposed to sequence of MAC addresses) are carried in the packet header. Thus, the overhead is reduced from 48+3=51 bits per hop to 3+3=6 bits per hop. Since in addition, most scatternets are planned to be relatively small, we estimate that the overhead of carrying information in the packet in RVM will be small and that the advantages of this scheme will outweigh the drawbacks.

V. Inter piconet Unicast Communication

A. Piconet identification and the Broadcast flag

Every piconet is identified by the Bluetooth address (48 bits) of its master. Since this address is too long for our purposes, we shall use local identifiers. Assuming that the maximal number of piconets a relay can be directly connected to is 7, we shall use in RVM local identifiers consisting of 3 bits. A relay assigns a 3-bit local identifier, denoted by LocID, to each directly connected piconet. This identifier will be used in the unicast and multicast routing through this relay. The value LocID=000 is reserved to identify the relay itself. This is illustrated in Fig. 2, where the numbers in italics are the LocID and the Roman bold numbers are the MacAddr's. For example, relay B received MacAddr=2,6,5 in the piconets whose masters are M1,M2,M5 respectively and has assigned LocID=4,3,7 to those piconets respectively.

In addition to the fields mentioned in Sec. III-A, we specify in the Layer 3 header a broadcast flag (BF) and a routing vector field (RVF) (see Fig. 2). The value BF=0 defines an inter-piconet unicast packet, BF=1 defines the packet as an inter-piconet broadcast packet. The broadcast case will be discussed in Sec. VI.

B. Routing of unicast packets

The protocol is similar to the source routing protocol of [8], [9]. When BF=0, the RVF contains a sequence of alternating id's LocID, MacAddr, LocID, MacAddr,etc. where LocID is a 3-bit local identifier and MacAddr is a 3-bit MAC address. When the packet is sent by the source node, the latter writes in the RVF field the sequence of id-s corresponding to the route to be traversed by the packet. The source node learns the route via the Route Discovery protocol described in Sec. VII. When a relay receives such a packet, it sends the packet to the master of the piconet corresponding to the first LocID, to be forwarded to the unit whose MAC address is given by the first MacAddr. Before sending it, it chops off the first pair (LocID, MacAddr). For example, consider the scatternet of Fig. 3, consisting of piconets M1-M7. The masters of the piconets are identified by the same notation as the piconets. A packet sent by unit A in Fig. 3 destined for E whose route is A-B-C-D-E contains in the RVF field the sequence 3,4,2,7,5,6,0 (see Table I). The packet is sent by A to M1 with MacAddr=5, FF=1, DA=2. master M1 includes the Layer 2 payload of the received packet in a packet with MacAddr=2 and sends it in piconet M1. When layer 2 of unit B receives the packet, it forwards it to layer 3. The latter learns from BF=0 that this is an inter-piconet unicast packet and from the first entry in RVF that the packet is to be forwarded in the piconet whose internal label LocID at B is 3, namely to piconet M2. Moreover, the packet is to be forwarded by master M2 to unit 4 (the second entry in RVF). After stripping the first two entries in RVF, relay B sends the packet to M2. The above procedure is repeated by the other relays and masters. When layer 3 at relay E receives the packet with RVF=0, it learns that the packet is intended for the unit itself and forwards it to the application layer.

VI. Inter piconet Broadcast

A. Unlimited Broadcast - the source routing method

Similarly to unicast routing, the packet collects the information about the units visited so far. If a unit has already seen this packet, it discards it, if it has not, the unit forwards the packet to all neighbors. Broadcast packets carry BF=1. In this case, the RVF field in the Layer 3 header contains a sequence of 48-bit Bluetooth addresses of the relays and masters the packet has visited so far. When a relay receives an inter-piconet broadcast packet, it looks in the RVF field. If its own Bluetooth address

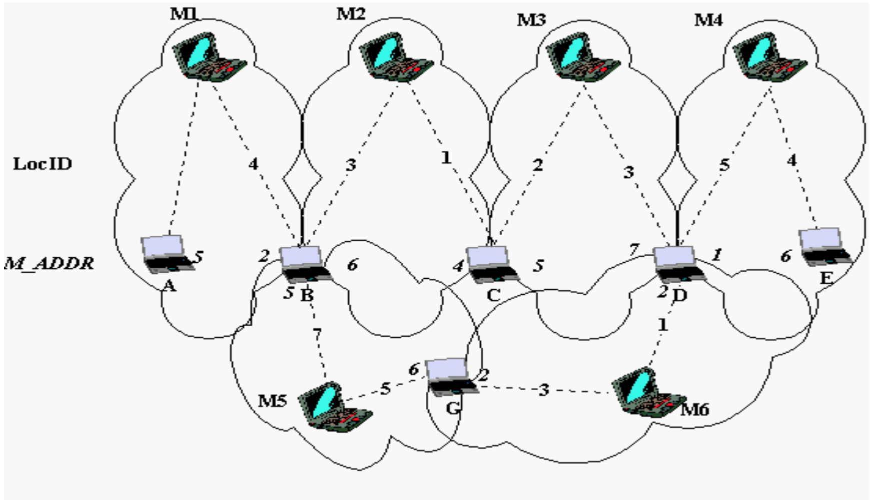

Fig. 3. A scatternet

TABLE I
HEADER FIELDS FOR THE EXAMPLE IN FIG. 3

	MacAddr	FF	DA	BF	RVF
A	5	1	2	0	3,4,2,7,5,6,0
M1	2				3,4,2,7,5,6,0
B	6	1	4	0	2,7,5,6,0
M2	4				2,7,5,6,0
C	5	1	7	0	5,6,0
M3	7				5,6,0
D	1	1	6	0	0
M4	6				0

is already in RVF, the packet has already visited this relay and hence it is discarded. If not, it adds its own Bluetooth address to RVF and sends the payload of the packet to the masters of all piconets it is directly connected to, except for the one it has received the packet from. The packets transmitted by the relay are sent as intra-piconet unicast packets intended to be broadcast in the piconet (FF=1, DA=000). If the source is a slave, it sends the packet to its master. masters broadcast the packet in their piconet according to the algorithm in Sec. III-B. Note that with this protocol, there is no need for sequence numbers at this layer.

B. *Unlimited Broadcast - the sequence number method (PI)*

In the source routing method, several copies of the same packet may be sent in any given piconet. This is because the copies have propagated on different routes and therefore the master cannot discover that a different copy of the same packet has already visited this piconet. In order to reduce the traffic of broadcast packets, we have to use master memory. The procedure is similar to the PI protocol of [10]. The source of the broadcast packet includes in the packet the source Bluetooth id (48 bits) and a sequence number. masters that have been upgraded to PI-capability remember the couples (id, sequence number) of the packets they have seen already and do not propagate the packets they have seen already. Note that if this procedure is included in addition to the source routing method of Sec.VI-A, then phased upgrading of masters to PI-capability is possible.

VII. ROUTE DISCOVERY

According to the source routing algorithm of Sec. V-B, user information packets are sent using a source routing algorithm, where the source writes the path to the destination in the RVF field of every packet it sends. Thus it must discover the route to the destination, where each hop is represented by a pair (LocID, MacAddr). Since scatternets are composed of mobile units, we propose a protocol where the source discovers two or more maximally disjoint routes and send each user information packet on all those routes.

A. *Discovery of the first route*

The source initiates a broadcast of SEARCH packets. SEARCH packets are Layer 3 control packets that are broadcast in the entire scatternet according to the protocol of Sec. VI. The payload of the SEARCH packets accumulates the list of pairs that represent the route from the destination to the source as follows. The source initiates the list with its own MacAddr. Each relay that propagates a SEARCH packet appends to the list the LocID of the piconet it has received the packet from and its MacAddr in the piconet it sends the SEARCH to. For example suppose that in Fig. 3, unit E searches for unit A. The headers and payload in some of the SEARCH packets are shown in Table II.

When the destination unit receives the first SEARCH packet of a given search, it returns a unicast REPLY to the source, using the path received in the SEARCH. In the example of Table 2, suppose that the first SEARCH packet received by destination A is the packets that has traversed E,M4,D,M3,C,M2,B,M1. The payload of the received SEARCH contains 6,5,7,2,4,3,2. This means that the REPLY packet and all unicast user in-

TABLE II

HEADER AND PAYLOAD FIELDS OF THE SEARCH PACKETS

	MacAdds	FF	DA	BF	RVF	payload
E	6	1	0	1	E	6
M4	0				E,M4	6
D on M3	7	1	0	1	E,M4,D	6,5,7
D on M6	2	1	0	1	E,M4,D	6,5,1
C on M2	4	1	0	1	E,M4,D,M3,C	6,5,7,2,4
B on M1	2	1	0	1	E,M4,D,M3,C,M2,B	6,5,7,2,4,3,2
M1	0				E,M4,D,M3,C,M2,B,M1	6,5,7,2,4,3,2

formation packets will be sent by A to E with DA=2 and RVF=3,4,2,7,5,6,0. The REPLY accumulates in its payload the path from source to destination in a similar manner as the SEARCH message. In the above example, the payload of the REPLY received by E will contain 5,4,6,1,5,3,1. This allows the source to get the route for user information packets going to the destination. Those will be sent by E with DA=1 and RVF=3,5,1,6,4,5. In addition, for the purpose of building the second route, relays on the first route remember that they are part of the first route.

B. Discovery of the second route

The second route is built similarly to the first, using SEARCH2 messages, except that relays on the first route delay the SEARCH2 packet transmission for a certain period (to be determined). In this way, the relays that are not on the first selected route are given the chance to be on the route over which the first SEARCH2 packet reaches the destination unit.

VIII. DISCUSSION AND CONCLUSION

The protocols presented here have been motivated by several practical design considerations. We envision an environment consisting of a collection of devices communicating with each other over a Bluetooth scatternet. Most instances of such networks (for example in conference rooms, offices, and in homes) will be short lived and small sized. Nodes may join or leave anytime, but we do not expect topology changes to occur very frequently. Further, we aim at making non-PC class devices (printers, slide projectors, light-switches, coffee makers, etc.) to be the first class participants of the network. From this perspective, minimizing soft-state as well as protocol complexity have been the primary goals of our design. We prefer on-demand route discovery over periodic route table advertisement [11] because only a small subset of nodes need to communicate with each other at any given time. We prefer to keep state in the packet (not in routers) in order to minimize state in intermediate nodes. Clearly, there is a limit beyond which such networks cannot be scaled, but we expect our design to suffice in the majority of the application scenarios.

Although, our discussion was limited to unicasting service, we anticipate the need for supporting additional services, such as:

a) finding all units that can be reached over the scatternet.

b) finding all services of a given type that can be reached over the scatternet.

c) finding any unit of a given type in the scatternet.

d) finding all services in the neighborhood of a chosen master node.

These are not new problems. In fact, in the previous work these problems were referred to as *connectivity test* in [10] and *query* in [12]. Using the primitives presented in this paper, problem a) can be solved in one of two ways:

(i) Every master periodically includes in a packet the list of active devices in its piconet and broadcasts the packet in the entire scatternet.

(ii) The unit that needs to discover all devices in the scatternet, referred to as the source, sends a broadcast request packet and in response, every master broadcasts the list of devices in its piconet.

Solution (ii) appears more suitable for Bluetooth, since it saves battery power at the master units. At first glance one may think that a more efficient solution would be to alter solution (ii) and request the masters to send the list of devices in their piconets to the source in unicast packets. However this variation would require the masters to have or discover routes to the source, which is not a trivial task. The problem b) of discovery of all services of a given type can be solved in a similar manner, where the masters include in the broadcast packets only the required type of services. Finding a simple solution to problem c) is still a subject of further research. The solution to problem d) of finding services or other information in a given remote piconet can be solved by first discovering a route to the master of that piconet and then sending special multicast packets whose destination is all units in that piconet.

Although a number of ad hoc routing schemes already exist [13], a unique set of design compromises need to be made for adapting known methods over Bluetooth scatternets. We are in the process of evaluating the effectiveness of our design choices through prototyping and simulation. We hope that our initial protocol design will bring forth the key issues which need to be debated before finalizing a specification for IP over Bluetooth.

REFERENCES

[1] J. Haarsten, "BLUETOOTH - the universal radio interface for ad-hoc, wireless connectivity," *Ericsson Review No.3*, no. 3, 1998.

[2] "Bluetooth technology." http://channel.intel.com/mobile/enterprise/bluetooth/.
[3] M. Corson and A. Ephremides, "A distributed routing algorithm for mobile wireless networks," *Wireless Networks No. 1*, no. 1, 1995.
[4] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers," in *ACM SIGCOMM*, Oct 1994.
[5] S. Murthy and J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *ACM Mobile Networks and Applications Journal*, 1996.
[6] J. Broch, D. Johnson, and D. A. Maltz, "The dynamic source routing protocol for mobile ad hoc networks." draft-ietf-manet-dsr-00, Internet draft, March 1998.
[7] V. Park and M. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *INFOCOM' 97*, April 1997.
[8] I. Cidon and I. S. Gopal, "Paris: An approach to integrated high-speed private networks," *Intern. J. on Digital and Analog Cabled Systems*, vol. 1, pp. 77–85, 1988.
[9] D. A. Pitt and J. L. Winkler, "Table-free bridging," *IEEE Journal on Selected Areas in Communications*, vol. SAC-5, pp. 1454–1462, Dec 1987.
[10] A. Segall, "Distributed network protocols," *IEEE Transactions on Information Theory*, vol. IT-29, pp. 23–35, Jan. 1983.
[11] J. Broch, D.A.Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking, ACM, Dallas*, Oct 1998.
[12] A. Baratz, I. Gopal, and A. Segall, "Fault-tolerant queries in computer networks," *IEEE Transactions on Communications*, vol. 42, pp. 100–110, January 1994.
[13] "Http://www.ietf.org/html.charters/manet-charter.html." IETF Working Group, Mobile Ad-Hoc Networks.