Wireless Netw (2014) 20:719–731
DOI 10.1007/s11276-013-0634-5

A novel real-time scheme for (*m*,*k*)-firm streams in wireless sensor networks

Bijun Li · Ki-Il Kim

Abstract The multimedia transmission based real-time applications have posed a big challenge to wireless sensor networks (WSNs) where both reliability and timeliness need to be guaranteed at the same time, to support an acceptable Quality of Service (QoS). The existing real-time routing protocols, however, are not able to meet the QoS requirements of realtime applications because of the inherent resource constraint of sensor nodes and instability of wireless communication. Therefore, we propose a real-time scheme in this paper, including a QoS-aware routing protocol and a set of fault recovery mechanisms, for (*m*,*k*)-firm based real-time applications over WSNs. A local status indicator which is specially devised for (*m*,*k*)-firm stream, is used for intermediate nodes to monitor and evaluate their local conditions. The proposed routing protocol takes into account of packet deadline, node condition and remaining energy of next hop, to make optimal forwarding decision. Additionally, according to the stream QoS and node condition, the proposed fault recovery mechanisms are utilized for nodes to handle the congestion, link failure and void problems occurred during transmission and remain the desired reliability and timeliness requirements. The proposed scheme has been well studied and verified through simulations. The results have proved the efficiency of the proposed scheme in terms of high successful transmission ratio, small end-to-end delay and long lifetime of network.

Keywords Real-time routing · Fault recovery · (*m*,*k*)-firm · Wireless sensor networks

1 Introduction

In recent decades, wireless sensor networks (WSNs) have been revolutionizing the way that people interact with the physical world by their diverse applications in different areas [1]. Specifically, real-time communication based applications have largely exploited the applied range and potentials of WSNs. For example, in a military surveillance system, the detection of a target must be transmitted to the base station as an alert within a very short time period. Fire detection also requires the packets to reach the monitoring station timely so that the fire-fighters could keep aware of current fire conditions. Moreover, the availability of enhanced nodes such as low-cost and miniature size cameras or microphones makes it possible for WSNs to provide more powerful functions. These nodes can capture multimedia data such as video and audio streams and still images in real-time applications. For example, a sensor network for health-care environment uses video transmission as sensory modality to identify patients' behavior [2]. The network architecture is shown in Fig. 1.

Supporting such real-time applications in WSNs, however, is a challenging work since WSNs differ dramatically from the traditional network systems such as wired networks or IP-based wireless networks. First, the link connections in WSNs are lossy and instable, so that they can be easily affected by surrounding environment. Precise delay prediction is difficult to run in WSNs [3]. Second, due to the limited resource constraints of WSNs, including power, processing and memory, a WSN protocol should minimize the energy consumption and overhead as well as

---

B. Li · K.-I. Kim (✉)
Department of Informatics, Engineering Research Institute,
Gyeongsang National University, Jinju, Gyeongnam, Korea
e-mail: kikim@gnu.ac.kr

B. Li
e-mail: libijun@gnu.ac.kr

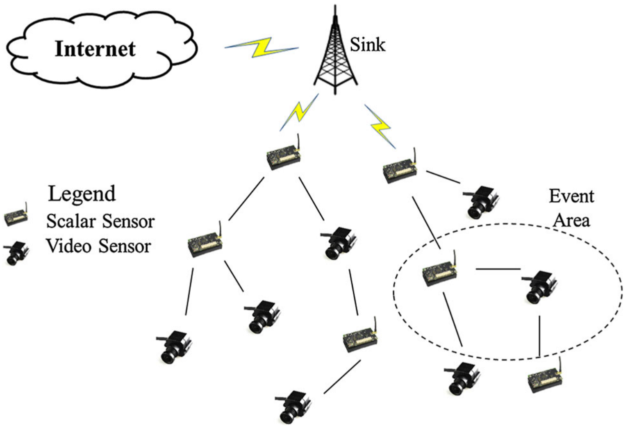


Fig. 1 Real-time application in wireless sensor networks

delay requirement when it is dealing with some mission-critical applications [4–6]. Third, applications may have different requirements in both timeliness and reliability areas. As a result, priorities should be assigned to the packets with shorter deadlines to make sure they would be delivered to the destination in time. Therefore, the techniques used in multimedia transmissions should consider not only energy efficiency and reliability but also timeliness to avoid dynamic failure of QoS and to facilitate specific service guarantees [7].

Generally, real-time tasks can be categorized into three classes: hard real-time (HRT), soft real-time (SRT) and firm real-time (FRT). In HRT task, 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 [2]. Due to the inherent constrains and lossy link connections of WSNs, it is impractical to fulfil HRT tasks in WSNs. In SRT task, a probabilistic guarantee is required and some deadline missing is tolerable so that the time-out packets are still useful and system would not crash. Most existing real-time routing protocols are supposed to guarantee SRT task in a hop-by-hop manner. FRT sets the criterion between HRT and SRT that the lateness of some packets is tolerable but may cause system performance degradation at the same time. Considering the inherent features of WSNs and application requirements, FRT is the optimal system for real-time communication over WSNs. An FRT task model called (*m*,*k*)-firm was proposed to measure the performance of real-time applications. The concept of (*m*,*k*)-firm was defined as follows: a real-time message stream is considered to have an (*m*,*k*)-firm guarantee requirement that at least *m* out of any *k* consecutive messages from the stream must meet their deadlines, to ensure adequate QoS [8] . Based on this concept, a priority assignment technology called Distance Based Priority (DBP) was developed to arbitrate between the streams in a system [8]. For each stream, the system maintains a state of the recent history of captured deadline meet and miss which are marked as M and m in Fig. 2, respectively. Then the state is denoted as DBP value of the stream. When a stream is close to the failing state, i.e. one of the grey states in Fig. 2, its customer will give it a high priority so as to increase its chances of meeting the deadline [8]. In the proposed scheme, DBP is calculated at sink and is used to evaluate the QoS of each stream rather than priority assignment.

Taking the advantage of this model, in [9] we have proposed a local status indicator (LSI) over WSNs to indicate the local condition of transmission status at each node. Unlike DBP assignment, LSI is used in a multi-hop network, aiming to evaluate transmission quality at each hop and to detect network faults, instead of assigning priority to streams. To the best of our knowledge, there's no existing similar works introducing FRT to real-time applications over WSNs. Based on LSI, we have introduced a set of fault recovery mechanisms for QoS maintenance in [10]. In this paper, we devise a real-time routing protocol for (*m*,*k*)-firm streams based on LSI as well, and work it under the fault recovery mechanisms to make a complete scheme for QoS guarantee of real-time applications. The proposed routing calculates optimal forwarding decision based on three metrics: packet deadline, LSI which is regarded as node condition, and remaining power of sensor node. Since packet deadline is used for node choosing, the QoS requirement is considered by each node. By choosing the forwarding nodes with qualified LSI, the timeliness of transfer is ensured. Awareness of remaining power additionally avoids fast drain of energy on often-used nodes and is supposed to prolong the lifetime of networks. Specific fault recovery mechanisms are then implemented for intermediate nodes to handle the problems occurred during transmissions, including congestion, link failure and void, by considering both stream DBP and LSI at each node. With the contributions of routing protocol and fault recovery mechanisms, the proposed scheme shows high performance in successful packet delivery ratio and deadline meet ratio, which have been studied and reported through simulations.

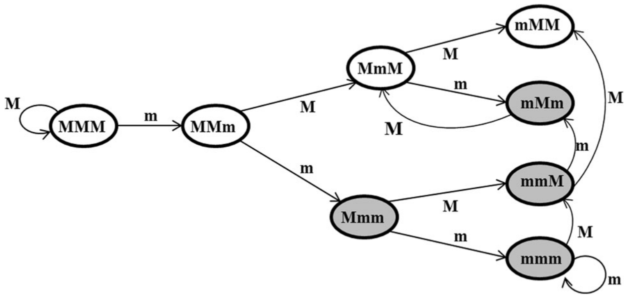


Fig. 2 An example of (2,3)-firm

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

2 Related works

The most common real-time routing protocols in WSNs are presented here. SPEED [11] is a well-known soft real-time routing protocol. It estimates the transmission speed between current node and candidate nodes, tries to establish a transmission path with all relay nodes maintaining a desired delivery speed. However, it doesn't take packet deadline of a real-time stream into account and has no fault-tolerant mechanism, which may consequently lead to severe dynamic failure. A multipath and multi-level SPEED routing protocol (MMSPEED) was proposed in [12], 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. Therefore, it is not suitable for large-scale long-distance transmission. Real-time Power-Aware Routing (RPAR) was proposed in [13], 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 velocities upstream node requires and downstream node provides. Energy consumption is considered as an important issue as well. A Scalable Hierarchical Power Efficient Routing (SHPER) was proposed in [14], in order to form an energy-efficient routing by electing the cluster heads according to the residual energy of the nodes. Based on it, authors of [15] 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 routing protocol with load distribution (RTLD) was proposed in [16], which makes forwarding decision based on link quality, packet transfer velocity and remaining power of next hop, aiming to ensure high packet throughput and long lifetime of networks.

A real-time fault tolerant routing protocol called FT-SPEED was proposed in [17] 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) [18] 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 doesn't support real-time communication due to its passive congestion detection manner. In [19], 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 [20], 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's not enough to meet the requirements. A priority based congestion control protocol was proposed in [21] 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's difficult to implement priority based scheduling to guarantee the bounded delay of specific real-time applications.

3 Design of proposed scheme

The proposed scheme is responsible for QoS-aware real-time routing and transmission fault recovery. When the optimal forwarding decision is calculated at each node by using the proposed routing protocol, fault recovery mechanisms are also activated to handle the problems occurred during transmissions, such as congestion, link failure and void. The situation is judged by the values of stream DBP and LSI, which stand for end-to-end QoS and local transmission status, respectively.

In general, the proposed scheme consists of two subsystems, and each subsystem includes several components, as shown in Fig. 3. The proposed routing protocol is composed of five components: routing mechanism, beacon exchange, orphan node backpressure, delay estimation and calculation of LSI. The routing mechanism makes forwarding decision and maintains a neighbor table. Beacon exchange is used to exchange information among nodes. Orphan node backpressure can prevent void occurrence. The delay estimation can calculate single-hop delay which is used for LSI calculation. At the meantime, the components of fault recovery will be activated if the end-to-end QoS performance of one stream cannot be met and the LSI calculation result shows that the local transmission is in a negative condition as well. To make it easier to understand how each component cooperates with others, Fig. 4 shows the work process of the proposed scheme.

In Fig. 4, ① stands for the establishment of a primary route which heads to the sink. This process is running based on the routing protocol we proposed in the following Sect. 3.4 It considers three metrics for making forwarding decisions: deadline, node condition which is represented as LSI and remaining power. All the necessary information can be acquired by using the mechanisms we introduced in Sects. 3.1–3.3. In case of congestion occurence,as shown as ② in Fig. 4, the congested node will inform its upstream nodes toreduce their traffic loads. The detail about congestion recovery is discussed in Sect. 3.5.1. ③ shows the process of link failure recovery, which helps nodes to recover link failure by generating limited redundancy over multipath. The detail about how to choose the optimal forwarding nodes is discussed in Sect. 3.5.2. When a node faces void problem, ④ will be active to handle it. Section 3.5.3 shows the detail of void recovery.

3.1 Beacon exchange

Following the proposed scheme, each node in the network periodically broadcasts beacons to its neighbors. This periodic beacon is used for each node to inform its existence to the neighbors. Since all nodes in WSNs are supposed to be stationary, the periodic beacon rate can be low, so that it will not involve too much overhead. In order to prolong the network lifetime and prevent some overloaded nodes from getting depleted much earlier than others, residual energy information is added in periodic beacons as well.

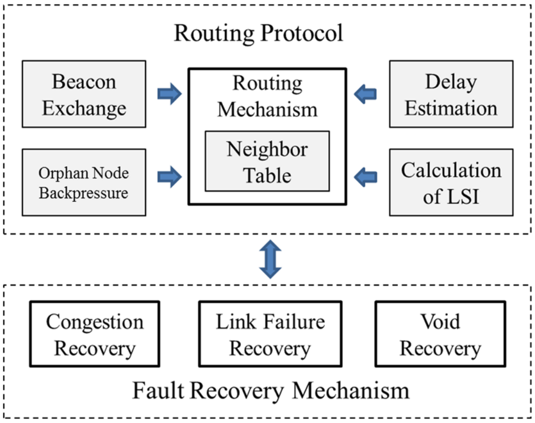


Fig. 3 Components of Proposed Scheme

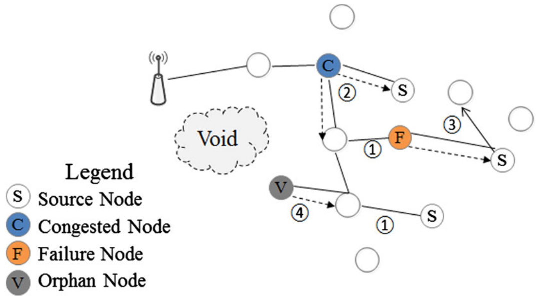


Fig. 4 Work Process of Proposed Scheme

In addition to periodic beacon, three types of on-demand beacons are used to support the functionalities. 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 back to source node as a feedback during transmissions in 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 routing decisions and implement fault recovery. 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 below: (*Neighbor ID*, *EnergyLevel*, *EstimatedDelay*, *ExpireTime*, (*m*,*k*), *Deadline*, DBP). The *EstimatedDelay* is obtained by Single-Hop Delay Estimation, and the detail is discussed in the Sect. 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 to detect whether or not congestion or link failure occurs in Sect. 3.3 The values of m and k are from the (m,k)-firm requirement of each stream. Also, the packet deadline is marked as *Deadline*. The stream DBP value informed by stream DBP beacons from sink is also recorded in neighbor table, named DBP, to show the QoS of stream. The calculation of stream DBP is done by the equation presented in [8] as follows:

$$\mathrm{DBP} = k - l(m,s) + 1 \tag{1}$$

where stream DBP is measured at sink, k comes from the required (m,k)-firm of one particular stream, $l(m,s)$ denotes the position (from the right) of the m^{th} deadline meeting in the current state s of the stream [8]. 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 1s in s, then $l(m,s) = k + 1$. For example, suppose stream has (1,3)-firm deadline. Then, $l(1, MmM) = 1$, $l(1, mMm) = 2$ and $l(1, Mmm) = 3$. The calculations are illustrated in Fig. 5.

3.2 Single-hop delay estimation

We use delay estimation mechanism, which was introduced by SPEED [11], 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:

$$Delay = RTT - T_{procACK} \tag{2}$$

where *Delay* is the estimated single-hop delay between upstream node and downstream node. *RTT* is the standard round-trip time calculated on upstream node, and $T_{procACK}$ stands for the processing time of ACK on downstream node. The current delay estimation is computed by combining the newly measured delay with previous delays via the exponential weighted moving average (EWMA) [22]. 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's possible that the network is congested even if buffer occupancy is low [23].

3.3 Calculation of local status indicator (LSI)

A key component of this paper is discussed in this section. In addition to stream DBP which is measured at sink for the QoS evaluation of each real-time stream, the proposed scheme employs a Local Status Indicator (LSI). It was first proposed in our prior work [9] and improved by this paper. LSI allows the intermediate nodes to investigate local transmissions to the next hop. It can efficiently detect the network fault occurrence such as congestion and link failure, therefore, it is used to help nodes make routing decision and handle the faults efficiently to prevent further degradation.

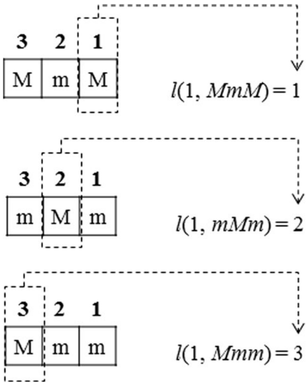


Fig. 5 Calculations of $l(m, s)$

The functionalities of LSI totally differ from stream DBP. As mentioned before, 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 of stream DBP that it could tell the distance to failure, in addition it makes the intermediate nodes be aware of the effect of its local condition to the end-to-end QoS guarantee, i.e. deadline missing caused by congestion or link failure. The value of LSI is calculated as follow. Formally,

$$\mathrm{LSI} = k - m - n(c,s) - n(f,s) \tag{3}$$

where LSI stands for the distance to failure on upstream node, k and m are set as the value of required (m,k)-firm; $n(c,s)$ and $n(f,s)$ denote the numbers of congestion and link failure in the current state s of the stream, respectively. The calculation of c and f are discussed below.

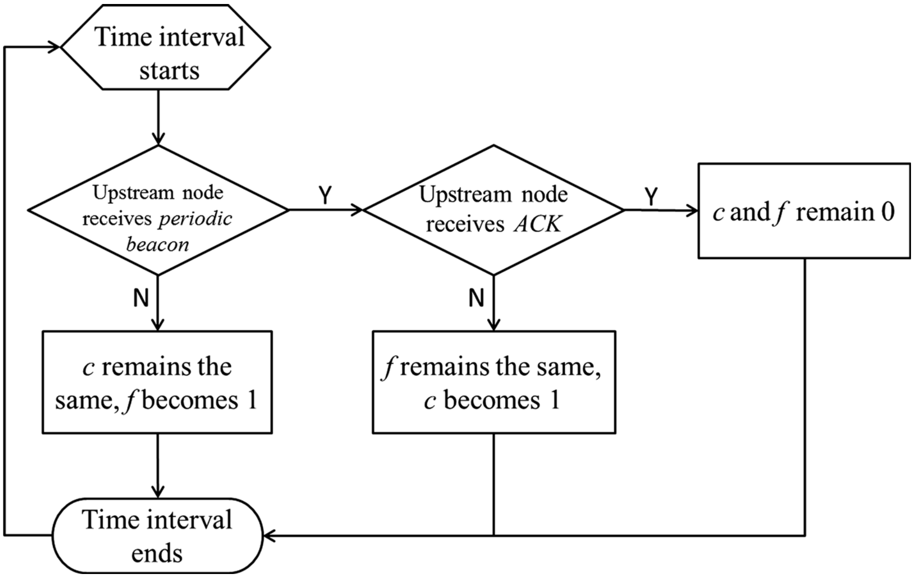
After an intermediate node receives the first packet, it starts a timer and forwards the packet to the next hop. At the time it receives ACK from the downstream node, the experienced delay is set to be a standard *RTT*, named as *ExpireTime*, as mentioned in Sect. 3.1, and stored into the corresponding 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 more 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. After an intermediate node forwards a packet, it will initialize the values of c and f as 0, and wait until the *ExpireTime* timeouts. The results of waiting can be categorized in Fig. 6.

In this paper, LSI will help intermediate nodes to get an evaluation of the local transmission status of each real-time stream. The greater this value is, the better condition current stream has. In case of negative value which shows the degradation of steam QoS may be caused by this node, LSI can distinguish between congestion and link failure as different causes of transmission faults. According to the congestion level $n(c,s)$ and link failure level $n(f,s)$, upstream node can quickly make local decision for routing, and implement fault recovery mechanisms as well. The details are discussed in Sects. 3.4 and 3.5.

3.4 Real-time routing protocol

The proposed routing protocol contains two processes: forwarding node selection and forwarding mechanism

Fig. 6 Calculation of *c* and *f* of LSI



implementation. The forwarding node selection is used to calculate a set of forwarding nodes based on forwarding metrics. The forwarding mechanism is aiming to implement the forwarding process according to the QoS requirements and forwarding node conditions.

In order to figure out the calculation of optimal forwarding, three forwarding metrics are considered: packet deadline, LSI and remaining power of nodes. According to the information provided before, we can easily measure three metrics, through *Deadline*, LSI and *EnergyLevel*, respectively.

Given by the three metrics of each downstream node, the upstream node could make forwarding decision, following the algorithm shown in Table 1. For an easier understanding, we would like to create a table for all parameters and variables which are used in the following 5 tables.

A 2-step algorithm runs when the upstream node receives a packet. Step 1 is used to select a forwarding set from the nodes listed in neighbor table. The selection is based on the three forwarding metrics we mentioned before. If the remaining time of received packet of stream *n* is longer than the deadline from node *i*, it's possible for upstream node to forward the packet to node *i*. That is, it's highly guaranteed that within $Deadline_i$ the packet could be delivered to sink. Also, the $\mathrm{DBP}_{s(i)}$ must be checked that only if it is in positive condition the $Deadline_i$ can be met. In other words, node *i* can guarantee that the packet would be delivered to sink within $Deadline_i$. Then the local condition should be checked.

To successfully forward a packet, LSI must be in a positive condition. Here's one thing to notice: even the values of LSI are the same, (*m,k*)-firm requirements may differ. It's possible that the required (*m,k*)-firm is stricter than the provided one, i.e. (3,4)-firm to (2,4)-firm, so the provided (*m,k*)-firm will not be considered even its LSI is in positive condition. The remaining power of node must be guaranteed higher than the defined threshold as well. The nodes which can meet all requirements will be added to a forwarding set. By considering these three routing metrics, both reliability and timeliness requirements can be guaranteed for real-time routing.

In Step 2, optimal forwarding node will be figured out if the forwarding set is not empty. In case of empty forwarding set, void recovery mechanism will be activated. We will discuss it in the following subsection. If stream DBP of current transmission is in positive condition which indicates the QoS requirement is met, the principle for next hop selection is energy efficiency, so that the node with largest remaining power would be the optimal forwarding node. In other case, the node with best condition, regarded as LSI, would be chosen.

Parameters and variables defined for algorithms

$Deadline_n$: deadline of stream from node *n*
$Deadline_{s(n)}$: deadline of packet of stream *n*
$Deadline_i$: deadline of stream from node *i*
$Deadline^j_{s(n)}$: deadline of stream *n* at node *j*
S_{node}: set of nodes in neighbour table
S_{fwd}: forwarding set
S^i_{cand}: candidate nodes set of node *i*
$\mathrm{DBP}_{s(n)}$: evaluated stream DBP value of stream from node *n*
$\mathrm{DBP}_{s(i)}$: evaluated stream DBP value of stream from node *i*
$(m_{s(n)}, k_{s(n)})$: (*m,k*)-firm requirement of stream from node *n*
$(m_{s(i)}, k_{s(i)})$: (*m,k*)-firm requirement ofstream from node *i*
$m_{s(n)}$: *m* value from required (*m,k*)-firm of stream *n*
$m^i_{s(n)}$: actual value of *m* from LSI of stream *n* at node *i*

continued

$LSI^i_{s(n)}$: evaluated LSI value of stream n at node i
LSI^i_{total}: total of LSI values of all streams passing by node i
$LSI^j_{s(n)}$: evaluated LSI value of stream n at downstream node j
E^i_{rem}: remaining power of node i
e_{thd}: lower threshold of power management
$f^j_{s(n)}$: link failure level of stream n at downstream node j
$k_{s(n)}$: k value from required (m,k)-firm of stream from node n
$r^i_{min,src}$: minimum source traffic rate of node i
r^i_{src}: current source traffic rate of node i
$r^i_{adj,src}$: adjusted source traffic rate of node i
$r^i_{adj,trs}$: adjusted transit traffic rate of node i
r^i_{trs}: current transit traffic rate of node i
r^i_{out}: outgoing traffic rate of node i
$buff_i$: transit traffic buffer status of node i
n^i_{strm}: number of streams passing by upstream node i
n^j_{strm}: number of streams at node j
n^j_{up}: number of upstream nodes of downstream node j
n^i_{down}: number of downstream nodes of node i
λ_i: weight of node i
$Delay_{i,j}$: delay of transmission between node i and j
$maxDelay_{s(n)}$: maximum allowable delay of stream n

3.5 Fault recovery mechanisms

We adapt the fault recovery mechanisms we proposed in prior work [10] to this scheme. Compared with existing fault tolerant mechanisms, our mechanisms make it available to handle fault recoveries with a bounded latency that it is guaranteed all solutions used to solve 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 to increase the rate of successful transmissions and QoS performance. In this paper, we use the QoS-aware fault recovery mechanisms to handle the congestion and link failure problems during routing. An orphan node removal backpressure for void problem is utilized as well. In the fault detection 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 Table 2 as follows:

In case of non-positive stream DBP value which indicates dissatisfaction of the required QoS, we will check the LSI at node i to figure out if the performance degradation is caused by transmission fault. If LSI is not positive and link failure level f is 0, the transmission fault is determined as congestion and corresponding congestion recovery mechanism is implemented, details are elaborated in Sect. 3.5.1. Otherwise, if it indicates link failure occurrence, link failure recovery which is discussed in Sect. 3.5.2 is activated to recover the fault.

3.5.1 *Congestion recovery mechanism*

Considering the property of WSNs transmissions, we defined a new node model in [10] for congestion recovery mechanism, as shown in Fig. 7. It provides two buffer queues 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

Table 1 Algorithm: real-time routing protocol

```
Pseudo-Code Executed by Upstream Node in Each Round
STEP 1
for there is node $i$ in $S_{node}$ do
    if $Deadline_i \le Deadline_n - elapsetime$ then
        if $DBP_{s(i)} > 0$ then
            if $LSI^i_{s(n)} > 0 \&\& m_{s(n)}/k_{s(n)} \le m_{s(i)}/k_{s(i)} \&\& E^i_{rem} < e_{thd}$ then
                node $i$ is in $S_{fwd}$;
            end if
        end if
    end if
STEP 2
    if $S_{fwd} \ne \emptyset$ then
        if $DBP_{s(s)} > 0$ then
            $nextHop = \arg\max_i \{E^i_{rem}\}$;
        else
            $nextHop = \arg\max_i \{LSI^i_{s(n)}\}$;
        end if
    else
        run $VoidRecoveryMechanism$;
    end if
end for
```

Table 2 Algorithm: fault detection

Psedo-Code Executed by Upstream Node in Each Round

```
if $DBP_{s(n)} > 0$ then
    $nextHop = keepcurrentnexthop$;
else if $LSI^{i}_{s(n)} > 0$ then
    $nextHop = keepcurrentnexthop$;
                              Congestion Recovery Mechanism
else if $f^{j}_{s(n)} == 0$ then
    run $CongestionRecoveryMechanism$;
                              Link Failure Recovery Mechanism
else
    run $LinkFailureRecoveryMechanism$;
end if
```

r^i_{src} and transit traffic forwarding rate r^i_{trs} separately. The outgoing traffic rate of node *i* can be calculated by adding the two traffic rates ($r^i_{out} = r^i_{src} + r^i_{trs}$).

Based on this node model, rate adjustments can be implemented efficiently on each node. Being different from others, our mechanism is supposed to handle congestion with the awareness of real-time stream QoS guarantee. Since rate adjustment is considered to be an efficient congestion control method in WSNs [24], our 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 transit traffic rate, respectively. Two algorithms are shown in the following parts.

3.5.1.1 Sink-source node system After the calculation of stream DBP using (1), sink sends back the measured DBP value and an adjusted source traffic rate $r_{adj,src}$ to the corresponding source node in a small pre-defined time interval. We argue that this feedback process can be easily achieved, as sink is considered to be full of computing resources and based wireless communications are widely used in WSNs. When source node receives the feedback of DBP value, it adds the value into the packets it generates. The adjusted source traffic rate $r_{adj,src}$ is calculated using the algorithm in Table 3, 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 lower threshold as the minimum source rate in order to limit the performance degradation caused by excessive low source traffic rate. When sink detects that the DBP is less than 0, which indicates the stream is in negative condition, and then it will adjust the corresponding source traffic rate $r^i_{adj,src}$ to a particular level, but not less than the minimum. The calculation of adjustment is based on the deadline meeting rate of the monitored consecutive packets. Then the adjusted source traffic rate will be sent back to the source node to implement traffic limitation.

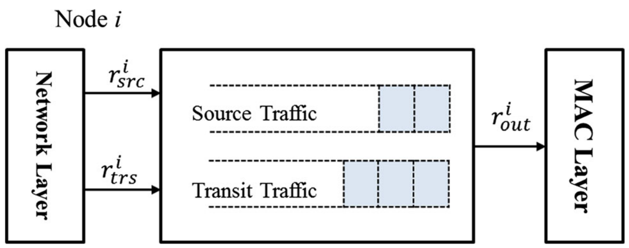


Fig. 7 Proposed Node Model

3.5.1.2 Intermediate node system Considering that big volumes of real-time 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 recovery. Our congestion recovery mechanism in [10] therefore uses the local system such as intermediate nodes, to participate in end-to-end QoS guarantee, by contributing an intermediate node congestion recovery algorithm.

Local congestion recovery mechanism is implemented at intermediate nodes, by reducing both source and transit traffic rates of intermediate node to adapt the local traffic load to the network capability. Usually it mitigates 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 [21]. We use LSI to detect the link-level congestion. The transit traffic buffer status of one node, named as *buff*, is used to monitor the node-level congestion. It could be sent as a beacon by one node to its neighbors. We argue that this beacon would not involve additional energy consumption since piggybacking scheme is used. The algorithm shown in Table 4 is supposed to detect both two types of congestions and then implement a 2-step mechanism to adjust source and 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 that only link-level congestion happens, upstream node

Table 3 Algorithm: sink-source node congestion recovery

Psedo-Code Executed at Sink in Each Round

```
if $DBP_{s(n)} \leq 0$ then
    if $r^i_{src} > r^i_{min,src}$ then
        $r^i_{adj,src} = \max\{r^i_{src} \cdot (1 - \frac{k_{s(n)} - |DBP_{s(n)} - 1|}{k_{s(n)}}), r^i_{min,src}\}$;
    end if
end if
```

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 is measured according to the total LSI values of all streams passing by and the outgoing traffic rate of downstream node as well.

3.5.2 Link failure recovery mechanism

This mechanism is used for nodes to recover link failure by choosing the optimal forwarding nodes for redundancy on multipath [10]. Different from existing fault tolerant schemes, our mechanism makes it available to establish multiple transmission paths with a bounded latency during transmissions. It is guaranteed that all selected nodes to forward 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. We therefore use this delay-aware link failure recovery algorithm to dynamically choose the optimal forwarding nodes which can guarantee both required reliability and bounded delay, to make it more adaptable for real-time applications than other works. The algorithm is listed in Table 5.

This algorithm shows how upstream node i makes decisions on which downstream node should be chosen as a candidate node. First, if both DBP and LSI values of stream n are smaller than 0, it indicates an unsatisfied stream end-to-end QoS guarantee and a high possibility of current node is the cause of performance degradation. Moreover if the link failure level f is not equal to 0, then the link failure recovery mechanism would be activated to figure out a proper set of candidate nodes from its neighbors for multipath establishment. The maximum allowable delay of current stream is calculated, and within this time period, packets arrived at sink could be considered as useful. For an upstream node i, to choose a proper forwarding candidate from all its downstream nodes, is to select the one that could keep the stream QoS guarantee. And then node i would add this node into its candidate nodes set. That is, all nodes in that set are supposed to be able to guarantee a bounded delay of packets.

However, not all nodes in that set are required in a case of densely employed network that there may be more than needed candidates are available. A calculation for the required number of forwarding paths should be done consequently, according to the actual situation. Two equations are used here for both source node and intermediate nodes to make decisions to choose the optimal number of alternative paths needed for redundancy, from their candidate nodes set, respectively.

For source node, the most useful information is the stream DBP value it receives as feedback from sink. So the adapted number of alternative paths could be calculated using the following equation:

$$P_{src} = \min\{|DBP - 1|, S_{cand}\} \tag{4}$$

where P_{src} is the optimal number of forwarding paths for multipath establishment on source node.

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 the following equation to calculate optimal number of alternative paths:

$$P_{int} = \min\{|LSI - 1|, S_{cand}\} \tag{5}$$

similar to (4), in (5) 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's possible that once an intermediate node detects link failure on primary path, it finds no

Table 4 Algorithm: intermediate nodes congestion recovery

```
Psedo-Code Executed by Upstream Node i in Each Round
if $DBP_{s(n)} \leq 0$ then
    if $LSI^i_{s(n)} \leq 0$ && $buff_i \neq overflow$ then
        if $r^i_{src} > r^i_{min,src}$ then
            $r^i_{adj,src} = \max\{r^i_{src} \cdot (1 - \frac{m_{s(n)} - m^i_{s(n)}}{m_{s(n)}}), r^i_{min,src}\}$;
        end if
    end if
Psedo-Code Executed by Downstream Node j in Each Round
    if $LSI^i_{s(n)} \leq 0$ && $r^i_{src} == r^i_{min,src} || buff_i \neq overflow$ then
        $LSI^i_{total} = \sum_{n=1}^{n^i_{strm}} (|LSI^i_{s(n)}| + 1)$;
        $\lambda_i = \frac{LSI^i_{total}}{\sum_{i=1}^{n^i_{strm}} LSI^i_{total}}$;
        $r^i_{adj,trs} = r^i_{out} \cdot \lambda_i$;
    end if
end if
```

Table 5 Algorithm: link failure recovery

```
Psedo-Code Executed at Upstream Node in Each Round
if $DBP_{s(n)} \leq 0$ then
    if $LSI^i_{s(n)} \leq 0$ then
        if $f^i_{s(n)} \neq 0$ then
            $maxDelay_{s(n)} = Deadline_{s(n)} - Delay_{i,j}$;
            for each $j in [1, n^i_{down}]$ do
                for each $i in [1, n^j_{strm}]$ do
                    if $Deadline^j_{s(n)} < maxDelay_{s(n)}$ && $LSI^j_{s(n)} > 0$ then
                        j is in $S^i_{cand}$;
                    end if
                end for
            end for
        end if
    end if
end if
```

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.

3.5.3 Void recovery mechanism

In WSNs, backpressure scheme is often used for re-routing or notification delivery. In [10], the void avoidance mechanism uses backpressure only to remove 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 routing schemes. Once an intermediate node updates its neighbor table and finds no candidates in the forwarding set, it will send backpressure beacon which is introduced in Sect. 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.

4 Performance evaluation

Performance of the proposed scheme is proved by simulations. We use NS-2 as the simulator. The simulation terrain is set as a 200 m × 200 m field. 3 source nodes are randomly selected, and the event area radius is set as 50 m. Sink is located at the lower right corner of the field so that 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 three scenarios for performance evaluation. In the first scenario, 2 source nodes are supposed to generate periodic traffic and the rest one generates aperiodic bursty traffic. This scenario is used to prove the adaptability of the proposed scheme when it faces a rapid change of data volumes. The second scenario contains various channel errors during transmissions in order to estimate the usability of the proposed scheme. In the third scenario, the packet rate increases monotonically, so the lifetime of the network can be evaluated under different workloads.

Evaluation results are presented as follows: (1) packet deadline missing ratio (PDMR), (2) stream dynamic failure ratio (SDFR), (3) energy-drained node ratio (EDNR). The first one refers to the timeliness of individual packet, which is considered as the most important feature in real-time application. The second one is supposed to measure the QoS guarantee in both reliability and timeliness, they are the main reasons of dynamic failure in real-time applications. The third one shows the efficiency of the proposed scheme in case of prolonging the lifetime of network. It's supposed to be energy efficient that it does not introduce much overhead to the resources, so the nodes will not drain fast.

4.1 Packet deadline missing

Both Figs. 8 and 9 plot the PDMR of three algorithms: SPEED, the proposed scheme with (3,5)-firm and (4,5)-firm guarantees. The packet deadline is set to be 40 ms for all three algorithms.

We chose the traffic of one source node from two periodical traffic nodes as target traffic, so that the horizontal axis of Fig. 8 stands for the ratio of the target traffic to all traffics in network, i.e. 30 % means 30 percent of total traffic is the target traffic and the rest 70 percent is background traffic. The less the value, the more traffic load it bears in network. Especially for the nodes which are closer to sink, the probability of congestion is much higher than other nodes. In Fig. 8 we can learn that the traffics transmitted using SPEED experience more than 30 % 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 the proposed scheme helps the intermediate nodes to build the primary route with optimal forwarding nodes, to meet timeliness requirement of the stream, and also handle the problems to reduce delay and remain an acceptable performance of QoS guarantee.

Simulation result of Fig. 9 shows when channel error happens and increases proportionally, PDMR raises dramatically in SPEED since it has no failure managements to handle the link failure. On the other hand, even under unstable network condition, LSI works well to indicate the distance to failure and distinguish between different faults. Based on both LSI and stream DBP, the proposed scheme is able to handle link failure efficiently, by make limited redundancy on guaranteed multipath, to reduce latency caused by link failure and increase reliability. The difference between (3,5)-firm stream and (4,5)-firm stream in Fig. 9 is that according to the mechanism of LSI, (4,5)-firm stream has more strict 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 dynamic failure

In Figs. 10 and 11, we evaluate the SDFR of three algorithms: SPEED, the proposed scheme with (3,5)-firm and (4,5)-firm guarantees. The packet deadline is also set to be 40 ms for all three algorithms.

The horizontal axes of Figs. 10 and 11 are the same as in Figs. 8 and 9, respectively. For real-time applications, the SDFR is mainly affected by packet deadline missing and packet loss. So in these two simulations, we try to prevent dynamic failure by reducing PDMR and packet loss. The significantly rising curves of SPEED in both figures demonstrate that without fault-tolerant mechanisms, it's hard to provide high performance of QoS in case of heavy traffic or instable network environment. Together with stream DBP, the proposed LSI plays a very important role in packets transmission since it makes all intermediate nodes to be aware of local transmission status to the next hop, so that it can make optimal decisions for forwarding and fault recovery. The congestion recovery mechanism and link failure recovery mechanism can effectively handle the problems to maintain the reliability and timeliness of transmissions, without introducing much extra overhead to latency. The proposed scheme would be highly desired by firm real-time stream applications. By distributing the duty of (*m*,*k*)-firm guarantee from sink to each intermediate node, LSI and stream DBP together make it possible to keep good QoS performance of real-time applications.

4.3 Energy efficiency

In Fig. 12, we evaluate the NLT of three algorithms: SPEED, the proposed scheme with (3,5)-firm and (4,5)-firm

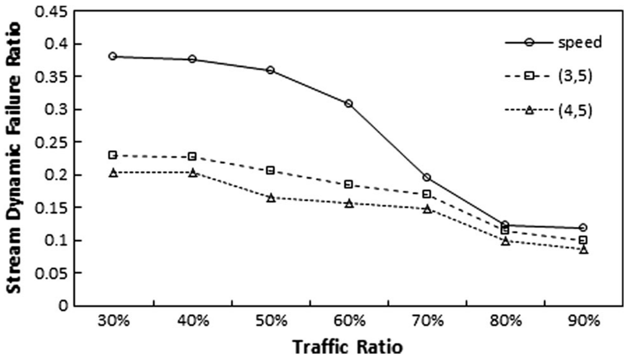


Fig. 8 PDMR under various traffic ratios

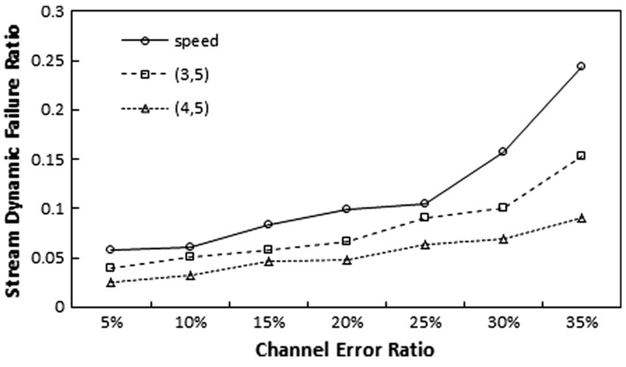


Fig. 9 PDMR under various channel error ratios

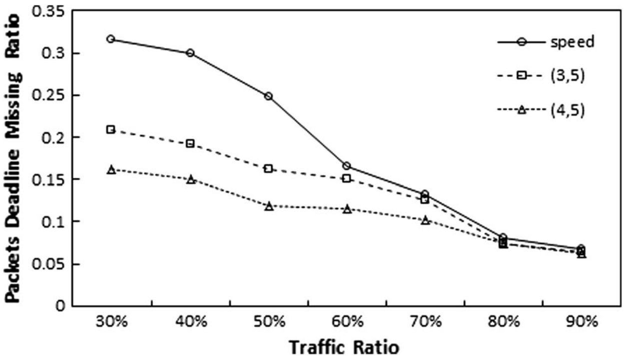


Fig. 10 SDFR under various traffic ratios

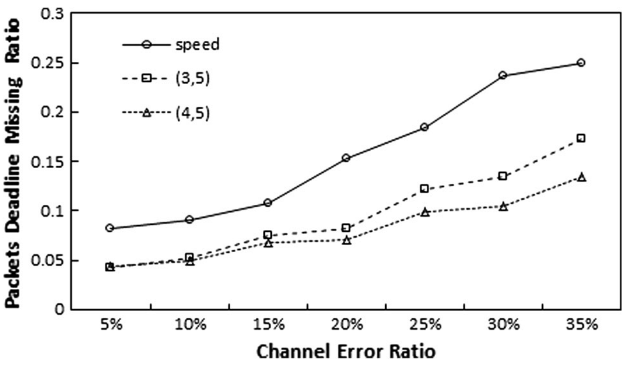


Fig. 11 SDFR under various channel error ratios

guarantees. We monotonically increase the packet generating rate, to figure out that whether or not the employed nodes will drain fast, under different packet rates.

Under the resource constraints, it is vital for sensor nodes to minimize energy consumption in radio communication to prolong the lifetime of the network. From Fig. 12, we argue that the proposed schemel tends to be energy efficient in packet transmissions, and compared with SPEED, it does not introduce much overhead to resources. When the packet rate increases, the proposed scheme with (4,5)-firm has nearly the same percentage of nodes drained as SPEED, because it has stricter QoS requirement than the one with (3,5)-firm, which is supposed to consume more energy for transmission to meet the requirement.Under heavy traffic condition, the proposed scheme still shows energy efficiency during packet transmission.

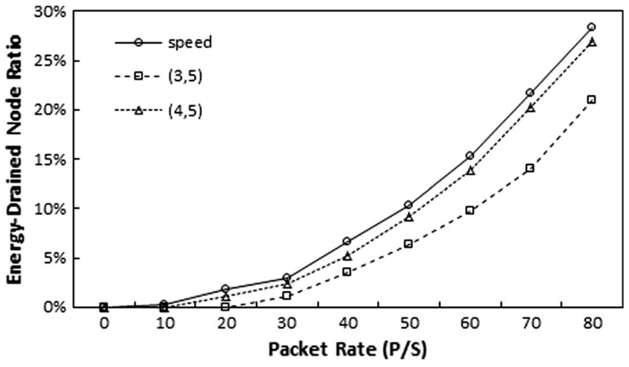


Fig. 12 EDNR under various packet rate

5 Conclusion and future works

The current challenge of WSNs is to implement real-time applications in a resource constrained network. It requires efficient fault-tolerant routing schemes since compared with non-real-time application, the QoS requirement of real-time applications are much more difficult to be satisfied due to the inherent constraints of WSNs. The proposed scheme uses 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, each node makes optimal forwarding decision, and implements different fault recovery mechanisms to handle congestion, link failure and void problems. This adaption capability makes the proposed scheme more functional in simulations, comparing to SPEED. Simulation results show that due to the contribution of each component, the proposed scheme performs much better in timeliness and QoS guarantee features with less deadline missing and dynamic failure, without introducing much overhead.

The future work goes to the cross-layer scheme design, for (*m*,*k*)-firm based real-time applications.

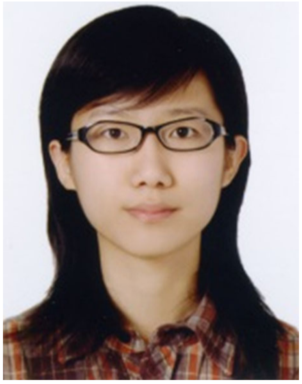
Acknowledgments This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2013R1A1A2A10004587) and the BK21+ program, Korea.

References

1. Li, Y., Chen, C.S., Song, Y.Q., & Wang, Z., et al. (2007). Real-time qos support in wireless sensor networks: a survey. In *7th IFAC international conference on fieldbuses & Networks in Industrial & Embedded Systems-FeT'2007*, 2007.

2. Teixeira, T., Culurciello, E., Park, J.H., Lymberopoulos, D., Barton-Sweeney, A.,& Savvides, A. (2006). Address-event imagers for sensor networks: evaluation and modeling. In *Proceedings of the 5th international conference on Information processing in sensor networks* (pp. 458–466). ACM, 2006.
3. Xiao, Y., Peng, M., Gibson, J. H., Xie, G. G., Du, D.-Z., & Vasilakos, A. V. (2012). Tight performance bounds of multihop fair access for mac protocols in wireless sensor networks and underwater sensor networks. *IEEE Transactions on Mobile Computing, 11*(10), 1538–1554.
4. Wang, X., Vasilakos, A. V., Chen, M., Liu, Y., & Kwon, T. T. (2012). A survey of green mobile networks: Opportunities and challenges. *Mobile Networks and Applications, 17*(1), 4–20.
5. Chilamkurti, N., Zeadally, S., Vasilakos, A., & Sharma, V. (2009). Cross-layer support for energy efficient routing in wireless sensor networks. *Journal of Sensors*, 2009.
6. Xiang, L., Luo, J.,& Vasilakos, A. (2011). Compressed data aggregation for energy efficient wireless sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2011. SECON 2011. 2011 IEEE Communications Society Conference on*, (pp. 46–54). IEEE.
7. Ehsan, S.,& Hamdaoui, B. (2012). A survey on energy-efficient routing techniques with qos assurances for wireless multimedia sensor networks. *Communications Surveys & Tutorials, IEEE, 14*(2), 265–27.
8. Hamdaoui M.,& Ramanathan, P. (1995). A dynamic priority assignment technique for streams with (m, k)-firm deadlines. *IEEE Transactions on Computers, 44*(12), 1443–1451.
9. Li, B., & Kim, K. I. (2012). A novel routing protocol for (m, k)-firm-based real-time streams in wireless sensor networks. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*. (pp. 1715–1719). IEEE, 2012.
10. Li, B.,& Kim, K. I. (2012). An-firm real-time aware fault-tolerant mechanism in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2012.
11. He, T., Stankovic, J.A., Lu, C.,& Abdelzaher, T. (2003). Speed: A stateless protocol for real-time communication in sensor networks. In *Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on*. (pp. 46–55). IEEE, 2003.
12. Felemban, E., Lee, C.G.,& Ekici, E. (2006). Mmspeed: multipath multi-speed protocol for qos guarantee of reliability and. timeliness in wireless sensor networks. *IEEE Transactions on Mobile Computing, 5*(6), 738–754.
13. Chipara, O., He, Z., Xing, G., Chen, Q., Wang, X., Lu, C., Stankovic, J.,& Abdelzaher, T. (2006) Real-time power-aware routing in sensor networks. In *Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on*. (pp. 83–92). IEEE, 2006.
14. Kandris, D., Tsioumas, P., Tzes, A., Nikolakopoulos, G., & Vergados, D.D. (2009). Power conservation through energy efficient routing in wireless sensor networks. *Sensors, 9*(9), 7320–7342.
15. Kandris, D., Tsagkaropoulos, M., Politis, I., Tzes, A., & Kotsopoulos, S. (2011). Energy efficient and perceived qos aware video routing over wireless multimedia sensor networks. *Ad Hoc Networks, 9*(4), 591–607.
16. Ahmed, A. A.,& Fisal, N. (2008). A real-time routing protocol with load distribution in wireless sensor networks. *Computer Communications, 31*(14), 3190–3203.
17. Zhao, L., Kan, B., Xu, Y.,& Li, X. (2007). Ft-speed: A fault-tolerant, real-time routing protocol for wireless sensor networks. In *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*. (pp. 2531–2534). IEEE, 2007.
18. Akan, O. B.,& Akyildiz, I. F. (2005). Event-to-sink reliable transport in wireless sensor networks. *IEEE/ACM Transactions on Networking (TON), 13*(5), 1003–1016.
19. Deb, B., Bhatnagar, S.,& Nath, B. (2003). Reinform: Reliable information forwarding using multiple paths in sensor networks. In *Local Computer Networks, 2003. LCN'03. Proceedings. 28th Annual IEEE International Conference on*, (pp. 406–415). IEEE, 2003.
20. Wu, G., Lin, C., Xia, F., Yao, L., Zhang, H.,& Liu, B. (2010). Dynamical jumping real-time fault-tolerant routing protocol for wireless sensor networks. *Sensors, 10*(3), 2416–2437.
21. Yaghmaee, M. H.,& Adjeroh, D. (2008). A new priority based congestion control protocol for wireless multimedia sensor networks. In *World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a*. (pp. 1–8). IEEE, 2008.
22. Kurose, J.,& Ross, K. (2006) *Computer Networks: A top down approach featuring the internet*. London: Pearson Addison Wesley
23. Wan, C.Y., Eisenman, S.B.,& Campbell, A.T. (2003). Coda: congestion detection and avoidance in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*. (pp. 266–279). ACM, 2003.
24. Paradis, L., & Han, Q. (2007). A survey of fault management in wireless sensor networks. *Journal of Network and Systems Management, 15*(2), 171–190.

Author Biographies



Bijun Li received the M.S. degrees in Informatics from the Gyeongsang National University, Jinju, Korea, in 2013. Her research interests include wireless sensor networks, and routing protocol.

Ki-Il Kim received the M.S. and Ph.D. degrees in computer science from the ChungNam National University, Daejeon, Korea, in 2002 and 2005, respectively. He is currently with the Department of Informatics, Gyeongsang National University. His research interests include routing for MANET, QoS in wireless network, multicast, and sensor networks.