A New Approach to Efficient Bluetooth Scatternet Formation In Adhoc Wireless Network

Hemali Kotak[1], Hiteishi Diwanji [2]
[1] A.V.P.T.I. /Comp. Deptt, RAJKOT,INDIA
Email: Hemali@yahoo.com
[2] L.D. ENGG. College/Comp. Deptt, Ahemedabad, INDIA
Hiteishi@hotmail.com

***Abstract*—Bluetooth enabled electronic devices connect & communicate wirelessly through short-range adhoc networks known as Piconets. Each device can also belong to several Piconets simultaneously. Piconets are established dynamically & automatically as Bluetooth device enter and leave radio proximity. The Bluetooth specification enables to form Scatternet from many devices but it does not specify a protocol for it. The Bluetooth Scatternet formation problem can be split into three parts connection setup, Scatternet building & Scatternet optimization. There are may be various approaches for forming Bluetooth Scatternet which will result in different topology for the same set of nodes. Here we address the problems of Scatternet building & optimizing, discuss problem with diff types of Scatternet & we suggest an approach for Scatternet building.**

***Index Terms*— Formation, optimization, Piconet, Scatternet, Topology**

I. INTRODUCTION

In these days, many people carry handheld devices such as personal digital assistants, laptop computers, and cell phones. In near future, people will share information using a network, which is formed spontaneously by these handheld devices, without the aid of pre-existing equipment such as centralized access points in meetings or conferences. This will be possible due to a mobile ad hoc network, which allows mobile devices to form a temporary network and to communicate with each other without any central infrastructures.

An ad hoc mobile network is composed of nodes that move throughout the network. So the topology changes continuously. In order to make it possible to communicate in the network, a routing protocol is needed to discover and maintenance routes between nodes. There are some routing protocols using distance vector or link state routing algorithms in conventional wired networks. These conventional routing protocols have some problems to be applied directly to adhoc network. The main cause of these problems is mobility of nodes. To solve these routing problems, many ad hoc routing protocols have been proposed. They must be adapted to node mobility that often changes the network topology drastically and unpredictably. The primary goad of the adhoc network routing protocol is correct and efficient route discovery and maintenance between a source and a destination so that messages can be delivered correctly in time. Route discovery and maintenance should be done with minimum overhead and bandwidth consumption. They can be classified into two categories: proactive and reactive protocols.

In proactive routing protocols (e.g., DSDV and CGSR etc.), each node maintains a table containing routing information to all destinations. The advantage of proactive routing protocols is that there is no latency in route discovery. However a lot of routing information may never use and periodical routing update scheme can easily overload the channel capacity in proactive routing protocols. On the other hand, reactive routing protocols (e.g., DSR and AODV etc.) create routes only when there is a need. These protocols do not have to maintain routing tables for all destinations. However, they may not be appropriate for interactive traffic as there is some latency in route discovery.

Recently, a new wireless technology has been developed to enable mobile devices to communicate with each other. Bluetooth technology enables wireless connectivity between mobile devices in an ad hoc fashion easily. Bluetooth embedded in mobile devices can form ad hoc networks, which are called Scatternet. Since Bluetooth Scatternet can be considered as a special kind of ad hoc networks, routing protocols for general ad hoc networks seem to be feasible for routing in Bluetooth Scatternet.

Bluetooth Scatternet has some different characteristics in comparison with general ad hoc networks. First, Scatternet need connection establishment process to communicate with others. If existing reactive ad hoc routing protocols is directly applied to Scatternet, source node will broadcast useless huge route request packets to find route when destination node moves to join another piconet. Therefore forming and reconstructing Scatternet must be considered in scatternet routing algorithm. Second, a node can only hear nodes that are in the same piconet. This means that even if nodes within communication range cannot listen to each other without having same frequency hoping sequences. Third, Flooding in Scatternet can be caused by relay nodes that are located between Piconets

Because of these characteristics of scatternet, general ad hoc routing protocols cannot be applied directly to Bluetooth Scatternet. Some adaptation is necessary. Bhagwat *et al.* proposed a routing vector method (RVM) for routing in Bluetooth Scatternet. RVM is based on source routing. Although RVM operates efficiently over small sized Bluetooth Scatternet, it has large routing overheads and long route search delays in large sized Bluetooth Scatternet. Sun *et al.* proposed algorithms to embed b-trees into a scatternet which enables such a network to become self-routing. However, its tree-shaped scatternet bears intrinsic inefficiency in terms of throughput.

The main discussion is as follows. First, the prior works related to routing protocols for mobile ad hoc network are briefly summarized. Secondly a new routing protocol for Bluetooth scatternet is proposed, and it is more efficient than the existing routing method in terms of throughput, routing overhead and route search delay, in Bluetooth

Compared to the prior works that focus either on the routing algorithm or on the formation algorithm, the proposed method deals with Bluetooth Scatternet as one system, in which mobile nodes form a scatternet and packets, are routed between a pair of nodes. The protocol consists of the formation process, the routing procedure and the mobility management

II. SCATTERNET FORMATION

Given a collection of Bluetooth devices, an explicit topology construction protocol is needed for forming Piconets, assigning slaves to Piconets, and interconnecting them via bridges so that the resulting scatternet is connected. Among some of the formation protocols Bluetooth Topology Construction Protocol (BTCP) that is the first attempt at building Scatternet is reviewed in this section.

Phase 1 : Coordinator Election

When mobile users press start button to connect other users, the first phase of BTCP is initiated. Each node has a variable called VOTE. The VOTE value is set to 1 when it is powered up. After initialization, each node enters INQUIRY or INQUIRY SCAN state alternately. If a pair of nodes discovers each other, they temporarily form a connection and compare their VOTES values. The node, which has larger VOTE, becomes a winner and the other node becomes a loser. (If they have equal VOTES, the node that has larger BD_ADDR becomes a winner) The loser transmits all the device FHS packets of the nodes it has obtained to the winner and disconnects with the winner. Then it enters PAGE SCAN state not to participate in the winner-loser process and the winner repeats this process. When timeout expires, a final winner becomes a coordinator, which knows the number, BD_ADDRS, and clock values of all the devices participating in scatternet configuration process.

Phase 2: Role Determination

In phase 2, the role of each node participating in scatternet configuration process is determined. At first, the coordinator checks the total number of nodes that has been discovered in phase 1. If it is smaller then 8, one piconet is formed and formation process is terminated. If it is larger then 8, the coordinator decides the role of each node. A default criterion of the role determination is to minimize the number of masters for fully connected scatternet. If specific scatternet formation criteria is needed, the participating nodes can be communicated to the coordinator during phase 1 in addition to the FHS information and the coordinator determines the role of the nodes to form a scatternet according to the specific topology using this additional information. After the role assignment, the coordinator forms a temporary piconet with designated masters and transmits to each designated master its connectivity list (SLAVESLIST, BRIDGELIST). Each entry of this list contains FHS packets so that the designated master can later page its slaves.

Phase 3: The Actual Connection Establishment

During this phase, each master x pages and connects the slaves and bridges defined in its SLAVESLIST(x) and BRIDGELIST(x) respectively and the formation protocol is terminated

III. EFFICIENT BLUETOOTH SCATTERNET ROUTING PROTOCOL (EBTCP)

Here an efficient algorithm for routing in scatternet-Efficient Bluetooth Topology Construction Protocol (EBTCP) is proposed. The key features of EBTCP are separating data flows from control flows in the hierarchical scatternet topology, and connecting routing information to a supervisor node.

The following are some terminologies used in EBTCP.

1)**Super Master**

A super master is a master of a piconet, which has masters of other Piconets as slaves.

2)**Relay Node**

When piconets share a slave , this shared slave node is a relay node.

3)***Piconet Information***

Piconet information is BD_ADDRs of nodes that construct a piconet (BD_ADDR of master and BD_ADDRs of slaves in a piconet).

A) Basic EBTCP

In Basic EBTCP, it is assumed that all the devices are within the range of each other. As a power class 2 that is designed for ordinary range (~10m) device is usually used in Bluetooth device, this range is within a radius of 5m or about 7m x 7m. A scenario model of this assumption is that many users in a single room wish to form a wireless network using their Bluetooth enabled devices.

B) Scatternet Formation

Existing Bluetooth scatternet formation methods can be classified into two categories by resulting structure: flat and tree structures as shown in Fig. 2. In the tree structure, all the nodes taking part in a scatternet can easily access to a root node. However, network flows are concentrated on a root link, so it can easily become a bottleneck. On the other hand, in the flat structure, there is no special bottleneck link.

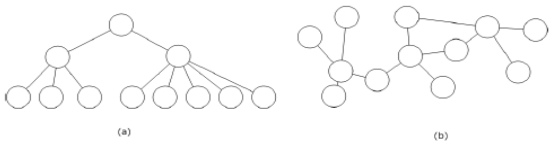


Fig 2 Examples of Tree and Flat structure: (a) Tree (b) Flat.

In EBTCP, a network is configured as a hierarchical structure, which is a hybrid of flat and tree structures. As shown in Fig. 3, layer 0 structure is the same as the flat structure. Also there is a node in layer 1, which is a super master of layer 0 masters. The purpose of forming the hierarchical structures is to make use the advantages of both structures. Here, data packets flow through layer 0 configuration, which has no bottleneck links, and control packets related with routing flow through layer 1 link to reduce routing delay and overhead.

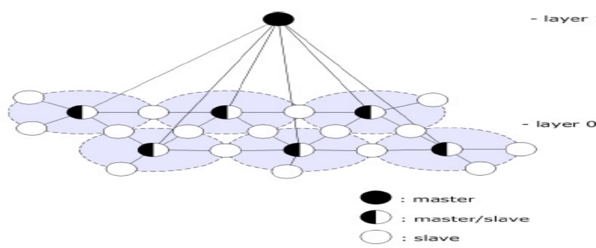


Fig 3 Two level hierarchical formation in Basic EBTCP

Now, let us consider how to form the hierarchical structures. EBTCP formation method is based on BTCP Let us call the coordinator in BTCP as a supervisor, It is assumed that all the relay nodes in layer 0 are slave/slave nodes and connect only two Piconets. Like BTCP, during formation process, a supervisor becomes to know BD_ADDRs and clock information of all nodes and hence knows the total number of nodes participating in the network configuration process. This information will be used not only in the formation process but also in the routing process. The formation process of Basic EBTCP is similar to that of BTCP except the phase 2.

At the start of the phase 2, the supervisor checks the total number of nodes that has been discovered in the phase 1.

If it is smaller than 8, one piconet is formed with the supervisor as the master and all other nodes as its slaves. If it is between 9 and 51, the supervisor determines the roles of the nodes to form a two level structure. As shown in Fig 3 layer 0 has at most 7 Piconets in two level structures. Therefore, there are at most 51 (maximum number of Piconets in layer 0 x maximum number of nodes in a piconet – minimum number of relay nodes + a layer 1 node = 7x8-6+1= 51) nodes in a two level structure. If it is between 52 and 352, the supervisor determines the roles of the nodes to form a three level structure. However, the two level formations will be sufficient in the range of 7m x 7m.

In addition to the role determination, the supervisor chooses a slave as a candidate that will be a supervisor when the present supervisor disappears. If the candidate is selected among masters, all the slaves that are connected to the candidate must rejoin other Piconets when the candidate becomes a supervisor. Therefore the candidate is chosen among slaves.

Finishing the role assignment, a piconet is formed with the supervisor as the master and the designated masters in layer 0 as its slaves. The supervisor sends connectivity list (SLAVELIST, RELAYLIST) to each designated masters. When the supervisor sends a connectivity list to a master that will have the candidate as slave, the supervisor also sends a master list (BD_ADDRs and clock information of masters) of layer 0 to the master. The master list is used when the candidate become a supervisor. In BTCP, the piconet, which has the supervisor as the master and the designated masters in layer 0 as its slaves, is tore down after the supervisor sends the connectivity list. However, in the EBTCP formation, this piconet is permanently connected for hierarchical topology.

Then, the designated masters in layer 0 page to and connect to its slaves. The master of the candidate forwards the master list to the candidate. Fig. 4shows this formation process of Basic EBTCP.

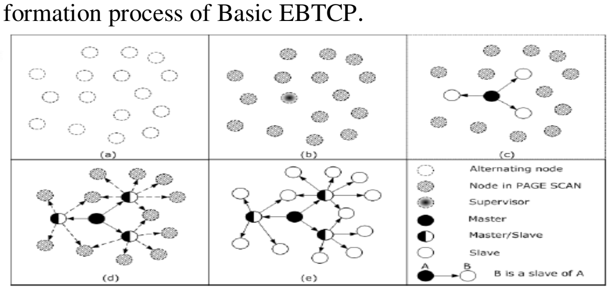
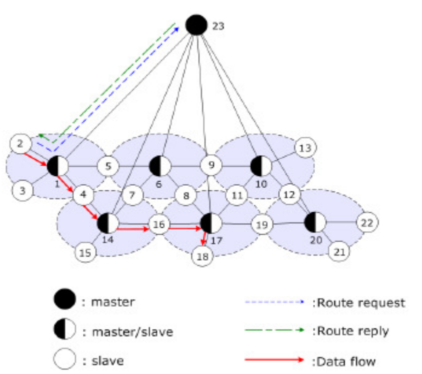


Fig 4 The formation process of basic EBTCP
(a) Start of Phase 1: All nodes start INQUIRY/INQUIRY SCAN to discover their neighborhood. (b) The supervisor is elected at the end of phase 1. (c) Phase 2: The supervisor forms a permanent piconet with the designated masters and sends them their connectivity lists. (d) Phase 3: Each master in layer 0 tries to connect to the designated slaves specified within its connectivity list. (e) Two level formation is configured as a result.

C) Routing tables and routing procedure

When a scatternet formation is finished, a supervisor constructs its routing table. This routing table consists of piconet information that is obtained during the formation process. If a slave belongs to two masters in this piconet information, it means that there is a link between two masters. Hence, piconet information contains the topology of the scatternet. Masters in layer 0 also have a routing table consisting of the BD_ADDRs of neighbor masters and its relay nodes. When the supervisor transmits the connectivity lists to the designated masters of layer 0, it also sends this information in the phase 2 formation process. The size of this routing table is very small, at most 84bytes (maximum number of relay nodes x 2 (corresponding neighbor masters) x 6byte (size of BD_ADDR) = 7 x 2 x 6). EBTCP is based on the concept of source routing . A sequence of BD_ADDRs, which is composed of masters and relay nodes, is carried in a packet header. If this small routing table is used, only the BD_ADDRs of the masters that are along the route are sufficient to forward packets. Consequently, using this small routing table, overheads of a packet header can be reduced by half.

A routing procedure of the EBTCP consists of two phases: route discovery and data delivery. When a slave has a packet to send to some destination, it first send route request message to its master. The route request message is made up of source add and destination add. A master receiving the packet or a master that is source itself forwards it to the supervisor. The supervisor checks whether a destination is a slave or a master. If it is a slave, the supervisor finds the master of the destination from its routing table. Because the supervisor knows the topology of the masters in layer 0, it can find the shortest path from a master of the source to a master of the destination using Dijkstra's shortest path algorithm. After finding the route, the supervisor writes the shortest path, which is consist of the BD_ADDRs of the masters, in a route reply packet, and sends it to the source. When the source receives the route reply packet, the source records the route in its cache. Data delivery is accomplished by source routing. The source writes the sequence of BD_ADDRs in a data packet header and transmits to the next master. A master receiving a data packet checks a packet header and forwards it to a relay node that is connected to the next master using its routing table.



Master	Slave
1	2,3,4,5
6	5,7,8,9
10	9,11,12,13
14	4,7,15,16
17	8,11,16,18,19
20	12,19,21,22

Routing table (piconet list) of node 23

Relay node	Next master
5	6
4	14

Routing table of node 1

Relay node	Next master
4	1
7	6
16	17

Routing table of node 14

Fig 5 An example of basic EBTCP.

An example of the routing table and the routing procedure is depicted in Fig. 5. Numbers of nodes represent BD_ADDRs of devices. In this example, node 2 is a source and node 18 is a destination. To find a route, node 2 sends a route search packet to its master (node 1). Node 1 receiving a route request packet forwards it upwards to the supervisor (node 23). The supervisor receiving a route request packet searches the master of the destination, which is node 17, from its piconet information. Then it finds a master list of the shortest path (1,14,17) from node 1 to node 17. This master list is delivered with a route reply packet to the source along the downward hierarchical path. When node 2 receives the route reply packet from supervisor, node 2 starts transmitting data packets to the node 1. The header of data packets contains the route (1,14,17), the source id (2) and the destination id (18). Node 1 receiving data packets find the next master (node 14) from a packet header and search the relay node (node 4) from its routing table. Then node 1 delivers data packets to node 4. Node 4 finds the next master from the headers of the data packets, and forwards them to the next master (node 14). In this way, data packet is finally delivered to node 18.

D) Mobility Management

In EBTCP, mobility management consists of three processes; formation maintenance, routing table update, route maintenance.

E) Formation Maintenance

As it is assumed that all nodes are within communication range of each other, mobility of nodes within this range does not change network topology. This means that if once a node connects to a piconet, disconnection does not happen even if it moves rapidly within this range. There are two cases that affect scatternet topology in Basic EBTCP. One is that a new node comes into the range; the other is that a node moves out from the range. When a new node comes into the range, it will enter INQUIRY state to find other nodes. In EBTCP, slave nodes that participate in the hierarchical formation do not enter INQUIRY SCAN state and only master nodes in layer 0 enter INQUIRY SCAN state periodically. Therefore, new nodes will find masters of layer 0 in INQUIRY state, and finally become a member of a piconet as a slave in the hierarchical formation. When a master in a layer 0 moves out of range, slaves of the corresponding master are connected to other Piconets or form a new piconet. This can be done by the supervisor. The supervisor can detect the disappearance of the master by using link supervision timers. In Bluetooth, the Link_Supervision_Timeout parameter is used to monitor link loss . If no Base band packets are received from the master for duration longer than the LST, the supervisor disconnects the connection. Then, the supervisor decides roles for the disconnected slaves. If the number of disconnected slaves is larger than some parameter k, the supervisor assigns new roles to the slaves to form a new piconet. Otherwise, it assigns new roles to the slaves to be connected to other existing Piconets. The parameter k is determined by a rule in formation of layer 0. After role assignment, the supervisor sends connectivity list to a new or existing master. The master pages and connects to the disconnected slaves specified in connectivity list and finally reformation is completed.

The supervisor also can move out of range. In this case, all sources that do not know a route must wait until a new supervisor is elected and scatternet is reformed. Therefore a fast recovery of scatternet is important. To do this, a candidate that is selected among slaves (not a master) in layer 0 during formation process is used. The candidate has master list (BD_ADDR and clock value of masters in layer 0). Whenever a master in layer 0 moves out or a new master is created, the supervisor sends the changes to the candidate. When the supervisor move out, masters in layer 0 are aware of this disconnection and change their mode into continuous PAGE SCAN mode (R0 mode). The candidate, which was informed of the disappearance of the supervisor by its master, disconnects itself with its master. Then using the master list that it has, the candidate becomes the supervisor and starts to page masters in layer 0. In this way, the candidate quickly connects to masters in layer 0 and becomes a new supervisor. When a candidate moves out of range, a master of the candidate informs this fact to the supervisor. Then the supervisor selects new candidate among slaves in layer 0, and sends the master list of layer 0 to the new candidate.

F) Routing table update

There are three kinds of nodes in a two level hierarchical formation: slaves in layer 0, masters in layer 0, and a supervisor. Let us consider routing table update for each kind of nodes. When a slave in layer 0 moves in or moves out of the range, the corresponding master of the slave is aware of this change and sends a routing update packet to the supervisor. A routing update packet contains three fields: a BD_ADDR of a mobile node, a BD_ADDR of the master that is connected with the mobile node and a flag to indicate the appearance or disappearance. If the slave is a relay node, a master sends routing update packets to the supervisor after it update its routing table. Upon receiving a routing update packet, the supervisor manages the mobility by adding or deleting the mobile node in its routing table. An example of a slave mobility management is depicted in Fig. 6

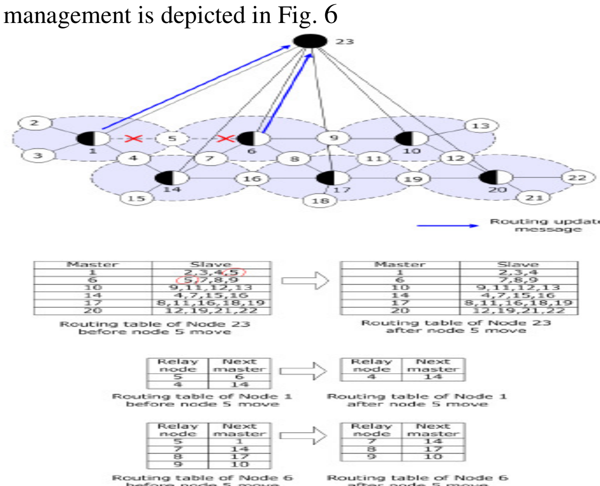


Fig 6 An example of mobility management node 5 moves out of the range.

In this example node 5 move out of the range, so a connection with node 1 and node 6 is disconnected. When a master in a layer 0 moves out of the range, the supervisor can detect disconnection with the master. Then the supervisor deletes piconet information related with the master in its routing table. After the supervisor assigns a role of slave of the disappeared master in formation maintenance process, the supervisor update its routing table according to the role assignment. When the supervisor moves out of the range, a candidate becomes a new supervisor in formation maintenance process. Whenever masters connect with a new supervisor, they send their piconet information to a new supervisor. Upon receiving piconet information, the new supervisor makes its routing table. If the candidate has the same routing table that the old supervisor has, this process is not necessary. But if it does, it is necessary to update the routing table in the supervisor and the candidate at the same time, and it may require lots of bandwidths.

G) Route maintenance

There is one more procedure to manage mobility. It is a route maintenance procedure. If a node along the route moves out of range, its upstream neighbor node cannot find a next master to forward data packets. In this case, it sends a route error message along the backwards path. The route error message contains a sequence of masters' BD_ADDRs along the backwards path, which is obtained from data packet header, BD_ADDR of a source, and BD_ADDR of a destination. When the route error message is propagated to the source, it erases the path from its cache. And then, it reinitiate route discovery for that destination if the route is still needed.

CONCLUSIONS

. Here a novel Bluetooth scatternet routing protocol has been proposed for Bluetooth ad hoc networks. The characteristics of the tree and flat structures were analyzed, and the hierarchical Scatternet topology, which is hybrid of the two structures, was proposed for routing in Bluetooth Scatternet. Using hierarchical formation, the paths for data flows and control flows are separated. As the control packet, which is related with finding a route and updating routing tables, flows through short paths, routing overhead and route search delay can be reduced. And control flows do not interfere data flows. Next, using the hierarchical structures, routing information is concentrated on supervisor node. When a node wants to find a route, it queries the route to the supervisor. Upon receiving the query, the supervisor replies the route to the node. The route is always the shortest path. Compared to other table driven routing methods in which all the node have routing table, EBTCP can reduce routing overhead by using this query-reply method. The Data delivery method of EBTCP is based on the source routing.

The mobility management is also provided for maintaining topology and routing in EBTCP. When a node moves and a network topology changes, this change is reported to the supervisor. Then, the supervisor updates its routing table to meet the change. In comparison with the most of other routing methods that use flooding scheme to update routing tables, EBTCP can reduce routing overhead and wastes of network bandwidth and power consumptions of devices by using this unicast update Scheme

Some suggestions for future works are addressed. First, the problem of defining scatternet formation criteria in the bottom layer is an important issue. Although the problem is out of the scope of this dissertation, it is the subject of future works. Second, the scheduling algorithm between lower layer master and super master is expected. When a master switches to its super master, its piconet members cannot communicate with others until the master switches back to its piconet. This causes decreasing of throughputs in the hierarchical formation scatternet.

.

REFERENCES

[1]] P. Bhagwat and Adrian Segall, "A Routing Vector Method (RVM) for Routing in Bluetooth Scatternets," in Proc. 6th IEEE Int. Workshop on Mobile Multimedia Communications (MOMUC'99), San Diego, USA, pp. 375 –379, November

[2] Bluetooth Resource Center, http://www.palowireless.com

[3] J. Bray and C. F. Sturman, Bluetooth: connect without cable, NJ: Prentice-Hall, 2001

[4] J. Broch, D. B. Johnson, and D. A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," IETF Internet draft, draft-ietfmanet- dsr-07.txt, February 2002.

[5] E. Dijkstra, "A note two problems in connection with graphs," Numerical Mathematics, pp. 269-271, 1959.

[6] C. Law, A. K. Mehta, and K. Siu, "Performance of a New Bluetooth Scatternet Formation Protocol," in Proc. ACM Int. Symposium on Mobile ad 2003/02/19 02:43:01 Bibliography

Authors

Hemali B. kotak

Sr. lecturer in Computer engg.

A.V.P.T.I. RAJKOT (Gujarat)

INDIA

(hemalikotak@yahoo.com)

ad hoc Networking and Computing, October 2001.

[7] *G. Pei and Mario Gerla, "Mobility Management for Hierarchical Wireless Networks," ACM/Kluwer Mobile Networks and Applications, vol. 6, no. 4, 2001*

[8] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," IETF Internet draft, draft-ietf-manet-aodv-07.txt, February 2002.

[9] E. M. Royer, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," IEEE Pers. Commun., vol. 6, no. 2, pp.46-55, April 1999.

G. V. Zaruba, S. Basagni, and I. Chlamtac, "Bluetrees-Scatternet Formation to Enable Bluetooth-Based Ad Hoc Networks", in Proc. IEEE Int. Conf. on Commun.(ICC'01), Helsinki, Finland, pp. 273-277, vol. 1, June 2001