Bluetooth Mesh Network Protocol

ONDŘEJ HYNČICA, FRANTIŠEK ZEZULKA, PETR FIEDLER
Department of Control and Instrumentation, Brno University of Technology
Kolejní 4, Brno
CZECH REPUBLIC

Abstract: Paper deals with simple and efficient protocols for forming and maintaining a Bluetooth network. The protocols are designed for mobile situations with frequent topology changes and for routing of translated messages between Bluetooth network members. To provide a simple, distributed solution to form an ad hoc network the protocols are based on a set of rules. The resulting topology is a mesh network with defined properties. For the network routing a modified DSDV protocol is employed. The proposed protocol enables self re-configuration of the network after communication failure and were tested in numerous simulated situations.

Key-Words: Bluetooth, ad-hoc, network topology, routing

1 Introduction

The Bluetooth technology has proved successful in many applications and its –amount – has significantly reduced the costs. This makes the use of Bluetooth attractive for areas and applications it was not primarily designed for. Such an application may be networks with large number of nodes.

The Bluetooth standard supports construction of larger network, but does not define how. The question of the network structure is very important, as it has significant effect on the communication properties. To meet different and specific requirements various network structures for Bluetooth have been proposed, like the tree topologies (Fig. 1) in [1, 2], star [4], ring [3] or several approaches that build mesh networks [5, 7, 6].

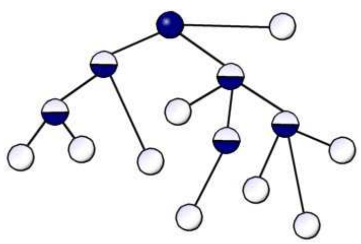

Fig. 1: A Bluetooth tree topology

The approach described here combines several existing protocols and focuses on mobile ad hoc operation with frequently required topology changes, lost links, but with relatively low traffic with no explicit pattern. The target application is communication for mobile platforms. The character of the exchanged information is periodical sensory data, status information and commands. The sensory data and the status information are expected to be broadcasted regularly.

The paper is organized as follows: section 2 describes some of the existing network structures and section 3 introduces the proposed structuring protocol. In section 4 is briefly explained the routing scheme used in the network, section 5 presents simulation results and discussions and the conclusion is brought in section 6.

2 Bluetooth Networks

Many different Bluetooth network structures have been proposed. They can be divided into two major groups, structured networks with a specific topology and networks without explicitly defined structure - mesh networks.

In [1] is described protocol that builds a "Bluetree" topology, a spanning tree that can be build either centralised (rooted) or decentralised. It is a simple and efficient protocol, which is well suited to the Bluetooth constraints. Another tree topology is shown in [2] - the TSF (Tree Scatternet Formation). Here the algorithm incrementally builds a tree structure of scatternets.

For all the tree-structures networks is the main advantage a simple, well defined structure enabling simple routing and scheduling. The nodes can be assigned unique addresses based on their position in the tree and significantly reduce the routing overhead [2].

A unique approach to Bluetooth topology is described in [3]. Here every node in the network acts as a master-slave bridge and has exactly two connections, forming a ring topology. Different ring approach is proposed in [8].

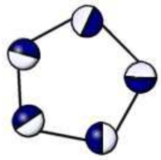

Fig. 2: BlueRing topology [3]

The second group, the mesh networks, takes out the problems with topology, but usually at the cost of increased routing and scheduling complexity. The idea behind the mesh networks is to define the desired properties (rules) for the network and build a network to meet these properties.

A network structuring protocol BTCP (Bluetooth Topology Construction Protocol) is described in [5]. The protocol sets rules for the resulting network that is built by an elected network coordinator. Another very effective protocol that is fully decentralised, the Bluenet, is presented in [6]. The network construction runs in several phases and builds a network to comply with defined properties. The BlueStars topology is described in [4]. The idea is to build a network consisting of piconets where every two piconets share one bridge. The resulting topology brings a reasonable trade off between the number of links and average network path.

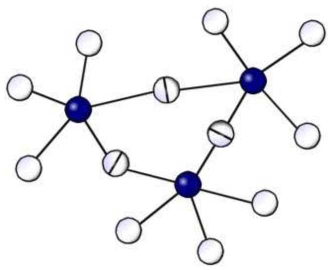

Fig. 3: Example of a BlueStars configuration

The networks with defined topology provide the advantage of simplified routing and scheduling, because the knowledge about network structure can be facilitated. The main disadvantage is handling of high mobility, since these networks need to restore their topology to recover and it can be difficult (merging of trees) or even impossible (a ring that cannot be closed).

The mesh networks can be more flexible, because they do not require a specific topology to work. The chosen properties can be tuned for a given application. Disadvantages are the network structure is meshed and no information about the resulting structure is available. For the expected application with high mobility a meshed topology seems better suited.

3 Structuring Protocol

The previous section introduced several from the many proposed network structures for Bluetooth. The protocol described here is intended for an application with high mobility of some of the nodes, with frequent link failures and topology changes. The key property is the connectivity, the traffic properties are not very important, as only smaller amounts of data are expected to be transferred.

The structures described above usually take an initial phase and then build a network with respect to the structure or rules, this often requires creating temporary links, which are later teared down and only few protocols care about later maintenance. But these are the key issues for a mobile network. The presented protocol aims to build a structure that has the properties as the Bluenet or BlueStars, but using an ad hoc protocol, based on the principles described in [7].

The protocol consists of two sets of rules - Connectivity rules and Structure optimizing rules. The Connectivity rules are responsible for maintaining a connected network when possible. The connectivity is achieved using the principle introduced in [7]:

> To ensure the reachability of each node in the network it is sufficient to provide the reachability of every visible node. This means when a node can reach its visible neighbors then it can reach the visible neighbors of these nodes, too, and so on.

Visible node is a node that is within Bluetooth radio range and was discovered during the Bluetooth Inquiry / Inquiry-scan procedure. A node is *reachable* if it can be reached through the network. *Reachability distance* is the number of hops between two reachable nodes. The connectivity rules can be basically expressed using the connected network definition above as “try to connect to any node that is visible but not reachable”.

The information about visible nodes is critical for maintaining a connected network, but unfortunately in Bluetooth is complicated, because Bluetooth does not support symmetric discovery protocol. Such a protocol must be defined on the user level by altering the Inquiry and Inquiry scan modes [4, 5] and will be referred to as seek procedure.

The link redundancy in the network can be controlled by limiting the network distance for which a node is considered reachable. This distance is denoted as reachability distance. The reachability distance is provided by the routing protocol that takes it as a metric. With this modification more connections can be established, because even if the node is reachable, when the distance exceeds a limit a new link will be created.

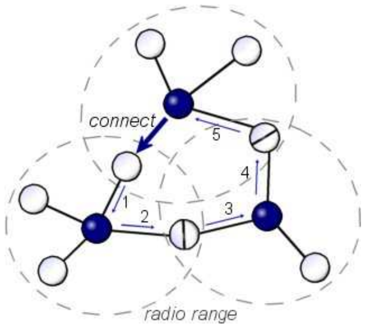

Fig. 4: Adding redundancy into the network, the connection will be set if the reachability limit is four or less

The rules depend on the state of the node. The nodes in network can be in three basic states: free, master, slave. The master and slave nodes can become bridging nodes, called slave bridge and master-slave bridge (acting as a slave in one piconet and a master in another) respectively (Fig. 5).

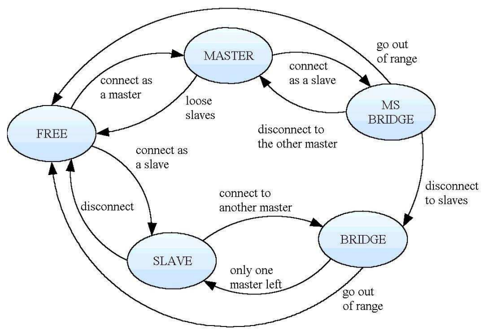

Fig. 5: States of a node in the Bluetooth network

Any free node tries to connect to any discovered node. To get connected it needs to discover nodes within range first, so it enters the seek mode. After it has found a node in range it initiates a link setup. As it does not know anything about the nodes who responded, it connects to the first discovered node. The connection is created with the role switch enabled. If the free node tries to connect to a master, the master will accept the connection (unless it has already seven active slaves) and perform the master-slave switch, so the free node will become a slave. If it tries to connect to another free node, the other node will accept and become a slave in this new piconet. If it connects to a slave node, it will become a master of a new piconet and the slave will become a slave bridge between the new and its old piconet. To decrease the number of created piconets, when the free node discovers that it has connected to a slave it tries to connect to its master. When successful, it joins the piconet and tears down the link with the slave. The simplified version of the rules is shown in Alg.1.

Algorithm 1 Free node connectivity rules

Seek;
IF(invited) Connect, become *SLAVE*
IF(discovered any) Try to connect;
IF(not connected) Go back to Seek;
ELSE IF(connected to master) become *SLAVE*
ELSE IF(connected to bridge with visible master)
 Try to connect(to its master);
 IF(connected) Drop the bridge, become *SLAVE*;
ELSE become *MASTER*;

Master nodes try to enlarge their piconets up to five slaves more slaves can be accepted if no other piconet is willing to take the free nodes and a new piconet would have to be created. This is not shown in the simplified algorithm for master nodes Algorithm 2. The master nodes do not become bridges, unless it is the only way to maintain connectivity.

Algorithm 2 Master node connectivity rules

IF(time to seek and slaves less than 5)
 Seek;
 IF(invited by a free node) Accept it as a slave;
 IF(found unreachable node) Try to connect;
IF(time for slaves to seek) Let slaves seek;
IF(still unreachable master found) connect as *MS-BRIDGE*;

The slave nodes are periodically ordered by their master to enter the seek mode and look for unreachable nodes and piconets. In such case they become bridges or master-slave bridges .

Algorithm 3 Slave node connectivity rules

IF(master order to seek) Seek;
IF(invited) accept, become a *BRIDGE*
IF(found unreachable)
 Try to connect;
 IF(connected) become *BRIDGE* or *MS-BRIDGE*

The connectivity rules themselves are sufficient to maintain a network. But to define more requirements, another set of rules to modify the network is proposed - the structure optimizing rules. These rules are implemented to:

- reduce the number of piconets
- remove unnecessary bridges
- provide shorter links

The structure optimizing rules are primarily executed by the master nodes. The rules are decentralized, without the need for a global knowledge of the network. The sources of information are the direct connections, the regular update messages provide reachability information and the seek procedure gives information on visible nodes. The rules are then executed locally, using the provided information. To avoid conflicts the rules are evaluated according to the following conditions:

1. Connectivity - rule can be applied only if it does not break connectivity
2. Priority - rules are evaluated in the order of importance (according to the listing above)
3. Piconet size - the master with more slaves wins
4. Distance - the connection over a shorter distance is preferred
5. ID - as the last measure is taken the unique ID

All the optimizations are based on moving nodes from one piconet to another. The structure optimizing rules may use the comfort of a connected network provided by the connectivity rules and negotiate the modifications. The negotiations are conducted by masters, using a take-join principle.

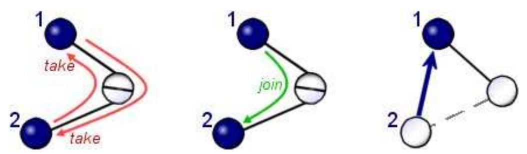

Fig. 6: An example of take-join procedure for the piconet number minimizing rule. First masters (1) and (2) activate the rule and send the TAKE messages. The master (2) has higher ID, so it does not respond to the request. The master (1) responds with JOIN and (2) accepts and joins (1)

The "take" request is send by a master who wants to give a slave or bridge to all visible masters. Any master that wants to take it, responds with a "join" offer. Finally the first master selects the most suitable master from those who responded with join (see the example for the first rule in Fig.6).

The first rule, minimizing the number of piconets, is done by merging piconets. This is done by the take-join mechanisms. Considered optimal size of piconet is at least three and maximally five. The maximal number is applied to leave a space for possible new nodes. Every master with less than 3 slaves wants to give up its piconet and merge with others. It sends the take message for each of its slaves to all visible and reachable masters. The masters who are ready to accept the incoming slave respond with join message. When the master has only one slave acting as a bridge left, it issues a take message for itself and becomes a slave in another piconet (this is illustrated in the example in the Fig.6).

The second rule, removing the additional bridges is applied to limit the redundancy in direct inter-piconet connections. When two piconets share more than the maximal allowed number of bridges (the results are presented for two) one of the masters disconnects to one of the shared bridges. The decisions which master should disconnect to which of the bridges are carried by the masters based on the conditions defined for the rules. The rule is also employed to decrease the degree of nodes in the network. When a bridge node has more than two outgoing links, it requests one of its masters to select another bridge. If the master connects to another bridge, then a redundant interconnection occurs and the rule will break one of the links, presumably the one of the node with the highest degree.

The rule 3 is employed because of the fact the communication links over shorter distance may provide higher quality and are more likely to last longer. For this reason if a slave has possibility to connect to another master over a shorter distance it tries to move to its piconet using the take-join negotiation. This enables handling the situation when a master node is moving from its piconet to enable the slaves to connect to other piconets before the connection is broken. The selection of a suitable master is carried out in each of the slaves, as the masters do not know what are the visible nodes for its slaves.

There are other situations that may be handled by additional rules. The protocol enables using more rules, only the basic guidelines need to be followed.

4 Routing Protocol

In the Bluetooth specification the routing is not addressed and has to be implemented in higher levels. The routing is needed for intra piconet routing and for inter-piconet communication in the scatternet. Routing is closely related to the network structure. There are some topologies designed with the purpose to facilitate the routing (trees, rings, clustered networks). But the advantage of simple or even trivial routing is payed by increased complexity of the network structure. For the proposed network structure a general and robust routing protocol needs to be designed.

The routing protocols can be categorized into two main groups - proactive (table-driven) and reactive (on-demand). The proactive routing maintains information about all the nodes in the network. Protocols from this group evaluate all the routes within the network, so a route is ready immediately when a packet needs to be forwarded. The available routes are stored in tables maintained at each node. The consistency of the tables is maintained by broadcasting updates.

The reactive routing creates routes only when they are needed, on demand. These protocols employ a global search to discover a path to forward a packet. Once the route has been discovered, it is being used. When a node needs to send a packet, first it searches for a path, usually by broadcasting route discovery packets and then forwards the packet along this route.

A typical member of the proactive protocols for wireless networks is the Destination-Sequenced Distance-Vector routing (DSDV), a table-driven algorithm proposed in [9]. It uses the Bellman-Ford algorithm improved to include freedom from loops by employing sequence numbers. Each node in the network stores routing tables listing all available destinations and number of hops to each. The number of hops is used as a metric.

The presence of periodic broadcast messages and status information expected in the application is the main guideline in the selection of the routing protocol. For these reasons is selected a proactive solution based on DSDV, because the periodic broadcasted messages can be utilized for the routing table updates, so no additional broadcasts for the purposes of routing are needed, the route is known in advance and it is a simple, distributed solution that provides optimal path.

The protocol builds a routing table consisting of entries where to forward a packet. If the route between two destinations is changed, the DSDV can not deliver the message before a next broadcast repairs the route.

With high rate of mobility the DSDV can have problems converging and that can lead to packet losses due to staled route information. To avoid this problem, an extension to the DSDV is proposed. When a node receives a packet that it considers to be undeliverable, it marks it as lost and broadcasts it to assure the delivery. The broadcast is done by flooding. This will deliver the packet if the lost node is within the network. It can be used only for small amounts of data otherwise the lost packets may flood the network after a major change in the topology. The extension to the DSDV with the lost packets enables to handle highly mobile situations, where flooding becomes the only possible solution.

5 Simulation and Results

The proposed protocols have been tested in a simulation. The simulation is limited to testing the rules for the initial situations and node movements requiring restructuring.

The simulation is done in discrete time steps, where the simulation cycles are inserted into node movement cycles. The simulation is done on the level of the Bluetooth baseband packets. In each cycle every node is able to transmit and receive a packet on all of its links. The polling, bridge scheduling and the Bluetooth power saving modes are not covered by the model. The inquiry and inquiry-scan is simplified that when a node performing inquiry-scan in a given cycle is in range of a node performing inquiry, it will be discovered.

The simulator uses a Matlab interface and enables step-by-step replay and statistical summaries. The important metrics of the Bluetooth network are the number of piconets, number of bridge nodes and the average and maximal network path.

Two basic series of tests have been conducted - tests of initial network formation (randomly located nodes, not moving) and test during a simulated operation where the nodes are moving through the area. The results of one of the initial formation tests are shown below in the table. The simulation was done for 10,000 random locations in an empty area 20×20 m (with the communication range of 10 m only 47 % of links in average were possible to be created).

Table 1: Initial formation simulation

Nodes	5	10	15	35
Piconets no.	1.7	2.9	4.1	9.6
Slaves per piconet	2.2	3.4	3.8	4.5
Bridges no.	0.5	1.7	3.3	14.5
Created links	3.9	13.7	24.0	79.1
Dropped links	0.7	4.6	9.0	36.5
Max. route	2.4	4.8	6.7	10.1
Avg. route	1.5	2.5	3.2	4.0

The results showed the protocol creates a connected network when possible very efficiently. The structuring rules were able to keep the number of piconets low and small bridging overhead while keeping the routes in the network reasonably short. One of the examples how the constructed network looks like and the effect of structuring rules is shown in Fig.7.

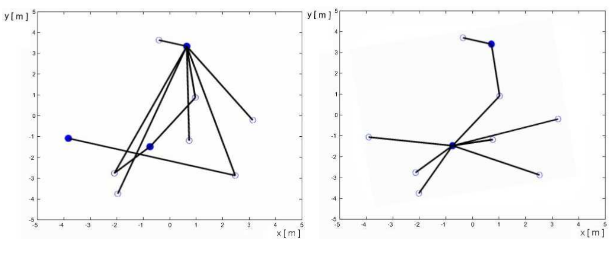

Fig. 7: Example network configuration, a) without structuring rules, b) with all rules applied

The success of the structuring rules in reducing the paths in networks can be seen in the figure Fig.8.

The next simulations were done for an “application run” were all the nodes were moving around the area. The results have shown that the connected network is constructed whenever possible, but the biggest influence has the seek procedure.

The problems may arise at several points. First it is the Bluetooth discovery itself that can take up to 10 s.

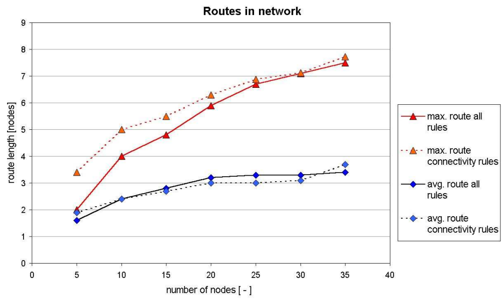

Fig. 8: Average and maximal paths with and without the structure optimizing rules

Then it is the discovery protocol. It is realised by a random changing of the inquiry and inquiry-scan modes [4, 5]. The last and very important matter is the timing of the seek procedure itself. The frequency of the seek differs with the node state. The seek procedure was not given much attention here, but the results have shown that it is necessary to give it close attention to get satisfactory results.

6 Conclusion

In this work a communication protocol for a mobile Bluetooth network has been proposed. The protocol includes routing and network structuring protocols.

The network structuring protocol is proposed to cover both initial network formation and maintenance. The proposed structuring protocol is selected not to build a specific topology, but a network with a specified properties given by a set of rules. The network construction is done by connectivity rules that assure connectivity and handle node mobility and link failures. To restrain the properties of the network the structure optimizing rules are defined. The rules reduce the number of piconets, the degree of nodes, the bridging overhead and the connection distance. The main focus is on connectivity.

Because the selected network structure does not provide any advantage for the network routing, a general routing protocol was adopted and modified. From the available ad hoc routing protocols, the DSDV table-based routing protocol has been chosen. The reasons were the expected tables existing at the nodes and regular broadcast messages required by the application. In this way, the main disadvantages of the DSDV (i.e. need for tables and regular update messages) are eliminated. Because the DSDV does not assure packet delivery in highly mobile situations, it was extended in a simple way to employ flooding principle for undeliverable packets.

The proposed network protocols aim to provide simple and efficient solution for maintaining a communication network. The realization based on rules can be easily implemented into simple systems and does not require storing of additional topology information. The problem of the rule based solution is that it is complicated to tune and the rules may interfere or collide. But the results for the intended application are very promissing and the protocol seems to meet all the prerequisites set.

Acknowledgements

This work was supported by the Brno University of Technology, Grant Agency of the Czech Republic grant GA 102/ 05/1097 and grant GA 102/05/0467. Without kind support of the above-mentioned agencies the research and development would not have been possible.

References:

[1] Záruba, G. V., Basagni, S., Chlamtac, I.: Bluetrees-Scatternet Formation to Enable Bluetooth-Based Ad Hoc Networks. Proc. IEEE ICC'01, June 2001.

[2] Tan, G., Miu, A., Guttag, J., Balakrishnan, H.: Forming Scatternets from Bluetooth Personal Area Networks. MIT-LCS-TR-826, October 2001.

[3] Chun-Choong F., Kee-Chaing, C.: Bluerings-Bluetooth Scatternet with Ring Structures. IASTED WOC 2002, July 2002.

[4] Petrioli, C., Basagni, S., Chlamtac, I.: Configuring BlueStars: Multi-hop Scatternet Formation for Bluetooth Networks. IEEE Trans. on Computers, special issue on Wireless Internet, 2002.

[5] Salonidis, T., Bhagwat, P., Tassiulas, L., LaMaire, R.: Distributed Topology Construction of Bluetooth Personal Area Networks. 2001.

[6] Wang, Z., Thomas, R. J., Haas, Z.: Bluenet - a New Scatternet Formation Scheme. Proc. IEEE HICSS 02, 2002.

[7] Félegyházi, M.: Development and Evaluation of a Dynamic Bluetooth Scatternet Formation Procedure. Master's Thesis. Budapest University of Technology and Economics. 2001.

[8] Lin, T.-Y., Tseng, Y.-C., Chang, K.-M., Tu, C.-L.: Formation, Routing, and Maintenance Protocols for the BlueRing Scatternet of Bluetooths. Proc IEEE HICSS'03. 2003.

[9] Perkins, C. E., Bhagwat, P.: Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. Comp. Commun. Rev., October 1994.