Design Methodology of a Wireless Sensor Network Architecture for Urgent Information Transmission

Tetsuya Kawai, Naoki Wakamiya, and Masayuki Murata
Graduate School of Information Science and Technology
Osaka University, Japan
{t-kawai, wakamiya, murata}@ist.osaka-u.ac.jp

ABSTRACT

Wireless sensor networks are expected to become an important social infrastructure which helps our life to be safe, secure, and comfortable. In this paper, we propose design methodology of a network architecture for fast and reliable transmission of urgent information in wireless sensor networks. In the methodology, several simple and fully-distributed control mechanisms which function in different spatial and temporal levels are introduced to each node to offer differentiated transmission to packets in accordance with their importance. We also show an example of a network architecture designed following the methodology. Our practical experiments showed that the architecture successfully improved the delivery ratio and delay of the urgent sensor information.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

Design, Performance, Implementation

Keywords

Sensor networks, Urgent information, Fastness, Reliability

1. INTRODUCTION

Wireless Sensor Network (WSN) technology is expected to play an essential role for our society in the near future. A WSN consists of a number of sensor nodes and a base station (BS). A node is equipped with a processing unit, a radio transceiver, and sensors. Nodes are deployed in a region to monitor and then environmental information detected by sensors is collected to a BS through wireless communication among sensor nodes [1].

WSN technology will be used for a wide variety of applications, such as agricultural, health, environmental, and industrial purposes. Among them, a WSN used as a social infrastructure to make our life safe, secure, and comfortable is one of the most promising. This sort of WSNs is supposed to carry various types of information, such as temperature, humidity, fire alarm, intrusion warning, seismic movement, image, and sound. The urgent information, a fire alarm for example, has to be transmitted through a WSN with higher reliability and lower latency than other non-urgent information. Since the capacity of a wireless network is limited, a WSN must be capable of differentiating and prioritizing packets depending on their urgency and importance of embedded sensor information, which are defined by an application. Furthermore, in the event of a large emergency, such as an earthquake attack, a lot of nodes detect the emergency and send urgent information at the same time. A WSN should adopt a network-level mechanism to mitigate serious congestion caused by this simultaneous packet emission.

There have been several proposals of QoS (Quality of Service) control [3, 19, 12] for WSNs. In ReInForM [4], each node stochastically relays received packets according to the different forwarding probability, and the reliability of transmission is improved through a multipath and retransmission mechanism. ESRT [15] regulates the emission rate at source nodes with feedback from a BS to accomplish the desired reliability. In [5], overheard packets are used for error correction in single-hop and multihop routing to improve the reliability. In [7], the authors propose a routing protocol to find the best path in terms of delay. Felemban *et al.* [6] expands this idea to provide QoS differentiation both in reliability and in latency using a multipath technique. For congestion mitigation in a WSN, CODA [17] combines hop-by-hop backpressure and end-to-end feedback techniques. In [8], a prioritized MAC protocol is introduced in addition to a hop-by-hop traffic control. IFRC [14] is developed to realise adaptive fair and efficient rate allocation by sharing information on the level of congestion among nodes.

The main contribution of this paper is proposal of design methodology of a network architecture for transmission of urgent sensor information. In comparison to the research works mentioned above, our approach is unique in that several simple mechanisms are incorporated above the network layer. In order to adapt to the scale of an emergency ranging from a small event like a gas leakage to a catastrophic event such as an earthquake attack, some simple mechanisms are embedded in each node, instead of a monolithic and complicated mechanism which controls its overall behavior. Those mechanisms work in different spatial and temporal levels and they autonomously and independently react to

WICON 2007, October 22-24, Austin, USA
Copyright © 2007 978-963-9799-12-7
DOI 10.4108/pwsn.2007.2265

its surrounding situation locally observed. When a small event happens, it is not necessary to involve all nodes to respond to the emergency. Instead, only nodes along the path from the node which detects the event to a BS participate in transmission of urgent information and adopt a hop-by-hop retransmission mechanism for example. As the scale of the emergency grows over time, additional mechanisms come into effect and more nodes become involved in the urgent information transmission. On the other hand, in the event of a large emergency, a lot of nodes detect the emergency at the same time and send urgent information simultaneously and independently from others. A WSN in this case immediately reacts as a whole to control serious congestion and ensures fast and reliable transmission of urgent information.

We also show a network architecture called UMIUSI (aUtonomous Mechanisms Integrated for Urgent Sensor Information), which incorporates five simple mechanisms following the above methodology.

The rest of this paper is organized as follows. Section 2 proposes design methodology for fast and reliable transmission of urgent information and introduces our UMIUSI architecture. Evaluation of the architecture by practical experiments is presented in Section 3. Finally, we conclude this paper in Section 4.

2. PROPOSED METHODOLOGY AND ARCHITECTURE

2.1 System Overview

In this paper, we consider a WSN deployed in a building or a house to monitor and control a living and working environment. A WSN consists of one BS and a number of immobile sensor nodes. The BS corresponds to a gateway server or a home server with power supply and sends sensor information to a monitoring station of a security company or an administration center. Sensor nodes are operated on a battery and equipped with a variety of sensors. Normally, each node observes its environment and reports obtained sensor information to the BS at regular intervals. A way that sensor information is transferred to the BS depends on a routing protocol or a data gathering mechanism employed. To save its power consumption, a sensor node sleeps and wakes up in accordance with a sleep scheduling algorithm such as that in [2]. Once a node detects an emergency or an unusual condition, it begins to emit packets containing urgent information at shorter intervals. On receiving the urgent information, the BS reports the emergency to the monitoring station through a regional network. Then, an acknowledgement is sent back to the BS. The ACK is forwarded to the source node of the urgent information and the node stops sending urgent information.

Although the methodology incorporates mechanisms above the network layer and does not depend on any specific MAC or routing protocols, we assume a contention-based MAC protocol and a multihop routing protocol. A time division multiple access (TDMA) protocol is also applicable, but we consider that it has many practical problems to be solved such as scheduling overhead and severe time synchronization requirements. As for the network layer, a multihop scheme with limitation on the radio transmission energy is usually preferred to avoid contention among wireless communication and prolong the lifetime of batteries.

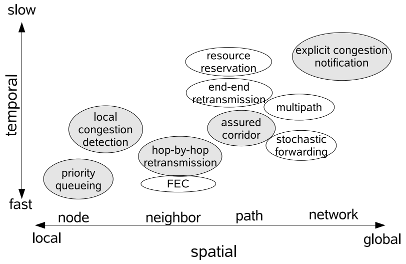

Figure 1: Examples of control mechanisms.

2.2 Design methodology

Since a large number of battery-driven nodes are deployed in a WSN, energy efficiency, fault tolerance, and scalability should be taken into account in designing a WSN architecture [1]. These factors need to be well considered also in such a WSN as is assumed in the previous subsection. However, in the event of emergency, urgent information must be transmitted as fast and reliable as possible, thus reliability and low latency are primary concerns. Therefore, we need a WSN architecture which satisfies requirements in both of normal and emergency conditions.

There have been a lot of excellent works on data gathering schemes which can be applied in normal situations, for example, [9]. We take an approach to incorporate mechanisms for urgent information transmission with any data gathering scheme well-designed for application-oriented communication. It means that a WSN operates on a data gathering scheme in the normal situation. Once an emergency occurs, an appropriate series of actions take place to deliver urgent information to the BS. Those nodes which are not involved in the emergency should keep their normal operation.

In summary, our design objectives of a WSN architecture for transmission of urgent sensor information are:

- *High reliability and low latency.* The reliability and latency of transmission of urgent information are the most important issues. Urgent information should be differentiated from other information and receive preferential controls according to their importance. We consider that energy efficiency can be sacrificed to some extent for transmission of urgent information.

- *Self-organizing and localized behavior.* The type and scale of an emergency and the number of simultaneous emergency events are unpredictable and dynamically change as time passes. A centralized architecture is infeasible in an emergency due to variations of traffic pattern and the level of congestion. Therefore, we need an architecture which is fully-distributed, self-organizing, and adaptive to dynamically changing conditions. As a consequence of localized reactions of each sensor node to the surroundings and local interactions among nodes, a globally-organized behavior of a WSN against detected emergencies emerges as a whole.

- *Simplicity.* Since a node has limited computational capacity and a small amount of memory, mechanisms to support fast and reliable transmission of urgent information must be simple enough. Simplicity also contributes to low energy consumption and less programming errors.

To satisfy the above requirements, we propose design methodology to combine several simple control mechanisms which function in different temporal and spatial levels. In Fig. 1, typical control mechanisms are arranged in accordance with their temporal and spatial effect. In general, larger the spatial area where a mechanism influences is, longer the time required to achieve effective control is. In the methodology, at least one mechanism is chosen for each of spatial levels. In our UMIUSI architecture, those five mechanisms indicated by gray circles are embedded in a sensor node.

When an emergency occurs, one or more mechanisms among them start working autonomously and independently as a reaction to the surrounding situation of a node. The collective behavior of these mechanisms offers appropriate preferential transmission of emergency packets. At the beginning of an event, quick-acting node-level and neighbor-level mechanisms contribute to reliable and fast transmission until slower path-level mechanisms come into effect. As the event develops and situation becomes more serious, additional mechanisms eventually become effective and network-level control is conducted.

We should note here that it is not possible to transmit all emergency packets with high reliability and low latency because the capacity of a WSN is limited. Therefore, it is necessary to classify sensor information into several classes in accordance with the required QoS in terms of delay and reliability. Classification and prioritization can be determined beforehand. Context-aware prioritization is also helpful to adapt to dynamically changing emergency conditions. Each packet has a field in its header to indicate its corresponding class and packets in different classes are treated in a different way in a WSN.

2.3 UMIUSI Architecture

As an example of a network architecture designed following our methodology, we propose the UMIUSI architecture. We consider three classes of sensor information as one normal class and two emergency classes and prioritize emergency class information over normal class information. The two types of urgent information are distinguished in more important and less important.

- *Normal Class.* Any non-urgent information belongs to this class. Normal class information is gathered to a BS at regular intervals of t_{norm}. Without an emergency, the latency and reliability of normal class information should satisfy application's requirements by the adopted data gathering scheme. An application can tolerate delay and loss of normal class information under emergency conditions. Packets of this class are called normal packets.

- *Important Class.* This class is for urgent information, but an application can tolerate loss and delay of important class information to some extent. Packets belonging to this class, called important packets, can be delayed or dropped depending on the level of congestion in an emergency. The interval of emission of important packets $t_{\mathrm{imp}} < t_{\mathrm{norm}}$ is determined by an application, but could be regulated to be larger than t_{norm} to mitigate congestion.

- *Critical Class.* This class is for the most urgent and important information which requires highly reliable and fast transmission to the BS. Critical packets are emitted by a node detecting an emergency at fixed regular intervals of $t_{\mathrm{cri}} < t_{\mathrm{norm}}$, which is determined by an application. The total amount of critical class traffic should not exceed the network capacity to guarantee a high delivery ratio and low delay of the required level. Therefore, the number of sensor nodes for critical information should be limited at the deployment, or some of them should be categorized into the important class.

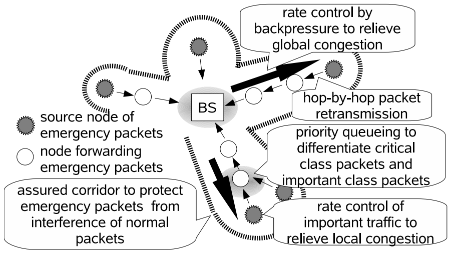
rate control by backpressure to relieve global congestion
BS
hop-by-hop packet retransmission
source node of emergency packets
node forwarding emergency packets
priority queueing to differentiate critical class packets and important class packets
assured corridor to protect emergency packets from interference of normal packets
rate control of important traffic to relieve local congestion

Figure 2: The mechanisms leveraged in UMIUSI.

Following the methodology, we choose five mechanisms for UMIUSI: priority queueing, rate control by local congestion detection, retransmission, assured corridor mechanism (ACM) [11], and rate control by backpressure. These mechanisms are simple and distributed, work independently of each other, and cover all the levels from node-level to network-level. Figure 2 briefly summarizes how and where they work. Although another combination with other mechanisms is also possible, we must carefully consider the relation among mechanisms. For example, we can also introduce in-network aggregation [13] for reducing the number of packets and thus suppressing congestion occurrence. However, adopting both hop-by-hop and end-to-end retransmission mechanisms is likely to be inefficient or redundant.

The detailed description of each mechanism is presented in the following.

2.3.1 Assured corridor mechanism (ACM)

The main purpose of this mechanism is to avoid loss of emergency packets caused by collisions with normal packets. In addition, ACM contributes to avoiding delay caused by sleeping nodes. ACM establishes an "assured corridor" from a source node to a BS, in which emergency packets are protected from normal packets.

An assured corridor consists of awake nodes, which is on the path from a source node to a BS, and surrounding silent nodes, which are in the range of radio signals of the awake nodes. In normal operation, all nodes are in the *NORMAL* state and operate in accordance with a data gathering scheme. Once a node detects an emergency, it moves to the *EMG_SEND* state and begins to periodically emit packets labeled as critical or important. On receiving an emergency packet for the first time, other node moves to either of the *SUPPRESSED* or *EMG_FORWARD* state. A node on the path to the BS is responsible for forwarding emergency packets to the BS. Therefore, it moves to the *EMG_FORWARD* state, cancels its sleep schedule to keep awake, and immediately relays emergency packets it receives. A node which receives an emergency packet but is not on the path moves to the *SUPPRESSED* state. A node in the *SUPPRESSED* state completely stops sending normal packets or decreases

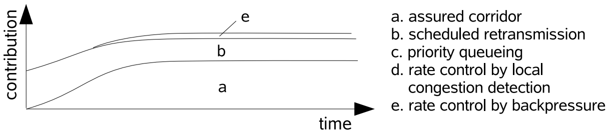

Figure 3: The contribution of each mechanism in a small scale event.

the sending rate of normal packets. Details of ACM with simulation results can be found in [11].

2.3.2 Retransmission

In order to recover a lost emergency packet while providing differentiated services, we introduce a prioritized scheduling algorithm to hop-by-hop retransmission. The algorithm can be incorporated with most retransmission mechanisms in either MAC layer or network layer, such as that in [16].

When a packet is lost, the first retransmission is scheduled after a backoff. To prioritize retransmission of a critical packet, the backoff timer for a critical packet is set shorter than that for an important packet, at 0.1 and 0.2 seconds respectively in our experiments. If the first retransmission fails, one or more trials are conducted by applying doubled backoff, *i.e.*, a binary backoff scheme, until retransmission succeeds or a next-hop node goes to sleep.

An emergency packet is discarded at a node when it receives the next emergency packet originating at the same source node. This is because that sensor data in the waiting packet is obsoleted by the new data. It is also possible to merge them and generate a new emergency packet.

2.3.3 Priority queueing

Each node has a priority queue for emergency packets, with which important packets are served only when there is no critical packet queued. This means that fast transmission of critical packets is accomplished at the sacrifice of longer transmission delay of important packets. Data aggregation such that proposed in [12] can be used to avoid this delay. Transmission of normal packets at a node during an emergency is delayed until the node returns to normal operation.

2.3.4 Rate control by local congestion detection

To mitigate congestion as fast as possible by local control, we introduce a rate control mechanism which is triggered by detection of local congestion. In order to keep the reporting rate of critical information at $1/t_{\mathrm{cri}}$, the rate control is applied only to important class traffic. When a source node of important packets detects congestion by, for example, monitoring packet reception rate, it increases the emission interval of important packets. Congestion detection can be done by other methods, including observation of queue occupancy and channel sampling such as proposed in [17, 8]. As a rate control algorithm, a TCP-like AIMD (Additive Increase and Multiplicative Decrease) algorithm, such as that in [18], is empirically employed in our experiments.

2.3.5 Rate control by backpressure

In an event of large emergency such as an earthquake, even if emission of normal packets is suppressed and source nodes of important packets regulate their emission rate, congestion cannot be fully avoided around a node belonging to multiple paths and around the BS, where many emergency packets concentrate on. We employ a backpressure mechanism for a network-level traffic control in UMIUSI.

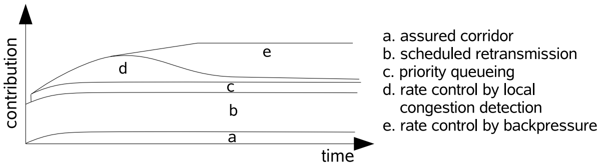

Figure 4: The contribution of each mechanism in a large scale event.

To suppress emission of important packets at their source nodes, a backpressure message is sent back to source nodes from a point of congestion by piggybacking on an emergency packet. Details are as follows. When a node detects congestion, it sets an explicit congestion notification (ECN) bit in the header of important packets which it relays toward the BS. By overhearing the packet, a node closer to the source recognizes that congestion occurs in the path to the BS. Then, it also sets an ECN bit of the following important packet which it relays to the next-hop node. Consequently, by means of overhearing, a congestion notification propagates to the source node. On receiving the notification, the source node reduces the emission rate of important packets, and the congestion is mitigated. The node which detected the congestion stops setting ECN bits once it confirms the mitigation.

2.4 Discussion

By integrating the five mechanisms, UMIUSI adapts to a variety of emergencies. With a small emergency event, only one or a few nodes would detect it. At the beginning of the event, the retransmission mechanism becomes effective immediately since an emergency packet is not protected from normal packets until an assured corridor is established. Once a corridor is established, the contribution of the retransmission mechanism becomes smaller as the number of packet loss is reduced by ACM. The priority queueing mechanism is not used since all emergency packets are likely to belong to one class. In addition, rate control of important class traffic by local congestion detection does not help much, since the number of nodes emitting important packets is small and the possibility of congestion is expected to be small. If multiple paths are established from a source node to a BS, there may occur collisions among emergency packets traversing different paths. In such a case, the rate control mechanism by backpressure is activated. Figure 3 is an intuitive sketch to show how much each mechanism contributes to fast and reliable transmission of emergency packets in a small scale event.

On the other hand, for a large scale emergency, many nodes become a source of emergency packets. Figure 4 illustrates the degree of contribution of the mechanisms for the case of large emergencies. Since most of nodes are involved in transmission of emergency packets as source nodes or forwarding nodes, an assured corridor to suppress the emission of normal packet does not help much. On the contrary, mechanisms to mitigate congestion within a corridor are effective. The priority queueing mechanism offers differentiated forwarding services to emergency packets in accor-

Table 1: The power consumption of nodes during 10 minutes. (mAh)

State	Node 1	Node 2	Node 3
NORMAL	8.786	8.814	8.759
EMG_FORWARD	8.795	8.834	8.778
EMG_SEND	8.795	8.836	8.779
SUPPRESSED	8.804	8.849	8.783

dance with their class. Rate control is first applied locally at a source node to mitigate local congestion among neighboring source nodes. Afterwards, congestion among emergency packets traversing different paths is solved by the backpressure mechanism.

All of these reactions against different types of emergency emerge as a consequence of autonomous and simple behavior of nodes. There is neither a mechanism to identify the type and scale of an emergency nor an explicit rule to choose and coordinate mechanisms.

3. PRACTICAL EXPERIMENTS

We implemented UMIUSI onto off-the-shelf sensor nodes provided by OKI Electric Industries Co., Ltd. and conducted experiments using two testbeds. Testbed A consisted of 26 nodes including a BS which were put in a 10 m × 6 m room. In Testbed B, 47 nodes were deployed over a floor of a building for feasibility demonstration. Before the practical experiments, we verified basic behavior of UMIUSI by thorough simulation experiments, but results are not shown in this paper for space limitation.

In the experiments, IEEE 802.15.4 non-beacon mode was employed for the MAC layer. The payload size of an emergency packets was 16 bytes including packet header with a class identifier, dummy sensor data, and a time stamp. For the network layer, we adopted the synchronization-based data gathering scheme [10] modified to ignore uni-directional links. It employs a tree-based routing, and timing of packet emission is the same among nodes of the same hop distance from the BS. In a normal state, all nodes adopt a sleep schedule. Nodes on the same hop distance wake up at the same time and receive packets from one-hop distant node. Then, they send packets to next-hop nodes. Finally, after overhearing packets emitted by the next-hop nodes, they go back to a sleep mode. In the experiments, we set the interval of normal packet emission t_{norm} at ten seconds, and the offset between emissions of adjacent nodes at one second. Routes from nodes to the BS dynamically changed for variations in radio environment.

For evaluation of transmission of emergency packets, we made one (small scale event) or eight (large scale event) nodes become source nodes, *i.e.*, with ACM, they moved to the *EMG_SEND* state. Each of them was scheduled to emit emergency packets at interval of $t_{\mathrm{emg}} = 0.5$ seconds, but the actual interval was about 0.58 seconds due to implementation constraints. Source nodes went back to normal operation with the sleep schedule ten minutes later.

Table 1 summarises the amounts of power consumed by randomly chosen three nodes in ten minutes for each of the four states on Testbed A. It was measured by a digital power meter (Yokogawa Electric WT210) attached to a DC input of a node. We find that the difference in power consumption among four states is relatively small compared to the consumed power. Therefore the active / sleep ratio in normal operation would determine the lifetime of a node, although we did not apply any sleep schedule in the experiments due to implementation constraints. A node adopts three AAA alkaline cells with serial connection. One cell has the capacity of about 1 Ah. Thus, if we apply a sleep schedule of the active / sleep ratio of 1/600, *i.e.*, being active for one second in ten minutes, the lifetime of a node can be estimated as 11,400 hours (= 1.30 years). Once an emergency occurs and a node stays in either of *EMG_SEND*, *EMG_FORWARD* or *SUPPRESSED* states for three minutes for example, it shortens the lifetime of a node by 30 hours. Developing a sleep schedule for these states is one of our future works.

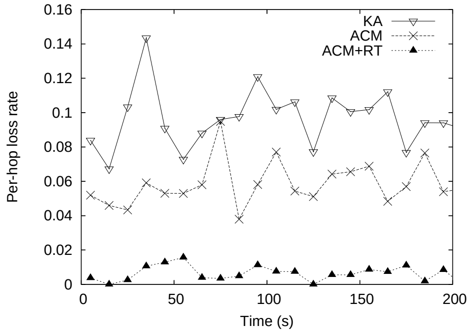

Figure 5: The per-hop loss rate of emergency packets (small scale event).

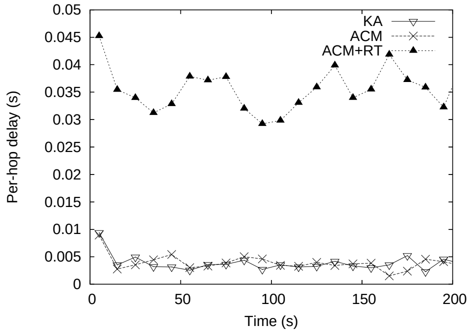

Figure 6: The per-hop delay of emergency packets (small scale event).

3.1 Testbed A

In Testbed A, 25 sensor nodes were arranged in a 5×5 grid topology with separation of one meter. A BS was put beside the grid with one meter separation. The transmission power was set to −27 dBm. The average delivery ratio of normal packets over ten hours experiments was around 80 %.

In order to evaluate the effect of mechanisms comprising UMIUSI, we compared five variants of combination of the mechanisms. One is KA (keep awake), in which nodes in a corridor keep awake but neither suppression of normal packets nor other mechanisms is conducted. The second is ACM, in which an assured corridor is established by suppressing emission of normal packets. The third one is ACM+RT (retransmission), in which ACM and the retransmission are applied. The fourth is ACM+RT+PQ (priority queueing), in which the priority queueing is additionally applied. The

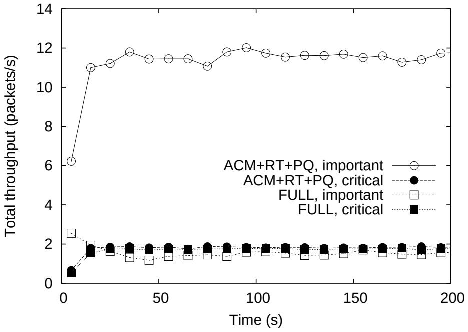

Figure 7: The total throughput of emergency packets (large scale event).

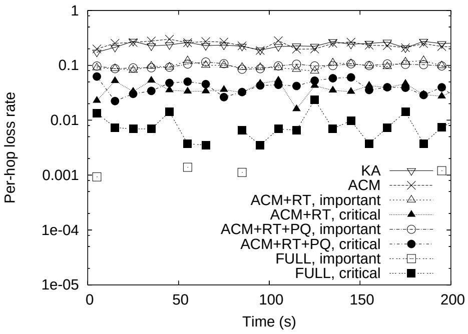

Figure 8: The per-hop loss rate of emergency packets (large scale event).

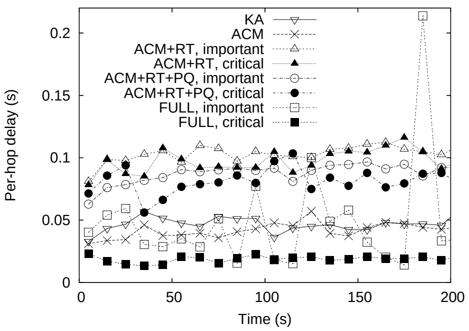

Figure 9: The per-hop delay of emergency packets (large scale event).

last one, FULL employs all of the mechanisms explained in Section 2.3. For the variants with RT, the first retransmission was scheduled at 0.1 seconds after the first transmission for the critical class and 0.2 seconds for the important class, respectively. A binary backoff scheme was applied to following retransmissions. In FULL, local congestion detection was done by monitoring packet reception rate at each node. When a node received more than 20 packets in recent two seconds, it considered that the wireless channel was highly loaded. We observed that the number of packet losses rose up sharply when the traffic exceeded this threshold in preliminary experiments. For the AIMD rate control, the parameters for multiplicative decrease and additive increase were 0.5 and 0.05 packets/s respectively taken from [18].

Since the hop distance from a node to the BS dynamically changed during the experiments, we employ the per-hop loss rate and per-hop delay as evaluation metrics. Letting n the hop distance from a source node to the BS and p_k $(k = 1, 2, \cdots, n)$ the per-hop loss rate in transmission of emergency packets at k-th hop, the loss rate P_n observed at the BS is given by

$$P_n = 1 - Q_n = 1 - \prod_{i=1}^{n}(1 - p_i)$$

where Q_n is the delivery ratio observed at the BS. In the experiments, packet losses were detected at the BS using a sequence number in the header of an emergency packet to obtain P_n. Then, assuming p_k is identical for all hops $(p_1 = p_2 = \cdots = p_n = p)$, the per-hop loss rate p is defined as

$$p = 1 - \sqrt[n]{1 - P_n} = 1 - \sqrt[n]{Q_n}.$$

The per-hop delay d is defined as

$$d = D_n/n,$$

where D_n is time taken from emission of an emergency packet at a source node to reception of the packet at the BS. Note that, since only a clock on the hundred milliseconds scale was provided for the application layer, time error in D_n observed was 100 ms at maximum.

3.1.1 Small Scale Event

In the scenario of a small scale event, one critical class node became a source node of critical packets and went back to normal operation ten minutes later. We conducted experiments twice for each of randomly chosen eight nodes.

Figure 5 shows the per-hop loss rate of emergency packets averaged over the 16 experiments. In these experiments, suppression of normal packets became effective immediately after a node began emitting critical packets, since the hop distance to the BS was so small, 1.6 on average, that it did not take long to establish an assured corridor. With retransmission, the per-hop loss rate was further improved to be less than 1 %.

However, retransmission led to increase of the per-hop delay (Fig. 6). Even if collision was mostly avoided by the suppression, packet losses frequently occurred (see ACM in Fig. 5) due to random channel errors. Therefore, a number of retransmissions were needed to recover those lost packets, which resulted in increase of the delay.

3.1.2 Large Scale Event

For the experiments of a large scale event, we considered eight sets of seven important class nodes and one critical class node. These eight nodes were moved to the *EMG_SEND* state in about ten seconds and went back to the *NORMAL* state about ten minutes later. We conducted experiments twice for each of the eight sets.

First, the total throughput of the critical and important class averaged over the experiments is illustrated in Fig 7. Here the total throughput is defined as the number of emergency packets received by the BS per second. Little difference was observed between the total throughput of ACM+RT and that of ACM+RT+PQ for both classes, thus results of ACM+RT are not shown in Fig. 7. With the rate control

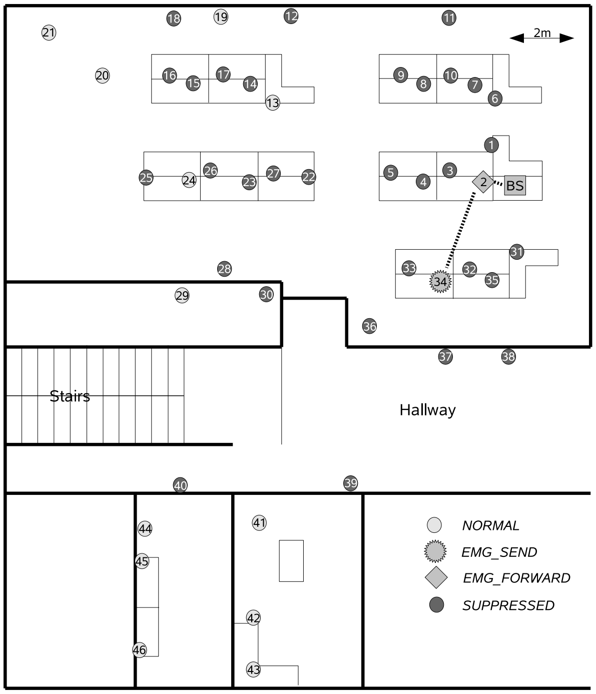

Figure 10: A small scale event in Testbed B. The thick lines represent concrete walls and steel doors. The rectangles with thin lines represent steel desks.

mechanisms, we can see that the total throughput of the important class decreased around 1.5 packets/s in 30 seconds while that without the rate control in ACM+RT+PQ was kept high at 11.5 packets/s.

The per-hop loss rate of emergency packets is illustrated in Fig. 8. Suppression of normal packets had little effect on reliability in a large emergency as can be seen in comparison between KA and ACM. Adding the priority queueing mechanism to ACM+RT was also not much helpful in this experiment settings, because paths of important and critical class traffic seldom overlapped with each other. In FULL, the per-hop loss rate of the critical class gradually decreased as the emission rate of important class was regulated by the rate control mechanisms. Since an important class packet had more chances to be retransmitted than a critical class packet due to the prolonged interval of emission by rate control, the per-hop loss rate of the important class was smaller than that of the critical class.

Figure 9 shows the per-hop delay of emergency packets. In FULL, the per-hop delay of the critical class gradually decreased in the first 30 seconds as the important class traffic was regulated by the rate control mechanisms. The emission interval of important class packets was prolonged by the rate control mechanisms, thus waiting time to be retransmitted at an *EMG_FORWARD* node could be as long as a few seconds. Such occasional large delay caused the large variation in the per-hop delay of the important class in FULL.

3.2 Testbed B

The purpose of the experiments in Testbed B is to verify feasibility of UMIUSI in practical settings. In Testbed B, 46 sensor nodes and a BS were deployed on a floor of several rooms and a hallway in a concrete building, as illustrated in Fig. 10. All of the five mechanisms of UMIUSI (FULL) were used, and parameters were the same as in Testbed A other than the transmission power of −7 dBm.

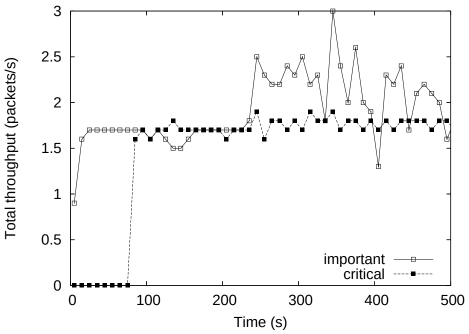

Figure 11: The total throughput of emergency packets in Testbed B.

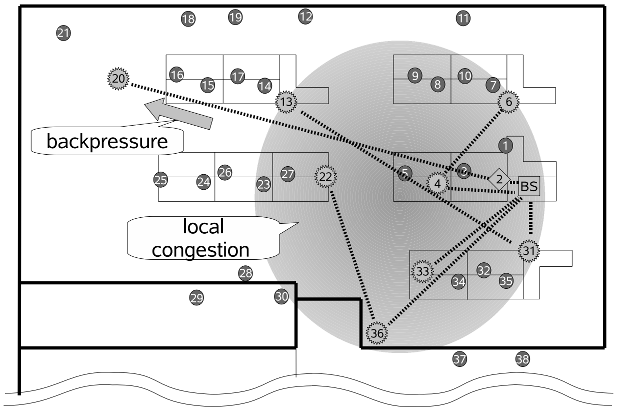

Figure 12: A snapshot in the large scale event in Testbed B ($t = 400$ s).

3.2.1 Small Scale Event

Figure 10 shows a snapshot where Node 34 detected an emergency. Node 34 reached the BS via Node 2 by a path established by the synchronization-based data gathering scheme. Therefore, Node 2 was an *EMG_FORWARD* node in the snapshot. Nodes indicated by dark circles could hear radio signals of Nodes 2 and 34 and moved to the *SUPPRESSED* state. Due to shadowing and fading, the geographical proximity does not necessarily correspond to neighbor relation. The average per-hop loss rate over eight experiments setting Nodes 4, 7, 16, 25, 30, 34, 41, and 46 an *EMG_SEND* node was 0.37%. The average hop distance to the BS was 2.75.

3.2.2 Large Scale Event

Next we considered a scenario where a small scale emergency grew to become large, such as a fire. In this scenario, Node 33 first detected an event, moved to the *EMG_SEND* state, and began sending important packets at time $t = 0$. At $t = 80$ seconds, Node 4 of the critical class next detected the event, followed by Nodes 22 and 36 at $t = 240$ seconds and Nodes 6, 13, 20, and 31 at $t = 340$ seconds. These six nodes were of the important class.

When Nodes 4 and 33 were in the *EMG_SEND* state, emergency packets were directly transmitted to the BS. As shown in Fig. 11 the average throughput was about 1.7 packets/s for both nodes. After Nodes 22 and 36 detected the event at $t = 240$ seconds, local congestion occurred and the three *EMG_SEND* nodes of the important class, *i.e.*, Nodes 33, 22, and 36, reduced the emission rate in order to mitigate congestion. The total throughput of the important class was controlled around 2.3 packets/s. At $t = 340$ seconds, other four nodes newly began to emit important packets and this caused congestion around Node 2. To reduce the important class traffic at its source, Node 2 sent a backpressure to Node 20 (Fig. 12). As a result, the total throughput of the important class was kept about the same level, and there was no loss of critical class packets throughout this ten minutes experiment. The loss rate of important class packets was 0.6 %.

As shown in this experiment, traffic control developed from path-level to network-level adapting to the growing scale of emergency without any centralized control in UMIUSI architecture.

4. CONCLUSION

Urgent sensor information is needed to be transmitted preferentially in a WSN used as a social infrastructure. In this paper, we presented the methodology for designing a WSN architecture for fast and reliable transmission of urgent information. We consider that several simple and fully-distributed mechanisms working in different spatial and temporal levels should be incorporated onto a node so that the collective control of these mechanisms offers preferential transmission of urgent information adapting to the scale of emergency.

We also showed a network architecture, called UMIUSI, designed following the methodology. In this architecture, sensor information is categorized into three traffic classes and five simple mechanisms collaborate consistently. We verified through practical experiments that the architecture successfully improved the delivery ratio and the delay of emergency packets regardless of the scale of emergency.

5. ACKNOWLEDGEMENTS

This research was supported by “Special Coordination Funds for Promoting Science and Technology: Yuragi Project” and a Grant-in-Aid for Scientific Research (A)(2) 16200003 of the Ministry of Education, Culture, Sports, Science and Technology in Japan.

6. REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, Mar. 2002.

[2] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic. Towards optimal sleep scheduling in sensor networks for rare-event detection. In *Proceedings of IPSN 2005*, pages 20–27, Los Angeles, California, USA, Apr. 2005.

[3] D. Chen and P. K. Varshney. QoS support in wireless sensor networks: A survey. In *Proceedings of ICWN 2004*, pages 227–233, Las Vegas, Nevada, USA, June 2004.

[4] B. Deb, S. Bhatnagar, and B. Nath. ReInForM: Reliable information forwarding using multiple paths in sensor networks. In *Proceedings of LCN 2003*, pages 406–415, Bonn, Germany, Oct. 2003.

[5] H. Dubois-Ferrière, D. Estrin, and M. Vetterli. Packet combining in sensor networks. In *Proceedings of ACM SenSys '05*, pages 102–115, San Diego, California, USA, Nov. 2005.

[6] E. Felemban, C.-G. Lee, E. Ekici, R. Boder, and S. Vural. Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks. In *Proceedings of IEEE INFOCOMM 2005*, volume 4, pages 2646–2657, Miami, Florida, USA, Mar. 2005.

[7] T. He, J. A. Stankovic, C. Lu, and T. F. Abdelzaher. A spatiotemporal communication protocol for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 16(10):995–1006, 2005.

[8] B. Hull, K. Jamieson, and H. Balakrishnan. Mitigating congestion in wireless sensor networks. In *Proceedings of ACM SenSys '04*, pages 134–147, Baltimore, Maryland, USA, Nov. 2004.

[9] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of ACM MobiCom 2000*, pages 56–67, Boston, Massachusetts, United States, Aug. 2000.

[10] S. Kashihara, N. Wakamiya, and M. Murata. Implementation and evaluation of a synchronization-based data gathering scheme for sensor networks. In *Proceedings of IEEE ICC 2005*, pages 3037–3043, Seoul, Korea, May 2005.

[11] T. Kawai, N. Wakamiya, and M. Murata. ACM: A transmission mechanism for urgent sensor information. In *Proceedings of IEEE IPCCC 2007*, pages 562–569, New Orleans, Louisiana, USA, April 2007.

[12] S. Pack, J. Choi, T. Kwon, and Y. Choi. Application aware data aggregation in wireless sensor networks. In *Proceedings of IEEE AWiN*, Nov. 2005.

[13] R. Rajagopalan and P. K. Varshney. Data aggregation techniques in sensor networks: A survey. *IEEE Communications Surveys and Tutorials*, 8(4):2–17, 2006.

[14] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis. Interference-aware fair rate control in wireless sensor networks. In *Proceedings of ACM SIGCOMM '06*, pages 63–74, Pisa, Italy, Sept. 2006.

[15] Y. Sankarasubramaniam, B. Akan, and I. F. Akyilidiz. ESRT: Event-to-sink reliable transport in wireless sensor networks. In *Proceedings of ACM MobiHoc 2003*, pages 177–188, Annapolis, Maryland, USA, June 2003.

[16] C.-Y. Wan, A. T. Campbell, and L. Krishnamurthy. PSFQ: a reliable transport protocol for wireless sensor networks. In *Proceedings of ACM WSNA 2002*, pages 1–11, Atlanta, Georgia, USA, Sept. 2002.

[17] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell. CODA: congestion detection and avoidance in sensor networks. In *Proceedings of ACM SenSys '03*, pages 266–279, Los Angeles, California, USA, Nov. 2003.

[18] A. Woo and D. E. Culler. A transmission control scheme for media access in sensor networks. In *Proceedings of ACM MobiCom 2001*, pages 221–235, Rome, Italy, July 2001.

[19] M. Younis, K. Akkaya, M. Eltoweissy, and A. Wadaa. On handling QoS traffic in wireless sensor networks. In *Proceedings of HICSS 2004*, Hawaii, USA, Jan. 2004.