Vol.2012-HCI-150 No.13
Vol.2012-UBI-36 No.13
2012/11/2

# HUSTLE: Deploying A Secure Wireless Sensor Network by Light Communication with Smartphone

NGUYEN DOAN MINH GIANG[†1,a)] TAKUYA TAKIMOTO[†2,b)] TAKURO YONEZAWA[†2,c)]
JIN NAKAZAWA[†2,d)] KAZUNORI TAKASHIO[†2,e)] HIDEYUKI TOKUDA[†2,f)]

***Abstract:*** Deploying Wireless Sensor Networks (WSN) securely still required users to have certain skills and efforts. In the near "sensor everywhere" future, much simpler method for deploying WSN is necessary for end-users. We propose a method called HUSTLE, which leverages a light-based communication among sensor nodes and Smartphone for adding multiple sensor nodes at the same time. To deploy wireless sensor network, users only need to turn on sensor nodes and shine Smartphone's flashlight on them. WSN ID and one-time security key can be transmitted to all of sensor nodes, and it can be added to with secure.

## 1. Introduction

Nowadays, with advances in hardware and wireless network technologies, we can make multi-functional sensor devices through low-cost, low-power methods. By using hundreds of sensor devices with several sensor node types, we can create a Wireless Sensor Network (WSN) across a geographical area. WSN can be used for collecting, processing and analyzing data in a large area, which is used by a lot of applications such as environment observation and surveillance on remote health services.

The "Sensor-everywhere" era is believed to come shortly and users are expected to be capable of installing an application on a server and setting-up the required sensor nodes by themselves in order to create the WSN. For that purpose, WSN has to meet simplicity and security simultaneously for its use. On the other hand, WSN becomes larger with a lot of sensor nodes, therefore reduce time to setting-up also become an essential requirement.

Although there have been researches efforts for secure WSN or WSN interface set-up [1-7], but it cannot meet security requirements or cannot meet simplicity or take a lot of time to setting-up with once one sensor node is added.

At the moment, we have a lot of applications, which need a light sensor at sensor node such as secure application, environmental monitoring application. In addition, Smartphone also becomes popular, modern Smartphone have a friendly interface with touch screen, and also have a flashlight, which is controllable by programming. Thus, we propose "HUSTLE" method that leverage light communication between Smartphone's flashlight and light sensor of sensor node.This light communication is only active in a small area, and also it can be used to send data to some sensor nodes with secure, then with this we can setting-up WSN easier, faster and more securely by only shine Smartphone's flashlight to new sensor nodes.

The remainder of this paper is organized as follows. Firstly, we describe about motivation, requirements of home environment's WSN and in the end of this section we write about some related works and problems of each when applied in a home environment. Third section is about my propose method, which is called HUSTLE, about concept and communication protocols. End of the third section is written about principle of light communication with Smartphone's flashlight and light sensor of sensor node. Experiment of HUSTLE in real is described in fourth section. Lastly, conclusion and future work is described in fifth section.

## 2. Motivation

To apply WSN at home environment for regular users, WSN need some requirements. There are:

- Easy to setting-up: Because it is used by regular users, who don't have skills and efforts about WSN, therefore it required a simple method which can help setting WSN simpler..
- Fast to setting-up: With a lot of applications, the number of sensor nodes has also increase to several hundreds of sensor nodes. Therefore, it required a faster deploy method.
- Maintaining secure of WSN: With home environment, data privacy also is one of the requirements. Examples can found in security and health monitoring applications. Thus, we have to maintain security of WSN.

To meet easy requirement and maintaining secure of WSN, it is necessary to automatically exchange identification of sensor node or identification of WSN, and also exchange one-time security key for maintaining security of WSN.

[†1] Presently with Keio University / Faculty of Environment and Information Studies
[†2] Presently with Keio University / Faculty of Policy Management
[a)] spider@ht.sfc.keio.ac.jp
[b)] tacky@ht.sfc.keio.ac.jp
[c)] takuro@ht.sfc.keio.ac.jp
[d)] jin@ht.sfc.keio.ac.jp
[e)] kaz@ht.sfc.keio.ac.jp
[f)] hxt@ht.sfc.keio.ac.jp

With once one sensor node method, this takes a lot of time to make a WSN with hundreds sensor nodes. Therefore to meet "fast to setting-up" requirement, it required a method, which can add several sensor nodes at the same time.

### 2.1 Related Work

Nowadays, we have some research efforts for secure WSN or WSN interface set-up. The first research is described in [1]. By using a sensor network kit called "Home Energy Tutor" and to install the application, users only have to place various sensors and scan barcodes on the sensor and in a printed catalog. This creates an association between sensor node and a room or appliance. However the problem is that with only barcodes there are no safe data to use as secure key. Especially when the device is placed outdoors. Therefore data could not be sent with safe in adding process. And also one-time only one sensor nodes are added, therefore this method can not meet setting-up time requirement.
In research [2], have a method that using the light sensor as a receiving data channel to configure the uPart sensor node. But it is only sensor reading cycle, compression values and what values should be transferred, and same with previous related research there is no safe data to use as secure, then we cannot keep secure of WSN.
Another we have research [3] and [4]. Before using, sensor nodes are configured with WSN information and secure protocol data. This information can help determine which WSN that sensor node should be added to. When a sensor node is turned on, it broadcasts the data to neighboring nodes. With this data, it is added to the corresponding WSN. But the problem of this method is about configuring process. This also required having certain skill. Besides, it is also difficult to reconfigure it again for use in a different WSN.
Different with previous related research, [5] is one of research that uses IrDA (Infrared Data Association) to transfer unique code to identify exact sensor node for adding to WSN. But it may cause some problem about cost of sensor node. To use IrDA sensor node requires having correlative hardware which is used only one-time at the installation step. This makes the sensor node more expensive and wasteful, especially with the existence of cheap sensor nodes.
Other method is described at [6]. In this method, camera of Smartphone is used to read QR-Code of each sensor node, with this we can identify exactly what sensor node is added. But however, haven't any safe data from QR-Code to use as secure key, therefore data could not be sent with safe in adding process, thus this method can not meet security requirement.

## 3. HUSTLE

To fulfill all of the requirements, we focus to leverage light communication among Smartphone's Flashlight and light sensor of sensor nodes. With light communication, we can send data to several sensor nodes at the same time, and with this we can be added several required sensor nodes at one-time. For HUSTLE method, we assume that WSN have some properties as follow:

- Each WSN has a unique ID for identification.
- Sink node can send a message to all of added sensor nodes.

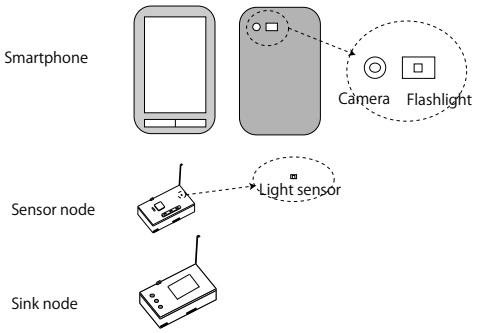


**Fig. 1 HUSTLE devices**

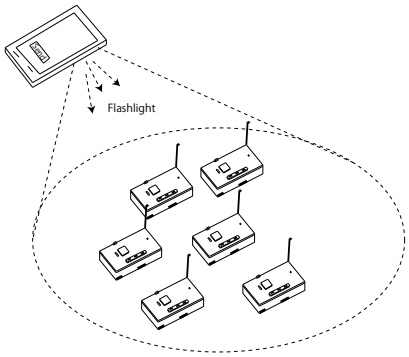


**Fig. 2 HUSTLE prototype: Send data to several sensor nodes**

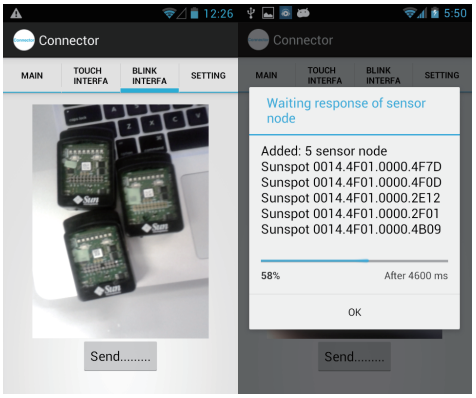


**Fig. 3 User interface: adding and feedback**

- Sink node and Sensor nodes can broadcast a message to another sensor node, include un-setting sensor node.
- All of sensor nodes have a light sensor and are programmable.

We describe HUSTLE, devices, prototype, user interface and adding protocol at below.

### 3.1 HUSTLE Concept

Structure of HUSTLE is shown in Fig.1, we have Smartphone, Sensor nodes and Sink node.
At Smartphone, we have a flashlight and a touch screen (Fig.1). The flashlight is put in back panel of the screen near the back camera and it is controllable by programming, then we can use flashlight to send information to all of sensor nodes and also show the status of this process on screen, like Fig.2.
At Sensor node, we have a lot of applications that use the light sensor (Fig.1). Examples can found in environment monitoring application, secure application. Therefore, we can use light sen-

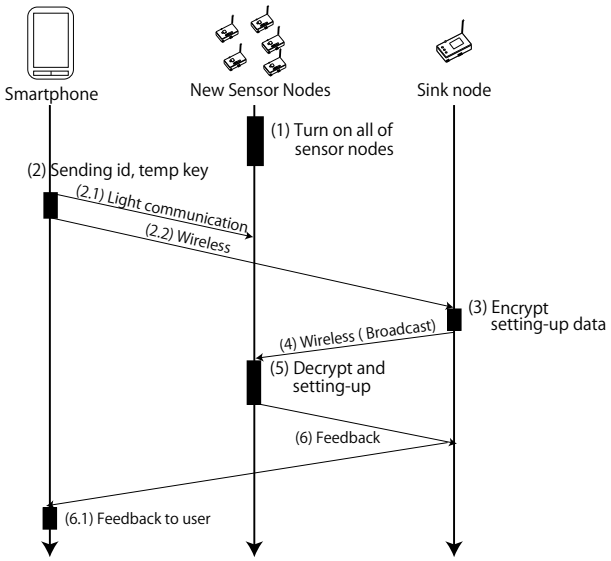


**Fig. 4 HUSTLE protocol**

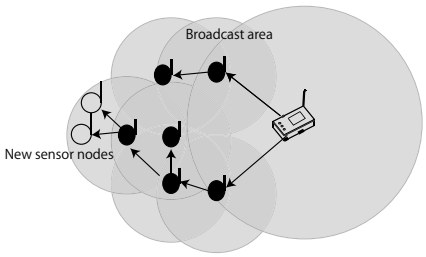


**Fig. 5 Broadcast setting-up message to new sensor nodes**

sor of sensor node to receive information from Smartphone and use it for setting-up process.

About user interface, this is shown at Fig.2 and Fig.3. When user wants to add some new sensor nodes, they turn on it and put it on a plane such as a table. Next step, user directs Smartphone flashlight to all of sensor nodes (Fig.2) and press Send button. Next step, sending process and adding process is automatically occur, user doesn't have to do anything. With HUSTLE, use only has to turn on sensor nodes and shine flashlight in it, this task like workaday task when use turn on Smartphone, capture video, this is very simple to perform. About how to HUSTLE can add new sensor nodes with simple interaction, this is described more specifically in the next part.

### 3.2 HUSTLE protocol

In this part, we describe about HUSTLE protocol with multi-hop routing WSN (Fig.4.) In case single-hop routing protocol is used, the protocol also similar.

The first, in step (1) user turns on all of sensor nodes by hand. After finishing step (1), the user uses Smartphone with HUSTLE program at step (2). In step (2), after user pushed send button, we split to two small processes (2.1) sending WSN id and one-time security key by light communication and (2.2) send this security key by wireless communication. In (2.1), we use Smartphone's flashlight to send WSN identity and the secure key to all of sensor nodes, which need to be added to. When (2.1) finished, the Smartphone sends the secure key to Sink node over the Internet (Wi-Fi connection or Mobile connection) in step (2.2).

With received the secure key in step (2.2), Sink node encrypts setting-up data (WSN information, routing information, secure information...) in step (3).

In step (4), Sink node broadcasts encrypted setting-up data, and to enlarge the broadcast area, all of added sensor nodes also broadcast this encrypted setting-up data ( Fig.5). In this, a message from Sink node to others added sensor nodes is encrypted by secure protocol, message broadcast from Sink node, added sensor nodes to new sensor nodes is encrypted by the secure key. At the new sensor nodes, it can use received secure key in step (2.1) to decrypt and setting-up with the decrypted data. Therefore, we can maintain secure of WSN. After this step, all of sensor nodes are a part of WSN.

In next step, to feedback to user about setting-up process, just added sensor nodes sends a feedback message to Sink node and Sink node forwards this message to Smartphone in step (6). And step (6.1), Smartphone screen is used to feedback to user about setting-up process, what sensor nodes just added (by sensor node ID, which is show at Fig.3).

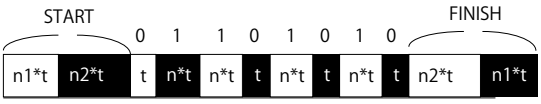


**Fig. 6 01101010 bits pattern**

**Fig. 7 Experiment hardware**

### 3.3 Light communication with flashlight

This part presents about principle of light communication with flashlight of Smartphone and Sensor node's light sensor. To send data with light signal, we have to encode data to light pattern. To do this, we convert data to binary data, 0bit and 1bit. Each bit has a correlative light pattern. And to detect start and finish of data, we also make START pattern and FINISH pattern.

And because we only can change state of flashlight of Smartphone from ON to OFF or OFF to ON, then for simplicity we make 0bit pattern is turned OFF or turn ON flashlight in time unit t. 1bit pattern is in time unit $n * t$. START pattern is the combination of two period time, turn ON in $n_1 * t$ then turn OFF in $n_2 * t$. FINISH pattern is the combination of turn ON in $n_2 * t$ then turn OFF in $n_1 * t$. With this principle, 01101010 bits are encoded to light pattern like Fig.6. In this, turn ON is described with white color; turn OFF is described with black color.

## 4. Experiment

In this section, we describe about experiment of HUSTLE. To

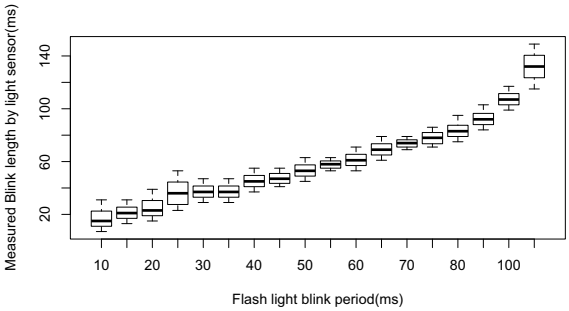


**Fig. 8 Flashlight blink period and measured time length**

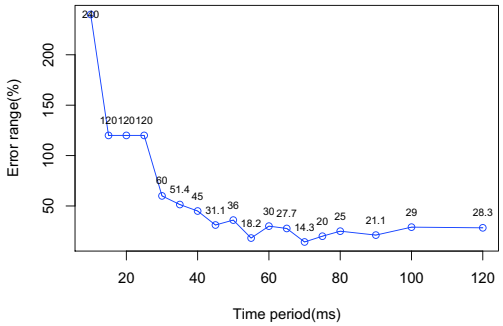


**Fig. 9 Flashlight blink period and measured error range**

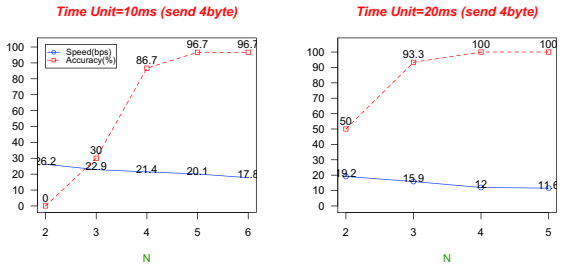


**Fig. 10 Accuracy and speed of light communication with different value of t and n when send 4 bytes data**

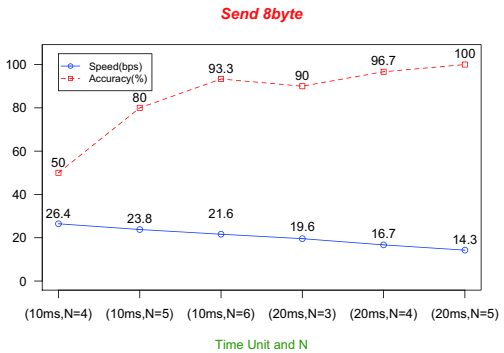


**Fig. 11 Accuracy and speed of light communication with different value of t and n when send 8 bytes data**

experiment HUSTLE method, we used Samsung Google Nexus S Smartphone, SunSpot Java Developer Kit that is shown at Fig.7. Galaxy Nexus S has a touch screen and a controllable flashlight. SunSpot Developer Kit includes SunSpot sensor node and Base Station that is used as Sink Node. SunSpot sensor node is programmable by the Java language and has a light sensor. We used the SunSpot sensor node to make a multi-hop WSN with Network-Wide-Key secure protocol [7]. We send 8bytes random one-time security key from Smartphone to sensor node for encrypting/decrypting data and for identifying. For detecting error when send data by light communication, we also add checksum at the begin of string bytes.

With light communication, to implement it and optimal it, we was tested error range of received blink light length when receive by light sensor of sensor node and send by Smartphone flashlight. And we also tested speed of light communication with different time units and light pattern's parameter.

We also make a testbed with 10 SunSpot sensor nodes and asked 10 participants to evaluate HUSTLE, about accuracy and usability of adding new sensor nodes. With this information, we can know how many sensor nodes can setting-up in one-time, and about can it useful for users.

### 4.1 Light communication

Firstly, when implement light communication we have to define the value of $t$, $n$, $n_1$ and $n_2$. To do that, we take an experiment with Smartphone flashlight and sensor node. We use Smartphone flashlight to blinking t unit period in 100 times, value of t is 10ms, 15ms, 20ms ... 120ms, then measure length time of received light signal by light sensor. We also calculate the error range of received blink time with $ER = (max - min)/t$. The results are shown at Fig.8 and Fig.9.

From Fig.8 and Fig.9, we can see that it is difficult to detect too short blink period. Examples can found with period 10ms, 15ms, 20ms and 25ms. With 10ms, error range is about 240%, from 7 to 31. And we can see that error range is reduced when enhance blink period. The reason of this problem is Smartphone's flashlight takes time to change from ON to OFF state and OFF to ON. When blink period too short, flashlight can not change to stable state, then light sensor can not sense the change of flashlight.

To select value of $n$, $n_1$ and $n_2$, we set $n_1 = 2n$, $n_2 = 3n$ and make an evaluation of light communication with different value of time unit period t and n. The accuracy of each value is present at Fig.10 and Fig.11. Like the mentioned reason, accuracy is increased when increase value of t and n. Compare Fig.10 and Fig.11, we also can see that, accuracy is decrease when send longer string bytes. From Fig.11 we can chosse time unit is 10ms and the value of n is 6. With this value, we have balance between speed (21.6bps) and accuracy (93.3%).

### 4.2 Evaluation HUSTLE: Adding new node accuracy

To evaluation accuracy of HUSTLE, we try to add from 1 new sensor node case to 10 sensor nodes case in one-time, each case is looped in 10 times. The results are shown in Tab.1. From Tab.1, we can know that, it is difficult to add too many nodes at one-time. Example with 10 sensor nodes case, 30% test only can add 5 nodes, and 50% test only can add 7nodes. This is because limitation of the range of Smartphone flashlight. To enlarge the range of flashlight we have to take smartphone at farther, it also makes

**Table 1 Test case and frequency of number nodes is added at one-time.**

| Case/Added | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 Node | 1 | 9 | | | | | | | | | |
| 2 Nodes | 0 | 2 | 8 | | | | | | | | |
| 3 Nodes | 0 | 0 | 1 | 9 | | | | | | | |
| 4 Nodes | 1 | 0 | 0 | 1 | 8 | | | | | | |
| 5 Nodes | 0 | 0 | 0 | 0 | 2 | 8 | | | | | |
| 6 Nodes | 0 | 0 | 0 | 0 | 1 | 3 | 6 | | | | |
| 7 Nodes | 1 | 0 | 0 | 0 | 0 | 3 | 4 | 2 | | | |
| 8 Nodes | 0 | 0 | 0 | 1 | 3 | 2 | 3 | 1 | 0 | | |
| 9 Nodes | 2 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | |
| 10 Nodes | 0 | 0 | 0 | 0 | 0 | 3 | 1 | 5 | 1 | 0 | 0 |

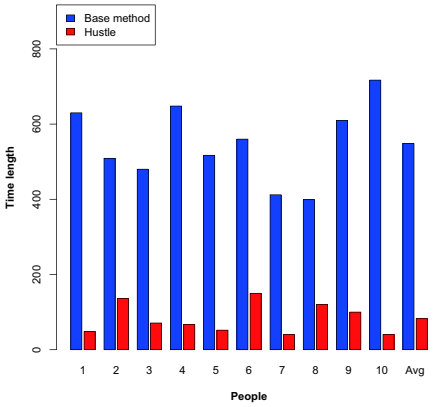


**Fig. 12 Setting-up time length**

**Table 2 Average question score**

| | Simply | Useful | Tire | Learnable | Handle error |
|---|---|---|---|---|---|
| Base method | 2.10 | 2.40 | 3.50 | 2.90 | 2 |
| HUSTLE | 3.70 | 3.50 | 1.70 | 3.80 | 2.8 |

the light signal more weak ($I = \frac{C}{d^2}$) and harder to handle.

### 4.3 Evaluation HUSTLE: Usability

For evaluate usability of HUSTLE, we asked 10 participants to evaluate the usability of HUSTLE method, and compare with base method. With base method, participants have to select true sensor node from list and input secure code of this sensor node. We measure the time length of each user when setting-up a WSN with 10 sensor nodes. After that, we also asked participants to fill out a questionnaire survey about each method with following questions: Was it simple to deploy? Was it useful? Was it tiring to deploy? Was it simple to learn? Was it simple to handle when have error? The results are shown in Fig.12 and Tab.2.

From Fig.12, we can see HUSTLE is about 7times faster than base method (548.2s and 82.5s). The reason is with HUSTLE users only have to use Smartphone to shine to all of sensor nodes, this is very simple to perform. But with base method, to add a new sensor node, participants have to select true sensor node, in addition input the secure key of each sensor node, this task take a lot of time to perform.

And with base method, we have the difference between each participant, someone can input security code faster and also familiar with hexadecimal. With regular users, it make difficult to input with hexadecimal and have limitation of input speed.

In Tab.2, almost of participants evaluate good about HUSTLE. With HUSTLE, we can make WSN with simple interaction, thus it has 3.70/4 point with Simply and 3.50 with Useful. In HUSTLE, one-time security key is sent automatically to all of sensor nodes and also it can check received data by check sum byte, thus it avoids wrong security code problem which usually happen with base method. Then HUSTLE got 1.70/4 point with Tire instead of 3.5 of base method (With Tire, lower score is better).

In Learnable and Handle error question, almost of participants evaluated HUSTLE good with 3.80/4 point of Learnable and 2.80/4 of Handle error, thus we can say that HUSTLE can be applied with nowadays WSN.

## 5. Conclusion and Future Work

In this paper, we propose HUSTLE, a simple interface to add new nodes to secure WSN. With HUSTLE, in once user can add several sensor nodes, thus this can reduce the time needed to setting-up a WSN. Through the experiment by using the Sunspot sensor node, we can see that, with HUSTLE any users can perform the construction of WSN faster, simpler and less tiring. On the other hand, HUSTLE uses only light sensor and does not require any special hardware, this does not make sensor node more expensive, an important requirement.

In future works, to make HUSTLE faster, we will improve speed of light communication. Other the hand, we desire to experiment with HUSTLE to add more than one type of sensor node within one WSN.

### References

[1] Chris. Beckmann, Sunny. Consolvo, Anthony. LaMarca, Nigel. Davies, and Elizabeth.and Mynatt.: Some assembly required: Supporting end-user sensor installation in domestic ubiquitous computing environments. UbiComp 2004: *Ubiquitous Computing*, pages 107-124, 2004.

[2] Michael Beigl, Albert Krohn, Till Riedel, Tobias Zimmer, Christian Decker, and Manabu Isomura.: The upart experience: The upart experience. In Proceedings of the 5th international conference on *Information processing in sensor networks*, IPSN '06, pages 366-373, New York, NY, USA, 2006. ACM.

[3] L.Selavo, A.Wood, Q.Cao, T.Sookoor, H.Liu, A.Srinivasan, Y.Wu, W.Kang, J.Stankovic, D.Young, and J.Porter.: Luster: wireless sensor network for environmental research. In Pro-ceedings of the 5th international conference on *Embedded networked sensor systems*, SenSys '07, pages 103-116, New York, NY, USA, 2007. ACM.

[4] Jeongyeup Paek, K.Chintalapudi, R.Govindan, J.Caffrey, and S.Masri.: A wireless sensor network for structural health monitoring: performance and experience. In Proceedings of the 2nd IEEE workshop on *Embedded Networked Sensors*, EmNets '05, pages 1-9, Washington, DC, USA, 2005. IEEE Computer Society.

[5] H. Baldus, K. Klabunde, and G. Musch.: Reliable set-up of medical body-sensor networks. *Computer Science*, pages 353-363, 2004.

[6] Simon Duquennoy, Niklas Wirstr?m, and Adam Dunkels. 2011. Demo: Snap: rapid sensornet deployment with a sensornet appstore. In Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys '11).

[7] MarcosA.Simplcio, Jr., PauloS.L.M.Barreto, CintiaB.Margi, and Tereza C.M.B.Carvalho.: A survey on key management mechanisms for distributed wireless sensor networks. *Comput. Netw.*, 54:2591-2612, October 2010.