社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006−DPS−129 (10)
2006／11／17

Reliable and Energy-efficient Data Transmission Procedures in Wireless Sensor-Actuator Networks

Kiyohiro Morita, Naohiro Hayashibara, and Makoto Takizawa
Tokyo Denki University, Japan
E-mail {kiyo, haya, taki}@takilab.k.dendia.ac.jp

In a wireless sensor-actuator network (WSAN), sensors and actuators are interconnected with wireless channels. Messages sent by sensors might be lost due to collision if multiple sensors simultaneously send the messages in a channel. In sensing applications, sensed values are in nature required to be sent to actuators in a real-time manner. If a message is detected to be lost, the lost message is retransmitted. It takes at least three rounds to retransmit a lost message. We have to reduce the number of messages retransmitted to realize the real-time communication. We newly propose a data transmission procedure where data in a message m_1 is redundantly forwarded to the destination node by another message m_2 sent by the receiver node of m_1. Even if the message m_1 is lost, the destination node s can receive the data in m_1 if s receives m_2. In addition, we have to reduce the energy consumption of a sensor where only some, of the sensors send sensed values.

無線センサ・アクチュエータ・ネットワーク (WSAN) における高信頼、低電力データ配送方式

森田 清宏, 林原 尚浩 滝沢 誠
東京電機大学大学院 理工学研究科 情報システム工学専攻

無線センサ-アクチュエータネットワーク (WSAN) は、センサとアクチュエータが無線リンクによって相互接続されたネットワークである。複数のセンサが同時にメッセージを送信すると、衝突によりメッセージが紛失する場合がある。センサ応用では、センサからアクチュエータにメッセージがリアルタイムに配送されなければならない。しかし、メッセージが紛失した場合、タイムアウトにより紛失検知後再送されるため、リアルタイムな配送は困難である。そこで、再送信を減らし、かつリアルタイム通信を実現するため、冗長なメッセージ配送手法を新たに提案する。センサの消費電力を低減するために、全センサがメッセージを送信するのではなく、必要数のセンサのみが送信する手法を提案する。

1 Introduction

In a wireless sensor-actuator network (WSAN) [1, 2], sensor nodes and actuator nodes are interconnected in wireless networks. A sensor gathers information about physical world and sends sensed values to actuators. An actuator makes a decision on what actions to be performed and performs the methods on actuation devices. Sensors are low-cost and low-energy devices which communicate with other nodes by taking usage of one broadcast channel. Radio emitted by a sensor is so weak that messages can be delivered to nodes in distance of several meters. An area where nodes can receive messages sent by a node c in a collision area of C. We discuss how to reliably deliver messages to nodes in one broadcast channel and how to reduce energy consumption of a sensor.

If multiple sensors simultaneously send messages in a broadcast channel, the messages are lost due to collision. In order to resolve the collision, some synchronization mechanisms like *CSMA* [7] and *CSMA/CA* [14] are used in sensor networks. If a message is lost, the message has to be retransmitted in traditional transmission protocols like *TCP* [8]. However, it takes time to detect messages lost by the timeout mechanism and retransmit the lost messages. Sensing applications like home network, traffic management, and meteorological observation in nature require real-time communication. That is, a value sensed by a sensor is required to be delivered to an actuator so as to satisfy some time constraint. In order to reduce the number of messages retransmitted, data and control information in a message from a sensor are redundantly transmitted in this paper. If a sensor node s_i sends a message m_1 in a broadcast channel, actuators and sensors receive the message m_1 in a collision area of the sensor s_i. After receiving the message m_1, a sensor node s_j sends a message m_2. Here, if the message m_1

can bring data in m_2, a node can receive data in the message m_1 if the node receives m_2 even if the node does not receive m_1. By using the redundant transmission scheme, the reliability of data transmission is increased in presense of collision. We discuss the *redundant* broadcast procedure which takes advantage of redundant data transmission.

A sensor is equipped with a battery and consumes the energy of the battery to send messages. A sensor sends sensed values if the sensor senses an event. Hence, every sensor sends a message on occurrence of an event. In this paper, we discuss an *energy efficient* data transmission procedure where only enough number of sensors send sensed values and the other sensors do not send to reduce the energy consumption.

In section 2, we present the system model. In section 3, we discuss the redundant data transmission procedure to realize the reliable real-time communication. In section 4, we discuss how to reduce the energy consumption.

2 System Model

A wireless sensor-actuator network (WSAN) W is composed of nodes interconnected with wireless channels. Let $C(c)$ be a collision *area* set of nodes to which a node c can deliver a message. There are two types of nodes, *sensor* and *actuator* ones. A sensor node gathers information in the physical world and then sends the sensed values to actuators [11]. Let **S** be a set of sensors $s_1, \cdots, s_n$ in a WSAN W. An actuator node a makes a decision on what actions to be performed based on values gathered by sensors. W is $\mathbf{S} \cup \{a\}$.

In this paper, we make the following assumptions on sensor-sensor and sensor-actuator communications:

1. One actuator a and sensors $s_1, \cdots, s_n$ $(n \geq 1)$ in W are interconnected in one broadcast channel.
2. Every sensor s_i can receive a message if an actuator a sends the message and no message collision occurs, i.e. $C(a) = W$.
3. A message sent by a sensor s_i can be received by only a limited number of nodes if there is no message collision, i.e. $C(s_i) \subseteq W$.

An actuator a sends a message with stronger radio channel than sensors and can deliver a message to every node as shown in Figure 1. On the other hand, a sensor node s_i can deliver a message to nodes in the collision area $C(s_i)$ of the sensor s_i due to weaker radio. It is noted $a \notin C(s_i)$ for some sensor s_i while $s_i \in C(a)$ for every sensor s_i. Every sensor may not directly deliver a message to an actuator a. A sensor node forwards a message to other nodes, i.e. through multi-hop communication.

If be a pair of different nodes c_1 and c_2 simultaneously send messages, nodes in the collision area $C(c_1) \cap C(c_2)$ cannot receive both the messages due to collision. They are well known *hidden* nodes [9]. Hence, if an actuator a sends a message m and a sensor s_i sends a message m_i, every node in $C(s_i)$ can receive neither m nor m_i while other nodes out of $C(s_i)$ can receive the message m from a.

In this paper, we discuss how messages sent by sensors are more reliably delivered to actuators.

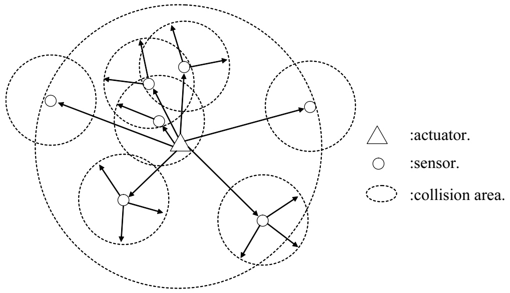


Figure 1. Collision areas of nodes.

3 Redundant Data Transmission Procedure

3.1 Basic procedure

We discuss the sensor-actuator communication to increase the reliability in presence of message collisions. If messages are simultaneously sent by a pair of sensors s_i and s_j in a common area $C(s_i) \cap C(s_j)$ of the collision areas, the messages are lost due to the collision. Let us consider four nodes s_1, s_2, s_3, and s_4 interconnected in a wireless channel as shown in Figure 2. Suppose the node s_1 sends a message m_1 and the node s_2 receives m_1. If another node s_4 sends a message m_3 while s_1 is sending m_1, m_1 and m_2 collide and the sensor s_3 can receive neither m_1 nor m_3 as shown in Figure 3. Suppose the node s_2 sends a message m_2 after receiving m_1 and the sensor s_3 receives m_2. Here, if the message m_2 carries data in the message m_1 to the node s_3, the node s_3 receives not only m_2 but also data of the lost message m_1 as shown in Figure 3a). Figure 3b) shows the traditional retransmission scheme where the node s_1 retransmits m_1 if timeout (TO) occurs. It takes at shortest three rounds to deliver a message if the message is lost. On the other hand, it takes two rounds to deliver a lost message m_1 and no message is retransmitted in the redundant transmission scheme of Figure 3a). In sensing applications, a sensor sends a sensed value, e.g. temperature value of two bytes. However, a message can carry longer data than the sensed value. Each message can carry more number of sensed values than one sensed value. For example, a message is 34 bytes long (variable) and a sensed value is just two bytes long in

Mica2dot [4]. If the identifier of a sensor is two bytes long, a pair of a value and a sensor identifier are four bytes long. Thus, each message can carry eight sensed values in addition to a value sensed by the sensor.

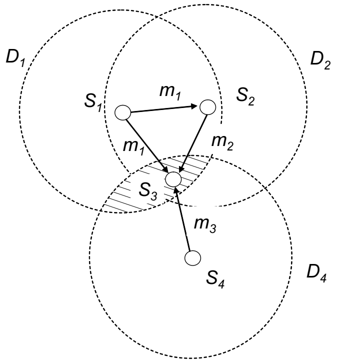
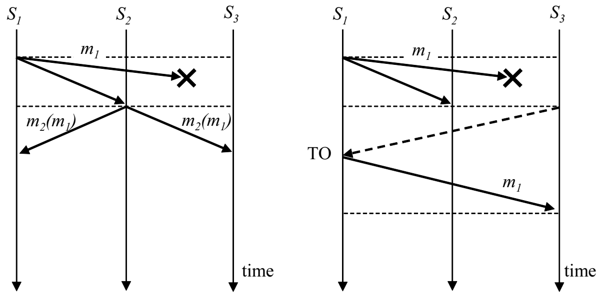


Figure 2. Collision of messages.



a) Redundant transmission.

b) TO/retransmission.

✕ :collision.

Figure 3. Recovery from message loss.

3.2 Message format

Each sensor s_i broadcasts a value v_i. Another sensor s_j also broadcasts the value on receipt of the value v_i. Finally, the actuator a receives the sensed value v_i. Thus, the sensor s_i not only sends a value sensed by s_i but also forwards received values to nodes.

Each message m sent by a sensor s_i is composed of the following fields:

- *m.src* = source sensor s_i of the message m.
- *m.seq* = sequence number of m.
- *m.val* = value sensed by the sensor s_i.
- *m.state* = ON if s_i knows the value $m.val$ is received by an actuator, else OFF.
- *m.data* = $\langle data_1, \cdots, data_{maxDT} \rangle$.
- $m.data_j$ = $\langle src, seq, val, state \rangle$ which s_i has received ($j = 1, \ \cdots, \ maxDT$).

If a sensor s_i sends a message m_2 after another message m_1, $m_1.seq = m_2.seq + 1$ and $m_1.src = m_2.src = s_i$. In addition to sending the value $m.val$ sensed by s_i, the message m carries the sensed values in $m.data$ which s_i has received from the other sensors. For each tuple $\langle s_j, seq, val, state \rangle$ in $m.data$, $state = ON$ if a sensor s_i receives the confirmation of the value val sent by s_j from an actuator a.

On receipt of a message m from a sensor s_i, an actuator a takes sensed values, not only $m.val$ of s_i but also val in $m.data_j (j = 1, \ \cdots, \ maxDT)$ which s_i receives from other sensors. Then, the actuator a sends the acknowledgment of the sensed values to every sensor. An actuator a sends an acknowledgment message m of the following fields:

- *m.src* = source actuator a of the message m.
- *m.seq* = sequence number of m.
- *m.ACK* = $\langle m.ack_1, \cdots, m.ack_{maxAK} \rangle$.
- *m.ack$_i$* = $\langle sid, seq \rangle$ where sid shows a sensor which sends a sensed value and seq is a sequence number of a message from the sensor sid, such that the actuator a receives every message whose sequence number is smaller than or equal to seq.

On receipt of a message m from an actuator a, a sensor s_i finds that the actuator a has received every message m_i where $m_i.seq \leq seq$ for some pair $\langle s_i, seq \rangle$ in $m.ACK$. In addition, the sensor s_i finds that the actuator a receives a message m_k from another sensor s_k where $m_k.seq \leq seq$ for some pair $\langle s_k, seq \rangle \in m.ACK$. If s_i received the message m_k, s_i marks m_k.

On receipt of a message m_j from a sensor s_j, a sensor s_i stores m_j in a receipt buffer BF_i. The sensor s_i forwards sensed values in the message m_j to other nodes as presented. A tuple $\langle s_j, m_j.seq, m_j.val, state \rangle$ is included in a message which s_i sends after receiving m_j. Here, if the message m_j is marked, i.e. s_i knows that the actuator a has received m_j, $state = ON$. Otherwise, $state = OFF$. Hence, if a sensor s_i receives a message m_k from another sensor s_k and $\langle p_j, seq, val, ON \rangle$ is included in $m_k.data$, the sensor s_i finds that the actuator a has received a message m_j where $m_j.seq \leq seq$.

3.3 Data transmission procedure

We discuss how sensors $s_1, \ \cdots, \ s_n$ and an actuator a send and receive messages in one broadcast channel. We use the following procedures to show the data transmission procedure:

- $enque(Q, m)$: a message m is enqueued into a queue Q.
- $m = deque(Q)$: a top message m is dequeued from a queue Q.
- $send(m)$: a message m is broadcast.

- $copy(Q, m)$: the top $maxDT$ elements in Q are copied into $m.data$.

Each sensor s_i has a receipt queue RQ_i where messages from sensors are stored in the receipt order. Each sensor s_i manipulates the following variables:

- seq_i = sequence number of a message which s_i has most recently sent.

Each sensor s_i sends a message m and receives messages from sensors as follows :

[Receipt from sensors] On receipt of a message m from a sensor s_j :

```
for each tuple $\langle s_j, q, v, st\rangle$ in $m.data$,
    if a tuple $\langle src, seq, val, state\rangle$ where
        $src = s_j$ and $seq = q$ is found in $RQ_i$,
        $state = ON$ if $st = ON$ and $state = OFF$;
    else $enque(RQ_i, \langle s, q, v, OFF\rangle)$;
```

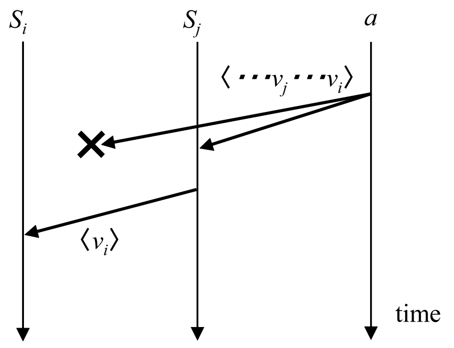


Figure 4. Forwarding of acknowledgment.

[Transmission] The sensor s_i sends a message m with a sensed value v to the actuator a :

1. $seq_i = seq_i + 1$; $m.seq = seq_i$; $m.src = s_i$; $m.val = v$; $m.state = OFF$;
2. $copy(RQ_i, m)$;
3. $send(m)$;

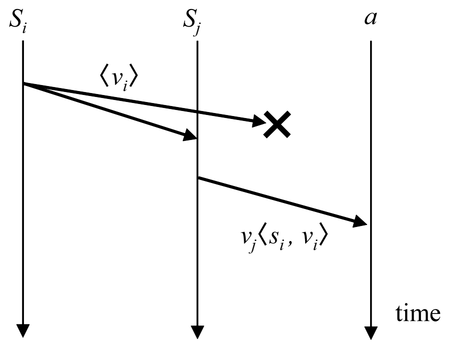


Figure 5. Forwarding of sensed values.

On receipt of a message m_j from a sensor s_j, a sensor s_i stores m_j in a receipt queue RQ_i. Then, the sensor s_i sends a message m_i which includes sensed values of messages in RQ_i. This means, not only the value sensed by sensor is sent but also a message received from another sensor is forwarded to the actuator a by using the flooding protocol [3].

An actuator a receives a message m_i from a sensor s_i. The actuator a broadcasts an acknowledgment message. There are the following variables to send an acknowledgment message in the actuator a;

- CT = counter, initially 0.
- CT_i = counter for each sensor $s_i (i = 1, \cdots, n)$.
- SEQ_i = sequence number of a message which the actuator expects to receive next from a sensor $s_i (i = 1, \cdots, n)$.
- V_i = value sensed by a sensor $s_i (i = 1, \cdots, n)$.
- $S_i = ON$ if the actuator a had sent acknowledgment for V_i, else OFF.

[Receipt from a sensor] On receipt of a message m from a sensor s_i :

```
if $SEQ_i = m.seq$, {
    $SEQ_i = SEQ_i + 1$; $V_i = m.val$; $S_i = OFF$;
    $CT_i = CT$; $CT = CT + 1$;
    for each $\langle s_k, sq, val\rangle$ in $m.data$,
        if $SEQ_k = sq$, {$SEQ_k = SEQ_k + 1$;
            $V_k = val$; $S_k = OFF$;
            $CT = CT + 1$; }
            $CT_k = CT$;
    }
```

[Transmission to sensors] An actuator a sends a message m to every sensor:

```
$h = 0$;
for $i = 1, \cdots, maxAK$ {
    find a sensor $s_j$ such that $CT_j$ is the minimum
    for $CT_j \geq h$ and $S_j = OFF$;
    $m.data_i = \langle s_j, SEQ_j\rangle$; $h = CT_j$; $S_j = ON$;
}
$send(m)$;
```

An actuator a broadcasts acknowledgment of sensed values to sensors. A sensor s_i receives the acknowledgment message m from the actuator a.

[Receipt from an actuator] On receipt of a message m from an actuator a;

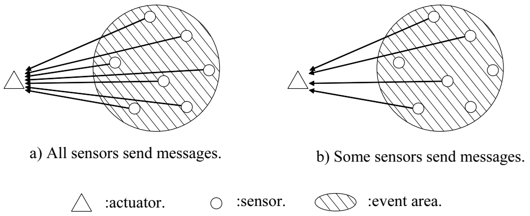
```
for $i = 1, \cdots, maxAK$,
    for every $\langle s_j, seq\rangle$ in $m.ack$,
    if a tuple $\langle s_j, seq_j, val, state\rangle$ is in $RQ_i$
    where $state = OFF$ and $seq_j \leq seq$,
        $state = ON$;
```

The size of the receipt queue RQ_i is limited. Some tuples in RQ_i are removed if RQ_i is full to store a new tuple. There are the following strategies:

1. A tuple $\langle sid, seq, val, ON\rangle$ is removed because it is sure that the actuator a has received the sensed value val.
2. If RQ_i is still full, the top tuple, i.e. oldest one is removed.

4 Energy-efficient Data Transmission

Sensors are low-cost, low-powered devices. We have to reduce the energy consumption of each sensor in a wireless sensor-actuator network (WSAN). Various types of technologies have been so far discussed [12, 13]. A sensor mainly consumes the energy to send and receive messages. In WSAN, multiple sensor nodes sense the same event and generate messages to deliver the sensed value to an actuator because they are closely located [2]. There are multiple sensors in collision areas. Hence, if every sensor which senses an event transmits a message, the collision occurs and lost messages are retransmitted as shown in Figure 6a). Hence, the more number of sensors send messages, the more amount of energy is consumed. Hence, only a limited number, not all of sensors which sense an event send messages to reduce the total energy consumption of sensors as shown in Figure 6b).



a) All sensors send messages.

b) Some sensors send messages.

Figure 6. Number of active sensors.

Clustering techniques have been so far proposed to limit the number of sensors which transmit messages. However, some devices like the beacon [15] and GPS [10] device are required in these approaches. In beacon systems, it is necessary to set up the base station to send out a beacon or sensor must send out a beacon itself. In WSAN, because sensors and actuators are geographically distributed, it is difficult to set up the base station. In addition, the battery of a sensor is burned earlier if sensors send out a beacon. On the other hand, in the GPS approach, every sensor should be equipped with the GPS device. However, GPS devices are too expensive to equip sensors. We have to develop methods to limit the number of sensors which transmit messages using neither the beacon nor GPS device.

If a node sends a message, multiple nodes can receive the message in a broadcast channel. Suppose a sensor node s_i senses a value v_i for an event. We take the following ways to reduce the energy consumption in sensors:

1. On receipt of a message m_j from another sensor s_j, the sensor s_i logs a tuple $\langle s_j, m_j.seq, m_j.val, m_j.state\rangle$ and every tuple in $m_j.data$ in RQ_i in the receipt queue RQ_i as discussed.
2. A sensor s_i senses a value v_i on occurrence of an event e. The sensor s_i first checks the queue RQ_i. If $|RQ_i| < \pi_i$, i.e. only a fewer number of sensors than π_i have sent sensed values, the sensor s_i sends a message m_i with the sensed value $m.val$. Otherwise, the sensor s_i does not send a message with the sensed value.

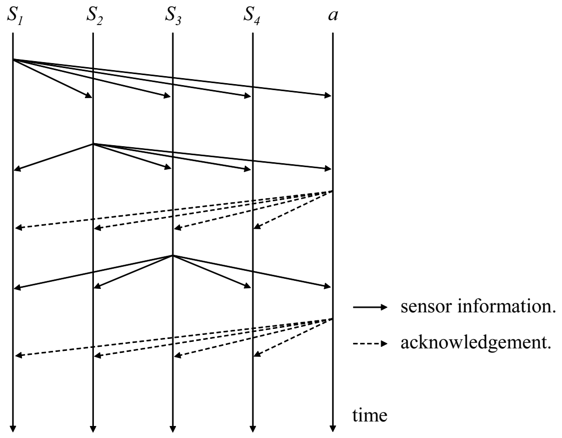


Figure 7. Repressed transmission.

In Figure 7, there are four sensor nodes $s_1, s_2, s_3,$ and s_4 and an actuator node a which are interconnected in a broadcast channel. Here, we assume each node can deliver a message to every node if no collision occurs. Suppose $\pi_i = \pi = 3$ for every sensor $s_i (i = 1, \cdots, 4)$. First, the sensor s_1 sends a message m_1 with the sensed value v_1. Every other node is listening to the transmission of the message m_1 and receives the message m_1 according to the *CSMA* scheme [7]. Every other sensor node s_i has two choices after the transmission of m_1; 1) s_i sends a message m_i with a value v_i or 2) waits. In this paper, we take the following strategies:

1. A sensor s_i first listens to the wireless channel before transmitting a message m_i.
2. If no node is transmitting a message in the channel, the sensor s_i sends a message m_i with the sensed value v_i.
3. If some sensor s_j or an actuator a is transmitting a message m, the sensor s_i receives the message m. If the sensor s_j finishes receiving the message m, the sensor s_i waits for random time units.

4. If the sensor s_i perceives the actuator a to receive messages from more than the number π_i of sensors, as discussed in the preceding subsection, the sensor s_i stops sending the message m_i.
5. If timer expires, the sensor s_i tries to send the message m_i at step 1.

As discussed in the preceding subsection, a message from an actuator a and sensors carry the acknowledgment information ack. By using the acknowledgment ack, each sensor s_i can know which sensed values which s_i sends and receives are acknowledged by the actuator a. In Figure 7, after the sensor s_1 sends a message m_1, the sensor s_2 sends a message m_2. Then, the actuator a sends the acknowledgment message for m_1 and m_2. The sensor s_3 sends a message m_3 and the actuator a sends the acknowledgment for m_3. On receipt of the acknowledgment, the sensor s_4 knows the actuator receives messages m_1, m_2 and m_3 from three sensors. Hence, the sensor s_4 stops transmission of a message m_4.

A sensor also consumes energy to receive messages. Then, an actuator a does not send an acknowledgment message on receipt of a message from a sensor. We take the delayed acknowledgment strategy. An actuator a sends an acknowledgment message for some number $maxAK$ of messages from sensors. The number $maxAK$ means each acknowledgment message can carry the acknowledgment of $maxAK$ sensor messages.

5 Concluding Remarks

We discussed the reliable and energy-efficient data transmission procedures in a wireless sensor-actuator network (WSAN). A message received by a sensor is redundantly carried by another message sent by the sensor. Hence, even if some messages are lost, nodes can receive data in the message if messages carry data in message which have been received. In addition, the total energy consumption of sensors is reduced where only a limited number of sensors send sensed values.

We are now evaluating the reliable and energy-efficient data transmission procedures discussed in this paper by using the simulator of Mica [5].

Acknowledgment

This research is partially supported by Research Institute for Science and Technology [Q05J-04] and Frontier Research and Department Center [16-J-6], Tokyo Denki University.

References

[1] I. F. Akyildiz and I. H. Kasimoglu. Wireless sensor and actor networks: research challenges. *Ad Hoc Networks journal (Elsevier)*, 2:351–367, 2004.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks journal (Elsevier)*, 38:393–422, 2002.

[3] Z. Cheng and W. B. Heinzelman. Flooding strategy for target discovery in wireless networks. *Proceedings of the 6th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems*, pages 33–41, 2003.

[4] Crossbow Technology, Inc. *Wireless sensor networks Getting Started Guide*, 2005.

[5] Crossbow Technology, Inc. *MPR2400J/420/520 MIB User's Manual*, 2006.

[6] N. Hayashibara, X. Défago, R. Yared, and T. Katayama. The φ Accrual Failure Detector. *Proc. of the 23rd IEEE International Symposium on Reliable Distributed Systems (SRDS-23)*, 1:68–78, 2004.

[7] I. Joe and S. G. Batsell. Reservation CSMA/CA for multimedia traffic over mobile ad-hoc networks. *2000 IEEE International Conference*, 3:1714–1718, 2000.

[8] J.Postel. *Transmission control protocol RFC 793.* 1981.

[9] S. Khurana, A. Kahol, and A. P.Jayasumana. Effect of Hidden Terminals on the Performance of IEEE 802.11 MAC protocol. *Proceedings of the 23rd Annual IEEE Conference on Local computer Networks*, pages 12–20, 1998.

[10] T.-H. Lin, Y.-S. Chen, and S. Lee. PCAR:A Power-Aware Chessboard-Based adaptive Routing Protocol for Wireless Sensor Networks. *Emerging Technologies:Frontiers of Mobile and Wireless Communication, 2004. Proceedings of the IEEE 6th Circuits and Systems Symposium*, 1:145–148, 2004.

[11] K. Ozaki, K. Watanabe, satoshi Itaya, N. Hayashibara, T. Enokido, and M. Takizawa. A Fault-Tolerant Model of Wireless Sensor-Actor Network. *Proceedings of the Ninth IEEE International Symposium on Object and Component-Oriented Real-Time Distributed computing (ISORC'06)*, pages 186–193, 2006.

[12] V. Paruchuri, A. Durresi, and L. Barolli. Energy Aware Routing Protocol for Heterogeneous Wireless Sensor Networks. *Proc. of 16th International Workshop on Detabase and Expert Systems Applications (DEXA2005)*, pages 133–137, 2005.

[13] C. Schurgers, M. B. Srivastava, and V. Tsiatsis. STEM: topology management for energy-efficient sensor networks. *Proceedings of IEEE Aerospace Conference Proceedings*, 3:1099–1108, 2002.

[14] Z. Yao, P. Fan, and Z. Cao. An enhanced CSMA-CA mechanism for multihop ad hoc networks. *Communications, 2004 and the 5th International Symposium on Multi-Dimensional Mobile Communications Proceedings*, 2:966–970, 2004.

[15] J. Zhang, B. Huang, L. Tu, and F. Zhang. An Energy-Efficient Cross-Layer Scheduling for Clustered Sensor Networks. *IEEE International Symposium on Communication and Information Technologies 2005 (ISCIT2005)*, 1:238–241, 2005.