

Research Article

An (m, k) -Firm Real-Time Aware Fault-Tolerant Mechanism in Wireless Sensor Networks

Bijun Li and Ki-Il Kim

Department of Informatics, Engineering Research Institute, Gyeongsang National University, Jinju 660-701, Republic of Korea

Correspondence should be addressed to Ki-Il Kim, kikim@gnu.ac.kr

Received 19 July 2012; Accepted 24 September 2012

Academic Editor: George P. Efthymoglou

Copyright © 2012 B. Li and K.-I. Kim. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many real-time routing mechanisms have been proposed to support the newly developed wireless sensor networks (WSNs) applications such as the transmission and retrieval of multimedia traffic. However, the inherent source constraints of sensor network and instability of wireless communication set quite a problem for the existing routing mechanisms to meet the quality of service (QoS) requirements of some specific QoS-aware applications. Hence, real-time fault-tolerant schemes are highly desired for WSNs to address these challenges. In this paper, we propose an (m, k) -firm-based real-time fault-tolerant mechanism, which helps routing mechanisms to achieve specific QoS requirement by employing a local status indicator (LSI) at each sensor node to monitor and evaluate the local conditions of node and network. Therefore, specific fault recovery mechanisms could be implemented for ensuring an acceptable QoS performance, according to the evaluated LSI values. By using this fault-tolerant scheme, each node dynamically adjusts its transmission capability to mitigate the performance degradation of real-time service caused by network faults and to maintain the desired reliability and timeliness. Simulation result shows that LSI cannot only help to reduce the effects of congestion, link failure, and void, but also reach higher successful transmission ratio and smaller transmission delay.

1. Introduction

Wireless sensor networks (WSNs) have been revolutionizing the way that people interact with the physical world by their diverse applications in different areas [1]. Especially, some real-time communication based novel applications have largely exploited the applied range and potentials of WSNs. For example, in a military surveillance system [2], the detection of a target must be transmitted to the base station as an alert, within a very short time period. Wild fire detection also requires that packets generated by sensor nodes reach the monitoring station timely so that fire-fighters could keep aware of current fire conditions. Moreover, the availability of low cost and miniature size hardware such as CMOS cameras and microphones made it possible to ubiquitously capture multimedia content from the environment [3]. For elderly and health care, with the incorporation of some telemedicine devices, it is possible to remotely monitor the patients' body temperature, blood pressure, breathing activity, and so forth [4]. However, supporting real-time communications in WSNs is a challenging work since WSNs differ dramatically

from the traditional network systems such as wired networks or IP-based wireless networks. First, link connections are lossy and instable in WSNs that they are easily affected by surrounding environment. Thus, it is difficult to implement the precise delay prediction in WSNs. Second, due to the limited resource constraints (power, processing, and memory) of WSNs, a WSN protocol should take minimum energy consumption and minimum overhead into account as well as delay requirement when it deals with some mission-critical applications. Third, various applications may have different requirements in both timeliness and reliability areas. As a result, priorities should be assigned to the packets which have shorter deadlines to make sure they would be delivered to the destination in time.

An efficient routing solution for real-time communications in WSNs is to use geographic routing mechanism. Unlike wired networks that delay is independent of physical distance between source and destination, in multi-hop wireless sensor networks, the end-to-end delay depends on not only single hop delay, but also distance a packet travels [5]. In this case, geographic routing could

effectively decrease the end-to-end delay by selecting the shortest path to destination. However, the void problem of geographic routing must be handled with the consideration of transmission delay and successful transmission rate. In this paper, we use SPEED [5] a well-known geographic soft real-time routing protocol, as the basic routing strategy for real-time communications. Thus, the proposed QoS-aware fault-tolerant mechanism works with SPEED, to provide performance control in both end system and intermediate system.

In general, real-time QoS guarantees can be categorized into three classes: hard real-time (HRT), soft real-time (SRT), and firm real-time (FRT). In HRT system, each packet will be checked with its deterministic end-to-end delay, named deadline, when it arrives at the destination. The arrival of a packet after its deadline is considered as system failure [1]. Due to the inherent constraints and lossy link connections of WSNs, it is impractical to guarantee HRT in WSNs. In SRT system, a probabilistic guarantee is required and some deadline missing is tolerable so that the time-out packets are still useful and system would not be crashed. Most existing real-time routing protocols are supposed to guarantee SRT in a hop-by-hop manner. The last category is FRT, which sets the criterion between HRT and SRT that the lateness of some packets is tolerable but it causes system performance degradation at the same time. Considering the inherent features of WSNs and corresponding application requirements, FRT is the optimal QoS guarantee for adapting real-time communication to WSNs.

In [6], an FRT model called (m, k) -firm was proposed to measure real-time application performance. The concept of (m, k) -firm was defined that a real-time message stream is considered to have an (m, k) -firm guarantee requirement that at least m out any k consecutive messages from the stream must meet their deadlines to ensure adequate QoS. Based on this concept, a priority assignment technology called Distance Based Priority (DBP) was devised to arbitrate between the streams in a system. For each stream, the system maintains a state which captures the recent history of the deadlines met and missed. This state is then denoted as the DBP of the stream. When a stream is close to a failing state, that is, one of the shaded states in Figure 1, its customer will give it a high priority so as to increase its chances of meeting the deadline.

Taking advantage of this model, we propose a novel local status indicator (LSI) which works at each node to indicate the local condition of transmission. In [6] the proposed DBP assignment was used for only one hop model, while in this paper we use LSI in a multi-hop network. Moreover, the purpose of using LSI is to evaluate transmission quality at each hop and to detect network faults, instead of assigning priority to each stream. To the best of our knowledge, there's no existing work introducing FRT to real-time applications in WSNs. To achieve this goal, hop-by-hop delay estimation is used for calculation of LSI. And then according to the value of LSI which shows the local condition of transmission and the DBP value of stream for the QoS performance which is measured at sink based on packet deadline, different fault recovery mechanisms would

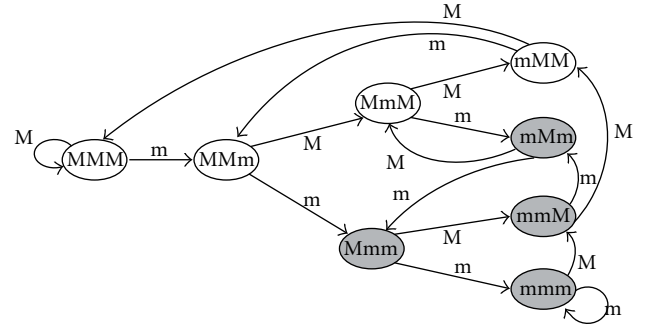


FIGURE 1: State transition diagram example of $(2,3)$ -firm.

be implemented for congestion and link failure. With LSI, the proposed fault-tolerant mechanism has some main features. First, it improves the robustness of existing routing protocol in both timeliness and reliability, with efficient fault-tolerant mechanisms. Second, it involves local system evaluation as well as end system for transmission condition monitoring without vast extra overhead. Third, the prompt fault recovery mechanisms it employed address the practical problems with the consideration of inherent features of WSNs.

The rest of this paper is organized as follows. Some related works are summarized in Section 2 and the proposed mechanism design is elaborated in Section 3. Section 4 shows the simulation results and analysis. We conclude the paper with open issues in Section 5.

2. Related Works

The growing of interest of WSNs applications has inspired the development of real-time routing protocol and fault-tolerant mechanisms. SPEED [5], as we mentioned before, is a classical soft end-to-end real-time routing protocol. It estimates the transmission speed between the current node and the candidate nodes and tries to establish a transmission path with all relay nodes maintaining a desired delivery speed. However, it does not consider the effect of congestion and link failure that when they occur, the information cannot be fed back to upstream nodes immediately, which thus results in the transmission delay and causes the relevant packets to be discarded. A multipath and multi-level SPEED routing protocol (MMSPEED) was proposed in [7], which supports service differentiation and probabilistic QoS guarantee. It dynamically selects the next hop according to the distance among the current node, neighbor node, and sink and sets up a tree structure with multipath for different QoS requirements of applications. However, the time complexity of this scheme is an exponential function of the distance between the current node and the sink node. Thus, it is not suitable for large-scale long-distance transmission. RPAR (Real-time Power-Aware Routing) was proposed in [8], in which the node transmitting power is dynamically adjusted according to its transmission condition and capability. The forwarding node selection is based on the delivery velocity which upstream node requires and downstream node provides. Energy consumption is

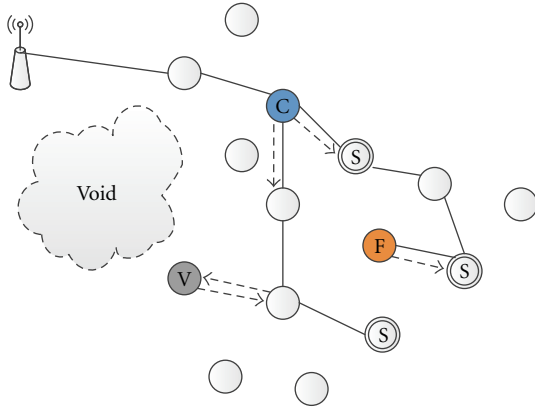


FIGURE 2: Application scenario.

considered as an important issue as well. However, RPAR is not robust enough that it does not consider the effect of link failure, on real-time transmission. A Scalable Hierarchical Power Efficient Routing (SHPER) was released in [9], in order to perform an energy-efficient routing by electing the cluster heads according to the residual energy of the nodes. Based on it, authors of [10] developed an innovative routing scheme named Power Efficient Multimedia Routing (PEMuR) for WMSNs aiming at achieving considerable reduction of energy consumption during routing along with high perceived video QoS.

A real-time fault tolerant routing protocol called FT-SPEED was proposed in [11] which also based on SPEED. It solves the problem of selecting forwarding path in the case that the current node faces a void area. The data can be sent to the sink via bypassing the void. FT-SPEED is supposed to be a fault-tolerant mechanism to reduce the impact of the void region, but the transmission path length maybe considerably long, which may ultimately cause deadline missing of transmitted packets. Event to Sink Reliable Transport (ESRT) [12] is a novel transport solution to achieve reliable event detection with minimum energy expenditure and congestion resolution. The sink is able to detect congestion based on local buffer level monitoring in sensor nodes while in sensor node, whose buffer overflows due to excessive incoming packets, sets congestion notification bit in the header of the packet it transmits. Nevertheless, it does not support real-time communication due to its passive congestion detection manner. In [13], a multipath-based reliable information forwarding protocol called ReInForM was proposed. It is used to deliver the data at desired levels of reliability to recover failures caused by channel errors. It controls the number of paths required for the desired reliability using only local knowledge of channel error rates and does not require any maintenance of multipath. However, the forwarding node selection mechanism of ReInForM considers only the required reliability so that it cannot be applied to meet the timeliness requirement of real-time applications. In [14], a dynamic jumping real-time fault-tolerant routing protocol (DMRF) was proposed to handle the potential fault of network such as failure, congestion and

void region. Each node could use the remaining transmission time of the data packets and the state of the forwarding candidate node set to determine the next hop. It is designed to guarantee the performance of real-time services, although only soft real-time can be satisfied due to its hot-by-hop transmission mode. For some specific applications such as multimedia transmission in WSNs it is not enough to meet the requirements. A priority based congestion control protocol was proposed in [15] that designed for multimedia application in WSNs. Queue length is used as an indication of congestion degree and the rate assignment to each traffic source is set based on its priority index as well as its current congestion degree. However, it should be noted that without MAC layer supports, it is difficult to implement priority based scheduling to guarantee the bounded delay of specific real-time applications.

3. Proposed Mechanism

The application scenario is described in Figure 2.

As we mentioned in Section 1 the proposed real-time fault-tolerant mechanism is adapted to existing real-time geographic routing protocol which is considered to be the optimal solution in real-time communication. Therefore, in this paper, we choose SPEED [5] as the basic routing protocol and implement all three fault recovery mechanisms based on it. As shown in Figure 2, in case of facing the problems of congestion, link failure or void area, the current node could promptly detect the fault occurring and effectively adopt measures to recover it.

The proposed fault-tolerant mechanism includes four components in which two of them are based on SPEED while the other two are executed using the information from them. The former two components from SPEED are beacon exchange and delay estimation, which also play important roles in SPEED. The location information of each node and other necessary parameters are provided by neighbor beacon exchange. Thus, single-hop delay estimation could be implemented using this information, and its output could be used for LSI calculation. Finally, the last component fault recovery will be activated only if the LSI calculation result shows the local transmission is in a negative condition and the end-to-end performance cannot meet its QoS requirement at the same time. The interaction of each component is depicted in Figure 3. This flowchart demonstrates the transmission process of the proposed mechanism. The details are elaborated in the following subsections.

3.1. Neighbor Beacon Exchange. Similar to other geographic protocols, each node in the proposed mechanism periodically broadcasts beacons to its neighbors. This periodic beacon is used to exchange location information among neighbors. In order to prolong the network lifetime to prevent some overloaded nodes from getting depleted much earlier than others, residual energy information is added in periodic beacons as well.

In addition to periodic beacon, three types of on-demand beacons are used to implement the functionalities.

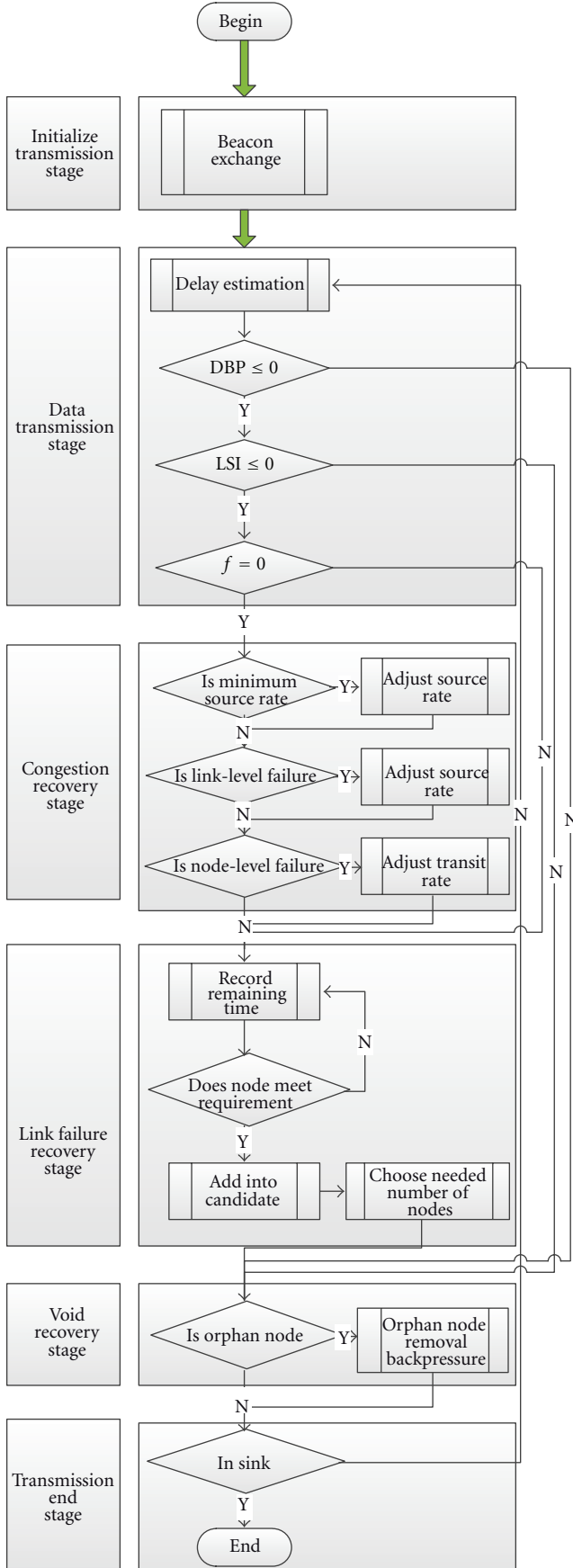


FIGURE 3: Transmission Progress of proposed mechanism.

The single-hop delay estimation beacon is used to measure the local transmission condition between current node and its corresponding node, while the orphan node removal beacon is used to avoid the inherent drawback of geographic protocol, the void region problem. Both will be discussed in the following subsections. Stream DBP beacons are sent from sink to source node as a feedback during transmissions at a regular interval. The value of stream DBP is added into the header of packets each source node generates, and is propagated to the intermediate nodes to help them make decisions for fault recovery in Section 3.4. We argue that the beaconing rate can be low when piggybacking scheme is used.

Based on the information provided by beacons, each node keeps a neighbor table and updates over time. The entries of this table are shown as follows: Neighbor ID, Position, EnergyLevel, EstimatedDelay, ExpireTime. The EstimatedDelay is obtained by Single-Hop Delay Estimation, and the detail is discussed in the Section 3.2. The ExpireTime is set to be a standard RTT (Round-Trip Time) for packet transmission between a pair of nodes. The value of ExpireTime is used for detecting whether or not congestion or link failure occurs by LSI in Section 3.3.

3.2. Single-Hop Delay Estimation. We use the delay estimation mechanism which was introduced by SPEED [5] to implement this function. In this mechanism, data packets passing is used for delay measurement. This delay estimation is calculated at the upstream node, as a metric to approximate the transmission condition between itself and the corresponding downstream node. Formally:

$$\text{Delay}_{i,j} = \text{RTT}_{i,j} - T_{j,\text{procACK}}, \quad (1)$$

where $\text{Delay}_{i,j}$ is the estimated single-hop delay between upstream node i and downstream node j . $\text{RTT}_{i,j}$ is the standard round-trip time calculated on node i , $T_{j,\text{procACK}}$ stands for the processing time of ACK on node j . The current delay estimation is computed by combining the newly measured delay with previous delays via the exponential weighted moving average (EWMA) [16]. Propagation delay is ignored. We use delay estimation instead of average queue size to measure the workload of nodes, since the shared media nature of wireless network, it is possible that the network is congested even if buffer occupancy is low [17].

3.3. Local Status Indicator (LSI) Calculation. The calculation of LSI is a key component of this paper. In addition to stream DBP which is measured at sink for the QoS evaluation of each real-time stream, the proposed mechanism employs a novel component called Local Status Indicator (LSI). LSI allows the intermediate nodes to investigate the local transmissions to the next hop. It can efficiently detect the network fault occurring such as congestion and link failure and help nodes handle the faults efficiently to prevent further degradation.

The functionalities of LSI and stream DBP are totally different. In this paper, stream DBP is calculated at sink to show the QoS performance of real-time stream by using the history of packet deadline missing. LSI follows the main idea

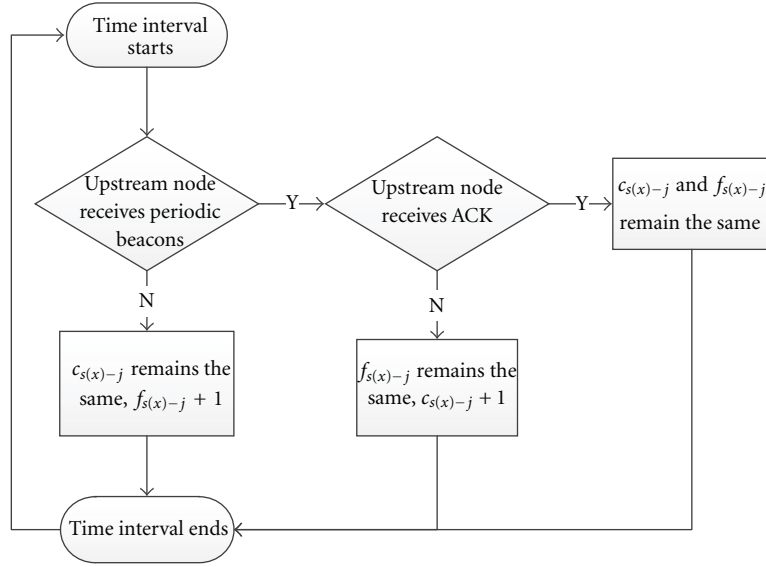


FIGURE 4: Results of waiting in one time interval.

of stream DBP that it could tell the distance to failure [6], in addition it makes the intermediate nodes be aware of the effect of its local condition to the end-to-end QoS guarantee, that is, deadline missing caused by congestion or link failure. The value of LSI is calculated as follow. Formally,

$$LSI_{s(x),i} = K_{s(x)} - m_{s(x)} - C_{s(x),j} - f_{s(x),j}, \quad (2)$$

where $LSI_{s(x),i}$ stands for the distance to failure on node i , k and m are set as the value of required (m, k) -firm; $C_{s(x),j}$ and $f_{s(x),j}$ denote the congestion and link failure levels of downstream node j , respectively.

After an intermediate node receives the first packet, it starts a timer and forwards the packet to the next hop using the routing scheme introduced in SPEED. At the time it receives ACK from the downstream node, the experienced delay is set to be a standard RTT, namely *ExpireTime* as mentioned in Section 3.1, and stored into the corresponding entry of local neighbor table. Since in sensor networks, the nodes which are located close to sink usually forward more packets than others, it is highly possible for them to face congestion or link failure. Therefore, the *ExpireTime* is not the same for all nodes, but proportional to the number of hops to sink. Every time after an intermediate node forwards a packet it will start a timer and wait until the *ExpireTime* timeouts. The results of waiting can be categorized as shown in Figure 4.

In this paper, by using the value of LSI, intermediate nodes could get an evaluation of the local transmission status of each real-time stream. The greater its value, the better condition this current stream has. In case of negative value, which shows the degradation of steam QoS may be caused by this node, LSI could distinguish congestion and link failure as different causes of packet deadlines missing. According to the values of $C_{s(x),j}$ and $f_{s(x),j}$, node i can quickly make local decision to implement fault recovery mechanisms. The details are discussed in Section 3.4.

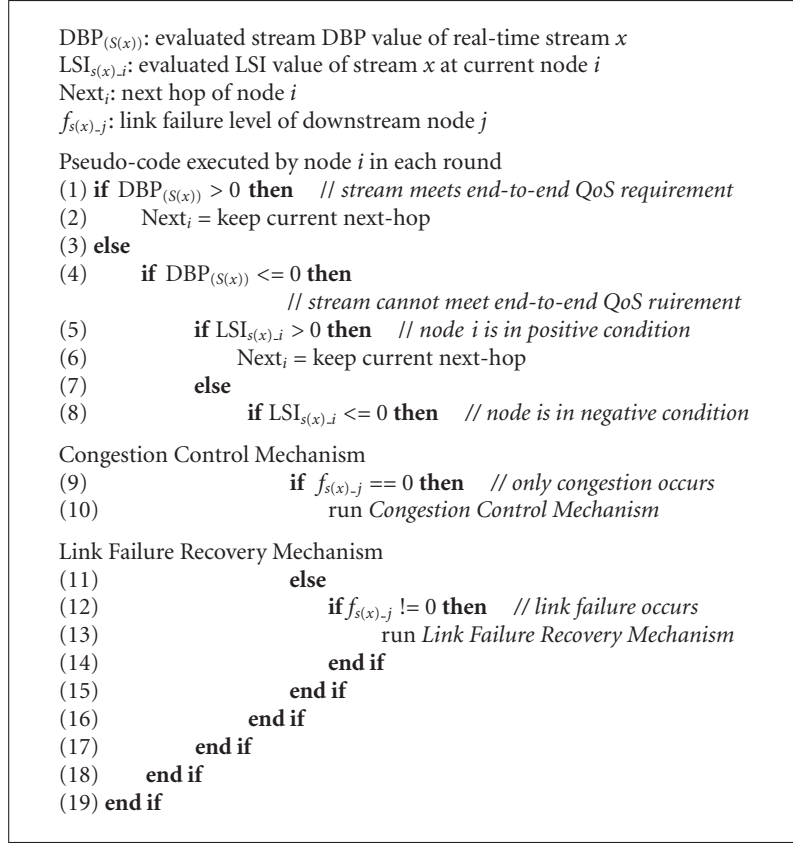
3.4. Fault Recovery Mechanisms. The principal contribution of this paper is the algorithms used for each node to locally recover fault. Compared with previous fault tolerant schemes, the proposed mechanism makes it available to handle the fault recovery with a bounded latency that it is guaranteed all solutions used to handle the problems would not involve excess delay to the transmission. Due to the features of real-time applications, both packet loss and packet deadline missing must be avoided for increasing the rate of successful transmissions and QoS performance. In this paper, we present a new QoS-aware fault recovery algorithm to handle the congestion and link failure problems, and an orphan node removal backpressure for void problem as well. As shown in Figure 3, in transmission stage, each node calculates the value of LSI and compares it with the stream DBP it gets from packets headers. It will make a decision that whether or not fault recovery mechanisms should be necessarily taken. The algorithm for this stage is shown in Algorithm 1.

First we check whether the value of stream DBP meets the end-to-end QoS requirement. The calculation is done following the equation presented in [6] as follows:

$$DBP_{s(x)} = k_{s(x)} - l_{s(x)}(m_{s(x)}, s) + 1, \quad (3)$$

where $DBP_{s(x)}$ is the measured DBP value of stream x at sink, $k_{s(x)}$ comes from the required (m, k) -firm of stream x , $l_{s(x)}(m_{s(x)}, s)$ denotes the position (from the right) of the m th deadline meeting in the current state s of stream x . When one packet is received, a 0 or a 1 is shifted in (from the right) depending on whether the packet missed or met its deadline. If there are less than $m_{s(x)}$ 1s in s , then $l_{s(x)}(m_{s(x)}, s) = k_{s(x)} + 1$. For example, suppose stream $S(1)$ has $(1, 3)$ -firm deadline. Then, $l_{s(1)}(1, MmM) = 1$ and $l_{s(1)}(2, MmM) = 3$.

Although we use the same equation as in [6], the functionality we defined for $DBP_{s(x)}$ differs. Here we calculate $DBP_{s(x)}$ value of each stream to estimate the performance, by



ALGORITHM 1: Data transmission stage algorithm.

checking and recording deadline met and missed of received packets, rather than stream priority assignment. Particularly, positive $DBP_{s(x)}$ value stands for stream meeting its QoS requirement, while nonpositive value stands for not meeting the requirement.

In case nonpositive stream DBP value appears, we check $LSI_{s(x),i}$ at node i to figure out if performance degradation is caused by transmission fault of node i . If $LSI_{s(x),i}$ is not positive and link failure level $f_{s(x),j}$ is 0, congestion is detected and corresponding congestion control mechanism is implemented (details are elaborated in Section 3.4.1.) Otherwise, if $f_{s(x),j}$ indicates link failure occurring, link failure recovery, which is discussed in Section 3.4.2, is activated to recover the fault.

3.4.1. Congestion Control Mechanism. With the considering of the property of WSNs transmission, we defined a new node model for congestion control mechanism, as shown in Figure 5. It provides two queuing buffers for (1) source traffic generated by node itself; (2) transit traffic that node receives from upstream nodes. By using this node model, one node i can adjust its source traffic sending rate $r_{src,i}$ and transit traffic forwarding rate $r_{trs,i}$ separately. The outgoing traffic rate of node i can be calculated by adding the two traffic rates ($r_{out,i} = r_{src,i} + r_{trs,i}$).

Based on this node model, rate adjustment can be implemented efficiently on each node.

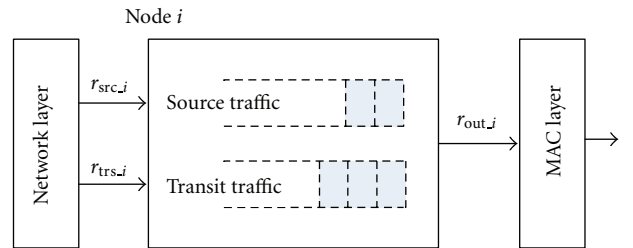


FIGURE 5: New node mode of the proposed scheme.

Different from other schemes, the proposed mechanism is supposed to handle congestion control with the awareness of real-time stream QoS guarantee. Since rate adjustment is considered to be an efficient congestion control method in WSNs [18], the proposed mechanism utilizes stream DBP and LSI values in two rate adjustment algorithms for sink-source node system and intermediate nodes system, to limit the source traffic rate and source/transit traffic rate, respectively. Two algorithms are shown in the following subsections.

Sink-Source Node System. After the calculation of stream DBP using (3), sink sends back the measured $DBP_{s(x)}$ and an adjusted source traffic rate $r_{adj,src,i}$ to the corresponding source node i in a small predefined time interval. We argue

$DBP_{s(x)}$: evaluated stream DBP value of real-time stream x
 $k_{s(x)}$: k value from required (m, k) -firm of stream x
 $r_{\min_src,i}$: minimum source traffic rate of node i
 $r_{src,i}$: current source traffic rate of node i
 $r_{adj_src,i}$: adjusted source traffic rate of node i

Pseudo-Code runs at Sink in each round

- (1) **if** $DBP_{s(x)} \leq 0$ **then** // stream cannot meet end-to-end QoS requirement
- (2) **if** $r_{src,i} > r_{\min_src,i}$ **then** // current source traffic rate can be reduced
- (3)
$$r_{adj_src,i} = \max \left\{ r_{src,i} \cdot \left(1 - \frac{k_{s(x)} - |DBP_{s(x)} - 1|}{k_{s(x)}} \right), r_{\min_src,i} \right\}$$
 //source traffic rate adjustment
- (4) **end if**
- (5) **end if**

ALGORITHM 2: Sink-Source node system congestion control.

that this feedback process can be easily achieved, as sink is considered to be full of computing resources and location based wireless communications are widely used in WSNs. When source node receives the feedback of $DBP_{s(x)}$, it adds the value into the packets it generates. The adjusted source traffic rate $r_{adj_src,i}$ is calculated according to $DBP_{s(x)}$ using the Algorithm 2, and is supposed to adapt the traffic load to network capability and acceptable QoS.

In order to reduce the network traffic load and to satisfy required QoS guarantee at the same time, source traffic rate is decreased to a predefined lower threshold as the minimum source rate, to limit the performance degradation caused by excessive low source traffic rate. Therefore, when sink detects that $DBP_{s(x)}$ is no more than 0, which indicates the stream x is in negative condition, it would adjust the corresponding source traffic rate to a particular level, but not less than the minimum source rate. The calculation of adjustment is based on the deadline meeting rate of the monitored consecutive packets. Then, the adjusted source traffic rate would be sent back to the source node to implement traffic limitation.

Intermediate Nodes System. Considering the feature of real-time applications that big volumes of data are generated in a very short period, it is possible that only sink-source node system rate adjustment is not sufficient to achieve congestion control. Thus, in the proposed congestion control mechanism, local system namely intermediate nodes also participates in end-to-end QoS guarantee, by contributing a local congestion control mechanism.

Local congestion control mechanism is implemented at intermediate nodes, by reducing both source and transit traffic rates to adapt the local traffic load to the node capability. Usually it could eventually mitigate the congestion. Due to the wireless natures and limited resources of WSNs, there exist two types of congestions: link-level congestion and node-level congestion [15]. We use LSI to detect the link-level congestion. The transit traffic buffer status of node i , named as $buff_i$, is used to monitor the node-level congestion. It could be sent as a beacon by node i to its neighbors. We argue that this beacon would not

involve additional energy consumption since piggybacking scheme is used. Algorithm 3 is supposed to be able to detect both two types of congestions and then implement a 2-step mechanism to adjust source/transit traffic rates. If congestion is not mitigated after this 2-step mechanism, a congestion notification will be propagated to the one-hop further upstream node in a backpressure manner, to make it execute the same algorithm to control the traffic.

Step 1. similar to sink-source node system, in case of only link-level congestion happens, upstream node i would first decrease its own source traffic rate according to the local transmission status. Thus, the outgoing traffic rate of node i can be reduced to an acceptable level based on the value of LSI and minimum source traffic rate.

Step 2. if congestion is not mitigated after the source traffic rate is reduced to a minimum acceptable level, or node-level congestion happens at downstream node, the second step will be taken to limit the transit traffic from upstream nodes to the congested downstream node. The weight of each upstream node i is measured according to the total LSI values of all streams passing by, and the outgoing traffic rate of downstream node as well.

3.4.2. Link Failure Recovery Mechanism. The proposed mechanism is used for nodes to recover link failure by choosing the optimal forwarding nodes for redundancy on multipath. Compared with previous fault tolerant schemes or real-time routing protocols, the proposed scheme makes it available to establish multiple transmission paths with a bounded latency during transmissions. It is guaranteed that all selected nodes for forwarding multiple copies of packets can relay the packets timely. Due to the features of real-time applications, packets loss would lead to not only decline of successful transmission rate, but also timeout of a certain amount of packets. The potential high latency which is involved by the use of multipath may severely influence the quality of packets received by sink. Therefore, we present a new delay-aware link failure recovery algorithm for dynamically choosing the optimal forwarding nodes


```

DBPs(x): evaluated stream DBP value of stream  $x$ 
LSIs(x),i: evaluated LSI value of stream  $x$  at node  $i$ 
LSIs(x),j: evaluated LSI value of stream  $x$  at downstream node  $j$ 
 $f_{s(x),i}$ : link failure level of stream  $x$  at node  $i$ 
deadlines(x): deadline of packet of stream  $x$ 
delaytrans,i,j: delay of transmission between  $i$  and  $j$ 
maxdelays(x): maximum allowable delay of stream  $x$ 
 $n_{dnode,i}$ : number of downstream nodes of node  $i$ 
 $d_{s(x),j}$ : deadline of stream  $x$  at node  $j$ 
 $n_{strm,j}$ : number of streams at node  $j$ 
 $s_{candi,i}$ : candidate nodes set of node  $i$ 

Pseudo-Code runs at node  $i$  in each round
(1) if DBPs(x) <= 0 then // stream cannot meet end-to-end QoS requirement
(2)     if LSIs(x),i <= 0 then // local fault happens
(3)         if  $f_{s(x),i}$  != 0 then // link failure happens
(4)             maxdelays(x) = deadlines(x) - delaytrans,i,j
(5)             for  $j$  from 1 to  $n_{dnode,i}$  // for all downstreams
(6)                 for  $x$  from 1 to  $n_{strm,j}$  // for all streams
(7)                     if  $d_{s(x),j}$  < maxdelays(x) && LSIs(x),j > 0 then
// good status of reliability and timeliness
(8)                          $j$  is in  $s_{candi,i}$ 
(9)                     end if
(10)                 end if
(11)             end if
(12) end if

```

ALGORITHM 4: Link failure recovery.

where $P_{src_fwd,i}$ is the optimal number of forwarding paths for multipath establishment.

This equation can be also used when source node receives backpressure from its downstream node, which indicates the failures of some links on the primary path and the intermediate nodes have no candidate to choose, so that it is necessary to start using multipath at the source node.

The local system includes all intermediate nodes and the links between them. Since LSI value is the most useful information for intermediate nodes to evaluate the transmission status, it is used in (5) for optimal number of alternative paths selection. The equation is shown as follows:

$$P_{itm_fwd,i} = \min\{|LSI_{s(x),i} - 1|, s_{candi,i}\}, \quad (5)$$

similar to (4), the number of forwarding paths is calculated adaptively with respect to candidate nodes set and actual situation.

In case of severe channel errors happening, or a sparsely employed network, it is possible that once an intermediate node detects link failure on primary path, it finds no candidates for multipath itself, so it sends backpressure to its upstream node. Therefore, the backpressure may finally reach the source node, and (4) would be executed for recovery as mentioned.

3.4.3. Void Avoidance Mechanism. In WSNs, backpressure scheme is often used for rerouting or notification delivery. In the proposed void avoidance mechanism, we use backpressure only for removing the orphan nodes which

are defined as the nodes without any downstream nodes in local neighbor tables, since these nodes may cause “void” problems in geographic routing schemes. Once an intermediate node updates its neighbor table and finds no downstream nodes left, it will send backpressure beacons which are introduced in Section 3.1, to notify its upstream nodes to remove it from their neighbor tables. We argue that the overload can be low since the beacon rate is low and using of piggybacking scheme.

4. Performance Evaluation

Performance of the proposed scheme is proved by simulation. We chose NS-2 as the simulator. 50 nodes are randomly placed in 200 m × 200 m field. 4 source nodes are randomly selected within an event area radius of 50 m. Sink is located at the lower right corner of the field. Thus the end-to-end hop-count ranges from 4 to 9 hops with an average of 6 hops. Each node has a radio range of 40 m. Propagation model is set to be Two-Ray Ground, protocols for physical and MAC layer are set to be wireless-phy and 802.11.

We set two scenarios to evaluate the performance. In the first one, 3 source nodes are supposed to generate periodic traffic and the last one generates aperiodic bursty traffic as well, to prove the adaptability of the proposed mechanism, when facing a rapid change of data volumes. The second one contains various channel errors during transmission in order to estimate the usability of the proposed mechanism.

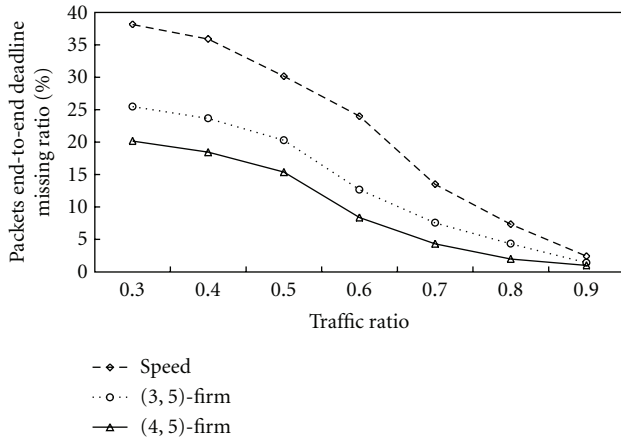


FIGURE 6: Packets end-to-end deadline missing ratio.

Evaluation results are presented as: (1) packets end-to-end deadlines missing ratio, (2) stream end-to-end dynamic failure ratio. The former one considers the timeliness feature of individual packet, while the latter one is supposed to measure the QoS guarantee in both reliability and timeliness which are the main reasons of dynamic failure in real-time applications.

4.1. Packets End-to-End Deadline Missing. Figures 6 and 7 plot the packets end-to-end deadline missing ratio of 3 different algorithms: SPEED, the proposed mechanism with (3,5)-firm and (4,5)-firm guarantees. The packets end-to-end deadline is set to be 50 ms for all 3 algorithms.

We chose the traffic of one node from those 3 periodical traffic nodes as the evaluation target, so that the horizontal axis of Figure 6 stands for the ratio of the target traffic to all traffics in network. The smaller it is, the heavier traffic load the network bears. Especially for the nodes which are closer to sink, the probability of congestion is much higher than other nodes. In Figure 6 we can learn that the traffics transmitted using SPEED experience more than 20% packet deadline missing when traffic ratio is about 60% and almost 40% deadline missing when traffic ratio reaches 30%. Considering only delivery speed as the routing metric without any fault-tolerant scheme, SPEED performs much worse than proposed mechanisms. With the help of LSI value based fault recovery mechanism, the proposed mechanism could efficient handle the problem and remain an acceptable performance of QoS guarantee.

Similar result comes from Figure 7, that in a scenario where channel error happens and increases proportionally, deadline missing ratio rises dramatically in SPEED since it has no failure managements. On the other hand, even under heavy traffic or unstable network condition, LSI works well to indicate the “distance to failure” and distinguish different faults; also based on both LSI and stream DBP, the proposed mechanism is smarter that it handles network faults efficiently. The difference between (3,5)-firm stream and (4,5)-firm stream in Figure 7 is that according to the mechanism of L-DBP, (4,5)-firm stream has more strict

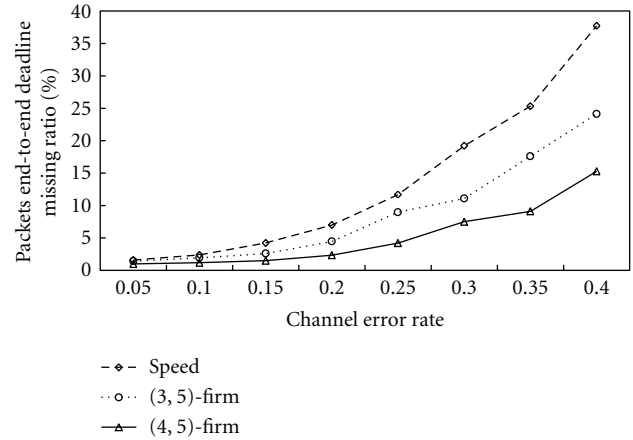


FIGURE 7: Packets end-to-end deadline missing ratio.

requirement, so that the upstream node is more sensitive to the transmission status changes, and it will make more agile reaction to change the downstream node with better condition.

4.2. Stream End-to-End Dynamic Failure. In Figures 8 and 9, we evaluate the stream end-to-end dynamic failure ratios, among 3 algorithms: SPEED, the proposed mechanism with different deadlines of 40 ms and 50 ms, respectively. We give a (3,5)-firm guarantee requirement for all 3 algorithms to test if they could meet their QoS guarantee.

The horizontal axis of Figures 8 and 9 is the same as in Figures 6 and 9, respectively. Simulation result shows that the dynamic failure ratio is closely related to packet deadline missing rate. In addition, for real-time applications, traffics may experience end-to-end dynamic failures even if the packet loss is less than requirement. The significantly rising curves of SPEED in both figures demonstrate that without firm real-time requirement and fault-tolerant mechanisms, it failed to apply good QoS performance in case of heavy traffic or instable network environment. Together with stream DBP, the proposed LSI plays a very important role in packets transmission that it makes all intermediate nodes to be aware of local transmission status with the next hop, and make correct decisions when fault occurs. The congestion control mechanism and link failure recovery mechanism effectively handled the faults, without introducing much extra overhead to latency and resources. The proposed mechanism could be highly desired by firm real-time stream applications. By distributing the duty of guarantee (m, k) -firm from sink to each intermediate node, LSI and stream DBP together make it possible to keep good QoS performance of real-time applications.

5. Conclusion and Future Works

For the rapidly developed real-time applications of WSNs, efficient and robust routing protocols are highly desired. Due to the inherent constraints of WSNs, it is difficult

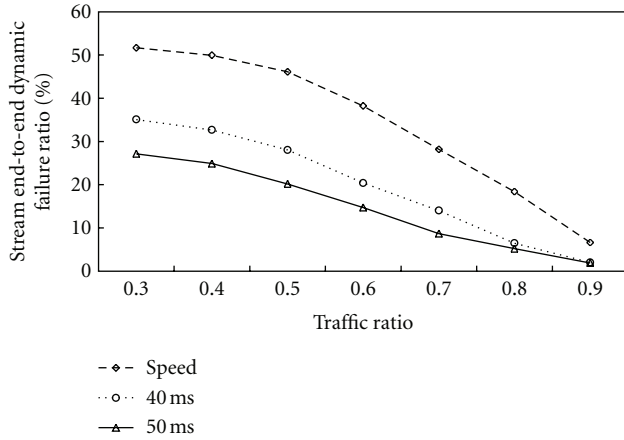


FIGURE 8: Stream end-to-end dynamic failure ratio.

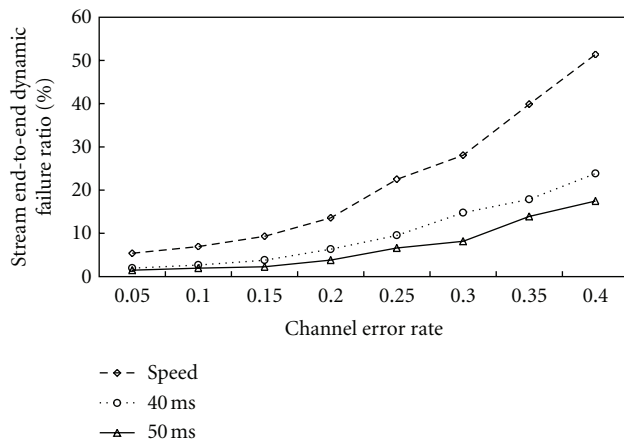


FIGURE 9: Stream end-to-end dynamic failure ratio.

for them to satisfy the requirements of some specific QoS-aware real-time applications. The proposed fault-tolerant mechanism is used to improve the existing real-time routing protocols with an (m, k) -firm based local transmission indicator (LSI) to make the intermediate nodes be aware of their local transmission conditions. According to the information provided by LSI and steam DBP, different fault recovery mechanisms are implemented to handle congestion, link failure and void. This adaption capability makes the proposed mechanism more functional in simulations, comparing to SPEED. Simulation results show that due to the contribution of each component, the proposed mechanism performs much better in both timeliness and QoS guarantee features with low end-to-end deadline missing ratio and low end-to-end dynamic failure ratio.

The future work will be focused on design of a new routing protocol with the adaption to (m, k) -firm and more simulation works for performance evaluation. Also, deployment issues will be taken for the further work.

Acknowledgment

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF)

funded by the Ministry of Education, Science, and Technology (2012-0001761).

References

- [1] Y. Li, C. S. Chen, Y. Q. Song, and Z. Wang, "Real-time QoS support in wireless sensor networks: a survey," in *Proceedings of the Annual 7th IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems (FeT '07)*, Toulouse, France, 2007.
- [2] S. Ehsan and B. Hamdaoui, "A survey on energy-efficient routing techniques with QoS assurances for wireless multimedia sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 2, pp. 265–278, 2012.
- [3] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921–960, 2007.
- [4] A. Sharif, V. Potdar, and E. Chang, "Wireless multimedia sensor network technology: a survey," in *Proceedings of the 7th IEEE International Conference on Industrial Informatics (INDIN '09)*, pp. 606–613, Cardiff, UK, June 2009.
- [5] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: a stateless protocol for real-time communication in sensor networks," in *Proceedings of the 23th IEEE International Conference on Distributed Computing Systems*, pp. 46–55, May 2003.
- [6] M. Hamdaoui and P. Ramanathan, "A dynamic priority assignment technique for streams with (m, k) -firm deadlines," *IEEE Transactions on Computers*, vol. 44, no. 12, pp. 1443–1451, 1995.
- [7] E. Felemban, C. G. Lee, and E. Ekici, "MMSPEED: multipath Multi-SPEED Protocol for QoS guarantee of reliability and timeliness in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 738–754, 2006.
- [8] O. Chipara, Z. He, G. Xing et al., "Real-time power-aware routing in sensor networks," in *Proceedings of the 14th IEEE International Workshop on Quality of Service (IWQoS '06)*, pp. 83–92, New Haven, Conn, USA, June 2006.
- [9] D. Kandris, P. Tsioumas, A. Tzes, G. Nikolakopoulos, and D. D. Vergados, "Power conservation through energy efficient routing in wireless sensor networks," *Sensors*, vol. 9, no. 9, pp. 7320–7342, 2009.
- [10] D. Kandris, M. Tsagkaropoulos, I. Politis, A. Tzes, and S. Kotsopoulos, "Energy efficient and perceived QoS aware video routing over Wireless Multimedia Sensor Networks," *Ad Hoc Networks*, vol. 9, no. 4, pp. 591–607, 2011.
- [11] L. Zhao, B. Kan, Y. Xu, and X. Li, "FT-SPEED: a fault-tolerant, real-time routing protocol for wireless sensor networks," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '07)*, pp. 2531–2534, Shanghai, China, September 2007.
- [12] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz, "ESRT: event-to-sink reliable transport in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 1003–1016, 2005.
- [13] B. Deb, S. Bhatnagar, and B. Nath, "ReInForM: reliable information forwarding using multiple paths in sensor networks," in *Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN '03)*, pp. 406–415, Bonn/Königswinter, Germany, 2003.
- [14] G. Wu, C. Lin, F. Xia, L. Yao, H. Zhang, and B. Liu, "Dynamical jumping real-time fault-tolerant routing protocol for wireless sensor networks," *Sensors*, vol. 10, no. 3, pp. 2416–2437, 2010.

- [15] M. H. Yaghmaee and D. Adjeroh, "A new priority based congestion control protocol for wireless multimedia sensor networks," in *Proceedings of the 9th IEEE International Symposium on Wireless, Mobile and Multimedia Networks (WoWMoM '08)*, Newport Beach, Calif, USA, June 2008.
- [16] J. F. Kurose and K. W. Ross, *Computer Networking A Top-Down Approach Featuring the Internet*, Pearson/Addison Wesley, 5th edition, 2009.
- [17] C. Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: congestion detection and avoidance in sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 266–279, Los Angeles, Calif, USA, November 2003.
- [18] L. Paradis and Q. Han, "A survey of fault management in wireless sensor networks," *Journal of Network and Systems Management*, vol. 15, no. 2, pp. 171–190, 2007.

