Proceedings of the 4th International Autonomous Minirobots for Research
and Edutainment Symposium, pp. 189-195, Buenos Aires, October 2007

A Bluetooth Scatternet for the Khepera Robot

Jia Lei Du, Ulf Witkowski, and Ulrich Rückert

System and Circuit Technology Research Group, Heinz Nixdorf Institute,
University of Paderborn, 33102 Paderborn, Germany
{jialdu, witkowski, rueckert}@hni.upb.de

Summary. Radio-based communication plays a vital role in multi-robot systems. Bluetooth is an energy-efficient communication technology suited for resource-limited mini-robots such as the Khepera. However, the maximum number of nodes in a Bluetooth piconet is limited, while scatternets - networks of piconets - have not been fully specified. In this paper we present a Bluetooth scatternet using Bluetooth communication sticks developed in our research group. In our solution, bridge nodes carrying two of such Bluetooth sticks are used to interconnect piconets. Beside the developed hardware, issues such as routing as well as topology control are addressed. Finally, data rate and latency measurements are presented for the implemented solution.

1 Introduction

It is argued that multi-robot systems can offer a number of advantages over single-robot systems, including speedup of parallelizable tasks, increased reliability and robustness due to specialization and simplification of the single team members, and the ability to solve tasks that single robots cannot such as distributed sensing. However, to fully exploit the potential, communication among the robots is essential. For communication, in most cases radio-based technologies such as WLAN or Bluetooth [1] are used. Particularly in resource-limited mini-robot systems like the Khepera robot [2], an energy-efficient communication standard such as Bluetooth is a suitable choice. However, Bluetooth has not been fully specified for larger networks, so called Bluetooth scatternets. In this project we implemented a form of Bluetooth scatternet using the Khepera robot and a Bluetooth communication module developed by our research group. The final objective is to provide a working Bluetooth scatternet for communication in a network of robots. The user is presented with a few simple access functions for sending and receiving messages, details such as topology control and routing are automatically executed and hidden from the user.

2 State of the Art

Various protocols for scatternet topology formation and/or routing have been suggested in literature ([3],[4],[5],[6],[7],[8]). However, with the exception of [8], evaluations of these solutions were only done in analysis or simulation. In [8] a scatternet implementation is presented with data rates of 50 kb/s for single-hop connections and 15 kb/s for multi-hop connections. Routing is based on a reactive routing protocol. The measured average per hop latencies ranged from 165 ms to 1400 ms, depending on the availability of node connections and routing table entries. The solution presented here has lower data transfer rates but offers comparatively constant (low) latencies as it is based on a proactive routing protocol and regularly maintains node connections and routing tables entries. This is a major advantage in time-critical applications.

3 Solution

In this section the implemented solution for our Bluetooth scatternet is described. First the used hardware components are described and then the issues of routing and topology control are addressed.

3.1 Used Hardware

Bluetooth is a communication technology standard for low-bandwidth, short-range, low-cost communication. A Bluetooth network is based on a master-slave architecture. A master controls up to seven slaves, forming a so-called piconet. In this project we use a Bluetooth stick developed by our research group [9]. It is based on a Bluetooth radio module with an integrated antenna from Mitsumi. The micro-controller of the Bluetooth stick communicates via a serial interface with the Motorola 68331 controller of the Khepera robot. The stick can be used both for connecting Kheperas to a PC as well as for communication among robots. Communication within such a piconet is based on Time Division Multiple Access (TDMA). All members synchronize their internal clocks with the piconet master. The master assigns communication time slots to each member of the piconet. Slaves of a piconet cannot communicate with each other directly, only via the master. This piconet structure is a limiting factor in the creation of larger Bluetooth networks. As medium access is based on TDMA, the communication latency increases linearly with the number of members in a piconet. For this reason, the maximum number of communication nodes in a piconet was arbitrarily set to eight. To form larger networks, the Bluetooth specification suggests joining multiple piconets in a so-called scatternet. However, in the specification only the basic idea is provided which partly explains the lack of real implementations up to date. To create connections between piconets, some network nodes must be members of at least two piconets. Thus, a scatternet can be created either by using a single Bluetooth radio module that is active in two or more piconets or, as described here, by using multiple Bluetooth radio modules per node (connected via the micro-controller of the Khepera in this case). To connect two Bluetooth sticks to the robot requires two serial interfaces. In our research group an FPGA board [10] for the Khepera has been developed including an adaptor which can carry up to four Bluetooth sticks (Fig. 1). The FPGA is used for interfacing the serial connections of the Bluetooth sticks via a FIFO buffer to the Khepera's K-Bus.

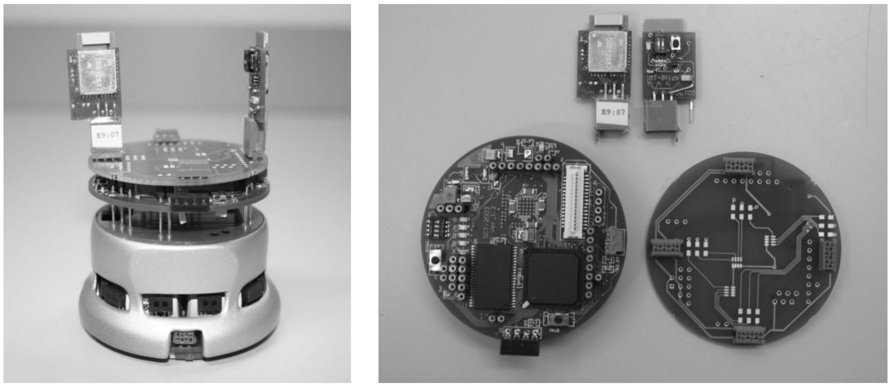

Fig. 1. A Khepera robot carrying two Bluetooth sticks serving as bridge node (left). The used FPGA/adaptor/Bluetooth modules (right).

3.2 Routing

Wireless mobile networks can be very dynamic, nodes are mobile and may enter and leave the communication range of other nodes or even the network at any time. Communication over long distances is based on multi-hop connections where nodes on the path between source and destination act as routers and forward messages. Various routing protocols have been developed and proposed for such networks, which can be classified in reactive, proactive and hybrid protocols. Proactive protocols built up routes in advance, whereas routing finding in reactive protocols is on-demand and only initiated when data actually has to be send.

Proactive protocols can offer low latencies as the routes are already available when needed. Destination-Sequenced Distance-Vector Routing (DSDV) is a typical example for a proactive routing algorithm [11]. Every node knows its direct neighbors and uses regular messages to maintain the connection to those neighbors. Each node in the network constructs its routing table by exchanging shortest path information with its neighbors (e.g. 'My shortest path to node X is via node R with cost of C.'). The constructed routing tables contain the address of every node in the network and an optimal route (based on some metrics, e.g. number of hops) to those nodes.

One disadvantage of proactive protocols is usually that traffic is required to maintain the routing tables even when no user data is being sent. However, since Bluetooth uses a master-slave architecture, the neighbors of a node are always known. As this neighborhood information can be used by the routing protocol at no cost, a proactive protocol can implemented without having to exchange messages for neighborhood detection. We implemented such a slightly modified DSDV protocol in this project. Only in the case of topology changes (new nodes, lost nodes) traffic is generated by the routing protocol for propagating routing information through the network.

3.3 Topology Control

Bluetooth networks are strictly organized due to the master-slave architecture. When the network is initiated, all bridge nodes start searching for unconnected nodes. Every bridge node tries to connect to at least one other bridge node to ensure that the network does not become fragmented. On the other hand, to limit the load on bridge nodes, avoid loops, and to connect as many ordinary nodes as possible, the bridge nodes try to not connect to more than two other bridge nodes. Only if no unconnected bridge nodes are within communication distance, all slots may be filled up with ordinary nodes. The environment is regularly probed for new bridge nodes. When a new bridge node is detected,

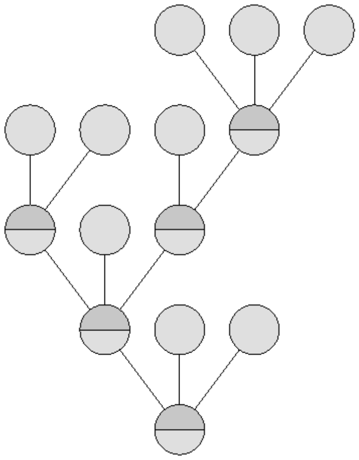

Fig. 2. Illustration of a possibly emerging topology when using the developed solution. Ordinary nodes that are connected to the network as slaves are in light grey. Bridge nodes ensure connectivity in the network and are half light grey (slave) and half dark grey (master).

it is integrated into the network. If necessary, a slave node may even be disconnected and reconnected via the new bridge node. If enough bridge nodes are available in the network, a fully connected network is established after some time. An example for a resulting network topology is shown in Figure 2.

It would be possible to equip bridge nodes with more than two Bluetooth sticks and thus join more than two piconets via a single bridge node. However, this has not been implemented as it would further increase the load on bridge nodes and the failure of such a node would affect three or more piconets at once.

4 Results

Sending and receiving messages in the network can be done using only a few simple user functions that hide the complexity of the underlying algorithms for topology control and routing which are executed in parallel background processes (Listing 1). In the following subsections the results of our data rate and latency measurements are presented.

Listing 1. Functions to send and receive data in the scatternet

```
void scatt_send_packet(char *destination, uint8 *ttl,
                       uint8 *id, char *data);

void scatt_receive_packet(char *source, char *id,
                          char *data);

// destination/source: BT address of target/source (12 byte)
// ttl (time to live): maximum number of hops (1 byte)
// id: packet ID (1 byte)
// data: data to be send/received data (46 byte)
```

4.1 Latency

For data transmission in real-time systems such as robots, latency is an important property. To measure communication latency in our solution, four nodes were used, two bridge nodes and two slave nodes. Each of the bridge nodes managed one of the slaves and the two bridges nodes were connected. All routing information was exchanged before our measurements. Every three seconds broadcast packets were send out by one of the slaves and all recipients of the broadcast replied with an acknowledgement. The time between the sending and the arrival of the reply was measured. So the first entry in Table 1 gives the latency for communicating with the own bridge node, the second entry gives the latency for communicating with the next bridge node, and the third entry gives the latency for communicating with the other slave in the neighboring piconet. All latencies are measured as round-trip times.

Number of Hops \ Latency	Minimum	Maximum	Average
2 Hops	336 ms	419 ms	388,5 ms
4 Hops	615 ms	673 ms	645,86 ms
6 Hops	1023 ms	1129 ms	1088,02 ms

Table 1. Latencies for communicating in a network of two piconets with two nodes in each piconet.

4.2 Data rate

With the used Bluetooth sticks, a up to three slaves can be connected to a master and a maximum data rate of 86 kb/s is achievable from slave to master and 9 kb/s from master to slave. These limitations are caused by the low performance of the controller in the Bluetooth stick. The lower data rate from master to slave can be explained by the additional demands to the controllers on master nodes, for example for piconet administration. We used broadcast packets to measure maximum throughput. The sending of a broadcast packet requires 27 ms, with a packet size of 73 bytes. Those 73 bytes are split into 2.67 fragments by Bluetooth with 8 additional bytes per fragment. Thus, the piconet master receives a maximum of 27.96 kb/s per slave, and with three slaves 84 kb/s can be achieved which corresponds to 98% of the maximum possible throughput. When messages are forwarded across piconets, the maximum data rate is 9 kb/s as there are always master-to-slave transmissions involved.

5 Conclusion

We showed how a scatternet solution using Bluetooth sticks was implemented and tested on the Khepera robot. This distinguishes our work from most related works that focus on simulation. Connections between piconets are created using bridge nodes carrying multiple Bluetooth sticks. To connect multiple Bluetooth sticks to the Khepera, an FPGA-based extension module was developed that interfaces up to four sticks with the Khepera's K-Bus. A modified DSDV routing protocol and an algorithm for constructing a scatternet topology in the network was implemented. The developed solution is transparent to the user and automatically constructs a network using the available bridge nodes and ordinary nodes in the environment. The average per hop latency is around 180 ms and the data rate varies from up to 84 kb/s for slave-to-master and 9 kb/s for master-to-slave communication and communication across piconets. Currently, the limiting factor regarding the throughput is the low performance of the Bluetooth stick micro-controller and higher data rates could be achieved by upgrading to a more powerful micro-controller. Using the developed solution, networks of Bluetooth devices beyond the size of a piconet can be constructed and the user can send and receive messages across piconet boundaries using a few simple functions. Thus, this solution is particularly suited to form communication networks for multi-robot systems of resource-limited mini-robots.

6 Acknowledgements

We thank Marc Struenkmann for his contributions to this project. This work was supported by the Sixth Framework Programme of the European Union as part of the GUARDIANS project (no. 045269, www.guardians-project.eu).

References

1. Bluetooth Special Interest Group (2003) Specification of the Bluetooth System. Nov. 2003
2. K-TEAM Corporation. Khepera II Robot. www.k-team.com
3. Salonidis T, Bhagwat P, Tassiulas L, LaMaire R (2001) Distributed Topology Construction of Bluetooth Personal Area Networks. In: Proc. of the 20th IEEE Infocom, Anchorage, Alaska, USA, 22.-26. Apr. 2001, vol. 3, pp. 1577-1586
4. Zaruba G V, Basagni S, Chlamtac I (2001) Bluetrees-scatternet formation to enable Bluetooth-based ad hocnetworks. In: Proc. of the IEEE Int. Conf. on Communications (ICC 2001), Helsinki, Finland, 11.-14. Jun. 2001, vol. 1, pp. 273-277
5. Wang Z F, Thomas R J, Haas Z (2002) Bluenet - a new scatternet formation scheme. In: Proceedings of the 35th Annual Hawaii Int. Conf. on System Sciences (HICSS 2002), Hawaii, USA, 7.-10. Jan. 2002
6. Lee Yong Liu M J, Saadawi T N (2003) A Bluetooth scatternet-route structure for multihop ad hoc networks. In: IEEE Journal on Selected Areas in Communications, 2003, vol. 21, no. 2, pp. 229-239
7. Pagani E, Rossi P G, Stefano T (2004) On-Demand Bluetooth Scatternet Formation Algorithm. In: Wireless On-Demand Network Systems, 2004, pp. 130-143
8. Kuntz A (2006) ScatterNetz-Routing (SNR) - Multihopkommunikation für medizinische Bluetooth ad hoc Netzwerke. Diplomarbeit, Institut für Biomedizinische Technik, Fakultät für Elektrotechnik, Universität Karlsruhe (TH), Mar. 2006
9. Grosseschallau M, Witkowski U, Rueckert U (2005) Low-cost Bluetooth Communication for the Autonomous Mobile Minirobot Khepera. In: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA'05), Barcelona, Spain, 18.-22. Apr. 2005, pp. 4205-4210
10. Chinapirom T, Kaulmann T, Witkowski U, Rckert U (2004) Visual Object Recognition by 2D-Color Camera and On-Board Information Processing for Minirobots. In: Proc. of the 2005 FIRA Robot World Congress, Busan, Korea, 27.-29. Oct. 2004
11. Perkins C E, Bhagwat P (1994) Highly Dynamic Destination- Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In: Proc. of the SIGCOMM Conf. on Communications Architectures, Protocols and Applications, Sep. 1994, pp. 234-244