Mobile Information Systems 4 (2008) 237–252
IOS Press

Design and evaluation of reliable data transmission protocol in wireless sensor networks

Ailixier Aikebaier[a,*], Tomoya Enokido[b] and Makoto Takizawa[c]
[a]*Department of Computers and Systems Engineering, Tokyo Denki University, Ishizaka, Hatoyama, Hiki, Saitama 350-0394, Japan*
[b]*Faculty of Bussiness Administration, Rissho University. 4-2-16, Osaki, Shinagawa, Tokyo, 141-8602, Japan*
[c]*Department of Computers and Information Science, Seikei University. 3-3-1 Kichijoji-kitamachi, Musashino-shi, Tokyo, 180-8633, Japan*

Abstract. A wireless sensor-actuator network (WSAN) is composed of sensor modes and actuator modes which are interconnected in wireless networks. A sensor node collects information on the physical world and sends a sensed value in a wireless network. Another sensor node forwards the sensed value to deliver to an actuator node. A sensor node can deliver messages with sensed values to only nearby nodes due to weak radio. Messages are forwarded by sensor nodes to an actuator node by a type of flooding protocol. A sensor mode senses an event and sends a message with the sensed value. In addition, on receipt of a message with a sensed value from another sensor mode, a sensor node forwards the sensed value. Messages transmitted by sensor nodes might be lost due to noise and collisions. In this paper, we discuss a redundant data transmission (RT) protocol to reliably and efficiently deliver sensed values sensed by sensor nodes to an actuator node. Here, a sensor node sends a message with not only its sensed value but also sensed values received from other sensor nodes. The more number of sensed values are included in a message, the more frequently the message is lost. Each message carries so many number of sensed values that the message loss ratio is not increased. Even if a message with a sensed value v is lost in the wireless network, an actuator node can receive the sensed value v from a message sent by another sensor node. Thus, each sensed value is redundantly carried in multiple messages. The redundancy of a sensed value is in nature increased since the sensed value is broadcast. In order to reduce the redundancy of sensed value, we take a strategy that the farther sensor nodes from an actuator node forward the fewer number of sensed values. We evaluate the RT protocol in terms of loss ratio, redundancy, and delay time of a sensed value. We show that about 80% of sensed values can be delivered to an actuator node even if 95% of messages are lost due to noise and collision.

Keywords: Wireless sensor-actuator network, WSAN, wireless network, data transmission protocol, reliable data transmission

1. Introduction

A wireless sensor-actuator network (WSAN) [1,2,6,14,15] is composed of *sensor* nodes and *actuator* nodes which are interconnected in a wireless network. A sensor node s_i gathers information on the physical world and sends a sensed value in a wireless channel. The sensor node s_i is referred to as *initial*

---

[*] Corresponding author: Ailixier Aikebaier, 3-3-1 Kichijoji-kitamachi, Musashino-shi, Tokyo 180-8633, Japan. Tel.: +81 422 37 3738; E-mail: alisher.akber@computer.org.

sensor node of the sensed value. Messages sent by a source sensor node are forwarded to an actuator node by another sensor node in a type of flooding routing algorithm [6]. There are many discussions on how sensor nodes forward messages to an actuator node so that energy consumption is minimized [6]. On receipt of a sensed value, the actuator node makes a decision on what action to be performed on the real world and performs the action on the real world. A sensor node can deliver a message to another node in its cell. A sensor node is a low-cost, low-energy device with wireless communication facility [5]. Typically, the radius of a cell is about six meters in a Mica Mote sensor node [4,5]. Hence, the farther a destination node is from a source sensor node, the more number of messages sent by the source sensor node are lost by the destination node due to noise in a wireless network. In addition, if multiple sensor nodes whose cells overlap simultaneously send messages, the messages are lost due to collisions. Only simple mechanisms for resolving collisions like the CSMA scheme [9] and TDMA [10] can be used in sensor nodes due to the limited computation power. It is significant to discuss how to reliably and efficiently deliver each sensed value to the actuator node from the source sensor node in present of message loss in the wireless network.

The authors discuss the *redundant data transmission* (RT) protocol [12,13] to reduce the loss ratio of sensed values sent by sensor nodes even if messages are lost due to noise and collision in a wireless network. If a sensor node s_i receives a message sent by another sensor node s_j, the sensor node s_i sends a message with not only a sensed value obtained by s_i but also sensed values received from other sensor nodes. Even if a message is lost, the sensed value in the lost message can be carried in other messages. The number of sensed values are carried in a message. The longer the massage length is, the higher possibility the message is lost. A message carries such a limited number of sensed values that the message loss ratio is not increased. It is noted that neither lost messages are retransmitted nor additional messages are transmitted in the RT protocol. Therefore, the number of messages transmitted in a wireless network is not increased in the RT protocol. Since each message can include only a limited number of sensed values, some sensed value may not be forwarded in another message. In addition, since each sensed value is flooding and is carried by multiple messages, the total bandwidth used by sensor nodes are increased. In this paper, we try to reduce the number of messages which carry each sensed value. We take a strategy that the farther sensor node from an actuator node forwards a sensed value with the smaller probability to reduce the number of messages carrying each sensed value.

In this paper, we implement the RT protocol in Mica Mote sensor nodes. We evaluate the RT protocol in terms of the loss ratio, redundancy and delay time of sensed values in presence of message loss. We show the loss ratio of sensed values can be reduced to about 20% even in an environment where 95% of messages are lost due to noise and collision in a wireless network. It is noted that the loss ratio of sensed values can be decreased even if lost messages are not retransmitted and any additional control message are not transmitted.

In Section 2, we present the data transmission procedure in the RT protocol. In Section 3, we evaluate the RT protocol in terms of data loss ratio, redundancy, and delivery time.

2. Redundant data transmission (RT) protocol

2.1. Redundant data transmission

We consider a wireless sensor actuator network (WSAN) where multiple *sensor* nodes $s_1, \cdots, s_n$ and one *actuator* node a which are interconnected in a wireless network. A sensor node s_i obtain a sensed value v by sensing an event occurring in an event area of a physical world and sends a message m with

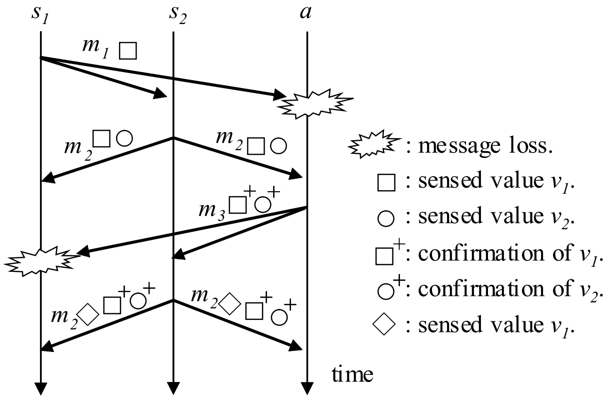


Fig. 1. Message loss.

the sensed value v in a wireless network. A *cell* is an area where a sensor node s_i can deliver a message. Another sensor node s_j in the cell of the sensor node s_i receives the message m and forwards the message m. Thus, a sensed value is forwarded by sensor nodes and delivered to the actuator node a.

Suppose the sensor node s_1 sends a message m_1 and the sensor node s_2 receives the message m_1 but the actuator node a does not receive m_1 due to noise and collision as shown in Fig. 1. Suppose the sensor node s_2 sends a message m_2 after receiving m_1 and the actuator node a receives m_2. Here, if the message m_2 carries a sensed value v_1 in the message m_1 in addition to the value v_2, the actuator node a receives not only the message m_2 but also the value v_1 of the lost message m_1. Data of the message m_1 carried by the message m_2 is referred to as *backup data* of the message m_1. The message m_2 is referred to as a *backup* message of the message m_1. On receipt of the message m_2, the actuator node a sends a confirmation message m_3 for the sensed values v_1 and v_2. The sensor node s_2 receives the message m_3 but the sensor node s_3 loses the message m_3. Then, the sensor node s_2 sends a message m_4 with a new sensed value v_3. Here, the message m_4 carries the receipt confirmation of the sensed values v_1 and v_2. The sensor node s_1 receives the message m_4 and knows that the sensed value v_1 is confirmed, i.e. the actuator node a receives the sensed value v_1.

On receipt of a sensed value v_j from another sensor node s_j, each sensor node s_i forwards the sensed value v_j in a message which the sensor node s_i sends. Thus, a sensed value is flooding through sensor nodes in a wireless network. A message can include only a limited number K of sensed values and each sensor node can store only a limited number M of sensed values. We have to reduce the number of sensor nodes which forward each sensed value. We take the following strategy:

1. Suppose a sensor node s_i sends a message m_i with a sensed value v_i and a pair of sensor nodes s_j and s_k receive the message m_i.
2. If the sensor node s_j is nearer to an actuator node a than the other sensor node s_k, the sensor node s_j has the higher possibility that the sensor node s_j forwards the value v_j than the sensor node s_k.

The *distance* δ_i between the actuator node a and a sensor node s_i shows how many sensor nodes a message hops to get to an actuator node a from the sensor node s_i. The longer the distance δ_i is, the smaller possibility a sensor node s_i forwards a receipt value.

On receipt of a sensed value v sensed by a sensor node s_i, the actuator node a sends a confirmation message m with a tuple $\langle s_i, v, ct\rangle$ where $ct = 0$ in the wireless network. If a sensor node s_j receives the confirmation data $\langle s_i, v, ct\rangle$, the variable ct is incremented by one if the sensor node s_j had not received any confirmation of the value v. Then, the sensor node s_j forwards the confirmation data to other sensor nodes. Each sensor node s_i collects the values of the variable ct carried in confirmation messages. The average value of the variable ct shows the distance δ_i of the sensor node s_i.

2.2. Message format

A sensor node s_i obtains a sensed value v and sends a message with the sensed value v in a wireless network. A message m sent by a sensor node s_i has the following attributes:

- $m.src =$ source sensor node s_i of the message m.
- $m.seq =$ sequence number of the message m.
- $m.val =$ value v sensed by the sensor node s_i.
- $m.state = ON$ if the source sensor node s_i knows that the value $m.val$ is received by an actuator node a, else OFF.
- $m.data =$ *backup* data $\langle data_1, \cdots, data_K \rangle (K \geqslant 0)$.
- $m.data_j =$ *backup* tuple $\langle sid, seq, val, state \rangle$ where sid is an initial sensor node of the sensed value val, seq is a sequence number of message which the sensor node sid sends, and $state$ is the state of the message whose sequence number is seq $(j = 1, \ \cdots, \ K)$.
- $m.dist =$ distance δ_i of the source sensor node s_i.

If a sensor node s_i sends a message m after sending another message m_1, the message sequence number seq is incremented by one, i.e. $m.seq = m_1.seq + 1$. In addition to the sensed value $m.val$, the message m carries the *backup* data $m.data$ which is composed of sensed values which the sensor node s_i has received from other sensor nodes. $m.data$ includes the number $K(\geqslant 0)$ of *backup* tuples. For each *backup* tuple $d = \langle sid, \ seq, \ val, \ state \rangle$ in $m.data$, $d.state = ON$ if the sensor node s_i receives the confirmation of the sensed value $d.val$ whose initial sensor node is $d.sid$.

An actuator node a receives messages with sensed values from sensor nodes. Then, the actuator node a sends the following confirmation message m to sensor nodes:

- $m.ack =$ receipt confirmation $\langle m.ack_1, \cdots, m.ack_A \rangle$ $(1 \leqslant A \leqslant n)$.
- $m.ack_j =$ confirmation tuple $\langle sid, seq \rangle$ showing that the actuator node a receives every message m' sent by a sensor node sid, where $m'.seq \leqslant seq$ $(j = 1, \cdots, A)$.

Suppose a sensor node s_i receives a confirmation message m which includes a confirmation tuple $\langle s_i, seq \rangle$. Here, the sensor node s_i knows that every message whose sequence number is smaller than or equal to seq and which the sensor node s_i sends is received, i.e. confirmed by the actuator node a.

2.3. Data transmission procedures

If a sensor node s_i had received a message m_j from a sensor node s_j, a tuple $\langle m_j.src, m_j.seq, m_j.val, m_j.state, m.dist \rangle$ is stored in the receipt queue RQ_i. A message m_j is referred to as *confirmed* in the receipt queue RQ_i and attributes $m_j.state$ of the message m_j is changed with ON if the sensor node s_i perceives the message m_j to be received by the actuator node a. A sensor node s_i knows that a sensed value v is confirmed not only by receiving a confirmation message of the sensed value v from the actuator node a but also a message from another sensor node which includes the confirmation information that the value v is confirmed. A sensor node s_i manipulates a variable SEQ whose initial value is zero. A variable D_i indicates the distance δ_i between the sensor node s_i and the actuator node a. A sensor node s_i sends a message m with the sensed value v as follows:

```
[SN-send($v$)] {
    $SEQ := SEQ + 1$;
    $m.seq := SEQ$;      /* sequence number of $m$ */
    $m.src := s_i$;      /* source sensor node */
    $m.val := v$;        /* sensed value */
    $m.dist := D_i$;     /* distance */
    $m.state := OFF$;    /* sensed value is not confirmed by an actuator node */
    $copy(RQ_i, m)$;     /* $K$ tuples in $RQ_i$ are stored in the backup data $m.data$. */
    $store(RQ_i, \langle m.src, m.seq, m.val, m.state, m.dist\rangle)$;
                         /* a tuple $\langle m.src, m.seq, m.val, m.state, m.dist\rangle$ is stored in $RQ_i$ */
    $send(s_i, m)$;      /* the sensor node $s_i$ broadcasts the message $m$. */
  }
```

In the procedure $copy(RQ_i, m)$, the top K of *backup* tuples in the receipt queue RQ_i are copied into the *backup* data of a message m. In the procedure $store(RQ_i, t)$, a tuple t is stored in the recept queue RQ_i. The sensor node s_i broadcasts a message m in the procedure $send(s_i, m)$ in a wireless network.

A sensor node s_i forwards sensed values of the receipt queue RQ_i in a message m' when the sensor node s_i sends the message m' by the procedure **SN-send**. In order to reduce the number of messages which carry the sensed values, the sensor node s_i does not always store the sensed values in the received message m. In this paper, a sensor node s_i decides if the sensor node s_i stores each sensed value of the message m in the receipt queue RQ_i by using the following function:

$$\delta(D_i, D_j) = \begin{cases} 1 & \text{if } D_j - D_i \geqslant 1. \\ 1 & \text{if } D_j = D_i \text{ and } rand(1) = 0. \\ 1 & \text{if } D_i > D_j \text{ and } rand(\lceil D_i - D_j + 1\rceil) = 0. \\ 0 & \text{if } D_i > D_j \text{ and } rand(\lceil D_i - D_j + 1\rceil) \neq 0. \end{cases} \tag{1}$$

Here, the function $rand(x)$ gives a random number from 0 to $2^x - 1$. If a sensor node s_i receives a sensed value v whose initial sensor node is s_j, the sensor node s_i stores the sensed value v in the receipt queue RQ_i only if $\delta(D_i, D_j)$ = 1 where D_i and D_j are distances δ_i and δ_j of the sensor nodes s_i and s_j from the actuator node a, respectively.

A sensor node s_i receives a message m from another sensor node s_j as follows:

```
[SN-SN-rec($m, s_j$)] {
    if $\delta(D_i, m.dist(= D_j))$ = 1,
      $enqueue(RQ_i, \langle m.src, m.seq, m.val, OFF, m.dist\rangle)$;
      for each backup tuple $d$ in $m.data$ {
        $s_k := d.src$;
        $end := False$;
        for every tuple $t$ in $RQ_i$ where
          $t.src = s_k$ and $t.seq \leqslant d.seq$ {
          $t.state := ON$ if $d.state = ON$;
          $end := True$;
          }
        if $end = False$,
          if $\delta(D_i, d.dist)$ = 1,
            $enqueue(RQ_i, \langle d.src, d.seq, d.val, OFF, d.dist\rangle)$;
```

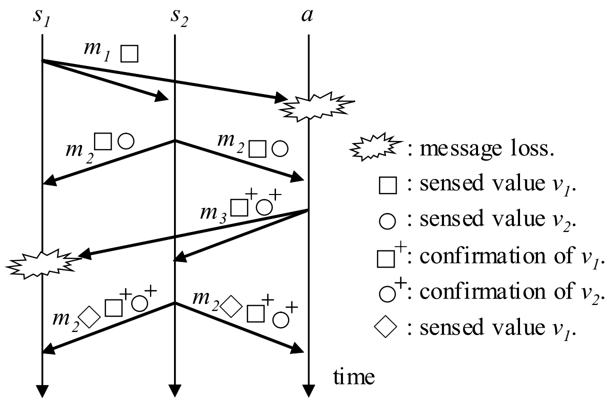


Fig. 2. Forwarding values.

```
      }
    }
  }
```

Suppose that a sensor node s_i receives a message m from another sensor node s_j. If the sensor node s_j is one hop farther from the actuator node a from the sensor node s_i, the sensor node s_i forwards the sensed value $m.val$ by broadcasting a message including the value $m.val$. If the sensor node s_j is at the same distance as the sensor node s_i, the sensor node s_i discards the value $m.val$ with probability 1/2. If the sensor node s_j is one hop nearer to the actuator node a, the sensor node s_i discards the value $m.val$ with probability 1/4. A fewer number of message are sent out to sensor nodes further from the actuator node a.

The receipt queue RQ_i can include at most M tuples. If the receipt queue RQ_i is full, one tuple is removed from the receipt queue RQ_i to make a space to store a new tuple. Here, a confirmed tuple is selected and removed in the receipt queue RQ_i. If the receipt queue RQ_i is still full, the top, i.e. oldest tuple of the receipt queue RQ_i is removed.

On receipt of sensed values, an actuator node a sends confirmation messages of the sensed values to sensor nodes. If a sensor node s_i receives a confirmation message m from the actuator node a, the following receipt procedure **SN-AC-rec** is performed in the sensor node s_i:

```
[SN-AC-rec($m, a$)] {
    for $i = 1, \cdots, A,$ {
      $b := m.ack_i$;
      for every tuple $t$ in $RQ_i$ where $t = src$,
      if $t.seq \leqslant b.seq, \quad t.state := ON$;
      }
  }
```

There are the variables SEQ_i, V_i, and S_i, for each sensor node s_i to send a confirmation message in the actuator node a $(i = 1, \cdots, m)$. The variable SEQ_i shows sequence number of a message which the actuator node a expects to receive next from a sensor node s_i. The variable V_i holds a value sensed by the sensor node s_i. The variable S_i is ON if the actuator node a had sent the receipt *confirmation* for the sensed value in the variable V_i to the sensor node s_i. Otherwise, the variable S_i is OFF.

An actuator node a receives a message m from a sensor node s_i as follows:

```
[AC-rec($m, s_i$)] {
    if $SEQ_i = m.seq$, {
      $SEQ_i := SEQ_i + 1$;
      $V_i := m.val$; /*sensed value from a sensor node $s_i$*/
      $S_i := OFF$;  /*a sensed value $V_i$ is not confirmed*/
      for each backup tuple $d$ in $m.data$,
      if $s_k = d.src$ and $SEQ_k = d.seq$, {
        $SEQ_k = SEQ_k + 1$;
        $V_k := d.val$;
        $S_k := OFF$;
        }
      }
    }
```

Sensed values from sensor nodes $s_1, \cdots, s_n$ are stored in a tuple $V = \langle V_1, \cdots, V_n \rangle$. An actuator node a sends a confirmation message m of sensed values in the tuple V to sensor nodes.

3. Evaluation

3.1. Assumptions

In the RT protocol, a message m sent by a sensor node s_i carries not only its sensed value $m.val$ but also to backup data $m.data$, i.e. sensed values in another message m' which the sensor node s_i has received before sending the message m. Even if a sensor node s_j loses a message m', the sensor node s_j can receive sensed values in the lost message m' if the sensor node s_j receives a backup message m of the last message m' from the sensor node s_i. Hence, it is significant to measure how much sensed values are lost. The *data loss ratio* (DL) is defined to be a ratio of the number of sensed values which an actuator node a loses to the total number of sensed values which the initial sensor nodes send. In traditional protocols, the data loss ratio DL is equal to the message loss ratio. In the RT protocol, the data loss ratio DL is smaller than the message loss ratio. Each sensed value is transmitted in multiple messages. The data *redundancy* (RD) of a sensed value v is defined to show how many times a message including the sensed value v is transmitted. The data redundancy RD shows how much bandwidth is spent to forward sensed values. Another point is how long it takes to deliver a sensed value to an actuator node a. We measure the minimum number of hops (mH) which a sensed value takes to deliver to the actuator node a. We evaluate the RT protocol in terms of data loss ratio (DL), data redundancy (RD), and minimum hop number (mH).

We make the following assumptions on the evaluation:

1. There are n sensor nodes $s_1, \cdots, s_n (n \geqslant 1)$ and one actuator node a. The nodes are located in an $m \times m$ mesh where the actuator node a is in the center. The sensor $s_1, \ldots, s_2$ are deployed in the $m \times m$ mesh. Figure 3 shows a 5×5 mesh with 24 sensor nodes. Here, $n = m^2 - 1$ and m is an odd number. In this paper, $n =$ 24 and 48 for $m =$ 5 and 7, respectively. Let f be the mesh size [m]. Here, $0 < f \leqslant 60$ [m]. The nodes are interconnected in a wireless channel. Let g show the mesh interval $g = f/(m-1)$.

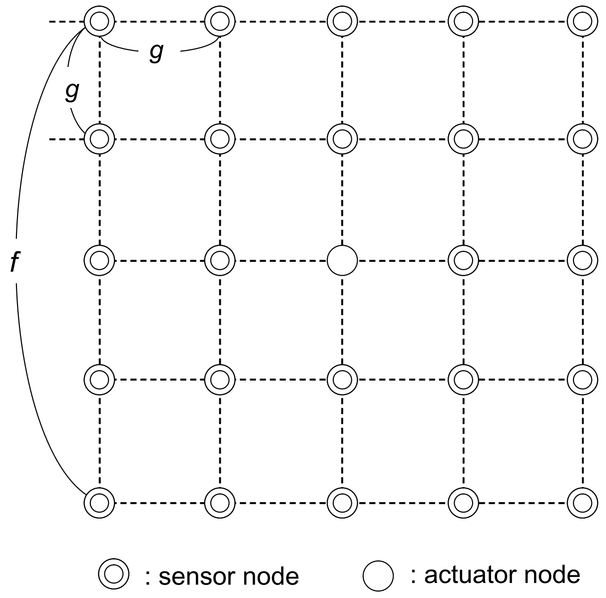


Fig. 3. Mesh structure ($n = 24$).

2. A sensor node s_i is realized in a MICA Mote [5]. The MICA Mote is a second generation mote module used for research and development of low power, wireless, sensor networks. The MICA mote was developed by UC Berkeley's research group on wireless sensors. It runs an operating system called Tiny OS. TinyOS is a small, open-source, energy efficient, software operating system developed by UC Berkeley which supports large scale, self-configuring sensor networks.
3. The transmission interval of each sensor node s_i follows the normal distribution of average τ [msec] and variance σ^2, i.e. $N(\tau, \sigma^2)$. Here, $\tau = 500$ [msec] and $\sigma^2 = 7500$. Suppose that a sensor node s_i sends a message m_2 after sending a message m_1. The probabilities that the sensor node s_i sends the message m_2 200, 500, and 900 [msec] after the message m_1 are 0.00114, 0.461, and 0.0000107, respectively.
4. Each of the sensor nodes $s_1, \ldots, s_n$ and actuator node a takes the CSMA [9] synchronization scheme against message collision in a wireless channel. Here, each node first listens to the wireless channel before transmitting a message. If the wireless channel is idle, the node start to send the message. In the CSMA/CD scheme [9,3,16], each node still listen to the wireless channel while transmitting the message. If the collision is detected, each node retransmits the message. Since each sensor node is too low-cost, low-power device to realize the CSMA/CD scheme, each node takes the simple synchronization scheme like CSMA.
5. Each message m sent by a sensor node s_i includes the number K of *backup* tuples $(K = 0, 1, \cdots, M)$. It is noted that "$K = 0$" shows the CSMA protocol, i.e. every message carries just one sensed value and the data loss ratio (DL) means the message loss ratio (ML). "$K \geqslant 1$" means the RT protocol, i.e. each message carries some sensed value in addition to its sensed value. The size of a sensed value (val) is two bytes long and the sensor identifier (id) is also two bytes long. That is, the length of one backup tuple is four bytes. Each sensor message carries its sensed value and additional K backup tuples. Each MICA message [5] has a header of thirteen bytes. Here, the length of a sensor message is $17 + 4K$ [byte].

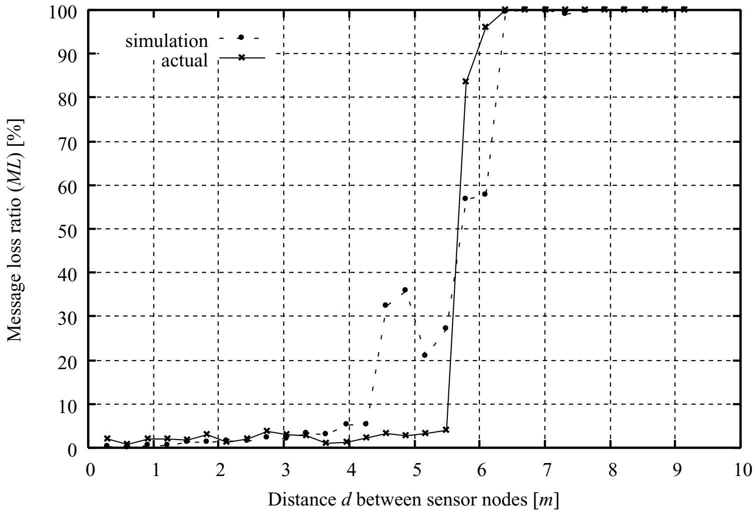


Fig. 4. Message loss ratio due to noise.

6. Each sensor node s_i can buffer at most five backup tuples of sensor messages in the receipt queue $RQ_i, M = 5$. On arrival of a message m, the oldest tuple is removed in the receipt queue RQ_i to store the information of the message m if the receipt queue RQ_i is full.

The RT protocol is realized in the nesC language [7] on the TinyOS operating system [8] of the Mica Mote sensor node. The RT protocol is simulated in the simulator TOSSIM [11] which takes usage of the CSMA scheme [9] as the basic message transmission protocol of a wireless network.

3.2. Noise and collision

First, we measure how many messages are lost with respect to the distance among a pair of Mica Mote sensor nodes [5] due to noise. Suppose a sensor node s_i sends a message m to another sensor node s_j which is d [m] from the source sensor node s_i in a wireless network. Figure 4 shows the message loss ratio (ML) for distance d, which is measured in actual sensor nodes and in the simulation. If $d < 5$ [m], 2.0 to 5.0% of the messages are lost due to noise. If $d \geqslant 5$, the message loss ratio ML is drastically increased.

Secondly, we measure how the message loss ratio (ML) changes for size K of a message. We consider three types of messages whose lengths are 17 ($K = 0$), 53 ($K = 9$), and 101 ($K = 21$) bytes. Figure 5 shows the message loss ratio ML for inter-node distance d in the simulation. If $d \leqslant 3$, the message loss ratio ML is smaller than 5% even if about 21 sensed values are included in a message. Here, the message loss ratio ML is independent of K, i.e. the length of message. This means that we can include more than one sensed value in a message without increase of message loss ratio ML.

3.3. Minimum hop number

In the RT protocol, sensed values can be delivered to the actuator node a even if some messages with the sensed values are lost. If the sensor node s_1 is within five meters of the actuator node a, the sensed value v can be directly delivered to the actuator node a as shown in Fig. 5. Here, it takes one hop. Even if a message with the sensed value v is lost by an actuator node a, another sensor node s_j which receives

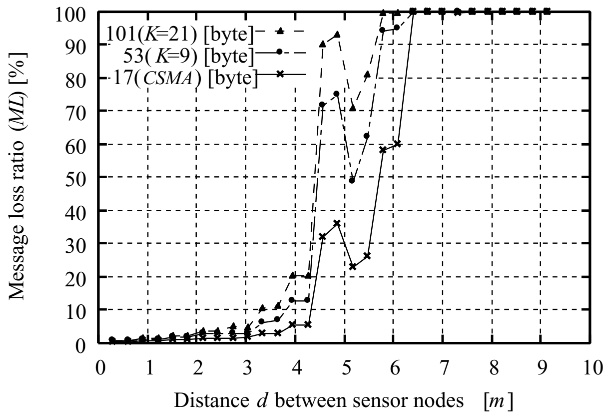


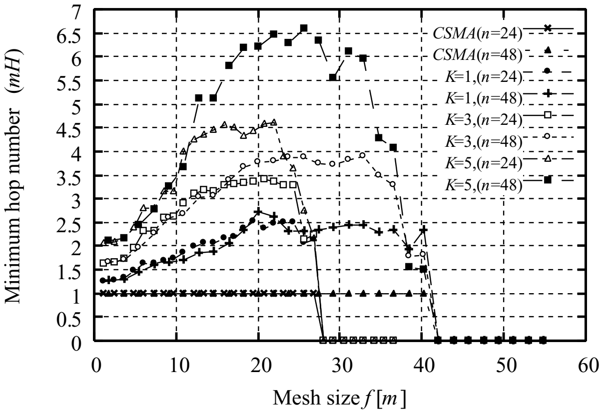
Fig. 5. Message loss ratio with backup tuples.

the sensed value v may forward the sensed value v to the actuator node a. Here, it takes two hops. Thus, it takes some number of hops to deliver a sensed value to an actuator node a. The larger the mesh interval g is, the more number of hops it takes. An actuator node a thus may receive the sensed value v from multiple sensor nodes. Each instance of the sensed value v takes different number of hops. Here, for each sensed value v which the actuator node a receives, the *minimum hop number* (mH) is taken out of all the instances of the sensed value v which the actuator node a receives. Here, each sensor node sends messages according to the assumption 2 for 50 seconds. Each sensor node sends 100 messages on average. It is noted that sensed values any of whose instances the actuator node a does not receive are not considered for the minimum hop number mH. Figure 6 shows the average minimum hop number mH for mesh size f $(0 < f < 60)$ where $n = 24$, 48. For $K = 0$ and $n = 24$, $mH = 1$ if $f \leqslant$ 26 [m]. Because every sensed value can be delivered to the actuator node a with one hop. If $K \geqslant 1$, the larger K is, the larger mH is. For example, $mH = 2.0$, 2.5, 3.1, 3.6, and 4.2 for $K = 1$, 2, 3, 4 and 5, respectively, with $n = 24$ and $f = 12$ [m]. For example, each sensed value is sent to the actuator node a in two messages on average for $K = 1$. In the longer mesh interval g, a sensor node can deliver messages to the smaller number of sensor nodes. Hence, the hop number is increased to deliver a sensor node to the actuator node. For example, $mH = 1.6$, 2.0, and 2.5 with mesh size $f = 8$, 12, and 20, respectively. For $n = 48$, the minimum hop number mH is 1.6, 2.5, and 2.7 with mesh size $f = 8$, 12, and 20, respectively. If $f > 28$ and $f \geqslant 40$, no message is delivered to the actuator node a since each sensor node s_i cannot send messages to any node which is six meters far from the sensor node s_i for $n = 24$ and 48, respectively, as shown in Fig. 4. Hence, the minimum hop number mH is zero. Figure 7 shows the minimum hop number mH for K where mesh size $f = 20$.

3.4. Data redundancy

In the RT protocol, a sensed value v sent by the source sensor node is carried in multiple messages. Since a source sensor node s_i of each sensed value v sends a message with the sensed value v in a wireless broadcast network, we calculate how many sensor nodes forward the sensed value v. The average number of sensor nodes which forward a sensed value v of the sensor node s_i is referred to as *data redundancy* (RD).

First, we do not take any mechanism to reduce the data redundancy. Figure 8 shows the data redundancy RD for the mesh size f where $n = 24$ and 48. Here, each sensor node send messages as explained in



* CSMA instead of K = 0.

Fig. 6. Minimum hop number (n = 24, 48).

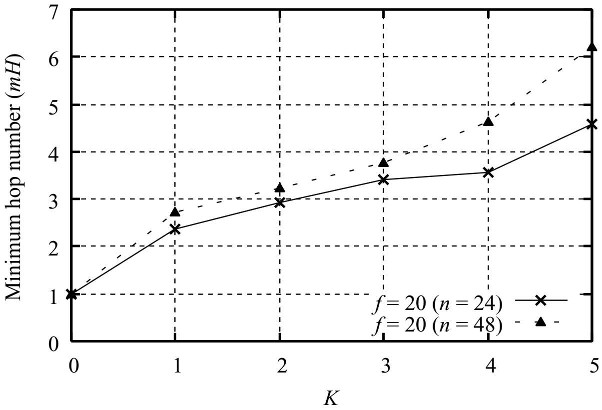


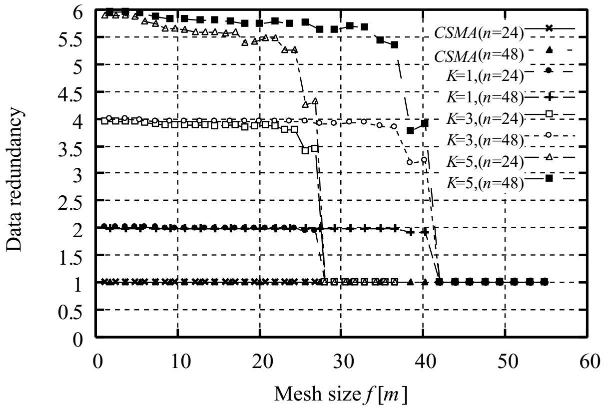
Fig. 7. Minimum hop number vs. K (n = 24, 48).

the preceding subsection. The larger K is, the higher the data redundancy RD is. For example, the data redundancy RD is 1, 2, 3, 3.8, and 4.8 for K = 0, 1, 2, 3, and 4, respectively, with f = 12 and n = 24. Three nodes send messages with each sensed value for K = 2 on average. The longer the mesh size f is, the smaller the data redundancy RD is. The longer the mesh size f is, the more number of messages are lost. Hence, some sensed values are lost since no sensor node receives the messages.

Next, we consider the redundancy reduction (RR) way as discussed in this paper. On receipt of a sensed value v from another sensor node s_{j-1} a sensor node s_i forwards the sensed value v with higher possibility if the sensor node s_i is nearer to the actuator node a than the sensor node s_j. Otherwise, the sensor node s_i may not forward the sensed value v. Figure 9 shows the data redundancy RD with n = 24. For example, about 10% of data redundancy can be reduced for K = 5, and f = 20 (g = 5).

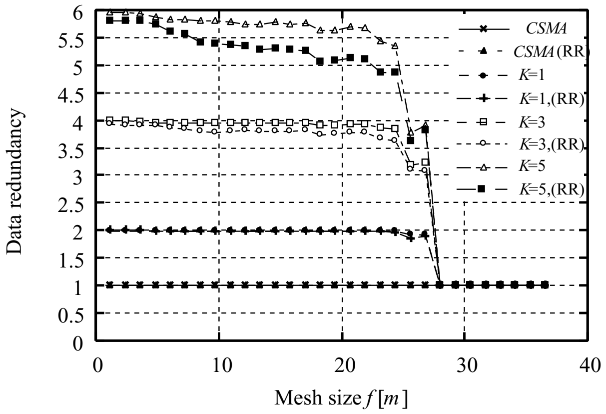
3.5. Data loss ratio

Finally, we measure the data loss ratio (DL) for n = 24 and 48. Here, each sensor node s_i sends messages according to the assumption 2. The data loss ratio DL for mesh size f is shown in Fig. 10



* CSMA instead of K = 0.

Fig. 8. Redundancy (n = 24, 48).

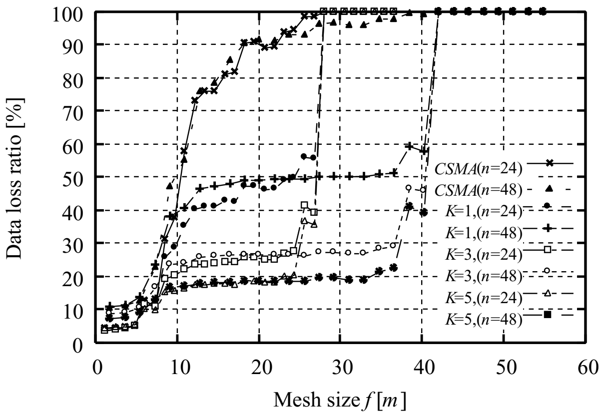


* CSMA instead of K = 0.

Fig. 9. Redundancy (n = 24).

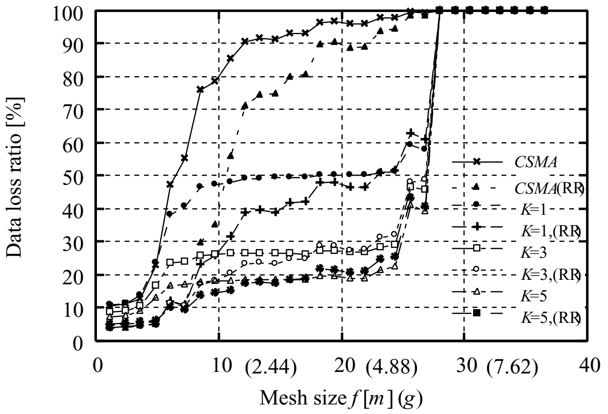
where the redundancy reduction (RR) scheme is not taken. Figures 12–13 show the data loss ratio DL with the redundancy reduction (RR) scheme. Here, each sensor node sends messages as explained in Fig. 6. If mesh interval $g \leqslant 2$ [m], the data loss ratio DL is independent of K because every sensor node can deliver a message to the actuator node a with one hop. Here, messages are mainly lost due to collisions, since sensor nodes are closely deployed. Since most messages are lost due to collision, no sensor node receives the messages. Hence, no sensor node can forward sensed values to other sensor nodes. The data loss ratio DL is the same as message loss ratio (ML) and is independent of K.

Next, let us consider case the mesh interval $g > 2$ [m]. Some sensor node cannot deliver a message to the actuator node a with one hop. Here, even if some message m is not received by the actuator node a, some sensor node might receive the message m and forwards the sensed value in the message m to the actuator node a. Hence, the larger K is, the smaller the data loss ratio DL is as shown in Fig. 10. For example, about 72% and 80% of the sensed values are delivered to the actuator node a for $K = 3$ and $K = 4$ although more than 95% of messages are lost where $n = 24$ and $f = 24$ [m]. For $K = 2$ and 1, the data loss ratio DL is 35% and 50%, respectively, with the mesh size $f = 24$. In addition, the data



* CSMA instead of K = 0.

Fig. 10. Data loss ratio ($n = 24$, 48).



* CSMA instead of K = 0.

Fig. 11. Data loss ratio ($n = 24$, RR).

loss ratio DL is increased from 30% to 100% and 25% to 55% with respect to the mesh size f, 8 [m] to 28 [m] for $K = 0$ and 1, respectively. However, the data loss ratio DL is 25% to 35% for $K = 2$ and 20% to 28% for $K = 3$. The data loss ratio DL is not so much increased with f for $K = 2$ and 3 as $K = 0$ and 1. For $g > 7$, i.e. $f > 28$ and $f > 42$ for $n = 24$ and 48, respectively, each sensor node cannot deliver messages to even a nearest sensor node since every pair of sensor nodes are too far. No sensed value is delivered to the actuator node a. In the RT protocol with redundancy reduction [RR], the data loss ratio DL can be reduced.

Following Figs 10 to 13, the data loss ratio (DL) can be reduced to 20 [%] $\sim$ 50 [%] in the RT protocol. Here, messages are mainly lost due to noise and some sensor node receives a message even if an actuator node a could not receive the message. It is noted the total number of messages transmitted for $K = 0$, 1, 2, 3, and 4 is the same. Without increasing the number of messages, the more number of sensed values can be delivered to the actuator node a in the RT protocol. If sensor nodes are very closely deployed, the data loss ratio DL cannot be decreased since most messages are lost due to collision and there is no chance to forward sensed values in another message. The minimum hop number mH shows

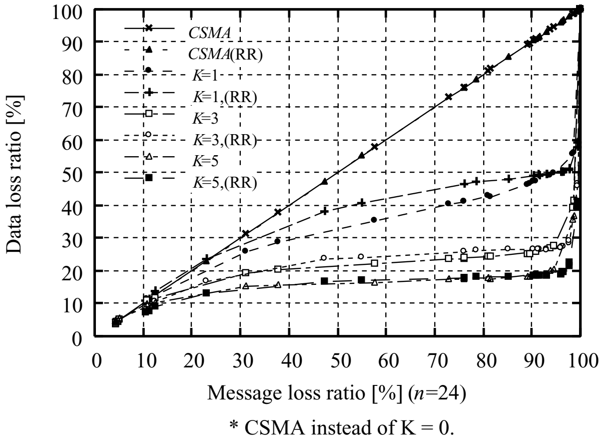


Fig. 12. Data loss ratio vs. message loss ratio.

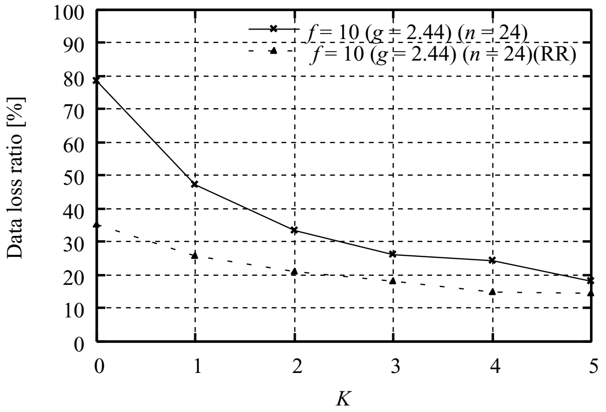


Fig. 13. Data loss ratio vs. K ($n = 24$).

the delivery time of a sensed value. For example, let us consider the mesh size f is 24 [m]. It takes 3.3 hops to deliver a sensed value to the actuator node a and 28% of sensed values are lost for $K = 3$ while it takes 2.5 hops and 50% of sensed values are lost for $K = 1$. For $K = 0$, the minimum hop number mH is 1 but 95% of sensed values are lost. There is thus tradeoff between the minimum hop number mH and the data loss ratio DL.

4. Concluding remarks

We discussed the redundant data transmission (RT) protocol in a wireless sensor actuator network (WSAN). Messages sent by sensor nodes are lost in a wireless network due to noise and collision. A message received by a sensor node is redundantly forwarded in another message sent by the sensor node in the RT protocol. Hence, even if some messages are lost, an actuator node can receive sensed values in the lost message if the actuator node receives other backup messages carrying the sensed values. It is noted that the number of transmitted messages is not increased in the RT protocol since messages are not retransmitted. In addition, we discussed how to reduce the data redundancy of sensed values transmitted by sensor nodes. On receipt of a sensed value v from another sensor node s_j, a sensor node s_i forwards the sensed value v if s_i is nearer to the actuator node than s_j. Otherwise, the sensor node s_i may not for ward the value v.

We evaluated the RT protocol where sensor nodes are deployed in the mesh structure. In the RT protocol, it is noted lost messages are not retransmitted since sensed values in the lost messages are carried by other messages. We showed that the data loss ratio, i.e. the loss ratio of sensed values can be reduced without retransmission of lost messages in the RT protocol. Only about 20% of sensed values are lost with $K = 4$ even if more than 95% of messages are lost. In a situation where 50% of messages are lost, only 22% of the sensed values are lost with $K = 3$.

Acknowledgment

We would like to thank Mr. Kiyohiro Morita, graduate student of Tokyo Denki University for helping us implement and evaluate the system.

References

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

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

[3] Y.-L. Chang and S. Shen, Design of priority schemes in csma/cd local are networks. *Proceedings of the 20th annual symposium on Simulation* (*ANSS'87*), 1987, 45–64.

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

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

[6] A. Durresi and V. Paruchuri. Geometric broadcast protocol for heterogeneous sensor networks. *Proc. of 19th IEEE International Conf. on Advanced Information Networking and Applications* (*AINA2005*), 2005, 1:343–348.

[7] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer and D. Culler, The nesc language: A holistic approach to networked embedded systems. *Proc. of the ACM SIGPLAN 2003 conference on Programming Language Design and Implementation* (*PLDI'03*), 2003, 1–11.

[8] C.-C. Han, R. Kumar, R. Shea, E. Kohler and M. Srivastava, A dynamic operating system for sensor nodes. *Proc. of the 3rd International Conference on Mobile systems, applications, and services*, 2005, 163–176.

[9] I. Joe and S.G. Batsell, Reservation csma/ca for multimedia traffic over mobile ad-hoc networks, *IEEE International Conference on Communications* **3** (2000), 1714–1718.

[10] K.H. Kim, K. Fujiwara, M.-C. Kim, L. Zheng, K. Watanabe and M. Takizawa, A ttf-based programming model and a support kernel running on a communicating sensor platform. *Proc. of the 8th International Symposium on Autonomous Decentralized Systems* (*ISADS2007*), 2007, 368–376.

[11] P. Levis, N. Lee, W. Welsh and D. Culler, Tossim: accurate and scalable simulation of entire tinyos applications. *Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, 126–137.

[12] K. Morita, K. Ozaki, K. Watanabe, N. Hayashibara, T. Enokido and M. Takizawa, Sensor-actuator communication protocols in wireless networks. *Proc. of the First International Conference on Network-based Information Systems* (*NBiS2007*), 2007, 11–19.

[13] K. Morita, K. Watanabe, N. Hayashibara, T. Enokido and M. Takizawa, Evaluation of reliable data transmission protocol in wireless sensor-actuator network. *Proc. of the 3rd IEEE International Workshop on Heterogeneous Wireless Sensor Networks* (*HWISE2007*), 2007, 713–718.

[14] V. Paruchuri, A. Durresi and L. Barolli, Energy aware routing protocol for heterogeneous wireless sensor networks. *Proc. of 16th International Workshop on Network-Based Information Systems* (*NBiS2005*), 2005, 133–137.

[15] V. Paruchuri, A. Durresi, L. Barolli and M. Takizawa, Three dimensional broadcast protocol for wireless networks. *Proc. of the 36th International Conference on Parallel Processing* (*ICPP2007*), 2007, 10–14.

[16] 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*, 2004, 2:966–970.

Ailixier Aikebaier was born in 1981. He received B.E degree in Computers and Systems Engineering from XinJiang University, China, in 2000. He is currently a graduate school student in the Department of Computers and Systems Engineering, Tokyo Denki University. His research interests include distributed systems, P2P network, consensus problems, and fault-tolerant systems.

Tomoya Enokido was born in 1974. He received B.E. and M.E. degrees in Computers and Systems Engineering from Tokyo Denki University, Japan, in 1997 and 1999. After he worked for NTT Data Corporation, he joined Tokyo Denki University in 2002. He received his D.E. degree from Tokyo Denki University in 2003. He is currently a lecturer in Faculty of Business Administration, Rissho University. He won the best paper awards of IEEE AINA 2004 and 2005. He worked for program co-chairs of ICPADS, MNSA2005, and UISW2005. His research interests include distributed systems, group communication, distributed objects, fault-tolerant systems, and distributed transaction management. He is a member of IEEE and IPSJ.

Makoto Takizawa was born in Tokyo, Japan, in 1950. He is currently a professor in the Department of Computer and Information Science, Seikei University since April of 2008. He was a visiting professor at GM-IASI (currently Fraunhofer), Germany (1989–1990) and at Keeled University, England (1990). He is also a visiting professor of Indian University, Xi'an, China since 2004. He joined JAPED (Japan Information Processing Center) in 1975 and developed a heterogeneous distributed system JUDD'S. He joined Tokyo Denki University in 1986. He is a Board of Governors (BOG) from 2003 to 2008 and a Golden Core member of IEEE Computer Society. He is a fellow of Information Processing Society of Japan (IMPS). He was a member of executive board of IMPS from 1999 to 2001 and chaired a SIDES of IMPS from 1997 to 2000. He was a general co-chair and a program co-chair of IEEE ICDCS (International Conference on Distributed Computing Systems) in 2002 and 1998, respectively. He also chairs many international conferences like IEEE ISCOR, MIPADS, and DEXA. He founded IEEE International Conference on Advanced Information Networking and Applications (AINA). He won the best paper awards at ICON-9 (1994), ICON-18 (1998), MIPADS (1996), DAMS (2001), ICPADS2005 (2005), BOCA (2007), and CRISIS(2008), and excellent paper award at AINA-2004. His research interests include distributed systems, group communications, fault-tolerant systems, and information security.


Advances in Multimedia
The Scientific World Journal
International Journal of Distributed Sensor Networks
Journal of Industrial Engineering
Applied Computational Intelligence and Soft Computing
Advances in Fuzzy Systems
Modelling & Simulation in Engineering
Journal of Computer Networks and Communications
Advances in Artificial Intelligence
Advances in Computer Engineering
International Journal of Computer Games Technology
International Journal of Biomedical Imaging
Advances in Artificial Neural Systems
Advances in Software Engineering
Journal of Robotics
Advances in Human-Computer Interaction
Computational Intelligence and Neuroscience
International Journal of Reconfigurable Computing
Journal of Electrical and Computer Engineering