DT-WBAN: Disruption Tolerant Wireless Body Area Networks in Healthcare Applications

Felix Büsching, Maximiliano Bottazzi, Wolf-Bastian Pöttner, and Lars Wolf
Technische Universität Braunschweig
Institute of Operating Systems and Computer Networks (IBR)
Mühlenpfordtstr. 23, 38106 Braunschweig, Germany
Email: [buesching | bottazzi | poettner | wolf]@ibr.cs.tu-bs.de

***Abstract*—Delay or disruption tolerant protocols can be used in several application areas. In the area of Ambient Assisted Living (AAL) or healthcare, these protocols are well-suited to provide a seamless handover between online and offline monitoring of vital parameters or activity data; this also implies a synchronization of offline gathered data.**

In this paper, we analyze the expected data rates for several medical sensors and applications and give an insight into how to dimension such disruption tolerant protocols and systems. We also developed a scenario for activity recognition in which we constantly monitor the acceleration data of a wearable sensor node. We implemented and evaluated this scenario with our disruption tolerant protocol for wireless sensor nodes and show its overall functionality, its utility, and its limits.

***Index Terms*—Delay Tolerant Networking; Disruption Tolerant Networking; Wireless Body Area Network; Wireless Sensor Networks; Bundle Protocol;**

I. Introduction

With Wireless Body Area Networks (WBANs), several different vital parameters or activity data – like movement or acceleration – can be recorded. Today's applications either store these data on a wearable device for later analysis or transmit the data to a central instance for further processing. In some field studies within the GAL project [1], activity data is recorded by a worn accelerometer over a longer period of time. This data is e.g. gathered in order to perform a fall risk assessment [2] of the person wearing the accelerometer. In other scenarios, a fall detection is performed [3], whereas in this case accelerometers are used as well. But, in contradiction to the first scenario where the data is stored on a microSD card and processed after the recording (offline), here the data is continuously transmitted via a wireless link to a processing Personal Computer (PC) and analyzed in realtime. In case of a detected fall, such system could immediately send an emergency message to the ambulance or healthcare professionals.

The concept of Delay or Disruption Tolerant Networks (DTNs) [4] has its origin in the interplanetary communication. When communicating with faraway satellites, probes, or Mars rovers, a continuous end-to-end connection between the distant devices and a local control center cannot be assumed. Rather only occasional links between two communication partners are common. This leads to huge disruptions and therefore long delays of the whole communication chain. In conventional (Internet) protocols (like Transmission Control Protocol (TCP) or User Datagram Protocol (UDP)) timeouts would occur. These protocols cannot handle such delays and disruptions as they rely on a permanent and stable end-to-end topology which is not present in outer space. In contrast, DTN protocols are designed to handle such harsh circumstances and to provide a reliable and secure communication where conventional protocols would fail. This is achieved by a dependable hop-by-hop communication between each two (or more) communication partners, whenever a physical radio link between these partners is established. The principle is as reliable as simple – the so-called "store, carry and forward" strategy.

In Figure 1 the basic functionality of a DTN is shown. Node A and node B are not located within each others communication range. Using common protocols, a communication between node A and B would not be possible. In a DTN, node A can forward its data to the moving node M which *stores* the data and *carries* it physically into the communication range of node B. Once arrived, node M *forwards* the data to node B.

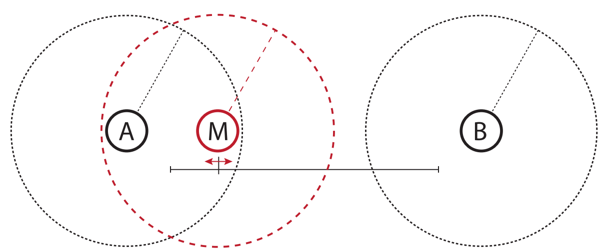

Figure 1. Node M moves between the communication radii of nodes A and B. In a DTN node M stores, carries and forwards the data.

The Bundle Protocol (BP) [5] is a standardized protocol for DTNs. There are several BP implementations for normal PC systems, like e.g. IBR-DTN [6]. In general, the data are packed in so-called *bundles* and forwarded from node to node, whereas these bundles can be of any size.

In many wireless monitoring scenarios – e.g. a patient equipped with a Body Area Network (BAN) for vital parameter monitoring – data is transmitted wirelessly to a sink. Compared to interplanetary communication, these Wireless Sensor Network (WSN) or WBAN scenarios have similar or even more sophisticated challenges to cope. Even in human activity monitoring scenarios, no continuous end-to-end connection between the data capturing BAN and the sink can be assumed.

While the movement of satellites and probes is clearly computable, the movement of humans is not predictable in most cases. First – due to shadowing, packet collisions, and other physical influences – even in (more or less) steady wireless connections, the data rate of the wireless channel can change. Second, when the monitored persons leaves the communication range of the sink, the data transmission completely collapses. In the first case, the quality of data would suffer; in the second case, data will be lost. That again may jeopardize a whole field study, as important data of certain times of the day of are missing, because the monitored person has left the building. All that could be handled by an appropriate dimensioned communication protocol.

The rest of the paper is structured as follows. In Section II some application areas for WBANs are shown and a simple use case of a disruption tolerant WBAN is presented. The expected data rates for selected sensors in WBANs are analyzed and pointed out in Section III. In Section IV, a capacity estimation for disruption tolerant WBANs in the considered area is given. The implemented system is delineated in Section V and evaluated in Section VI. Section VII concludes the paper and gives an outlook on future work.

II. Motivation and Use Case: Disruption Tolerant Healthcare Applications

In countless field studies and surveys, BANs or WBANs are used to record vital parameters, activity data and other information. These data are either processed offline or online, but, sooner or later, the data are transferred to a sink where the further processing is performed. This transfer is either accomplished physically, e.g. by the exchange of an SD card, or wirelessly by a radio connection, whereas there seems to be a trend towards wireless transmission.

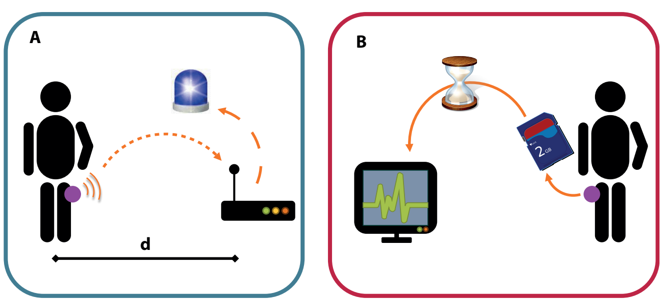

Figure 2. Two different use cases with similar hardware involved. An online fall detection is performed in use case A. In B activity data is continuously recorded on an SD card and processed offline.

Let's think about the following two common uses cases which both assume an elderly person, living at home alone. In the first use case, an online fall detection shall be performed. Data from a wearable 3-axis accelerometer is recorded and directly transmitted to a base station. At this base station, the data are analyzed online and – in case of a detected fall – an automated emergency message to some call center is placed immediately. This surely only works within the communication range of the base station. This use case is shown as A in Figure 2.

In the second use case, the activity and the gait of a person shall be measured and, by that, the individual fall risk shall be assessed. This use case relies on the data from a wearable 3-axis accelerometer, as well; but, here the data are analyzed offline as an immediate action is not necessary. Instead, it is desired to have a complete dataset for the whole monitored period, without any missing data. Thus, a normal continuous radio transmission of this data is not possible as outside the communication range of the base station – e.g. when the monitored person leaves the house – data would get lost. In the future, this use case shall be extended by extra sensors for additional degrees of freedom. A Microelectromechanical System (MEMS) gyroscope will be used to measure orientation in three axes. And a highly precise barometric pressure sensor enables to detect variations in altitude of few centimeters; this allows to determine, whether a person is going up- or downstairs. All of these sensors are already combined in the INGA [7] wireless sensor node, which was designed for that purpose. This use case is shown as B in Figure 2.

So, both use cases include at least one wearable device with similar sensors, whereas the first case requires a constant wireless connection to a sink to perform the fall detection and to send alarm messages; and the second case requires the complete data to be written on a SD card, without any losses. Looking at the accelerometer, even the same measured values could be used for a fall detection and an activity or gait assessment. So, why record this data twice with two devices, just because one dataset is to be analyzed offline and the other dataset has to be analyzed online?

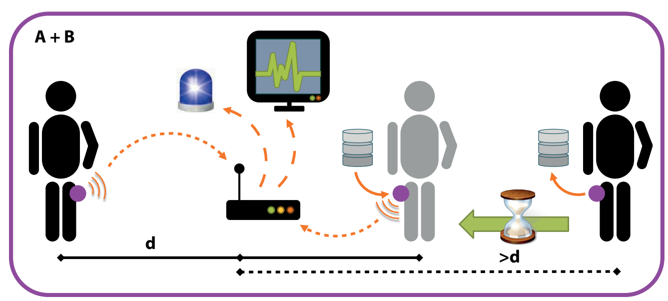

Figure 3. With a DTN protocol the exemplary use cases A and B can be combined. The online fall detection works within the communication range (d) of the base station. The continuous recording stores the data on the node when out of range ($> d$) and synchronizes the data to the base station when back in the range of the base station ($\leq d$).

Here, a disruption tolerant communication protocol comes into play. If the same data of the same sensors are used, a unified data transmission would be the next step. The basic idea is to immediately transmit data, if in communication range of the sink, and to store the data locally on the node, if not. Applications that rely on immediately transmitted data would still function as always – as long as the device is within the communication range of the sink. Additionally, these applications could benefit from an improved fault tolerance, as short disruptions due to fading or a briefly disturbed radio channel would also be handled by such a protocol. Applications that need constantly recorded data also work as always, with the additional benefit that this data are automatically transferred to a base station if in range. Thus, an inconvenient exchange of SD cards will have to be performed no longer, as an implicit synchronization is performed by the DTN protocol. The combination of these two use cases by utilizing a DTN protocol is shown in Figure 3.

A DTN protocol therefore is supposed be transparent to the user and handle all processes automatically. Connection setup, transmission, storing and caching of data, routing, etc. is performed by the DTN layer. The user or software developer just sends the desired data to the DTN layer which is responsible for all the rest.

III. Data Rates

The capability of a DTN protocol depends on the generated data rate of the used sensors and on the achievable payload data rate of the wireless communication channel. In this section, we will give a short insight into both.

A. *Data Rate of Radio Channel*

The data rate of the radio channel depends on the underlying technology. For us, the maximum achievable payload data rate for the application – net data rate – without overhead of other protocols is the most interesting point.

Standard WiFi (IEEE 802.11b) has a net data rate of ~ 5.5 MBit/s [8], whereas the nominal gross data rate is with 11 MBit twice as high. The same applies for newer and improved versions of the WiFi standard where the net data rate is – under optimal conditions – round about half of the respective nominal gross data rate [9]. But, due to its high energy consumption, WiFi is inappropriate for WBANs. Bluetooth (IEEE 802.15.1) in its earlier versions shares the drawback of a comparatively high energy consumption. The current Bluetooth 4.0 is able to transmit at much lower power consumption but still shares another drawback with its predecessors; in a Bluetooth piconet only 8 devices can be active at the same time – in many current and future WBAN scenarios this is just not enough. This is why, in the areas of WSNs and WBANs, IEEE 802.15.4 has been established as a standard.

The nominal data rate of IEEE 802.15.4 in the 2.4 GHz band is at 250 kilobit/s. The maximum achievable net data rate depends on the specific implementation of the used protocols. In our μDTN implementation [10] we could achieve a maximum net data rate of ~ 50 kilobit/s; in Section V and VI we will provide some details.

B. *Generated Data Rate for Monitoring Vital Parameters and Human Activity*

Normally, vital parameters or activity data are continuously sampled at a specific data rate. Whereas, depending on the measured variable, different sensors are able to sample at different data rates. At this point, just some general examples for selected sensors are given. In the concrete case, most sensors of our INGA sensor node are included, as they are part of at least one of the presented use cases. Many sensors do not provide values of whole bytes, but these sensors normally use whole bytes for the representation. E.g. some sensors have a sampling resolution of 10 bit, which is presented as 2 byte (low-byte and high-byte, 16 bit). Here, we assume a respective source coding to achieve that only the sampled data rate will have to be transmitted and stuff bits are discarded before the transmission.

Accelerometer ADXL 345: In a detailed study [11] we determined that Analog Devices' accelerometer ADXL 345 is best suited for our purposes. Due to its low static noise and low energy consumption, it is part of the INGA wireless sensor node. The digital sensor allows for sampling three axes in several discrete rates in the area of 0.1 to 3200 Hz. The resolution depends on the sensitivity (± 2 g to ± 16 g) and varies between 10 and 13 bit. Thus, the lowest possible data rate of this sensor is $0.1\,\text{Hz} \cdot 3\,\text{axes} \cdot 10\,\text{bit} = 3\,\text{bit/s}$. According to that, the highest possible data rate is $3200\,\text{Hz} \cdot 3\,\text{axes} \cdot 13\,\text{bit} = 124\,800\,\text{bit/s}$. In the area of healthcare, both extremes are useless. The authors of [12] argue that under normal conditions there are no accelerations with a frequency > 20 Hz at a human body. Taking the Nyquist-Shannon sampling theorem and some guard space into account, in [11] 50 Hz has been proposed as a reasonable sampling rate for the considered scope. This leads to a typical and realistic data rate of 1500 bit/s for the accelerometer in the desired use case.

Gyroscope ST L3G4200D: Three additional degrees of freedom can be detected by a 3-axis gyroscope. This specific gyroscope has a sensitivity of up to 2000 degree per second and a resolution of 16 bit. The sampling rate can be set to either 100, 200, 400 or 800 Hz, and thus, the data rate of the gyroscope can vary between 4800 and 38 400 bit/s. In the considered field of application, only few studies take gyroscope data into account. While the authors of [13] use a sampling rate of 200 Hz, in [14], [15], and [16] the data are sampled at 100 Hz. As the authors of [17] show that at a sampling rate of 200 Hz, a calibration during usage is possible, we take these 200 Hz as a typical sampling rate for gyroscopes. This leads to typical data rate of 9600 bit/s for the gyroscope.

Barometric Pressure Sensor BMP085 (incl. Temperature Sensor): The authors of [18] use a barometric pressure sensor to determine the altitude of a subject inside a building. Thus, these sensors can be used for activity detection, as well. In [3] such sensor is used for a fall detection, whereas it was sampled at 1.8 Hz and 19 bit resolution. In this specific case, a data rate of 34.2 bit/s is generated. INGAs pressure sensor (manufactured by Bosch) takes 25.5 milliseconds for sampling at the highest resolution of 19 bit. That leads to a maximum sampling rate of 39.2 Hz, which again would lead to a maximum data rate of 741 bit/s. The integrated temperature sensor has a resolution of 16 bit and is able to deliver a new measured value every 4.5 milliseconds. This would lead to a maximum sampling rate of the temperature sensor of 222 Hz and, by that, to a maximum data rate of 3552 bit/s.

As neither temperature nor barometric pressure change that quickly in the desired field of application, such high sampling rates would not make much sense. We consider a sampling rate of 2 Hz reasonable for the barometric pressure sensor and the temperature sensor; that leads to a typical combined data rate of $2\,\text{Hz} \cdot 19\,\text{bit} + 2\,\text{Hz} \cdot 16\,\text{bit} = 70\,\text{bit/s}$ for these two sensors.

Heart Rate and ECG: Not present on INGA, but typically encountered in the area of patient monitoring are measurements of a persons heart rate or an Electrocardiogram (ECG). The highest heart rate ever observed was 1511 beats per minute ($\sim 25\,\text{Hz}$) – at an Etruscan shrew. For humans beings, more than 220 bpm are not expected, thus a resolution of 8 bit is fully sufficient for heart rate monitoring. Since the heart rate does not change within a heartbeat, a transmission with more than 1 Hz is not reasonable. Quite different is the ECG. In [19] a wearable and wireless 1 channel ECG is presented; It works with a resolution of 8 bit and a sampling rate of 500 Hz, which leads to a data rate of 4000 bit/s. The 3 channel ECG in [20] samples every channel with a rate of 125 Hz and a resolution of 14 bit, which again leads to an overall data rate of 1750 bit/s. If the raw data of a fully equipped 12-channel ECG with a sampling rate of 500 Hz and a resolution of 16 bit is to be transferred, an overall data rate of 96 kbit/s would be needed. A typical ECG in (remote) monitoring applications has a bandwidth of up to 100 Hz, which leads to a sampling rate of 200 Hz; sampling three channels with a resolution of 14 bit then leads to an overall data rate of 8400 bit/s.

Table I
MINIMUM, MAXIMUM AND TYPICAL DATA RATES OF EXEMPLARY SENSORS.

Sensor	D_{min}	D_{max}	D_{typ}
Accelerometer	3 bit/s	124 800 bit/s	1500 bit/s
Gyroscope	4800 bit/s	38 400 bit/s	9600 bit/s
Pressure & Temperature	35 bit/s	4392 bit/s	70 bit/s
Heart rate	8 bit/s	32 bit/s	16 bit/s
ECG	800 bit/s	96 000 bit/s	8400 bit/s

As to be seen in Table I, data rates differ from sensor to sensor and from application to application. Thus, the lowest data rate is at 8 bit/s for just transmitting the heart rate every second. With 124 Kbit/s the highest data rate of the considered sensors would occur if a 3-axis accelerometer is sampled at full speed and resolution. When combining sensors, data rates between and beyond these extremes can be generated. Therefore, in the following section, we try to develop a model which is independent of the generated data rate.

IV. Capacity Estimation for Disruption Tolerant Networks

First of all, some variables have to be defined to calculate the capacity a DTN and its capability to cope with disruptions. Data are generated at the generator data rate D_{gen}. According to the use case presented in Section II, an accelerometer generates data at a typical data rate $D_{gen} = 1500\,\text{bit/s}$ as presented in Section III. D_{good} is the net throughput of the application layer (goodput). As mentioned before, our implementation of a DTN protocol for wireless sensor nodes was able to achieve a goodput of $D_{good} \approx 50\,\text{kilobit/s}$.

If the generated data rate is permanently higher than the capacity of the communication channel ($D_{gen} > D_{good}$), only a local storage (e.g. on an SD card) at the recording system is feasible, as the accruing data could never be transmitted completely. This may be overcome by an efficient reduction (e.g. compression) of the raw measured values, but this it out of scope for this publication.

In case of $D_{gen} = D_{good}$ a radio transmission of the data is possible in principle, but, any degradation of the radio channel would lead to data loss; so, a certain oversupply of bandwidth is necessary in any way.

Thus, $D_{gen} < D_{good}$ is the most desirable case, as here (shorter and longer) disruptions can be handled and previously recorded data can be delivered.

The manageable duration of a disruption also depends on the node's available memory for caching S_{node}. The fill level of this storage will be called S_{fill}; at an ongoing disruption of duration $t_{disrupt}$, the storage will be filled with generator data rate D_{gen}, as to be seen in Equation (1).

$$S_{fill} = t_{disrupt} \cdot D_{gen} \tag{1}$$

With lasting duration of disruption $t_{disrupt}$, S_{fill} increases. Once data has been cached in S_{node} ($S_{fill} > 0$), the transfer of the cached data (synchronization) can start. But, as constantly new data are generated, not the whole goodput D_{good} is available for the synchronization.

This is why in Equation (2), where the needed time for synchronization t_{sync} is calculated, in the denominator D_{gen} has to be subtracted.

$$t_{sync} = \frac{S_{fill}}{D_{good} - D_{gen}} \tag{2}$$

In fact, all new generated data will be cached and enqueued after the already stored data to guarantee the correct order of data. If the correct order of data can be achieved differently (e.g. by timestamps) it is also possible to transmit the latest data at first; this would allow for an immediate online monitoring after returning in transmission range of the sink.

A. Special Solution

According to Equations (1) and (2), the contents of Table II can be calculated. In this table, the achievable synchronization times and the needed memory for $G_{gen} = 1500\,\text{bit/s}$ and $D_{good} = 50\,\text{kilobit/s}$ are displayed for different periods of disruption. It is easy to see, that short disruptions in the range of minutes can more or less be instantly synchronized. Even a whole day of disruption can be synchronized in less than 45 minutes. To handle longer disruptions, the node has to be equipped with a sufficient amount of memory. In this scenario, one year of disruption could be synchronized in 11.3 days, but, 6 GB of storage are needed for this period; however, nowadays SD cards should be able to handle this.

Table II
TIME OF DISRUPTION AND SYNCHRONIZATION, AND NEEDED MEMORY FOR $G_{gen} = 1500$ bit/s AND $D_{good} = 50$ kilobit/s.

Disruption	$t_{disrupt}$	t_{sync}	S_{fill}
1 Second	1 s	0.03 s	187.5 byte
1 Minute	60 s	1.86 s	11.25 kB
10 Minutes	600 s	18.56 s	112.50 kB
1 Hour	3600 s	111.34 s	675.00 kB
8 Hours	28 800 s	890.72 s	5.40 MB
1 Day	86 400 s	2672.17 s	16.20 MB
1 Week	604 800 s	18 705.15 s	113.40 MB
1 Month	2 629 744 s	81 332.28 s	493.07 MB
1 Year	31 556 926 s	975 987.40 s	5.92 GB

If we combine Equations (1) and (2), the dependency of the storage fill level S_{fill} can be be dropped and we get Equation (3).

$$t_{sync} = \frac{t_{disrupt} \cdot D_{gen}}{D_{good} - D_{gen}} \tag{3}$$

B. General Solution

Finally, to dimension such disruption tolerant systems, only the ratio of D_{gen} to D_{good} is relevant. Thus, in Equation (4), this ratio is substituted by η. For the special case mentioned above, $\eta = 0.03$.

$$\eta = \frac{D_{gen}}{D_{good}} \tag{4}$$

After substitution of η, Equation (5) results from Equations (4) and (3).

$$t_{sync} = \frac{\eta \cdot t_{disrupt}}{1 - \eta} \tag{5}$$

With this equation, different ratios of generated data rate and application layer throughput η can be addressed, and thus, points regarding capacity and capability to cope with disruptions can be made. In Figure 4 the synchronization time for different values of η for a disruption of one hour are shown. Additionally the memory consumption for $D_{gen} = 1500$ Bit/s is displayed. While the memory usage increases linear, the synchronization time shows exponential growth.

In Figure 5 the synchronization time for different values of η is presented over the disruption time. As can be seen, for small η only very short synchronization times are needed. At $\eta = 0.5$ the synchronization time equals the disruption time. With growing η the synchronization time can be far higher than the disruption time.

V. IMPLEMENTATION

We implemented the described use case (see Section II) for an ongoing field study within the GAL project. The setup is comprised of several parts.

A. MSHP – A Multi Services Home Platform

The so-called Multi Services Home Platform (MSHP) [1] acts as the base station in this experiment. It is mostly consisting of a small standard PC (x86 architecture) in a set-top box [21]. It is running a standard Linux and the Java based

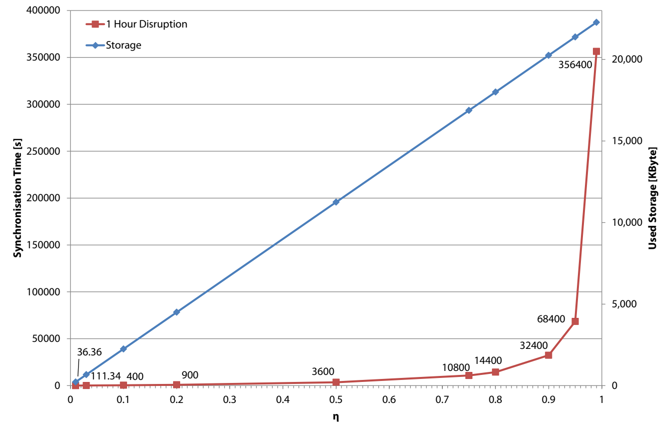

Figure 4. Synchronization time (t_{sync} - left axis) and memory usage (S_{fill} - right axis) after a disruption of one hour ($t_{disrupt} = 1h$) and different η; at a constant generator data rate $D_{gen} = 1500$ bit/s.

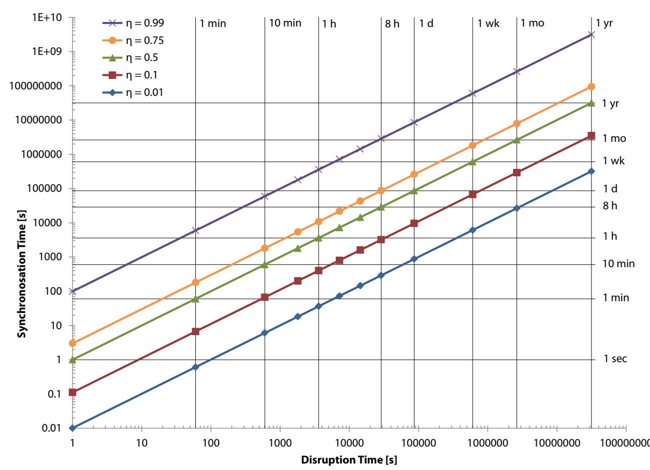

Figure 5. Logarithmic synchronization time over disruption time at different values for η.

OSGi Service Platform, which is just the normal middleware that is used in the entire GAL project. The household of each monitored person is to be equipped with one MSHP, each. Due to the security and privacy concept [22], all recorded data is processed and stored on the specific MSHP in a local database. The online fall detection is performed inside the OSGi service platform as a so-called OSGi bundle, based on the data delivered from the WBAN sensor. Alarm messages can be sent via different channels, like E-Mail, SMS or telephone call. Several other sensors and actuators are attached to a MSHP, but, for our use case we only concentrate on the WBAN.

B. INGA – Inexpensive Node for General Applications

The WBAN consists of INGA wireless sensor nodes. INGA [7] has been developed for human activity monitoring and is equipped with several sensors. In addition to the already in Section III mentioned accelerometer, gyroscope and barometric pressure sensor, INGA also has the capability to measure its own power consumption and to monitor the battery

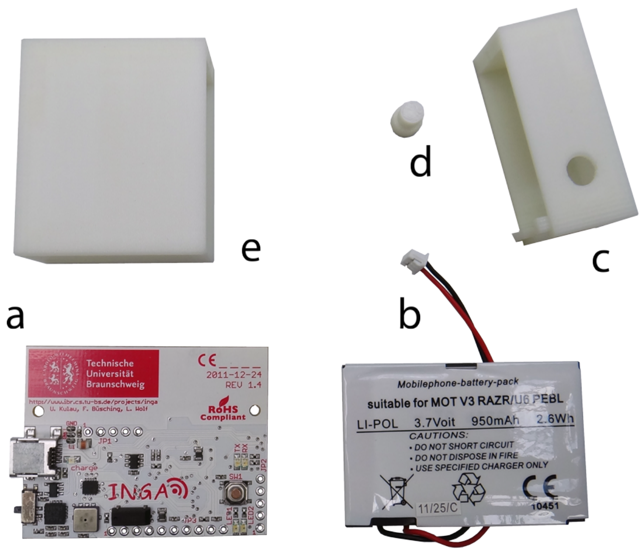

Figure 6. INGA wireless sensor node (a), 950 mAh LiPo battery (b), and 3D printed housing (c, d, e).

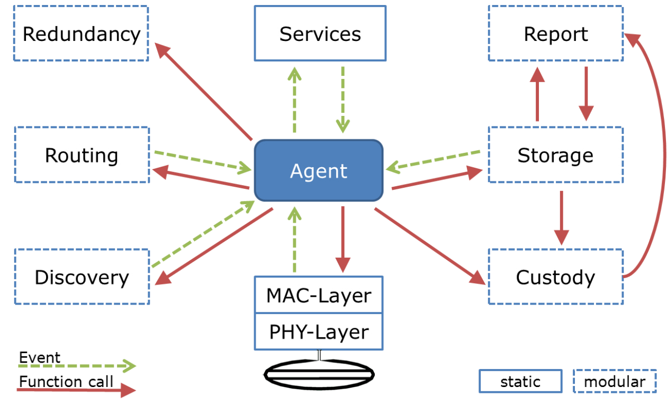

Figure 7. Block diagram of μDTNs architecture.

voltage. INGA is controlled by an 8 bit ATmega 1284P microcontroller, working at 8 MHz and with internal 128kilobytes of flash memory and 16kilobytes of RAM. There is also an external flash memory (16 megabit) and a slot for a microSD card (which can handle cards of up to 32GB). The integrated charging controller allows to charge an attached LiPo battery e.g. via an USB power supply.

INGAs dimensions are 39 x 50 x 7 mm^3 and we manufactured some experimental cases for INGA with a 3D printer, as can be seen in Figure 6.

The rudimentary network consists of two INGAs, running the Contiki [23] open source operating system for wireless sensor nodes. One INGA node is equipped with a battery and attached to the persons body, using a beltbag. It continuously samples acceleration data of the built-in accelerometer at 50 Hz.

Acting as a gateway, another INGA node is connected to the MSHP via the USB port and transmits all received data via an USB-to-serial connection to the receiving OSGi process inside the MSHP. This data is stored in a local database for later analysis and contemporary forwarded to the fall detection service.

For the data transmission between those two wireless sensor nodes we used our μDTN implementation.

C. μDTN – A Bundle Protocol Implementation for WSNs

According to the DTN concept in [4] and the BP specification in [5], we implemented a DTN protocol for WSNs called μDTN [10]. The implementation for Contiki is compatible to Contiki's network stack. Because of implementation efficiency for wireless sensor nodes two compromises in compatibility to the BP specification were necessary: First, only Compressed Bundle Header Encoding (CBHE) of μDTNs primary block is supported, because CBHE leads to smaller bundles that are easier to process and have less communication overhead. Instead of using the (specified) string-based EIDs, μDTN also uses CBHE-style addresses in these bundles. Second, the current version of μDTN does not support fragmentation of bundles. Consequently, only bundles that fit into one IEEE 802.15.4 radio frame can be processed. At the moment, this BP implementation can handle bundles with a payload of up to 88 bytes. When considering these two limitations, μDTN is interoperable with other standard complying BP implementations, e.g. IBR-DTN [6], running on Linux.

In Figure 7 the general structure of our our DTN protocol for WSNs is shown. The *agent* is the central instance that handles the other modular components. For the desired use case with just two communication partners involved, the *routing* module does just simple flooding. Thus, whenever the *discovery* detects an appropriate partner, all new bundles are forwarded to this specific partner. In the current implementation, only RAM *storage* is implemented; thus, only a very limited number of bundles can be stored on the node. We are currently working on a more sophisticated variant that includes the SD card for nearly "unlimited" storage capacity.

Most other bundle protocol implementations are located in the application layer; in contrast, our approach is located just above the MAC layer of IEEE 802.15.4. This saves the overhead of the intermediate layers and allows for an efficient implementation for small microprocessors. In [10] a detailed description of this μDTN protocol is given.

VI. Evaluation

Besides some experiments in the lab, data regarding the capability of using DTN protocols in the desired use cases have been recorded and evaluated in a field study.

A. Prior Studies

In a prior proof-of-concept implementation [24], we used INGA with μDTN in an experiment in which an elevator physically carries temperature data from the rooftop to our lab in the 3^{rd} floor. The elevator was only intermittently in use and then more or less randomly moved between the 14 stories of the building. Nevertheless, all data generated on the

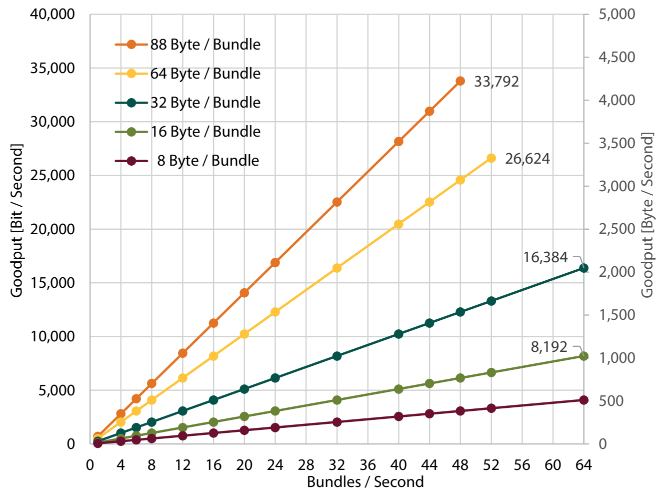

Figure 8. End-to-end application layer throughput (goodput) of the realistic use case.

rooftop made it to our lab, little by little. Thus, we showed the general functionality of our DTN implementation for sensor nodes and we also showed that μDTN was able to handle disruptions without any packet loss.

In a lab setting, we measured the application-layer throughput of μDTN on INGA. With a bundle payload size of 80 bytes a throughput of 47.58 kilobit/s could be achieved. The round-trip in this singlehop scenario was 23.13 ms.

B. Field Study

The realistic use case – consisting of all the components mentioned above – was also evaluated regarding its DTN capabilities. In Figure 8 the results for different payload sizes and different bundle rates are given. In contrast to the measurements in the lab, here the whole chain from the source to the sink is covered; data acquisition, coding, transmitting and receiving via radio link, decoding, converting, transmitting and receiving via serial connection

As can be seen, the maximum achievable payload data rate (goodput) is with $\sim$33 792 bit/s lower than the one in the pure lab setting mentioned beforehand. This is due to the fact that the receiving node not only has to handle the radio connection but also has to "unpack" the bundles, transform the data into plain text, and send it over a serial connection. In Figure 9 we illustrate the transmission chain of the whole implemented and evaluated system. The *printf* over the Universal Asynchronous Receiver/Transmitter (UART) connection is the main responsible for the decreased throughput of the system. For the future we plan to overcome this bottleneck by using a USB adapter that is capable of directly delivering IEEE 802.15.4 frames to the operating system of the MSHP.

Nevertheless, for now and for our use case of a continuously sampled 3-axis accelerometer, the targeted source data rate of 1500 bit/s can easily be transferred via the DTN. As the implementation is able to transport $\sim$33 792 bit/s of payload when transmitting 48 bundles per second with the maximum

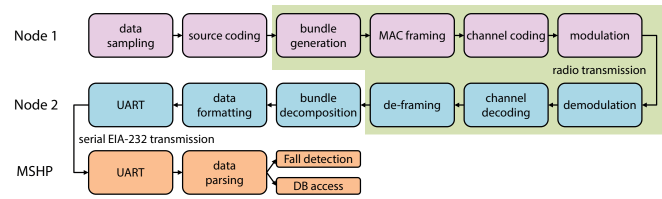

Figure 9. Communication chain of the implemented use case. Only the shaded parts have been considered in the prior lab experiments.

bundle payload of 88 bytes, more than 22 times the data of the accelerometer could be transferred. Thus, for our scenario $\eta \approx 0.044$, which guarantees for relative short synchronization periods. According to Equation 5, the data from one hour disruption should be synchronized after <3 minutes in the implemented use case.

Unfortunately, the storage capacity is the limiting factor in our setup. As the current implementation is not yet capable of utilizing INGAs external memory or SD card, only the RAM storage can be used for caching bundles. In the particular case, the remaining RAM is able to store 60 bundles. Thus, data generated during a disruption duration of some seconds only can be cached and accordingly synchronized, at the moment.

VII. Summary & Conclusions

We have presented two typical use cases for patient monitoring and activity detection that could take advantage of disruption tolerant protocols for data transmission. In fact, we claim that in many use cases in the area of AAL and healthcare, the overhead of DTN protocols would be exceeded by its benefits by far. Pure data collecting use cases can profit from the implicit synchronization. Online monitoring scenarios can benefit from the DTNs capability to also handle short disruptions or interference on the radio link; thus, in such scenarios the loss of data is avoided by the DTN protocol.

The expected data rates for several sensors and applications in the considered area have been exposed. In any case, if the application layer throughput of the DTN implementation is higher than the expected data rate of the considered sensors, a DTN is advantageous. In the area of IEEE 802.15.4 WBANs at 2.4 GHz, for our implementation this limit is currently at $\sim$33 792 bit/s. If the combined data rate of the desired sensors is above that limit, either the DTN implementation will have to be improved or bandwidth will have to be saved. While the first task is ongoing work, the second task can easily and efficiently be implemented by a reasonable coding of the sensors raw data. Delta coding, e.g., takes the dependency of prior values into account and by that is able to lower the needed goodput significantly. However, future implementations will be more efficient and we are pretty sure to be able to push that limit to $\sim$50 Kb/s, which should be sufficient for most use cases.

Thus, if data are recorded in a BAN and these data are either to be stored for future analysis or immediately transmitted for an online analysis, a DTN protocol will make life much easier and even allows to combine these two requirements.

INGA is open source hardware; all resources can be accessed at http://www.ibr.cs.tu-bs.de/projects/inga. μDTN is open source software; its source code can be accessed at http://www.ibr.cs.tu-bs.de/projects/mudtn.

A. Future Work

To overcome the bottleneck of the error-prone serial communication between receiving node and base station, we plan to use an appropriate USB IEEE 802.15.4 adapter, that is able to forward IEEE 802.15.4 frames to the operating system of the host computer. Thus, the decomposition of the bundles and the data formatting can be performed on a powerful device and the overall performance of the system would increase significantly.

The major drawback of our implementation is μDTNs storage module, which is not yet capable of utilizing INGAs external flash or microSD card for caching bundles, but, this is ongoing work.

ACKNOWLEDGMENT

The Lower Saxony research network "Design of Environments for Ageing" acknowledges the support of the Lower Saxony Ministry of Science and Culture through the "Niedersächsisches Vorab" grant programme (grant ZN 2701).

REFERENCES

[1] M. Eichelberg, A. Hein, F. Büsching, and L. Wolf, "The GAL middleware platform for AAL," in *e-Health Networking Applications and Services (Healthcom), 2010 12th IEEE International Conference on*, 2010, pp. 1 –6. [Online]. Available: http://dx.doi.org/10.1109/HEALTH.2010.5556589

[2] M. Marschollek, K.-H. Wolf, M. Gietzelt, G. Nemitz, H. Meyer zu Schwabedissen, and R. Haux, "Assessing elderly persons' fall risk using spectral analysis on accelerometric data - a clinical evaluation study," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, 2008, pp. 3682 –3685. [Online]. Available: http://dx.doi.org/10.1109/IEMBS.2008.4650008

[3] F. Bianchi, S. Redmond, M. Narayanan, S. Cerutti, and N. Lovell, "Barometric pressure and triaxial accelerometry-based falls event detection," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 18, no. 6, pp. 619 –627, dec. 2010. [Online]. Available: http://dx.doi.org/10.1109/TNSRE.2010.2070807

[4] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-Tolerant Networking Architecture," RFC 4838 (Informational), Internet Engineering Task Force, Apr. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4838.txt

[5] K. Scott and S. Burleigh, "Bundle Protocol Specification," RFC 5050 (Experimental), Internet Engineering Task Force, Nov. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc5050.txt

[6] S. Schildt, J. Morgenroth, W.-B. Pöttner, and L. Wolf, "IBR-DTN: A lightweight, modular and highly portable Bundle Protocol implementation," *Electronic Communications of the EASST*, vol. 37, Jan 2011. [Online]. Available: http://dx.doi.org/10.1145/1409985.1410008

[7] F. Büsching, U. Kulau, and L. Wolf, "Architecture and evaluation of inga - an inexpensive node for general applications," in *Sensors, 2012 IEEE*. Taipei, Taiwan: IEEE, oct. 2012, pp. 842–845. [Online]. Available: http://dx.doi.org/10.1109/ICSENS.2012.6411295

[8] A. Kamerman and G. Aben, "Throughput performance of wireless lans operating at 2.4 and 5 ghz," in *Personal, Indoor and Mobile Radio Communications, 2000. PIMRC 2000. The 11th IEEE International Symposium on*, vol. 1, 2000, pp. 190–195 vol.1. [Online]. Available: http://dx.doi.org/10.1109/PIMRC.2000.881416

[9] Y. Xiao, "Ieee 802.11n: enhancements for higher throughput in wireless lans," *Wireless Communications, IEEE*, vol. 12, no. 6, pp. 82–91, 2005. [Online]. Available: http://dx.doi.org/10.1109/MWC.2005.1561948

[10] G. von Zengen, F. Büsching, W.-B. Pöttner, and L. Wolf, "An Overview of μDTN: Unifying DTNs and WSNs," in *Proceedings of the 11th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN)*, Darmstadt, Germany, 9 2012. [Online]. Available: http://www.ibr.cs.tu-bs.de/papers/buesching-fgsn2012.pdf

[11] F. Büsching, U. Kulau, M. Gietzelt, and L. Wolf, "Comparison and validation of capacitive accelerometers for health care applications," *Computer Methods and Programs in Biomedicine*, vol. 106, no. 2, pp. 79 – 88, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169260711002884

[12] A. Willemsen, J. van Alsté, and H. Boom, "Real-time gait assessment utilizing a new way of accelerometry," *Journal of Biomechanics*, vol. 23, no. 8, pp. 859 – 863, 1990. [Online]. Available: http://www.sciencedirect.com/science/article/pii/002192909090033Y

[13] K. Aminian, B. Najafi, C. Büla, P.-F. Leyvraz, and P. Robert, "Spatio-temporal parameters of gait measured by an ambulatory system using miniature gyroscopes," *Journal of Biomechanics*, vol. 35, no. 5, pp. 689 – 699, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021929002000088

[14] R. E. Mayagoitia, A. V. Nene, and P. H. Veltink, "Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems," *Journal of Biomechanics*, vol. 35, no. 4, pp. 537 – 542, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021929001002317

[15] T. Liu, Y. Inoue, and K. Shibata, "Development of a wearable sensor system for quantitative gait analysis," *Measurement*, vol. 42, no. 7, pp. 978 – 988, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0263224109000372

[16] I. Pappas, T. Keller, S. Mangold, M. Popovic, V. Dietz, and M. Morari, "A reliable gyroscope-based gait-phase detection sensor embedded in a shoe insole," *Sensors Journal, IEEE*, vol. 4, no. 2, pp. 268 – 274, april 2004. [Online]. Available: http://dx.doi.org/10.1109/JSEN.2004.823671

[17] S. Scapellato, F. Cavallo, C. Martelloni, and A. M. Sabatini, "In-use calibration of body-mounted gyroscopes for applications in gait analysis," *Sensors and Actuators A: Physical*, vol. 123–124, no. 0, pp. 418 – 422, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924424705001755

[18] G. Lammel, J. Gutmann, L. Marti, and M. Dobler, "Indoor navigation with mems sensors," *Procedia Chemistry*, vol. 1, no. 1, pp. 532 – 535, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S187661960900134X

[19] R. Fensli, E. Gunnarson, and O. Hejlesen, "A wireless ecg system for continuous event recording and communication to a clinical alarm station," in *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, vol. 1, sept. 2004, pp. 2208 –2211. [Online]. Available: http://dx.doi.org/10.1109/IEMBS.2004.1403644

[20] J. Proulx, R. Clifford, S. Sorensen, D.-J. Lee, and J. Archibald, "Development and evaluation of a bluetooth ekg monitoring sensor," in *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*, 2006, pp. 507–511. [Online]. Available: http://dx.doi.org/10.1109/CBMS.2006.74

[21] F. Büsching, M. Doering, and L. Wolf, "Integration of an environments for aging platform in soho-routers," in *Proceedings of the 14th IEEE International Symposium on Consumer Electronics (ISCE2010)*, Braunschweig, Germany, June 2010. [Online]. Available: http://dx.doi.org/10.1109/ISCE.2010.5522751

[22] F. Büsching, M. Bottazzi, and L. C. Wolf, "The GAL monitoring concept for distributed AAL platforms," in *2012 IEEE 14th International Conference on e-Health Networking, Applications and Services (Healthcom) (IEEE Healthcom 2012)*, Beijing, P.R. China, Oct. 2012. [Online]. Available: http://dx.doi.org/10.1109/HealthCom.2012.6379463

[23] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors," in *Proceedings of the First IEEE Workshop on Embedded Networked Sensors (Emnets-I)*, Tampa, Florida, USA, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1109/LCN.2004.38

[24] W.-B. Pöttner, F. Büsching, G. von Zengen, and L. Wolf, "Data elevators: Applying the bundle protocol in delay tolerant wireless sensor networks," in *The Ninth IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2012)*, Las Vegas, Nevada, USA, Oct. 2012. [Online]. Available: http://dx.doi.org/10.1109/MASS.2012.6502520