IEEE Instrumentation and Measurement
Technology Conference
Como, Italy, 18-20 May 2004

# Energy-Efficient Data Collection for Bluetooth-Based Sensor Networks

Matthias Handy, Jan Blumenthal, Dirk Timmermann
Institute of Applied Microelectronics and Computer Science, University of Rostock
R.-Wagner-Str. 31, 18119 Rostock, Germany
Phone: +49-381-4983534, Fax: +49-381-4983601, E-mail: matthias.handy@etechnik.uni-rostock.de

***Abstract –*** *A wide range of sensor network applications deals with an issue often referred to as* data collection*: sensor nodes periodically transmit data to a base station. The base station analyzes incoming data for interesting events. In this paper, we introduce a novel Data Collection Protocol (DCP) for wireless sensor networks. DCP is tailored to Bluetooth-based sensor nodes and therefore enables sensor network applications based on inexpensive COTS-hardware. DCP is scalable, robust, and not limited to piconet or scatternet structures. As a potential application of DCP, we describe a wireless sensor network deployed as flood prevention system.*

***Keywords –*** *Bluetooth, energy conservation, microsensors, routing*

## I. INTRODUCTION

The collaboration of countless tiny sensor nodes in a network promises an enormous potential of novel applications. Advances in miniaturization and integration of electronic and mechanical components will enable sensor nodes with a size of a few cubic millimeters in the near future. At the same time, an ongoing price decline will allow the deployment of sensor networks covering thousands of nodes and, as a consequence, replace conventional wired sensors in many areas.

A possible application of wireless sensor networks is *flood prevention*. A thawing period in adjacent mountains and heavy rainfall at the same time can cause rising water levels of nearby rivers. Hence, natural or artificial dikes have to be reinforced to keep flood waters at bay. Typically, sandbags are piled along hundreds of meters or kilometers along a river or lake.[1] These dikes of sandbags have to be monitored permanently during the critical phase. Any site of leakage has to be detected as soon as possible in order to reinforce it with additional sandbags. Humans continuously walking along the dikes usually monitor the site. However, this kind of monitoring does not provide sufficient protection against breaking of dikes. A wireless sensor network could help to detect leaks earlier and thereby prevent floods. Each sandbag can be equipped with at least one sensor node. Each sensor node (or *node* for short) typically consists of a microcontroller, a communication component, an energy supply, and one or more sensors.

This paper's opening scenario belongs to a specific group of sensor network applications. Here, data transfers are mostly directed to the base station. Only control messages are directed from base station to nodes. Furthermore, communication among sensor nodes is limited to the exchange of control messages, except for forwarding sensor information towards the base station. Since most of the communication is directed from sensor nodes to base station, we can refer to this kind of data flow as *unidirectional*. The base station itself has to take care of receiving all relevant information from the network and to collect sensor information from nodes. This issue is often referred to as *data collection*.

In this paper, we introduce and discuss a novel data collection protocol (DCP) for Bluetooth-based sensor networks that can be adopted for several sensor network applications such as flood prevention. The applicability of Bluetooth in wireless sensor networks is subject of recent research activities, disclosing several properties of Bluetooth. These properties qualify this technology to be a sensor node's RF-interface. However, some characteristics of Bluetooth make its application in sensor networks difficult. In [1], Leopold et al. discuss various advantages and drawbacks of Bluetooth concerning its applicability in sensor networks. In their conclusion, the authors describe Bluetooth-based sensor networks as applicable for a niche of applications where data transfers appear infrequently but at high rates. The interference reducing FHSS-technique and a fully implemented MAC-layer are the qualifying properties of Bluetooth. Bluetooth drawbacks cover the issue that in order to transfer data between two Bluetooth nodes a connection has to be established in advance. In addition, the size of Bluetooth-piconets is limited (max. eight active members, one master and up to seven slaves), a fact that can complicate the assembly of large sensor networks. Overlapping piconets, so-called scatternets, can solve this problem. The following chapters will yield more advantages and drawbacks of Bluetooth concerning its applicability in wireless sensor networks.

Bluetooth devices can form a so-called piconet. Each piconet consists of one master and up to seven active slaves. The master coordinates piconet communication. Even two connected Bluetooth devices form a piconet, requiring a role assignment. One device becomes master, the other device acts as slave. Overlapping piconets form a so-called scatternet. Scatternets imply that some Bluetooth devices are member of more than one piconet and have been a theoretical construct for a

[1] In the summer of 2002, a "great wall of sandbags" was piled along hundreds of kilometers around China's Dongting lake and the Yangtze river, bringing relief to residents.

long time, not supported by Bluetooth device vendors. Recently, Bluetooth devices are available with limited scatternet support.

Bluetooth technology differentiates, among others, between *inquiry* substate and *page* substate. The inquiry substate is used by a unit that wants to discover new devices. A Bluetooth device switches to page substate in order to create a connection to another device. A master can only connect slaves with an enabled page scan mode.

The remainder of this paper is organized as follows. Our novel data collection protocol is described in Section II. This section covers both protocol phases and the discussion of several special cases. Section III presents simulation results. Related work is discussed in Section IV, followed by concluding remarks in Section V.

## II. DATA COLLECTION PROTOCOL

### A. General Remarks

A simple application of our protocol is a network consisting of one base station and several sensor nodes. Not all nodes have to be in communication range of the base station. Therefore, distant nodes have to use routing nodes to reach the base station. Role assignment, e.g. the selection of nodes as routing nodes, has to be repeated periodically to evenly distribute energy consumption among nodes.

If no events occur, it is sufficient for the base station to receive sensor information in a regular manner. However, if an event occurs, it is essential to transmit information about spatial and temporal characteristics of the event to the base station as fast as possible.

Our protocol can be divided into two phases. During set-up phase, the network is formed. Base station and sensor nodes explore their vicinity and search for neighboring nodes. At the end of the set-up phase each sensor node knows at least one *Packet Forward Address* (PFA). If a sensor node is not able to reach base station directly, it transmits its data to a PFA instead. The PFA then is responsible for data forwarding.

Furthermore, roles are assigned to nodes during set-up phase. When the set-up phase has finished, the network is formed and steady-state phase is started. During steady-state, sensor information is transmitted to the base station. After a certain time, the protocol enters set-up phase again and reorganizes the network. The reorganization interval depends on various network parameters, such as type and frequency of topology changes or the energy level of nodes.

Our protocol distinguishes sensor nodes according to their role in the network. Each node holds one of the following two roles. *Cluster members* gather sensor data and forward the data to a cluster head. *Cluster heads* collect sensor data from cluster members and forward it to the base station. The whole sensor networks then consists of a set of clusters; each cluster consists of a cluster head and at least one cluster member. Before

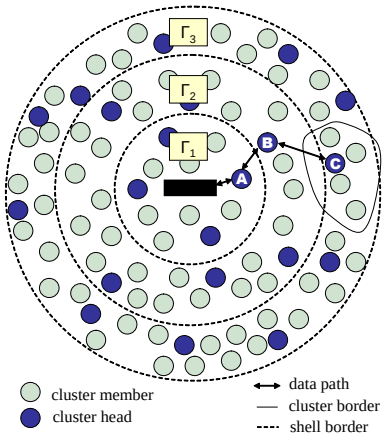


Fig. 1. Idealized view of shells in a sensor network. The base station is situated in the center of the network.

cluster members join a cluster head, they are also referred to as *simple nodes*. As described above, a Bluetooth node can either be a master or a slave. Each piconet consists of a master and up to seven slaves. A cluster head of our protocol is not necessarily a master and a cluster member not necessarily a slave. Our protocol is flexible by not stipulating master and slave roles.

Our protocol uses the theoretical model of *shells*. Each shell $\Gamma_k$ is a set of cluster heads in the plane with a specific characteristic. A node belongs to a certain shell depending on the number of hops $k$ to the base station. Let $N$ be the set of all sensor nodes in a network and $C$ the set of cluster heads in the same network with $C \in N$. Let $dist(c)$ be the number of hops from cluster head $c$ to the base station. Given a cluster head $c$, then $c \in \Gamma_k$ only if $dist(c) = k$. An idealized view of shells in a sensor network is given in Fig. 2, where the base station is situated in the center of the network. Notice that, shells are only circles in theory.

### B. Set-up Phase

The set-up phase of our protocol, i.e. the formation of the network, is divided into seven steps. Depending on the number of shells in the network, step 5 to step 7 have to be repeated. At the beginning of the set-up phase, all nodes have their inquiry scan mode and page scan mode enabled.

Initial value: $k = 1$ (number of a shell)

1. Nodes decide themselves whether to become a cluster head or not by means of a cluster head selection algorithm. (Cluster head selection strategies are discussed in subsection D) Let $p$ be the assumed cluster head probability. Then, $p \cdot N$ nodes become a cluster head and $(1-p) \cdot N$ nodes become a cluster member.
2. After step (1), the base station initiates an inquiry to discover all nodes in the vicinity.
3. After step (2), the base station successively creates connections to all sensor nodes having responded to the inquiry, i.e. cluster heads of shell 1 and cluster members in

the range of the base station. Each of the connected nodes stores the base station's device address as packet forward address, whereas the base station maintains a table of discovered node addresses. Thereafter, all connections to the base station are closed.

4. Each cluster head of shell $k$ then changes its visibility mode, so that it can not be discovered by inquiry scans (inquiry scan disable). After the completion of set-up phase, the inquiry scan mode is enabled again.
5. Subsequently, each cluster head of shell $k$ initiates an inquiry. The changed visibility of cluster heads of shell $k$ prevents them from mutual discovery. The inquiry can yield the following responses:
   (a) Cluster heads not discovered by cluster heads of shell $k-1$ (cluster head of shell 0 is the base station) and thus $k+1$ hops away from the base station (cluster heads of shell $k+1$)
   (b) Simple nodes in range (cluster members)

   If no cluster heads were discovered, proceed to step (7).
6. Each cluster head of shell $k$ then successively creates connections to all cluster heads of shell $k+1$ that responded to the inquiry. Each cluster head of shell $k+1$ stores the device address of at least one cluster head of shell $k$ as packet forward address and subsequently disables its inquiry scan mode. Each cluster head of shell $k$ maintains a table of dependent cluster heads of shell $k+1$.
7. Each cluster head of shell $k$ forms a cluster with all discovered simple nodes as cluster members. Cluster members store the device address of their cluster head as packet forward address. The cluster head maintains a table of all cluster members. Notice that connections between cluster head and cluster members are only established on demand.

If step (6) was skipped the set-up phase is finished. Otherwise, increment $k$ by 1 and go back to step (5).

### C. Steady-State Phase

When the set-up phase is finished and each node has received a packet forward address, the steady-state phase begins. In this phase, each cluster member periodically transmits its sensor data to its cluster head. The cluster head then forwards the collected data to the base station and uses other cluster heads as routing nodes if needed. The following steps are performed during steady-state phase.

1. Each cluster member periodically transmits its sensor data to its cluster head.
2. Cluster heads preprocess sensor data by means of a data fusion or compression algorithm. Hence, cluster heads have already eliminated redundant data and have furthermore reduced the amount of data being forwarded to the base station.
3. Each cluster head transmits its aggregated sensor data to its PFA.
4. If a sensor node receives aggregated sensor data from a cluster head of a higher shell, it forwards the data to its own PFA.

### D. Selection of Cluster Heads

The selection of cluster heads is not a trivial problem since cluster heads have to comply with several requirements. As an example, cluster heads have to be equipped with sufficient energy reserve in order to perform energy depleting routing and data fusion mechanisms. Moreover, cluster heads should be situated in the center of a cluster in order to minimize intra-cluster energy consumption.

From an abstract point of view, each cluster is a group of sensor nodes with a certain assignment of roles. For such a group of nodes, Liu et al. introduce in [2] the term *collaboration group* (or *group* for short). Referring to Liu et al., each group is a set of entities, e.g. sensor nodes, that encapsulates two properties: *scope* and *structure*. The scope of a group defines its members, e.g. all sensor nodes within a certain range. A group's structure defines the roles each member plays in the group. Formally, a group is a 4-tuple

$$G = (A, L, p, R) \tag{1}$$

where $A$ is the set of entities, $L$ is the set of roles, $p : A \rightarrow L$ is a function that assigns each entity a role, $R \subseteq L \times L$ are the connectivity relations between roles. In the case of our protocol, two roles can be distinguished:

$$L = \{cluster\ head,\ cluster\ member\} \tag{2}$$

Additionally, one connectivity relation exists:

$$R = \{(cluster\ member,\ cluster\ head)\} \tag{3}$$

This relation describes, that each cluster member transmits its sensor data to a cluster head. The set of sensor nodes belonging to a cluster is the set of entities forming group $A$. Consequently, the whole sensor network consists of a set of groups (clusters). For a complete determination of the 4-tuple, each entity (sensor node) has to be assigned a role, i.e. $p$ has to be defined. In the case of our Data Collection Protocol, cluster heads are stochastically determined similar to the algorithms described in [3] and [4]. Each sensor node determines a random number between 0 and 1 and compares it to a predefined threshold. If the random number is less than the threshold the node becomes a cluster head, otherwise not.

When the role assignment is finished, simple nodes have to join a cluster head and become cluster members in order to complete group forming. Assume each sensor node to be able to measure the strength of a received signal and each cluster head sends with the same output power. Moreover, assume the strength of a received signal to be inversely proportional to the distance between two nodes. Consequently, each simple node

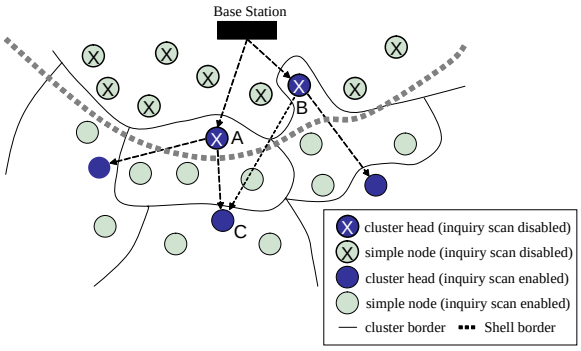


Fig. 2. Cutting from the first and second shell after completion of set-up phase. The base station forms a cluster with simple nodes in transmission range. When cluster heads of the first shell perform an inquiry, they neither discover other cluster heads of shell 1, nor simple nodes already connected by the base station due to their disabled inquiry scan mode (marked by an X). Cluster head C stores the addresses of cluster heads A and B as possible PFA. C chooses A as primary PFA since it is closer than B. If node A fails C uses B as PFA (multipath routing).

has to join the cluster head with the maximum signal strength. Our Data Collection Protocol benefits from this idea.

### *E. Generation of the 1st Shell*

The formation of the first shell is a special case for our protocol, since the base station can act as cluster head in this shell. (Of course, the base station must not be the only cluster head of shell 1.) As described in Subsection B, at the beginning of the set-up phase, the base station attempts to discover sensor nodes in the vicinity. When the base station's inquiry is finished, it connects to all discovered cluster heads (step 3). For a synchronous protocol cycle, cluster heads of shell 1 should defer their own inquiry until step 3 and 4 are completed. Otherwise, a cluster head that received its PFA from the base station would immediately disable its inquiry scan mode and start an inquiry although step 3 and 4 are still running. We can avoid this behavior by providing each cluster head of shell 1 with a delay time. After a cluster head received its PFA from the base station, it waits for this delay time before it starts the inquiry. Assume the base station to connect to cluster heads successively. The delay time $T_c$ of cluster head $c$ then is calculated by

$$T_c = T_g \cdot r \tag{4}$$

where $T_g$ is the basic time, needed for connection establishment, data transfer (PFA) and $r$ is number of cluster heads of shell 1 not yet connected by the base station. According to experiments the basic time $T_g$ is not larger than 5 seconds for commercial Bluetooth USB modules.

### *F. Generation of Subsequent Shells*

When the first shell is formed, its cluster heads start an inquiry to discover nodes in the vicinity. Either simple nodes or cluster heads can respond to this inquiry. The disabled inquiry scan mode of nodes that already received a PFA prevents them from being discovered again. Fig. 2 illustrates this process. Consider the case where a sensor node receives multiple PFAs since more than one cluster head connects with this node. For our protocol, such a behavior is advantageous and, therefore, explicitly welcomed. It increases the robustness and efficiency of our protocol. On the one hand, a multipath-routing scheme can be applied. If a cluster head fails depending nodes choose another PFA. On the other hand, sensor nodes with multiple PFAs can choose the one with the shortest distance in order to reduce energy consumption.

If a connection is established between two Bluetooth devices, each of them can measure a RSSI-value of the connection (RSSI=Received Signal Strength Indicator). RSSI measures the signal strength of a received signal [5]. The shorter the distance between two Bluetooth devices, the larger the RSSI-value. Multiple PFAs allow a sensor node to choose the one with the largest RSSI value. Hence, the sensor node can adapt its output-power to the shorter distance and thus save energy. Notice that many commercially available Bluetooth devices adapt output power automatically and not controllable by an application. However, a RSSI value is useless if we don't know the corresponding output power. Consequently, the output power of Bluetooth-based sensor nodes must be controllable by the sensor node application.

Our protocol does not maintain connections during steady-state phase. When a data transfer is finished the nodes disconnect immediately. The advantages of this behavior are:

- Clusters are not limited to piconet size. A cluster can consist of more than seven cluster members.
- Assuming a low data transfer rate, energy can be saved if nodes disconnect immediately after data transfers.
- If all connections in our sensor network were maintained during steady-state phase a large scatternet consisting of many piconets would be formed. Although Bluetooth uses frequency hopping, too many piconets would interfere each other [6].

However, delay times for our protocol are higher than for a connected piconet. Alternatively, cluster heads or network regions with temporary high data rates could autonomously decide to maintain a piconet or local scatternet.

## III. SIMULATION RESULTS

Based on detailed simulations, a first impression of our protocol's performance can be gained. For the simulations, we use a self-developed simulation tool that models protocol behavior depending on various system parameters. In Fig. 3 to Fig. 6, several parameters are determined for various topologies. In the legends of the figures six different cases are specified. The parameters in squared brackets are node density and node's communication range in meters. Node density describes the number of nodes/m. For node density 0.01, network dimensions are 100m·100m. 100 Nodes are randomly distributed

over this area. The base station is located in the center of the network at position (50m, 50m). For node density 0.04, 100 sensor nodes are distributed over an area of 50m·50m. The base station is located at position (25m, 25m). For all presented experiments, cluster head probability is varied between 0 and 1 in steps of 0.1. Each cluster head probability is examined in 100 simulation runs whereupon average values and standard deviation are calculated.

Fig. 3 depicts the maximum number of hops to base station, i.e. the longest path of the network. Notice that the longest path is shorter for large communication ranges, which is not surprising. Moreover, the maximum number of hops is not strictly monotonic increasing if cluster head probability is incremented. Except for the combination [0.01; 10], a maximum can be found for every curve. As an example, the input parameter combination [0.04; 15] shows a maximum at cluster head probability 0.2. A further increment of cluster head probability does not result in longer paths.

Fig. 4 illustrates the average number of hops to base station, i.e. the length of the average path. Again this figure reveals for each combination of input parameters a maximum. Thus, for each combination of input parameters, an upper bound for cluster head probability can be found. If we increment cluster head probability further, the average and longest paths will not increase.

Fig. 5 depicts the percentage of unconnected nodes after set-up phase. Notice that input parameter combinations [0.01; 10] and [0.04; 10] are not usable, since even for high cluster head probabilities, the percentage of unconnected nodes is extremely high. The input parameter combinations [0.01; 20] and [0.04; 10] are two-edged cases. A satisfying percentage of unconnected nodes is only accomplished with a large cluster head probability. However, for cluster head probabilities larger than 0.3, clusters are too small to operate efficiently. As an example, a cluster head probability of 0.5 results in clusters with one cluster head and one cluster member at 100% connectivity.

Fig. 6 illustrates the number of nodes that directly depend on a cluster head. This includes both cluster members and cluster heads of higher shells that forward data to the examined cluster head. To summarize, for an efficient operation of our Data Collection Protocol, input parameters have to be chosen wisely. Cluster head probability should be used as instrument for fine tuning only since a variation of this parameter in regions larger than 0.3 leads to inefficient mini-clusters.

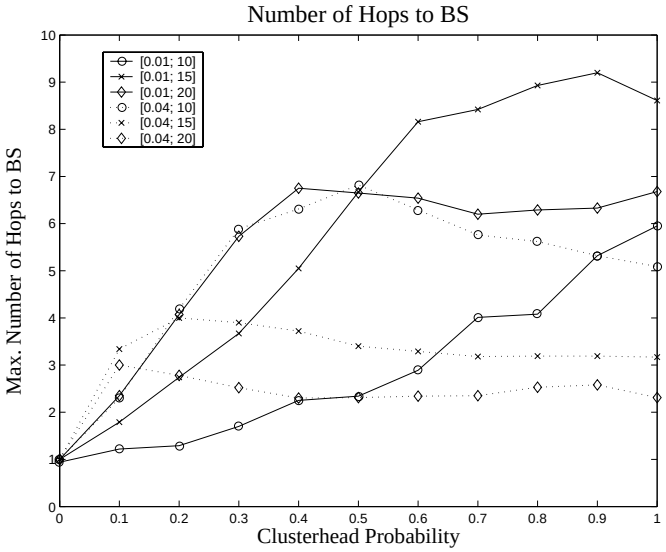


Fig. 3. Maximum number of hops to base station. In squared brackets: [node density; node's communication range].

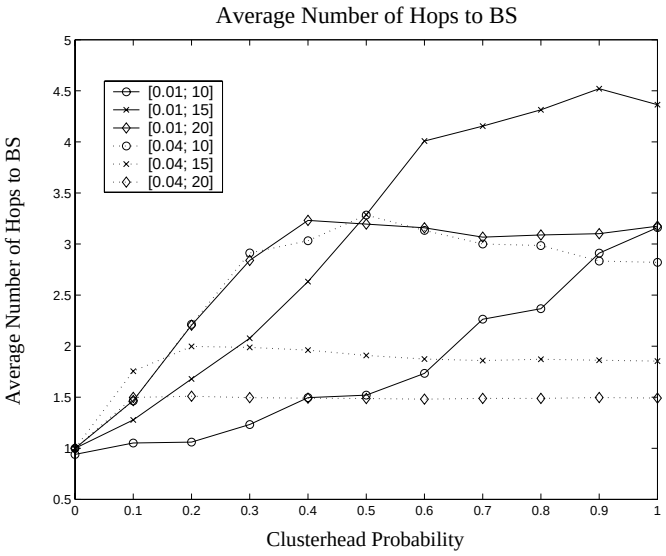


Fig. 4. Average number of hops to base station. In squared brackets [node density; node's communication range].

## IV. RELATED WORK

A wireless sensor network is a special case of an ad hoc network. Every protocol for sensor networks has therefore to be compared to existing protocols from the ad hoc domain. In the case of the network layer, reactive and proactive protocols can be distinguished [7]. Our Data Collection Protocol works proactively since routes are discovered during set-up phase. However, ad hoc routing protocols are not suitable for our intended applications of sensor networks because they are optimized for multi-directional data transfers. DCP is mainly designed for unidirectional data transfers. In our intended sensor network applications, data transfers are mostly directed to the base station. For ad hoc routing protocols, communication among nodes is more important. Therefore, ad hoc routing protocols do not require a base station. Additionally, ad hoc routing protocols are optimized for networks with highly mobile nodes; routes have to be updated more frequently.

There is one characteristic of ad hoc protocols that disqualifies them from an adoption for Bluetooth-based sensor networks. Ad hoc routing protocols mostly require a broadcast

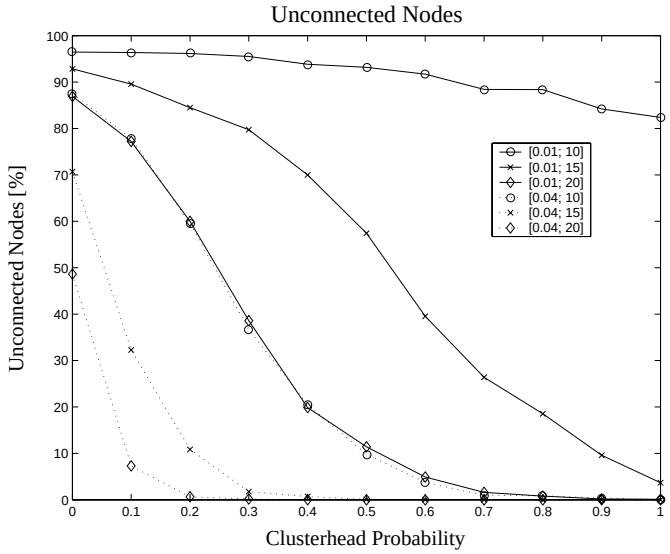


Fig. 5. Percentage of unconnected nodes in a network. In squared brackets [node density; node's communication range].

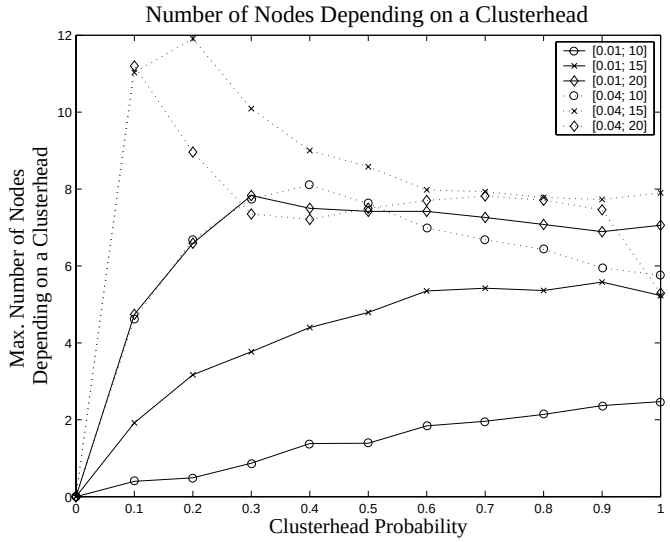


Fig. 6. Number of nodes depending on a cluster head. In squared brackets [node density; node's communication range].

mechanism. However, Bluetooth's broadcast support is limited. A general broadcast to all nodes in communication range is not possible. Only Page and Inquiry are similar to a broadcast, though transmitted ID-packets may not contain payload data. The master of a Bluetooth piconet can initiate a piconet broadcast. This requires a connection between master and all slaves.

Although our protocol does not depend on scatternets, formation protocols for scatternets might be an alternative [8]. Algorithms for scatternet formation are optimized for a high connectivity and maintain connections during operation. This results in shorter delay times but also in higher energy consumption. In contrast to scatternet formation protocols, our DCP allows clusters containing more than seven members. Furthermore, a disconnection after data transfer is more energy efficient for our intended applications.

## V. CONCLUSION

In conclusion, we introduced DCP, a scalable and robust protocol for energy-efficient data collection in large Bluetooth-based sensor networks. DCP's main advantage is independence from Bluetooth piconet and scatternet limitations. This yields a more flexible and scalable network infrastructure. A periodically repeated role assignment within the network leads to a uniform distribution of energy consumption. Additionally, energy consumption is reduced by a technique that only maintains connections between sensor nodes if data transfers occur. With these characteristics, DCP enables a wide range of sensor network applications with inexpensive hardware.

## ACKNOWLEDGMENT

This work is supported in part by the Gottlieb Daimler- and Karl Benz-Foundation, Ladenburg. We would like to thank Rico Möckel who developed the simulation environment and carried out the simulations.

## REFERENCES

[1] M. Leopold, M. Dydensborg, and P. Bonnet, "Bluetooth and sensor networks: A reality check," in *Proceedings of SenSys 2003*, November 2003.
[2] J. Liu, M. Chu, J. Liu, J. Reich, and F. Zhao, "State-centric programming for sensor-actuator network systems," *IEEE Pervasive Computing*, vol. 2, no. 4, pp. 50–62, 2003.
[3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. of the 33rd International Conference on System Sciences (HICSS '00)*, January 2000.
[4] M. Handy, M. Haase, and D. Timmermann, "Low-energy adaptive clustering hierarchy with deterministic cluster-head selection," in *Proc. of IEEE Int. Conference on Mobile and Wireless Communications Networks*, September 2002.
[5] *Specification of the Bluetooth System, Version 1.1*, February 2001.
[6] J. Bray, "How many bluetooth piconets fit in a room," Prentice Hall PTR, URL: http://www.informit.com, May 2001.
[7] C. Perkins, *Ad Hoc Networking*. Addison-Wesley, December 2000.
[8] S. Basagni, R. Bruno, and C. Petrioli, "A performance comparison of scatternet formation protocols for networks of bluetooth devices," in *Proc. of the 1st IEEE International Conference on Pervasive Computing and Communications*, March 2003.