Background Data Acquisition and Carrying: The BlueDACS Project

Thomas Wieland, Martin Fenne, and Benjamin Stöcker

Department of Electrical Engineering and Computer Science,
University of Applied Sciences Coburg,
96450 Coburg, Germany
{wielandt, fenne, stoeckeb}@fh-coburg.de

Abstract. As an alternative to ad hoc wireless sensor networks, we propose to utilize the mobile devices that are carried by people who walk along the site on which the sensors are deployed. Each sensor establishes a wireless connection to a mobile station and transmits its current measurements. On encounters two mobile stations can exchange their data for increasing redundancy and likeliness of delivery. When a mobile station approaches the sensor network server, it unloads all data which is saved there in a data base, processed, and published in local or global networks. In this paper, we describe our realization of this approach, the communication mechanisms that have been developed for it, as well as its potential usage. This approach is mainly characterized by the smooth integration of an additional data service in existing work processes utilizing only commercial off-the-shelf components.

1 Introduction

Wireless sensors are usually deployed in a significant number so that they set up an ad hoc network [1, 2]. The measurements of the sensors are transmitted over this network from one sensor to another until an interface station is reached. These stations are typically connected also to a LAN or have other means of processing the received information. Although the promise of sensor networks lies in the simplicity and multitude of sensors, the electronic components available today turn out to be rather expensive – especially when purchased in small quantities. In addition, there are numerous scenarios where a few number of actual sensors is sufficient; more would only be required for data transmission.

So if the sensors are singularly deployed in a distance that they cannot reach each other, different ways of data transport have to be found. This might cause a large delay in the overall communication route. But on the other hand the capacity of the network as well as its robustness can increase [3]. One way can be utilizing electronic devices that are in some way brought close to the sensors.

The set-up of a wireless sensor network infrastructure requires more than merely the sensor nodes. In a small network with limited spatial expansion all measurements are collected and processed directly by a server. When the number of nodes grows or when they are deployed in disjoint areas, regional relay stations must be used. These mobile stations act as concentrators (collectors) for an area of sensors and forward all

W. Grass et al. (Eds.): ARCS 2006, LNCS 3894, pp. 56 – 68, 2006.

data gathered in this area to the sensor network server (SNS). This architecture has proven to be useful for instance in environmental research [4].

The basic idea in this paper is to install isolated sensors that have wireless network connectivity, but are unable to reach any other sensor. The collectors are mobile devices carried by people who come into sufficient proximity. When two of these mobile devices come in the transmission range of each other, they can mutually exchange the data they have currently stored. As soon as a device is close enough to an SNS, it delivers its data and removes it from its RAM.

In the following sections, we are going to describe the setup of our prototype. The system that has been developed is called '*BlueDACS*' as an acronym for 'Bluetooth-based Data Acquisition and Carrying System'. It uses its own protocol on the application layer for a light-weight binary data transmission. Various solutions to several technical problems that had to be found are also explained. We describe our experiences with the overall system and describe some application scenarios. Finally we give a perspective for future work.

2 System Components

The entire system comprises three subsystems: the sensors nodes, the mobile phones as collectors, and the server domain. The phones are the data carrying units, communication with the other subsystems and with each other. The server domain consists of the server itself and optional relay stations, allowing a more frequent data delivery with shorter paths to go. The overall communication relationships are shown in Figure 1.

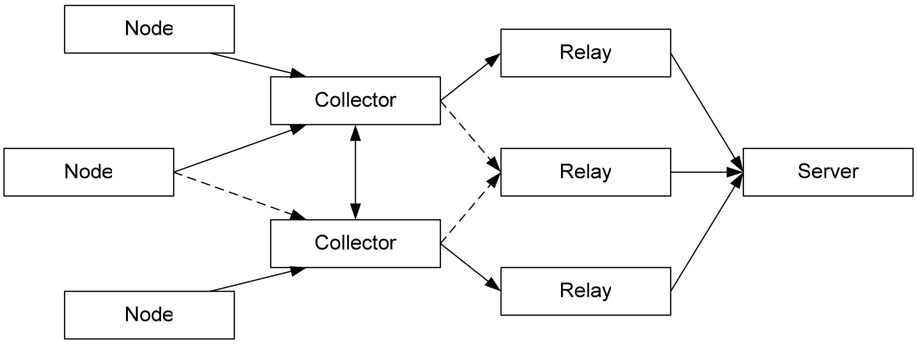

Fig. 1. Communication paths of the BlueDACS system. The nodes forward their data to the carrier which may exchange it among one another and eventually send to a receiver station.

2.1 Radio Communication

In sensor networks where several nodes are to coexist, a spread-spectrum approach on the radio communication layer can reduce interferences. A widely used incarnation of this technology is the Bluetooth protocol (IEEE 802.15). Originally designed as a cable replacement for home and office peripheral devices, it can be found today in a vast number of environments at steadily decreasing costs. This is one reason why Bluetooth has already been used for sensor networks [5-7].

However there are a few general problems with the usage of Bluetooth for sensor networks:

- Of the Bluetooth protocol stack, a sensor should at least use the network layer, maybe the LLC layer too. Depending on the type of nodes and their usage, this might impose a considerable overhead.
- The entire Bluetooth protocol is considered to be rather complex and thus power-intensive.
- The proposed scatternet functionality that would be valuable for sensor networks is currently not included in commercial products.
- Bluetooth-based nodes have to establish a connection before exchanging data. In this connection, one node acts as master and the other as slave. So a rigid role concept is necessary.

There is, however, an outstanding advantage: Bluetooth is included in numerous consumer devices on the market. So for realizing a showcase that is supposed to work with such components, Bluetooth is today the only choice for the radio communication.

Some of the aspects mentioned above may, however, even turn out as benefits: The standardized network layer, e.g., allows the developer to rely on a consistent communication channel with media access control and error detection.

2.2 Sensor Nodes

As sensor components we use ATmega128 RISC devices from Atmel [8]. They provide 128 kB self-programming flash program memory, 4 kB SRAM, 4 kB EEPROM, and 8 Channel 10-bit A/D-converter, as well as two RS232 ports. The device can be programmed in the C programming language and consumes acceptably little energy. Sensors can be attached to it directly at the A/D-converters or digitally via a separate sensor aggregation board.

For communicating with the collector units an external Bluetooth adapter, the Stollmann BlueRS, connected to the serial interface, is used [9]. For sending the data to the mobile station, a simple serial communication using the RFCOMM layer of Bluetooth is sufficient. In our configuration, the sensor node acts as a Bluetooth slave and does therefore not emit inquiry scans for other communication partners. In fact it sends inquiry packets once in a while to allow the mobile node to detect it. The sensor node is designed to deliver its data to the mobile stations exclusively. This procedure can still be optimized and leaves room for further work: Due to security reasons the sensors should not send any inquiry packets at all, at least in some scenarios. In this case, the mobile stations have exact location information about the sensors and may detect them with AGPS. This would also allow the mobile station to reduce the frequency of their inquiry scans.

The nodes use their local buffers for their own measurements. If there is an overflow, the oldest data set is skipped, acting like a normal ring buffer.

For the scenarios that are anticipated for this network configuration, a permanent power supply (long-lasting battery, dynamo, solar cell, or power outlet) can be assumed.

2.3 Mobile Phones as Data Collectors

The collectors are mobile phones that are carried by people walking across the site. From the view of the sensor network, their walking routes may be random. The deployment of a node should be planned in a way that there is a real chance that at least once a day at least one person comes in the reach of this node. It is obvious that this requirement is essential, making data gathering in less frequently visited areas unfeasible. It should, however, be realistic for many interesting applications.

As soon as such a mobile station comes in reach of a sensor node, a Bluetooth connection can be established. A handshake mechanism, a simple form of authorization, guarantees that only qualified devices get access to the data and accidentally approaching Bluetooth devices of other persons are not affected.

The mobile phones act as Bluetooth masters when communicating with the sensor nodes and the server or its relay stations. One may also take the role of a slave when talking to another mobile node. In order to fulfill their master role, the devices send out regularly an inquiry scan for detecting other devices in reach. The frequency of the scans depends on the actual usage scenario. Commonly, one scan every two or three minutes is a good trade-off between power consumption for the scan and reachability of other nodes. With this feature enabled, the power reserve of a mobile phone lasts between 4 and 8 hours.

We used Nokia 6600 and Siemens S65 phones as off-the-shelf components. Both have Bluetooth capabilities and can be programmed in Java utilizing the Mobile Information Device Profile (MIDP), version 2.0. The drawback is that currently on most devices such Java applications cannot run in background and have to be activated manually. Thus after a phone call, the bearer has to start the BlueDACS program again. Some phones with the Symbian operating system (like the Nokia 6600) are able to put the virtual machine in background during other activities; but the VM is suspended during this time. At least the application is resumed automatically and need not be restarted. This type of problem will more and more disappear as the operating systems of the mobile devices get more powerful. Alternatively, PDAs or pocket computers may be used; they have typically a larger power reservoir and allow background applications. We also made some tests with HP IPAQ PocketPCs.

The usage of standard, wide-spread consumer devices represents the special advantage in this approach. Relaxing the need for sensors to communicate with one another makes the entire system more flexible and less expensive. Although sensor nodes are thought to be deployed as rather tiny electronic components in large numbers, the current costs for such nodes – either off-the-shelf or self-configured – are still considerable, especially when purchased in small quantities. On the other hand, the solution proposed here gives the existing devices an additional task and leverages previous investments with an added value.

2.4 Relay Stations and Sensor Network Servers

As there is only one sensor network server for all sensor nodes of a specific group or even for all the sensors in the system, the chance that a mobile station comes in reach of Bluetooth transmission range of this server is rather low. (It may be installed in a place every worker must pass, like the factory gate, but this leads to other drawbacks and is not practicable for any application.) So it is helpful to install some additional relay servers that act as local points of delivery for the data. The relay servers are connected to the sensor network server (SNS) by LAN or over the Internet. They are therefore similar to the notion of "infostations" introduced in [10].

The SNS is realized as a usual PC with Bluetooth connectivity. It stores the data in an XML data base and allows access through various interfaces depending on the actual application, the data privacy, and the technical infrastructure, e.g. web-based reports, XML web services, and specialized applications making use of the data. Such applications may generate daily profiles from the measurements, graphical summaries, analyses of threshold exceedances with a hierarchical alarming system, etc. There are, of course, even much more sophisticated applications possible (see below).

From the communication perspective a relay station shows the same behavior as a sensor node in the BlueDACS system. It acts as a Bluetooth slave and sends out inquiry signals periodically to allow mobile nodes to find it.

3 The Communication Process

The description of the overall communication process can be divided into three different aspects: The format for storing the data, the procedures performed for transmitting, and the protocol used in this transmission.

3.1 Data Format

A uniform binary format that is used in all subsystems simplifies the handling of the acquired data and avoids reformatting. A BlueDACS data packet consists of the following elements:

- The number of data sets in the packet (*n*, 2 bytes)
- The length of each data set (*m*, 1 byte)
- A device identification number specifying the node that has acquired the data (*DID*, 2 bytes)
- A data acquisition number enumerating all packets generated by this node (*DAN*, 1 byte)
- The actual data sets, usually containing one or more measurement values and a time stamp, with length *m* and quantity *n* as indicated by the numbers above
- An error check array of *m* values, the result of an XOR operation among the data sets

The width of the DAN of 1 byte is clearly a restriction. It means that the node can generate a maximum of 256 packets until it has to overwrite its memory. It is, however, no real handicap for the practical usage. As each packet may consist of 65536 data sets at the most, a single node can still create more data than a carrier might be able to retrieve. And after all, the memory of the node is fairly limited such that these constraints are fully acceptable.

As we have designed here communication packets for the application layer, we do not have to include a very strong error-checking, but should be able to rely on what is already done on the network layer. So a simple XOR-generated code will certainly be sufficient.

3.2 Transmission Procedures

As soon as a mobile station comes in reach of a sensor node, a Bluetooth connection can be established. The sensor node transmits its most recent measurements via the existing connection to the mobile station. The data exchange protocol uses acknowledgements to ensure the success of the transmission. The mobile station checks the XOR field of the message and sends an ACK if everything is all right. Only after receiving this ACK, the node may delete the transmitted data sets from its buffer.

Like the sensor nodes, also the mobile stations keep all data that has not been transmitted as long as there is sufficient memory. The data is stored in the flash memory of the devices; thus it is still available when the phone has been switched off or has run out of energy.

If two mobile stations come close to each other (e.g., when their bearers walk by each other), they can forward mutually all measurements they currently store. They establish a Bluetooth connection, verify their authorization, and send the respective data sets. The communication is again safeguarded by XORs and ACKs to ensure that only uncorrupted data is stored. Data sets that are already stored on the respective device are discarded.

The data exchange of two mobile stations does not mean the forwarding of the complete data packets. Thus the data sets are not removed from each other’s buffer after the transmission. This procedure just increases the redundancy of the entire system and the chance that any mobile station delivers the data to the SNS in the end.

As soon as a collector comes close to a relay station or the SNS, both sides establish a Bluetooth connection. The relay server just forwards all datagrams. If the transmission, again using XOR and ACKs, is successful, the mobile station removes the data sets from its buffer. The protocol is similar to the one described above for the communication between sensor node and mobile station. The SNS saves the measurements in a data base, provided that there has been no record with the same DID/DAN combination in the last 24 hours.

As pointed out in [11] and [12], there are numerous variants of this procedure. E.g., the transmission between two mobile stations may only be performed with certain likelihood, the stations may record which packets they have passed on, and they may even keep information about the IDs of those packets that – as far as the station knows – have already been successfully delivered to the SNS. For our prototype, the procedures described above seemed sufficient. Nevertheless it is clear that it strongly depends on the actual application which kind of strategy is used here.

3.3 Communication Protocol

A BlueDACS communication session is done in rounds. At the beginning of each round both partners negotiate who will be sender and who will be receiver. For this purpose, both partners send one byte (in the structure shown below) and the partner with the higher value – interpreted as an integer –becomes sender. Now a data packet is transmitted. Afterwards a new negotiation is performed, which includes an acknowledge flag for confirming the correct reception of the last packet. If the comparison of the bytes indicates that no further communication is necessary, the connection is closed. In the case that the ACK fails, the packet in question is re-sent.

The structure of the negotiation byte is as follows:

WS	RA	RA	RA	RA	RA	WR	ACK

The individual bits have the following meaning:

- WS: "wants to send". This bit is set if the sender has data to transmit.
- RA: random. A random number is used to clarify the precedence of transmissions. The partner with the higher number is to send first. A sensor node sets all bits except the last one to 0, making sure that he has the primary right to send. A relay station sets all bits to 0, as it does not have anything to send at all. The mobile stations always set the first bit to 0 and the last bit to 1, while selecting a random combination for the other ones.
- WR: "wants to receive". This bit is set if the sender is ready to receive data. Sensor nodes, e.g., set it to 0.
- ACK: acknowledge. This bit is set if the last transmission was successful (not used in the handshake procedure).

This arrangement of the flags allows interpreting the byte as a number. The communication partner with the higher number gets the right to send first. A communication is only established if one partner is willing to send (WS = 1) and the other is willing to receive (WR = 1). If this is not the case, no data exchange is necessary and the connection is closed.

If this handshaking procedure should result in equal random bits, the negotiation is repeated. This case can only happen between two mobile phones.

The big advantage of this protocol is its simplicity. It is easy to implement on any platform, meets all communication requirements, and uses the resources very efficiently. The header is rather small, i.e. reduced to the absolutely necessary issues. This is especially important for applications of the proposed type, which imply low data rates. A small header assures that the data size /packet size ratio and thus the effective bandwidth is high.

Moreover the sensor nodes are small embedded systems without much processing power. For this platform a protocol should be preferred that allows a succinct implementation. This fact also leads to a reduced error probability – which is a considerable benefit for embedded devices that might be produced and deployed in numbers. Updates of the firmware could become difficult and expensive.

4 Results and Experiences

The system has been realized as described above and tested on a prototype level. A number of experiences during development and testing could be gained.

The on-board memory of the sensor boards and the mobile stations were fairly sufficient. The code of the applications has a footprint of less than 5 kB which leaves enough memory for data storage, even on the Atmel boards.

A more serious limitation was that the Atmel boards we used do not have onboard clocks. So the tagging of the packets can only be performed with a counter. In extreme cases this might cause overflow confusions. To allow an accurate and unique identification of the packets, an additional time-stamp would be desirable. In the present hardware configuration, this could, however, only be achieved by external clock components.

The energy consumption of the Bluetooth adapter at the nodes may reach, according to the manufacturer, up to 45 mA with a BT connection, but no traffic, and 70 mA with data traffic at 115 kBit/s. As we assume a reliable power supply for the nodes, these values (which are considerably lower than the ones reported in [7] for another BT-based node) are acceptable for the anticipated usage scenarios. If the application requires a low power consumption, the BT module may be switched off after one successful delivery for some time until enough measurements are gathered again that need to be forwarded.

The programming interfaces both of the BT adapter and the mobile phones do not allow the manipulation of the radio communication on a low level. So it was not possible to test the influence of time slot aggregations and variations of encoding redundancy. For small payload packets, however, this influence is normally negligible.

The limitations of the Bluetooth protocol with respect to multiple communication partners turned out to be rather uncritical for the transmission between a node and a mobile station is, since there are hardly any other devices around that could interfere the connection. The same holds partially for the communication between mobile stations, provided they do not meet in very crowded areas. The contact between the mobile station and the SNS (or relay stations) may be more difficult. But as our experience from the test bed shows, the establishment of a connection succeeds in most cases, even if sometimes a little delay has to be accepted.

An unexpected result of the laboratory showcase experiments was that the inquiry scan, which is executed regularly by the mobile stations, takes a relatively long time, up to 30 seconds. The performance can be improved if the addresses of known devices are stored on each mobile station and only connections to these are tried periodically. This procedure assumes that the information about the devices in the system is configured at deployment time. This step makes the deployment more complex, but improves the overall performance significantly. In a repetition of the experiment with stored addresses the entire communication sequence took only a few seconds.

The mobile stations have the most complex task in the BlueDACS systems since they have to act as servers waiting for incoming connections as well as clients trying to connect to other stations. The communication is done via sockets on exactly this basis. So making data exchange during random encounters possible requires a sophisticated multithreading, switching between an open server socket and opening a client socket. The Java implementation on the mobile phones used in the prototype fulfilled this task without major problems.

The system tends to be rather reliable. The crucial aspect is the probability of the encounter of the communication partners. Theoretical calculations show that even if the probability of a mobile station meeting a node is less than 1% for a system with five carriers, all data of the node can eventually be delivered and no overflow occurs. In experiments in the university all sensor nodes could forward their measurements completely, thus fulfilling the task of the entire system satisfactorily. The behavior of all subsystems were as required; it was even possible that the sensor nodes generated data packets event-driven, i.e. only when the measured value changed. Long-term experiences can reasonably only be gained in the context of a real application. So the set-up of a real-world test bed and the evaluations of the respective results are still part of ongoing and future work (see below).

All in all, the prototype showed that the approach proposed above can actually be realized with off-the-shelf standard components, especially with standard consumer devices. The rather intensive energy consumption on the mobile phones and the long delays during BT inquiry scans and sometimes also during connection establishment give room for further research in optimization strategies. An extension in progress is, e.g., to make the server maintain a list of all communication partners in a BlueDACS environment and their BT data. This list is distributed as configuration information by the relay stations to the mobile phones which forward it to one another if necessary. So all devices know about the valid communication partners and can confine themselves to making BT connections with these only.

5 Usage Scenarios

The BlueDACS system is not thought for the classical sensor network scenarios, as these generally rely on a real-time wide-range data distribution [1, 2]. Yet there are various fields of usage where the longer time that the data might take to travel from the sensor to the server is acceptable. Some of these fields are discussed in the following.

5.1 Home Care of Elderly People

The Western industrial societies are more and more aging. An increasing number of persons need individual care, but they also want to stay in their familiar environment as long as possible. The care of these persons is often done by specially trained nurses and other assistants who visit their clients regularly, giving them the care they actually need.

The caregivers usually carry a mobile phone with them. More and more of this group also have PocketPCs for accounting the services they give. On the other hand the clients can have electronic devices measuring some physiological quantity, e.g. blood sugar or cardiograms. These devices can act as the sensor nodes described above, recording the recent measurements and transmitting it wirelessly. The mobile station in the pocket of the nurse visiting the client establishes a Bluetooth connection with the devices at the respective home and stores the data it receives.

At the end of the day the nurse returns to the central office of her organization where the corresponding relay station with BT connection is located. Her mobile device transmits the data it has gathered throughout the day to the server. There the health conditions of the clients can be analyzed based on the knowledge about their maladies and medications. If some discrepancies are detected, an alarm is issued, e.g. recommending the client to see a doctor.

The advantage of BlueDACS in this scenario is that it exactly fits in the processes that are currently run. Without much additional effort, a considerable improvement of the home care could be achieved. The challenge in this scenario is to fine-tune the sleep and wake-up sequence of the Bluetooth module in the client's device to certify that the data is actually delivered to the caregiver, without wasting two much energy by sending all day long.

5.2 Environmental Measurements

The quality of weather forecasts and the evaluation of water level changes at creeks and rivers crucially depend on measurements of the environment. The more measurements of temperature, wind speed, and air pressure are available, the more accurate a forecast can be. The same holds true for the prediction of floods, allowing detailed warnings and countermeasures. But the measurement places are often out of reach of public data networks. In the more dense populated areas of Europe, the data is frequently transmitted via cellular networks which causes air-time costs. In other areas often a visit of the measurement stations by staff persons is necessary. They denote the current values on pencil and paper, entering them later in their computers by hand.

This tedious task could be automated and made more reliant by a system like BlueDACS. The measurement stations act as the sensor nodes, receiving their energy from solar panels, possibly storing hourly measurements for one or two days. The data gathering task could be completed by people that pass the station anyway, offering a small fee for their service. If this is unfeasible, either more ore less regular visits are still necessary, but can be done faster and more efficiently. In any case, a Bluetooth connection with a mobile device is used to transmit the measurements. When the person comes home or to his/her office, a PC there could act as a relay station, taking up the data from the mobile phone and forwarding it to the nation-wide weather office (or other authorities) for further processing.

The data is usually not very sensible; thus no special security measures have to be taken. All in all, BlueDACS can automate the tasks and complement other data gathering technologies. In this scenario, the trade-off between the probability to reach a carrier and the energy consumption is also an issue. But as these outdoor devices will usually be powered by solar cells, a higher consumption of the radio unit is acceptable.

5.3 Monitoring of Operation Times of Heavy Machine

On construction sites of roads, bridges, and other complex buildings heavy and very specialized machines are used, e.g. steamrollers or tar-sealing machines. Often they are required just for one step in the entire work process, but are designed to fulfill this task very efficiently. For construction companies the purchase of such machines is often uneconomically; so they obtain them under leasing or renting contracts. More and more renting of such machines is done on a per-work-hour basis. The renting company has only to pay for the time the machine is actually in action. But for a correct and instant accounting the information about the operation times has to reach the machine owner in a reasonable time frame.

This problem could be solved with a system like BlueDACS. The machines would have built-in (and hopefully manipulation-resistant) clocks for recording the operation times. At least the foreman on the construction site carries a mobile phone which receives the operation data from the machines via a Bluetooth connection. If he meets other people involved in the construction, like architects or engineers, his device may exchange the data with their mobiles. As soon as one of them comes back into his office, the data can be transmitted to a relay station that is deployed there. This will then send the information about the operation times to the machine owner so that the billing can be processed in a precise and timely manner.

Again, BlueDACS does not require an abrupt change, but integrates well in the current processes. It is not essential how long the data actually takes from the machine to the owner. This electronic way will be in any case certainly much faster than written notices. But security is an important issue here that has not been discussed so far. The construction machine should encrypt the data, e.g. with a public key, leaving only the address of the owner permanently readable. At the server decryption can be done with the respective private key. So privacy and accuracy are protected on the way.

6 Related Works

A similar approach has already been described in related works. The basic idea of the protocol to bridge disconnected parts of ad hoc networks by mobile nodes is known as the epidemic routing protocol [13]. Much of the work in this field followed this idea, but mostly in a naïve, straightforward way. The approach of Chen and Murphy [14], the so-called Disconnected Transitive Communication paradigm, is certainly similar to ours, but describes more a general framework than a practical realisation. Musolesi et al. [15] concentrate on the routing algorithms and how to make them context-aware, demonstrating their results only by simulations. On the other hand, Lindgren et al. [16] emphasise the probabilistic aspect in their routing mechanisms, trying to connect entire clouds of nodes. They rely on the transitivity of the model and calculate the probability of delivery based on this assumption. In our approach, transitivity is also supported; the system does, however, not rely on it indispensably. In fact, it depends on the actual usage scenario whether the data exchange between the mobile stations is essential or not.

One of the first examples of the application of routing in partially disconnected ad hoc networks involving wireless sensor nodes is the ZebraNet project [17]. It employed sensors attached to animals for studies of wildlife behavior using flooding-like routing protocols. The idea of animals carrying data was also seized in DataMules [18] where, rather similar to our approach, mules are thought to visit sensor nodes and forward their measurements. The probability of delivery depended crucially on the visiting frequency of the mules. Small and Haas [11, 12] who proposed to attach the sensor nodes to whales focused primarily on theoretical comparisons between several data dissemination strategies.

Our results, however, differ in a several aspects. First of all, we have realized a working prototype that proves the feasibility of this concept; only a few of the approaches cited above did that. Second, this prototype does not rely on special hardware, individually adapted to this specific problem. As shown above, we used standard mobile phones as collectors, connecting via IEEE 802.15 (Bluetooth) protocol. And third, we separated the roles of sensors and collectors such that the sensors do not communicate with one another and the collectors do not have sensing capabilities, respectively. But most of all our system is designed to be non-intrusive. As illustrated by the usage scenarios, BlueDACS can integrated smoothly in the processes that are carried out anyway. The usage of BlueDACS does not require any additional efforts, especially not from the people carrying the mobile stations. It just introduces a significant additional value to a process that is already there.

7 Conclusion

Our experiments show that Bluetooth can be used for transmitting data acquired by sensors and other measurement devices. The approach of data gathering and carrying by more or less encountering mobile devices does certainly not replace the sensor network concept, for the delivery times are magnitudes larges than with direct wireless connection. As outline in the scenarios, however, there may be a couple of situations where the drawback of the long over-all transmission time is acceptable. Particularly when the data transmission integrates smoothly in existing processes, it can be a complimentary method worth evaluating.

The prototype described above has proven the feasibility of this concept. For utilizing standard consumer devices, Bluetooth is currently the technology of choice for the radio communication. But as we have outline, there are certain disadvantages with this protocol that raise the need for looking for alternatives. A possible candidate can be ZigBee (IEEE 802.15.4); this protocol was designed for energy-efficient transmission of small data streams. It is thus perfectly suitable for the applications described here. Maybe with the wider dissemination in the industrial area it may also find its way into mobile devices.

The BlueDACS protocol proposed here seems to be fairly sufficient for the anticipated use cases. The only missing issue is security. This could be added by taking an encrypted data set as payload for a packet. But the integration of security in the overall system has clearly to be the subject of further research.

Acknowledgements

The team that realized the prototype comprised the authors as well as Tobias Beer, Wladimir Beylin, Christoph Neumann, and Markus Wachter. So thanks go to all participants for their contributions.

This work was in parts supported by Siemens Corporate Technology, Munich, whom we are also grateful.

References

1. Akyildiz, I.; Su, W. et al.: A Survey on Sensor Networks. IEEE Comm., vol. 40 (2002) 102-114
2. Zhao, F.; Guibas, L.: Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann Publishers, St. Louis, MO (2004)
3. Grossglauser, M., Tse, D.N.: Mobility Increases the Capacity of Ad Hoc Wireless Networks. IEEE/ACM Transactions on Networking, vol. 10/4 (2002) 477–486
4. Martinez, K.; Hart, J.; Ong, R.: Environmental Sensor Networks. IEEE Computer, vol. 37/8 (2004) 50-56
5. Milanovic, N.; Radovanovic, A. et al.: Bluetooth Ad hoc Sensor Network. In: Proc. IX. Telecommunications Forum TELFOR, Belgrad, Serbia (2001)
6. Leopold, M., Dydensborg, M.B., Bonnet. P.: Bluetooth and Sensor Networks: A Reality Check. In: Proc. 1st ACM Conf. Embedded Networked Sensor Systems (SenSys 2003). ACM Press, New York (2003)
7. Beutel, J., Kasten, O. Mattern, F. et al.: Prototyping Wireless Sensor Networks with BTnodes. In: 1st European Workshop on Wireless Sensor Networks (EWSN 2004), Springer LNCS, vol. 2920, Berlin (2004) 323-338
8. www.atmel.com
9. www.stollmann.de
10. Goodman, D.J., Borras, J., Mandayam, N.B., Yates, R.D.: INFOSTATIONS: A New System for Data and Messaging Services. In: Proc. IEEE VTC ’97 (1997) 969-973
11. Small, T., Haas, Z.: The Shared Wireless Infostation Model - A New Ad Hoc Networking Paradigm (or Where there is a Whale, there is a Way), Proc. 4th ACM MobiHoc, Annapolis, MD (2003) 233-244
12. Small, T., Haas, Z., Purgue, A., Fristup, K.: A Sensor Network for Biological Data Acquisition, Chapter 11 in: Ilyas, M. (ed.): Handbook of Sensor Networks, CRC Press, (2004)
13. Vahdat, A., Becker, D.: Epidemic Routing for Partially-Connected Ad Hoc Networks. Duke University Technical Report CS-2000-06 (2000)
14. Chen, X., Murphy, A.L.: Enabling disconnected transitive communication in mobile ad hoc networks. In: Proc. of Workshop on Principles of Mobile Computing, collocated with PODC'01, Newport, RI (2001) 21-27
15. Musolesi, M., Hailes, S., Mascolo, C.: Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks. In: Pro. 6th IEEE Intern. Symp. World of Wireless Mobile and Multimedia Networks (WoWMoM'05), Messina, Italy (2005) 183–189
16. A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. SIGMOBILE Mobile Computing and Communications Review, vol. 7/3 (2003) 19-20
17. Juang, P., Oki, H., Wang, Y. et al.: Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In: Proc. of 10th intern. conf. on Architectural support for programming languages and operating systems, San Jose, CA (2002) 96-107
18. Shah, R., Roy, S., Jain, S., Brunette, W.: Data MULEs: Modeling and Analysis of a Three-tier Architecture for Sparse Sensor Networks. In: Elsevier Ad Hoc Networks J., vol. 1/2-3 (2003) 215-233