Sensors & Transducers, Vol. 18, Special Issue, January 2013, pp. 188-202

www.sensorsportal.com

Sensors & Transducers

ISSN 1726-5479

http://www.sensorsportal.com

On QoS Guarantees of Error Control Schemes for Data Dissemination in a Chain-based Wireless Sensor Networks

Zahra Taghikhaki, Nirvana Meratnia, Paul J.M. Havinga
Pervasive Systems, University of Twente, Enschede, 7522 NB, The Netherlands
(z.taghikhaki, n.meratnia, p.j.m.havinga)@utwente.nl

Abstract: Time-critical applications of Wireless Sensor Networks (WSNs) demand timely data delivery for fast identification of out-of-ordinary situations and fast and reliable delivery of notification and warning messages. Due to the low reliable links in WSNs, achieving real-time guarantees and providing reliable data is quite challenging. Reliable data dissemination is traditionally performed by applying error control mechanisms. However, these mechanisms are not always suitable for time critical applications of WSNs, in which packet burst loss occurs.

In this paper, we compare different error control schemes in terms of their reliability, energy efficiency and real-timeness guarantees to assess the appropriateness of each method facing different conditions. Additionally, we introduce and evaluate an enhanced version of a real-time error control scheme, called READ-MN, in terms of its Quality of Service (QoS) guarantees.

Keywords: QoS, reliability, real-timeness, error control, energy efficiency, duty cycle, wireless sensor network.

1. Introduction

Wireless sensor networks are one of the promising technologies for monitoring applications such as structural health monitoring. Monitoring operational performance of large linear civil engineering (infra)structures such as bridges, tunnels, and highways restricts sensor network deployment to long stretch of narrow and elongated spreads. As the length of these (infra)structures is often much greater than their width, their topologies resemble a long chain. The elongated area of interest may extend from a few tens of meters to a few hundreds of kilometers in length. Long linear chain-type sensor networks have often a large number of hop counts. To operate for a long time, they usually need to work on a low duty cycle. The large number of hop counts challenges existing data dissemination protocols already designed for WSNs, while the low duty cycle introduces extra delays.

Time-critical applications highly depend on the availability of real-time data as in these applications data is not valuable if it is received after its Time To Live (TTL). Outdated data is not only useless but may also be harmful as it may have negative impacts on the decisions made by providing invalid information. Moreover, transmitting expired data depletes the energy of relaying nodes inappropriately. Due to the harsh transmission environment, providing real-time guarantees and data reliability in WSNs is quite challenging. Most of existing real-time algorithms applied networks other than WSNs assume network is reliable and packets are not lost because of unreliable links. Therefore, they cannot be directly applied to WSNs. The higher the packet loss due to unreliable links, the lower the performance of a real-time WSN.

Reliable data dissemination is traditionally performed by applying error control protocols, which

could provide an adequate degree of quality even in the presence of errors. There are two key error control strategies in WSN for maintaining reliable communication over noisy channels. The first one is Forward Error Correction (FEC) [1], which relies on transmission of redundant data to allow the receiver node to reconstruct the original messages. The second strategy is Automatic Repeat Request (ARQ) [2], in which high-rate detection codes are normally used and a re-transmission is requested if the received data is found to be erroneous. In other words, ARQ tries to retransmit the lost or erroneous packets, while FEC adds some redundancy to the original message to be able to recover the lost or erroneous packets. The main disadvantage of ARQ is that it wastes time waiting to receive ACKs, which in turn leads to low throughput. FEC, on the other hand, imposes a permanent bandwidth overhead for the redundant information regardless of the channel condition. Additionally, FEC is designed to tolerate the expected worst-case error rate and it is not robust enough to handle packet burst loss, which is likely to occur in wireless links. FEC is often used in the networks, in which errors tend to modify just a few bits at a time but it cannot guarantee full reliability in networks with high error rates unless it is coupled with ARQ. The scheme combining ARQ with FEC is called Hybrid ARQ (H-ARQ) [2], which is an approach aiming to recover from lost or erroneous packets for near real-time communications. H-ARQ, however, cannot assure a delay bounded transmission.

Motivated to overcome the drawbacks of error controlling schemes and make them suitable for the unreliable and delay bounded transmission, in this paper we introduce a new error control scheme, which unlike existing techniques combines real-time and reliability guarantees for each packet and increases hit ratio (the percentage of the packets received by the base station before their deadline expire). To deal with the energy consumption and in order to enrich data, we utilize data aggregation on the intermediate nodes as far as it does not influence packet deadline. The packets that are more likely to not reach the Base Station (BS) within their TTL are dropped in order to save energy of the intermediate nodes. Moreover, duty cycling is employed to make sensor nodes capable of operating for a long time.

The rest of this paper is organized as follows. Firstly we briefly discuss the state of the art in Section II. Then some preliminaries of this study will be presented in section III, followed by detailed description of the READ, READ+, READ-MN. Performance evaluation will be presented in Section IX, while we draw some conclusions in Section X.

2. Related work

2.1 Reliable and Real-time Data aggregation

Several data aggregation protocols have been proposed for WSNs in the past. However only a very few of them consider both reliability and timeliness and aim to ensure them simultaneously. Real-time guarantees are usually provided through either real-time scheduling or real-time routing. SPEED [4] is a well-known protocol addressing soft real-time guarantee in WSNs in such a way that packet deadline is mapped to a velocity requirement. The node with a velocity higher than a specified requirement is more likely to be chosen as the upstream node. MMSPEED [5], which is an enhanced version of SPEED, aims to meet reliability and timeliness requirements together while utilizing multipath routing to handle reliability such that number of paths is in direct proportion with the required reliability. Timeliness is supported by combining the SPEED idea with packet prioritization, which is done on the basis of the required speed for each packet. R2TP [6] uses a reliable and real-time data dissemination, in which reliability is satisfied by sending several copies of one packet through multiple paths such that sum of the reliability of the considered paths is equal or higher than the requested reliability. The packet is dropped by the intermediate nodes if the elapsed time of a given node is greater than the delivery time requirement. Otherwise, it forwards that packet through multi paths using the given node's table, which stores the delay of different paths. Soyturk et al. [7] present a reliable data acquisition approach for time-critical application of WSNs. Reliability is provided similarly to techniques of [5][6] leveraging multipath approach, while real-time concern is supported by prioritization of the packets. This technique, therefore, deals with the priority scheduling in order to handle queuing delay, which is the main cause of making end-to-end latency. Almost all of the aforementioned approaches support reliability by sending several copies of a packet through different paths. To the best of our knowledge, there is no well-explored work to address these two quality of service (QoS) parameters, i.e., reliability and timeliness together in a chain-based WSN, in which only one (or a few) path(s) can be established between source and destination nodes. Moreover, since approaches of [5][6] are proposed for data dissemination rather than data aggregation, they must employ other methods to filter out redundant data in case of availability of duplicate sensitive aggregation functions like sum or average. QoS-ACA [3] aims to fast, reliably, and energy efficiently aggregate data in a chain-based WSN and send the aggregated value to a BS. To ensure reliability, it leverages the benefits of retransmission without using any acknowledgement. It utilizes the optimum number of retransmissions to ensure the required reliability. It also considers the residual and required energy of each sensor node and the distance between node and the BS as two main criteria to select a node as an aggregator. However, it does not guarantee delivery of a packet to the BS within its deadline.

2.2 Error control

Wireless networks often apply error control mechanisms as wireless channels can be easily affected by unpredictable factors such as weather, obstacles, shadowing, and mobility. ARQ and FEC are two main error control approaches often used. Generally speaking considering the way retransmission takes place, there are three types of ARQ protocols, namely, stop-and-wait (S-W), go-back-n, and selective repeat [1][2]. Stop-and-wait is the simplest version of ARQ, in which the sender transmits the packets, stops, and waits (idling) for an acknowledgement (ACK) or Non-acknowledgement (NACK) from the receiver before it continues with further transmissions. The idling time waiting for receiving the acknowledgement makes this scheme inefficient. The advantage of stop and-wait ARQ is that it only requires a half-duplex channel. As go-back-n and selective repeat are continuous in ARQ, they require a full duplex channel because packets/codewords are sent continuously until a NACK is received. In addition to the acknowledgement overhead and the need of return channel, losing ACK or NACK packets which is more likely to occur in unreliable WSNs contributes to inefficiency of ARQ. Almost all existing ARQ protocols assume the acknowledgement packets are never lost, which is an unrealistic assumption for WSNs. If the acknowledgement packet is lost or becomes erroneous due to link/network failure, sender continues sending copies of the received data even if data is already received. This leads to high energy dissipation and wasting bandwidth. If NACK packets are lost, sender will never be informed about erroneous or loss packets and thereby the reliability cannot be ensured. FEC is another error control approach performed by adding redundancy to the transmitted information using a predetermined algorithm. There are different FEC encoding schemes utilized to mathematically generate parity data from source data. Each FEC scheme has a different complexity level and different error recovery efficiencies. The simplest way to generate parity is the use of exclusive OR (XOR) [8], which generates one parity for specific amount of original data. The XOR encoding has very low processing complexity but it can only repair a single codeword/packet loss in a transmission group. Reed-Solomon (RS) [8] code is a famous technique to generate multiple parities for each transmission group in order to provide better and efficient protection against losses. This better flexibility rather than XOR of FEC approach comes at the expense of higher processing and memory usage. FEC functions well in presence of random packet loss but it is not robust enough to handle packet burst loss, which is likely to occur in wireless channels. A drawback of FEC is that regardless of information correctness, the decoded information is always delivered to the destination. As the basic FEC cannot be adapted for time-varying channel states, a fixed coding scheme is chosen to encode some information packets. By doing so, bandwidth is wasted in case of low error rate of the channel as there is no need to have the redundant information. ARQ approach, on the other hand, is suitable in case of having return channel which may not be available and also works well for the delay tolerable applications such as file transfer. The main advantage of ARQ over FEC is that it has a simpler decoding. All in all it can be said that although compared with FEC, ARQ can provide higher reliability, it wastes more time for receiving ACKs. This results, in turn, in higher delay and makes ARQ not suitable for delay constrained data dissemination.

To address above challenges, an error control is needed which is able to (i) shorten the delay of the ARQ, (ii) alleviate the impact of lost acknowledgements, (iii) maintain the reliability of ARQ, (iv) ensure energy efficiency, and (v) guarantees packet delivery before their TTL expires.

3. Preliminaries

3.1 Quality of Service Parameters

3.1.1 Real-timeness

An increasing number of WSN applications require real-timeness as their QoS parameter. Applications may have one of the following four notions of time:

- Time-unrestricted: which indicates no dedicated deadline exists and application at hand is not time critical.
- Soft Real Time (SRT): based on which the usefulness of a packet received after its deadline decreases, which in turn results in a graceful degradation of the performance. SRT-based approaches aim to reduce deadline miss ratio of the packets and are common in WSNs because of the unpredictability nature of these networks.
- Firm Real Time (FRT): on which, the usefulness of a packet received after its deadline is Zero. FRT methods can tolerate infrequent deadline misses.
- Hard Real Time (HRT): HRT applications highly rely on receipt of all packets before their deadline ends.

3.1.2 Reliability

Another QoS parameter requirement of many WSNs applications is reliability. One commonly used approach to ensure reliable data delivery in a failure prone environment is sending several copies of one packet from a single source node towards the destination node. To know whether data is received by the destination, one of the following techniques is used:

- Sending an acknowledgement: in this technique if the acknowledgement packet is lost due to link/network failure, source node continues sending copies of the received data, which leads to high energy dissipation.
- Sending multiple copies without sending any

acknowledgement: although this approach reduces the acknowledgement overhead, it requires a solution to ensure data reach to the destination after sending *n* copies of a packet.

3.1.3 Energy efficiency

Energy efficiency has the highest priority in WSNs to ensure long network life time. As one of the most energy-expenditure operations is transmitting data, each sensor node often turns its radio off and goes to asleep state most of the time to obtain significant energy saving. In a duty-cycle-based power management scheme, each sensor node goes to sleep and wakes up periodically. The proportion of the time that each sensor node spent in sleep mode has direct impact on the data delivery delay, packet loss, and throughput. The shorter the duty cycle, the lower the event detection probability and the longer detection delay. In a scheduling scheme, a sensor node is allowed to switch between three operation modes:

- Sleep mode: which results in low power consumption. In this state the radio of a node is turned off but the sensors may be operational.
- Active mode: which itself includes two operational states: receiving state (RX), and transmitting state (TX).
- Idle state: in which radio is ready to receive or transmit data. According to the conditions, the radio is changed to the appropriate active state.

Fig.1. presents the state diagram illustrating the main states of the radio and the ways state transitions occur. Once the sleeping time (T_S) is over, the radio must undergo a transition to idle state. On the other hand, the radio of a node must be switched off as soon as the active time (T_A) is finished. It is worth noting that these four states have different levels of energy consumption, which differ from one radio model to another.

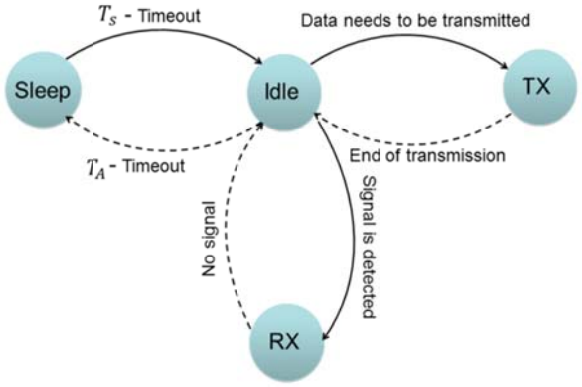

Fig. 1. State diagram for radio states

4. Network Model

We make the following assumptions regarding the WSN. The WSN consists of *N* sensor nodes deployed in a linear topology and one BS is located at the end of the chain. As each chain should have a chain leader through which sensor data is forwarded to the BS, the role of being leader rotates among sensor nodes considering some criteria, which will be explained later. In case of not being a chain leader, sensor nodes can only communicate with their direct neighbors, hence the power level of them is adjusted by taking the distance to the closest upstream neighbor into account. The location of sensor nodes and the BS are fixed and are known a priori. We have chosen for this network model as this is the case in many structural health monitoring applications, in which sensor nodes are placed at known and fixed locations of a bridge, for instance, at critical locations in a long linear array form and send their data periodically or upon detection of abnormal situations via relaying their data to neighboring nodes to the BS. Our network has a three-tiers architecture in which the first tier includes regular sensor nodes, the second tier includes chain leaders, and the third tier includes BS. It is clear that such hierarchical architecture can be easily expanded to include more chain leaders in the second tier and more BSs in the highest tier for a long range chain-type sensor networks (Fig.2.). Additional tier can also be added between chain leaders and BSs based on specific application need. Therefore, the proposed architecture is easily scalable to increase the size of the network. In this case, the size of the network corresponds to the length of an elongated topology. Without loss of generality, in this study we assume that there is only one chain leader (in the second tier) and one BS (in the third tier). Managing second and third tiers has been explained in detail in [3].

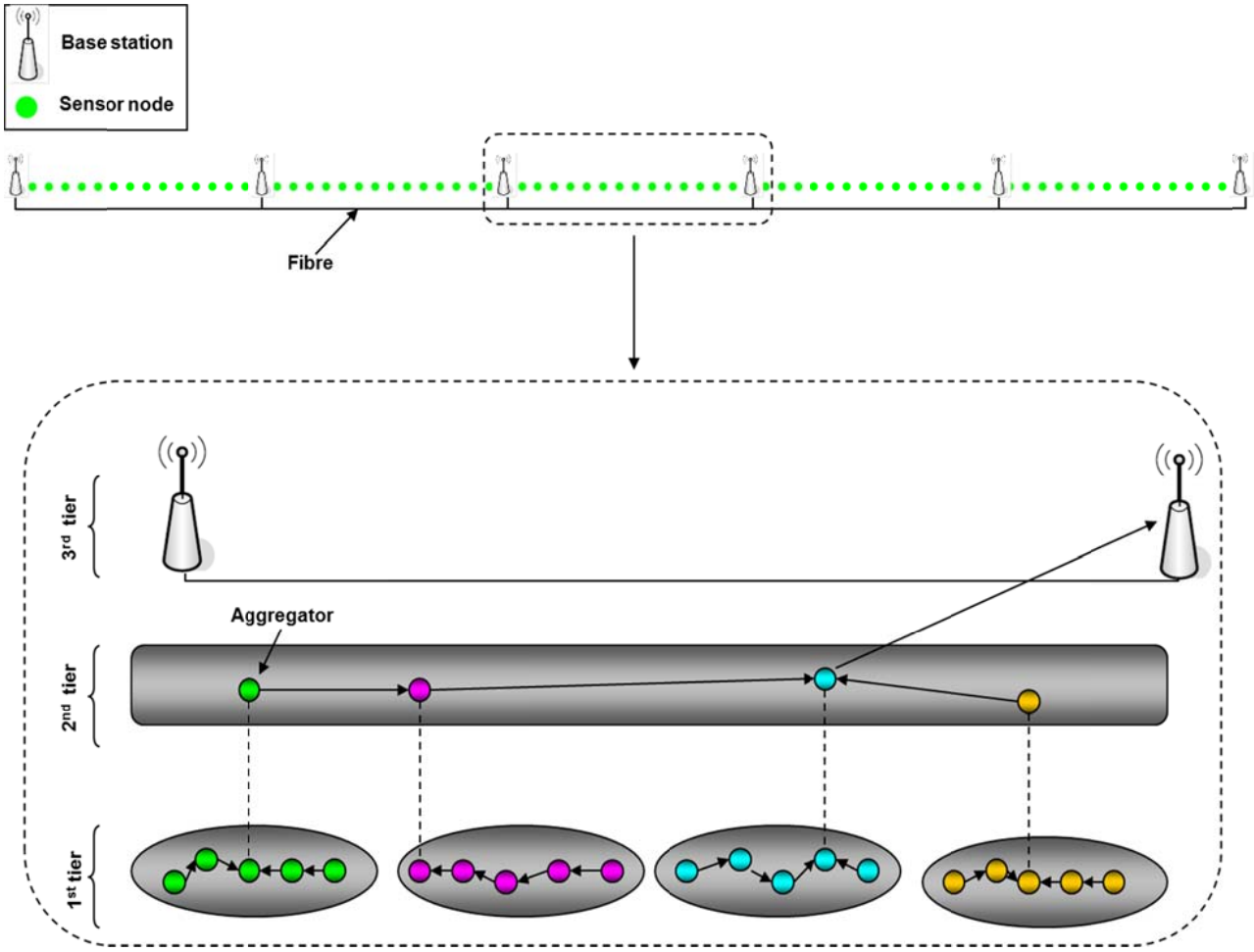

Fig. 2. Three-tiers architecture model

Every sensor node in a chain must send its data to its upstream neighbor, which is selected in the chain construction phase. Intermediate nodes along the path to the chain leader aggregate the data received from the downstream nodes with their own data (if any) and forward the local aggregated value towards the chain leader. The chain leader, also called the aggregator, must perform final aggregation on the data received from two sides of the chain and then forward the result to the BS directly.

To motivate the need to address both data reliability and real-timeness in our protocol, let us consider the network illustrated in Fig. 3., which consists of six sensor nodes such that one of them is selected as the chain-leader and a packet, whose TTL is 10s, should be forwarded from S_0 towards the leader. Let us assume that time required to deliver a packet from S_0 to the leader is 3s and from the leader to the BS is 1s. Clearly, this packet will be received by the BS after 4s. This implies that 6s from its TTL is remained, which can be exploited to achieve higher network performance. We can spend this time for either (i) increasing aggregation degree of the leader or (ii) improving transmission reliability of the network. If the network has high reliable links and it is almost guaranteed that the packet is received by destination through the first transmission, it is better to spend this remaining time for the aggregation process and to increase aggregation degree of the leader. In this case, leader can put the received packet on hold and perform aggregation on other packets which are on the way and will be received within limited time duration of the waiting packet. The remaining TTL time can also be used to improve transmission reliability by utilizing a retransmission mechanism and sending several copies of the given packet. This is particularly useful when network suffers from packet loss.

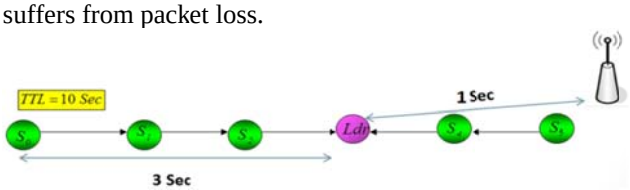

Fig. 3. An example of a chain based network

5. An overview of READ

To eliminate the delay and transmission overheads introduced by acknowledgements in ARQ, we aim to

assure reliability by sending multiple copies of one packet without sending any acknowledgement. Even though this approach reduces the acknowledgement overhead and delays, it requires a solution to ensure data to reach to the destination after sending some copies of a packet. QoS-ACA [3], which is an approach to guarantee reliability by sending several copies of one packet, estimates the optimal number of retransmissions for each link based on the requested reliability of the application and packet loss rate of the given link. However, QoS-ACA does not care about real-timeness and only aims to ensure high reliability for a delay tolerable application. Thereby, we cannot utilize this approach and we require to estimate number of copies for each link having reliability of the links in mind while keeping an eye on the packet TTL. Since receiving a packet after its deadline is not only useless but also depletes energy, it is highly preferable to drop such packets to prevent wasting energy of the intermediate nodes relaying the packet. A key question here is how to assign the remaining TTL of a given packet to relaying nodes for their retransmission or in another word, for how long a packet can be delayed on the intermediate nodes so that the reliability gain and on-time end-to-end delivery ratio can still be maximized. We answered this question by introducing READ [9] that is a fair and simple heuristic which allocates the available packet TTLs proportionately to the packet loss probability of the links along the forwarding path to judiciously and fairly uses the packet TTLs on intermediate nodes in such a way that reliability gain and on-time end-to-end delivery ratio are maximized.

6. A Detailed Description of READ

READ starts with chain construction using PEGASIS algorithm proposed in [11]. In a given chain, one node must be selected as the leader in order to do the final aggregation and to send the aggregated local-view data stream to the BS. Two QoS parameters, i.e., reliability and energy consumption as well as two assigned weights will be considered to make different criteria for electing a leader. To this end, we introduce the following equations:

$$B^T(S'_j) = (B^R(S'_j))^{W_R} \times (B^E(S'_j))^{W_E} \quad (1)$$

$$B^E(S'_j) = \left(\frac{RsdEg(S'_j)}{IniEg(S'_j) \times RqEg(S'_j)}\right) \quad (2)$$

$$B^R(S'_j) = \frac{1}{N-1} \times \sum_{i=0}^{N} EER(S_i, S'_j) \quad (3)$$

$$EER(S_i, S_{CL}) = \begin{cases} \prod_{k=i}^{CL-1} HHR(S_k, S_{k+1}) & CL > i \\ \prod_{k=CL}^{i-1} HHR(S_k, S_{k+1}) & CL < i \end{cases} \quad (4)$$

Where S' represents a set of sensor nodes, which are able to directly communicate with the BS and CL represents the candidate leader. The hop-by-hop reliability (HHR) between two sensor nodes will be obtained using $HHR(S_i, S_{i+1}) = 1 - p_{pktloss}(S_i, S_{i+1})$. By having the hop-by-hop reliabilities, BS must evaluate the appropriateness of each member of S' to be an aggregator. To this end, BS first calculates the end-to-end reliability from each sensor node to the designated leader by employing equation (4). At the second step, BS finds the benefit of each candidate leader in terms of reliability (B^R) by averaging sum of the end-to-end reliability of each sensor node to the designated leader using equation (3). This selection ensures the maximum reliability that this chain can provide. BS also finds the benefit of each candidate leader in terms of prolonging lifetime (B^E) using equation (2) where $RsdEg(S'_i)$ denotes residual energy of S'_i, $IniEg(S'_i)$ is initial energy of S'_i and $RqEg(S'_i)$ denotes the required energy of S'_i if being selected as the leader. After finding all the benefit values in a chain, BS selects the sensor node, which provides the maximum benefit as the leader for a given chain using equation (1). The higher the benefit value of equation (1), the higher the probability of being selected as a leader. Due to application specific nature of WSN, different applications have different requirements. Therefore, assigned weights (*w*) to each QoS parameter of equation (1) can be changed in order to satisfy the application requirements.

BS is responsible to find out the packet loss of each link by looking at the packet loss statistics reported by the neighboring node of each node in order to well and fast adapt the portion of each node from TTL of the packet based on the last reported links state. For doing so, each sensor node by comparing the sequence number of the packet (or packet copy) receives from its downstream node with the one expects to receive could easily calculate the packet loss of its adjacent link. Afterward, each sensor node puts its view about its adjacent link situation along with the data it must relay, in a packet and sends it toward the BS. BS makes a packet conveying the new portion of each node after finding the last situation of the links quality based on the new link reliabilities and sends it to the leader who must send it as well as to all nodes on both sides of the chain and inform sensor nodes about their new portion of the TTL. Each node receiving this packet picks its portion and then forwards the packet down to the adjacent neighbor as long as the neighbor node

receives it.

To find out optimal number of copies which must be sent through each link, we follow the following steps:

As we consider duty cycling in order to save energy we should take sleeping times, which greatly influence remaining TTL of the packet, into account. We assume that the duty cycle of the node is in such a way that if one node sends the first copy of the packet to its upstream node, that node is awake at that time but it is likely that the upstream node goes to sleep mode before finishing transferring all copies of a given packet. Therefore, we first should find the number of time slots in one awake time period (n_S) by having transmission time (TT) of one packet and awake time period (AwT) using $n_s = \frac{AwT}{TT}$. It is worth noting that having duty cycle (DC) and toggle period (TP), the (AwT) can be calculated easily using $AwT = TP \times DC$.

Then we need to calculate number of time slots that each packet requires (rS) to be able to transmit all its copies along the path towards the BS. As we are allowed to send (or receive) each copy of one packet in one time slot, the number of time slots corresponds to the number of copies. Therefore, having required time slots for a given TTL is enough to know the number of copies, which must be transmitted to increase reliability while TTL requirement of the packet is met. To find (rS), first we need to calculate the number of required awake cycle (nRc) to transmit all copies through different nodes, using equation (5) while (AsT) represents the time when the node is in sleep mode.

$$nRc = \frac{TTL}{ns \times TT + AsT} \tag{5}$$

where $AsT = TP \times (1 - DC)$.

Each time slot for a given node represents one receipt/transmission for that node. Leveraging equations (5) and (6), required time slots (rS) for the given packet are calculated.

A given source node can calculate the TTL of a packet in terms of time slots using equation (7).

$$rT = TTL - (nS \times TT + AsT) \times nRc \tag{6}$$

$$rS = \frac{rT}{TT} + nRc \times nS \tag{7}$$

Where (rT) denotes remaining time of the packet after using nRc awake cycles to transmit copies. Then, the optimal number of sent copies for node S_j to meet deadline requirement of the packet by considering the packet loss probabilities of the upward links can be obtained by BS using equation (8). The first term of the right part of equation (8) (n'_j) represents the portion of (S_j) from remaining TTL of the packet. The second term of equation (8) ($\log^{1-RqRL}_{PL(S_j,S_{j+1})}$) puts an upper bound for the number of packet copies for each link only by looking at the packet loss rate of the given link and the reliability requested by the application.

$$n_j = \min(n'_j, \log^{1-RqRL}_{PL(S_j,S_{j+1})}) \tag{8}$$

$$n'_j = \frac{PL(S_j, S_{j+1})}{PL(S_{LID}, BS) + \sum_{i=j}^{LID-1} PL(S_i, S_{i+1})} \times lS_j \tag{9}$$

$$\text{Where } \begin{cases} lS_{SourceNode} = rS \\ lS_j = lS_{j-1} - C_{j-1} \\ 0 < C_{j-1} \le n_{j-1} \end{cases} \tag{10}$$

S_{i+1} represents the upstream node of S_i in the chain, $PL(S_j, S_{j+1})$ denotes the packet loss between S_j and S_{j+1}, LID is leader identification in the second tier, n_j represents the number of copies of a given packet which should be transmitted by the node S_j and $RqRL$ is the requested reliability by the application for the links. Each sensor node upon receiving a packet must also update remaining or left time slots (lS_j) of the packet employing equation (10), using which required time slots to send C copies of a packet from one node to its upstream node is subtracted from the available time slots of the packet. As we do not know which copy is received first, upstream node can easily recognize C by looking at the copy number of the packet.

Applying this equation for a long chain where more than one chain leader is located in the second tier as shown in Fig.2, a few modifications should be made which results to the following equation:

$$n'_j = \frac{PL(S_j, S_{j+1})}{\sum_{\substack{K=LID \\ K \in SecondTierNodes}} PL(L_k, L_{k+1}) + PL(L_{LastL}, BS) + \sum_{i=j}^{LID-1} PL(S_i, S_{i+1})} \times lS_j \tag{11}$$

The first and second terms in the denominator, which are the modified parts, show the EER in the second tier between the leader of the given chain (L_{LID}) and BS, while L_k denotes the leader ID in the second tier, L_{k+1} is the upstream leader of L_k and L_{LastL} is the closest leader to the given BS .

Fig. 4. shows the psuedocode of READ protocol.

Initialization

1. Construct chain using PEGASIS
2. Find $S' = \{S_i \mid S_i \text{ is in Commuicati on range of } BS\}$
3. $Leader = \{S_i \mid S_i \in S' \& \text{best satisfy Equation 1}\}$
4. Duty cycling schedule;
5. $Ptn = \{\bigcup_i ptn_i \mid ptn_i \text{ is portion of } S_i \text{ from TTL}\}$
6. $BS \xrightarrow{(Ptn, LID)} S_{LID}$
7. $S_{LID} \xrightarrow{(Ptn, LID)} S_{LID-1}, S_{LID+1}$
8. S_i receives (Ptn, LID)
9. **Repeat** {
   Repeat { S_i sends (Ptn, LID)
   } **until** (S_{i-1} receives (Ptn, LID))

READ Protocol

1. **if** (event detected by S_i)
   S_i calculates n_i using equation 11
2. **if** (($S_i \neq LID$)) {
   a. **Repeat** {} **until** ($State(S_{i+1}) == Awake$)
   b. $numberofSentCopies_i = 0$
   c. **Repeat** { $S_i \xrightarrow{(Data_i)} S_{i+1}$
   $numberofSentCopies_i ++$
   d. } **until** (($State(S_{i+1}) == Asleep$) **or** ($numberofSentCopies_i == n_i$))
   e. **if** ((($State(S_{i+1}) == Asleep$)) **and** ($numberofSentCopies_i \neq n_i$))
   {**Repeat** {} **until** ($State(S_{i+1}) == Awake$);
   Go to step 2.b }}
3. **else if** (($S_i == LID$)) {
   a. $S_{i+1} = BS$; Run step 2.b to 2.d
   b. **if** ($RsdEg_{LID} < \theta \times IniEg_{LID}$) { ($hE = \{S_i \mid S_i \in S' \& RsdEg_i > \theta \times IniEg_i\}$)}
   c. **if** ($hE == \emptyset$){ **for** (each $S_i \in S'$)
   { $IniEg_i = RsdEg_i$ }

Fig. 4. Psuedocode of READ

7. Updating link reliability in READ+

As we presented in [10], to deal with inherently non-deterministic quality of wireless links while adhering to the delay requirements of the packets, packet loss rate of the links need to be continuously updated. This updating procedure can be accomplished either at the BS which has a global view of the whole network or at the nodes. However, since nodes only have local information about their links quality, BS may seem to be the best place to update packet loss rates (PLRs). On the other hand, it is quite possible that BS does not have recent statistical information about PLRs of the links if packets are not received by the BS. Updating PLR at the BS is also not efficient in case of having a long chain, which frequently experiences link quality changes. In this case, updating PLR locally seems promising as each node is aware of the PLR of its adjacent downward link. In the local updating scheme, PLR can be calculated by stamping source data packets with a sequence number and assigning each copy of a packet with a copy number. The new portion of each node from available TTL of a packet can then be calculated using equation below:

$$n_j^{new} = n_j^{old} \times \frac{PL^{new}(S_j, S_{j+1})}{PL^{old}(S_j, S_{j+1})} \quad (12)$$

When BS is responsible for calculating the packet loss of each link, it needs the packet loss statistics reported by the neighboring nodes of each link in order to well and fast adapt the portion of each node from TTL of the packet based on the last reported state of the links. For doing so, each node puts its view about its adjacent link situation along with the data that must be relayed in a packet and sends it towards the BS. After calculating the new packet loss of the links, BS assigns a new portion of TTL for each node to use for its upcoming packets and sends it to the leader to inform sensor nodes about their new portion of the TTL. Each node upon receipt of this information takes its portion of TTL and then forwards the packet down to the adjacent neighbor. Locally updating PLR increases the ratio of the number of received packets to the total packets. However due to lack of a global view, it is possible that when equation (13) is true, TTL of some of the received packets has expired. These situations need to be avoided as they have significant impacts on lowering down the hit ratio and energy efficiency.

$$\sum_{j=0}^{LID-1} PL^{old}(S_j, S_{j+1}) < \sum_{j=0}^{LID-1} PL^{new}(S_j, S_{j+1}) \quad (13)$$

where *LID* is chain leader's ID.

8. Relaying through multiple neighbors in READ-MN

As mentioned before, in READ and its enhanced

version READ+ all regular sensor nodes adjust their power levels as low as possible so that they could communicate with their closest neighbor node in order to save more energy. In other words, all copies from a given packet are sent through the closest neighbor nodes. Even though, this policy seems to provide an energy efficient data collection approach, using one path to relay all packet copies could be a bottleneck. To solve this problem, we introduce a modified version of READ+ protocol called, READ-MN (READ- Multi Neighbors).

READ-MN allows each node to send different copies of one packet through different neighbors instead of the closest one as done in READ+. Using multiple neighbors to send packet copies brings about some difficulties which must be handled in advance.

First, in READ+, the transmission path is quite specified in advance as there is only one choice for each node to relay its data through. Therefore, TTL of the packet could easily be assigned to each intermediate node since $\sum_{i=j}^{LID-1} PL(S_i, S_{i+1})$ is known for the BS in advance and equation(9) can easily be calculated. Calculating $\sum_{i=j}^{LID-1} PL(S_i, S_{i+1})$ is difficult if we do not know from which path data will be received. This is due to the fact that each node S_i has different neighbors and S_{i+1} is one of them. Therefore, as the path along which the data will reach the BS is not known in advance, calculating $\sum_{i=j}^{LID-1} PL(S_i, S_{i+1})$ is a challenge. To combat against this arising problem, we calculate $\sum_{i=j}^{LID-1} PL(S_i, S_{i+1})$ on the basis of the longest path (in terms of number of nodes involved). This is similar to what READ+ does. By doing so, if one packet reaches to the BS it will definitely be on-time as we consider the worst case which is the longest path for $\sum_{i=j}^{LID-1} PL(S_i, S_{i+1})$.

In READ-MN, each sensor node is allowed to send its copies via φ different neighbors, while φ could be a predefined parameter by the user or it could even be self-adapted by looking at the number of packet copies each node has to send and hence, φ_i will have different values for different nodes (i).

Regardless of the way φ is calculated, each sensor node has to adjust its power level in such a way to be able to reach all φ neighbors. The bigger the φ, the higher the power level nodes may use, and thereby the more energy dissipation.

It is worth mentioning that if φ=1, READ+ and READ-MN are the same and that is why we mention READ+ in all following graphs when $\varphi = 1$. Therefore, by READ-MN we mean $\varphi > 1$.

One of the path through which one event packet may reach to the BS is presented in Fig. 5. for $\varphi = 1$, $\varphi = 2$ and $\varphi = 3$. In case of event detection, just relaying and aggregating data along the path is enough and thereby no need to have all nodes' contribution (C,E,G,I in Fig. 5.b or B,C,E,G,I in Fig. 5.c.) however, having them helps better understand the situation.

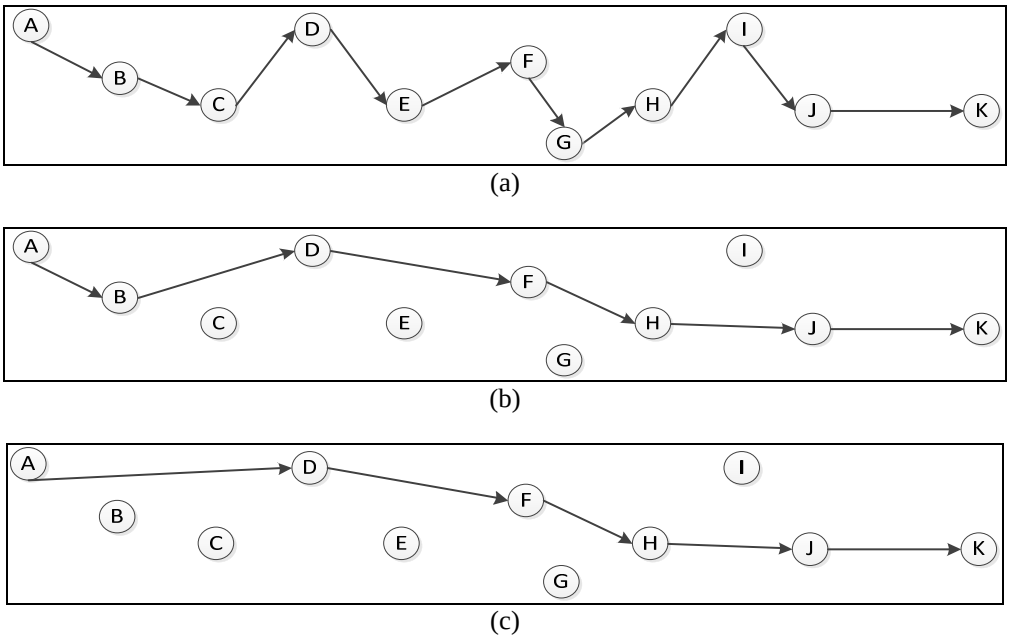

Fig. 5. A given relaying path for a received packet a) $\varphi = 1$ (READ+) b) $\varphi = 2$ c) $\varphi = 3$

In a monitoring application, BS expects to have all nodes data after a sampling interval. In other words, the contribution of nodes C,E,G,I in Fig. 5.b. which are not necessary in event detection, is quite essential for monitoring application. In case of monitoring, each node aggregates its data with the downstream node's data which is received following a predefined schedule. If the downstream node's data has not been received according to the schedule, the given node (say node C in the Fig. 5.b) has to send its raw data to the upstream neighbor (node D or E). The arising overhead here is that the receiving node, say D, must again relay packet arrived from C once relayed packet is arrived from B. This twice dissipates node D's energy. The closer the node to the BS, the higher the energy consumption. This is because of the heaviest burden on relaying nodes to transmit data from downstream nodes.

Even though, READ-MN can well mitigate the bottleneck problem, this advantage brings about the following problems:

- IN READ-MN, it is possible that more than one copy of a given packet reach the BS. This requires filtering the duplicates.
- READ-MN has a higher memory overhead than READ+ as it requires to store the packet loss of several paths instead of one as done in READ+. In other words, depending on the path through which each node has to relay the packet copies, the different n'_j is calculated according to equation (9), and hence the packet loss of different neighbor links must be known for each node in advance.
- READ-MN has to change the scheduling of the nodes so that each node has to be ON longer than required by READ+. This is due to the fact that in READ+ ($\varphi = 1$) only the immediate upstream nodes are allowed to relay data, while in READ-MN, a given node may want to relay its data through further nodes as $\varphi > 1$. Therefore, in READ+ each node has to be switched ON after its immediate downstream node does, while in READ-MN each node has to be switched ON after φ^{th} downstream nodes. In other words, duty cycle, which is in direct relation with φ, in READ-MN is longer than READ+.

9. Performance Evaluation

In this section, we compare the presented error control scheme with two existing and well-known error control schemes, i.e., ARQ and FEC. The ARQ we consider is a hob-by-hob S-W ARQ, which provides reliability by sending acknowledgements. The FEC scheme we consider is systematic hop-by-hop XOR-FEC (HH-FEC), which is a one-dimensional version of (n, k) FEC where k=n-1, and in which intermediate nodes have to perform XOR-FEC encoding/decoding function individually at each hop (if needed). For the sake of completeness, we also compare our protocol with a protocol without error controlling, in which only the original data without any parity or redundancy is aggregated and forwarded along the path to the destination. In XOR-FEC, the packets received without error can be processed and forwarded along the path. If, however, one packet is received erroneously, it has to wait till the last packet which carries the XOR of the group reaches the node. The number of packets in each group is calculated using equation introduced in [12].

Moreover, we examine READ-MN with different φ to evaluate different QoS parameters.

We use Java JDK6 to perform simulations for different TTLs, average link reliabilities, and φ. Each simulation is executed 100 times.

9.1 Performance Metrics

We consider hit ratio μ and energy efficiency η as two performance metrics. Hit Ratio is a metric that describes the efficiency of a real-time protocol and is defined as the percentage of the packets received by the BS before their deadline expire. Energy efficiency is defined by the amount of useful energy (E_{eff}) spent to disseminate packets received by the BS before their TTL expire to the total energy (E_{total}) spent to send all packets, i.e., $\eta = \frac{E_{eff}}{E_{total}}$, $E_{total} = E_{eff} + E_{urr} + E_{op} + E_{rnit}$ where E_{urr} represents energy spent for disseminating the un-received packets, E_{op} is energy spent for the imposed overhead (parity) of the received packets, and E_{rnit} is energy wasted on the packets received after their TTL expired.

9.2 Simulation parameters

For simulation, we consider a chain consisting of 16 randomly distributed nodes in a linear topology. BS is located one hop away from the rightmost node of the chain. In all simulations, the source node is the leftmost node, data rate is one sample per five seconds and updating PLRs is done locally by the upstream nodes. The quality of half of the random links change after reading almost 20 samples in average and toggle period (TP) is assumed to be 5000 ms. Other simulation parameters are listed in Table I.

Table 1. Simulation Parameters

Mac layer IEEE	802.15.4
Transmit bit rate	250 kbps

Operation frequency	2.4 GHZ
Radio model	TI CC2420
Transmission range	10-90 m

9.3 Simulation study of READ and READ+

We plot the achieved hit ratio and energy efficiency as the packet TTL increases from $(ChL \times TT)$ to $(ChL \times TT \times 800)$, where ChL is length of the chain and TT is transmission time.

Fig. 6. illustrates attained hit ratio and energy efficiency versus packet TTL for the given chain for three average PLRs in the network when DC=0.04. We have chosen three PLRs: 0.02, 0.15 and 0.45 in order to study the impact of different levels of packet loss. It can be seen that READ+ either outperforms S-W ARQ or has more or less the same hit ratio. The energy efficiency of ARQ, however, is comparably lower than READ+. READ+ also often performs better than HH-FEC. This could be justified as: First, HH-FEC needs to keep early lost packets in a group waiting long for the parity packet to be able to reconstruct them. Although the lost or erroneous packets may be reconstructed or corrected but due to long waiting time, their TTL expire. In case of longer chains or having less reliable links, it is also possible that one reconstructed packet undergoes more losses along the way. This brings about more delay. Secondly, as XOR-FEC is able to correct only one lost/erroneous packet, it cannot handle and manage losing more than one packet in a group. Lower duty cycle values implies longer waiting time for the packets that are ready to send their data as they also need to wait till nodes wake up again. Lower duty cycle values result in sending smaller number of copies or less retransmissions. As S-W ARQ wastes half of the awake mode time waiting for Acks, READ+ has higher hit ratio in presence of low duty cycles as it uses all awake time to send packet copies. The right side graphs of Fig. 6. provide a comparison among these approaches by looking at the energy efficiency metric. READ+ also is the most promising approach in terms of η particularly in case of high PLR and short TTL. When PLR is low and TTL is long, HH-FEC outperforms READ+ due to its lower overhead. Energy efficiency of S-W ARQ even in the best condition cannot exceed 0.5 that is due to acknowledgement overhead which must always be used. Compared with S-W ARQ, READ+ is more energy efficient specially when encountering packets with small TTLs. It can be argued that most of the very delay-constraint packets received by the BS using S-W ARQ scheme are expired because of the extra delay introduced by the use of acknowledgement messages. However, for very delay-constrained records or when PLR is pretty low (i.e. around 0.02), READ+ is as energy efficient as HH-FEC. This is due to the fact that in these cases READ+ also sends less packets. Fig. 7. compares hit ratio of READ+ when PLR is updated locally by nodes or globally by BS while PLR=0.45 and DC=1. As it can be seen, local update functions a little better than global update in terms of hit ratio.

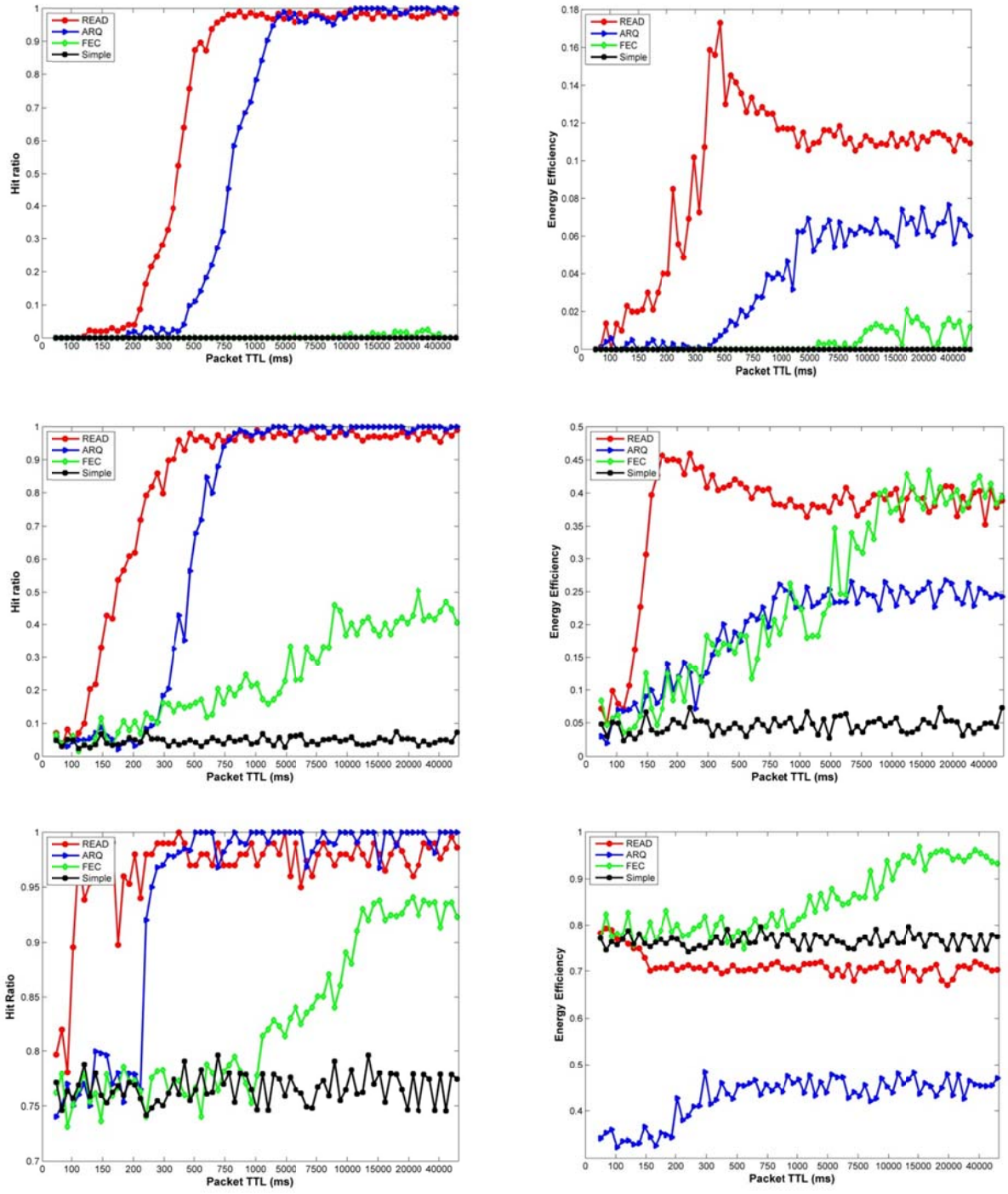

Fig. 6. Hit ratio (left-side graphs) and energy efficiency (right-side graphs) when DC=0.04 for PLR=0.45 (top), PLR=0.15 (middle), PLR=0.02 (bottom)

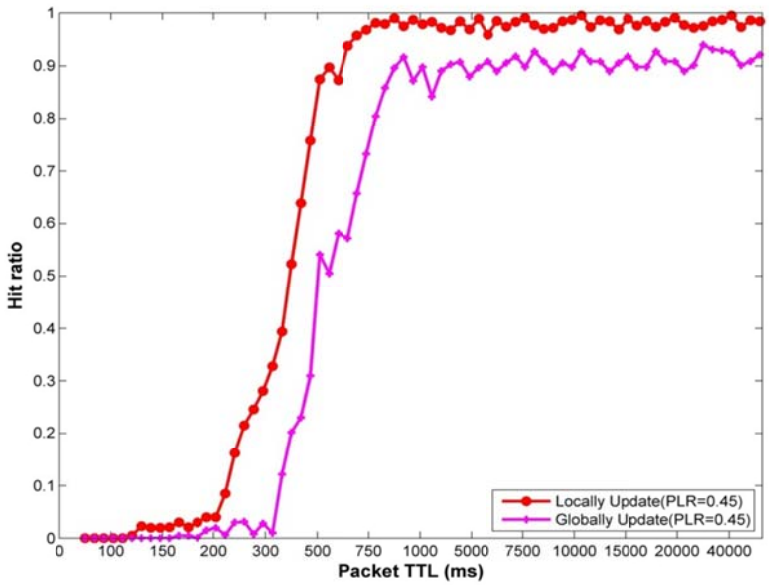

Fig. 7. Hit ratio of locally and globally updating PLRs

9.4 Simulation study of READ-MN

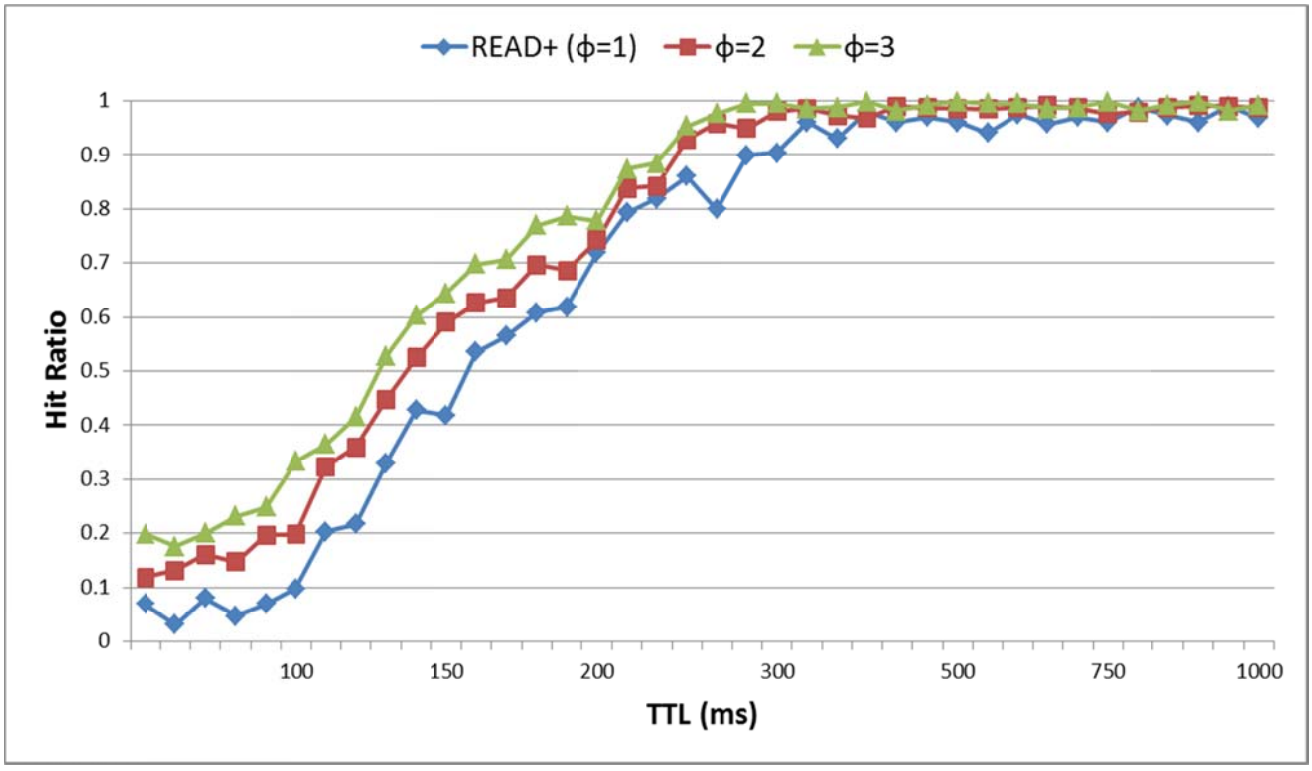

Fig. 8. Hit ratio vs. TTL

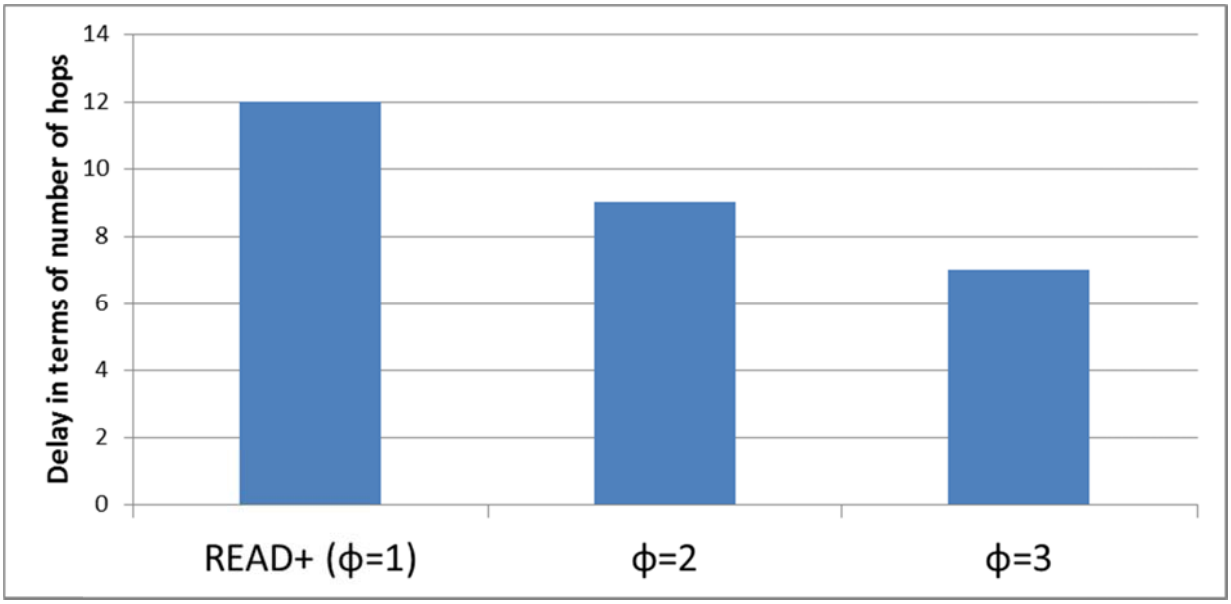

Fig. 9. Delay in terms of number of hops vs. φ

The Fig. 8. represents the achieved hit ratio when different number of neighbors (φ) is employed.

It can be seen that the more the neighbors, the higher the hit ratio. Using less intermediate nodes (big φ) being involved in packet relaying may result in small end-to-end delay for the given packet since data may reach the BS via longer hops (Fig. 9.). Thereby, more copies of a given packet can be sent if it is relayed via less intermediate nodes. However, different from our expectation, there is no significant difference among these three approaches in terms of hit ratio. To account for this issue, several aspects must be taken into account:

- Collision may arise as several nodes are sending their packet simultaneously. As it can be seen in Fig. 10. it is quite possible when node D is receiving B's packet through C, node B is sending the second copy of the given packet directly to D which could result in collision. The bigger the φ, the more collisions may occur. Therefore, the bigger φ does not always bring higher hit ratio as the collision increases as well.
  To mitigate the collision two or more different frequencies can be used for the sequential neighbors. Most likely if $\varphi + 1$ different frequencies are used, no collision occurs.

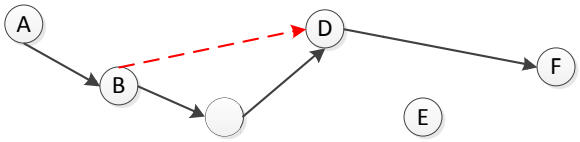

Fig. 10. A given chain-based network

- In updating part, the more statistical information the BS has about links quality, the more accurate n'_j it can suggest to the sensor nodes for the next time intervals, and the higher hit ratio can be achieved. The bigger the φ, the less statistical information about the given links may be available. For example, in case of Fig. 10, if $\varphi = 1$, all packet copies of A must be sent via node B and then more statistics about link AB will be sent along with the data packet toward the BS. In contrast, if $\varphi = 3$, in average one third of the packet copies are relayed through B and the rest must be sent through C and E and thereby, less statistical information about links quality will be available. Therefore, lack of enough statistical information brings about lower hit ratio as an inaccurate estimation about n'_j may be obtained.

Fig. 11. shows the contribution rate which is defined as the number of sensor nodes contributing to the packets' content which are received by the BS, for two different applications; event detection and monitoring. In case of event detection, as soon as one node receives an event packet, it should aggregate it with its own data and relay it to the upstream node. Therefore, as in READ+ we have $\varphi = 1$, all sensor node along the path must receive, aggregate and relay the given event packet one by one, so that they all have to contribute to a given received packet. By doing so, the BS better understands the context and has better insight about the environment state as the received data is as rich as possible since all nodes contribute to. When φ increases, event packet can reach the BS through less intermediate nodes (Fig. 5.b. and Fig. 5.c.) and then the contribution of all nodes in the received packet decreases. The bigger φ, the less contributor since data may reach the BS via longer hops, which in turn results in a shorter delay.

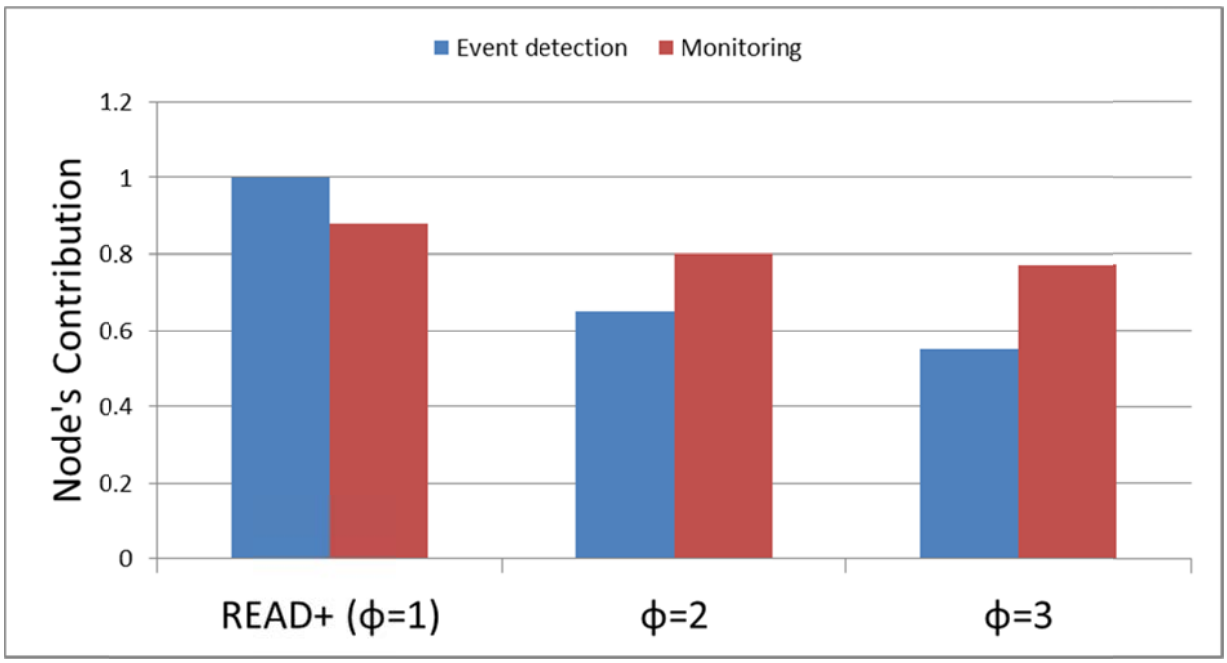

Fig. 11. Node's contribution vs. φ

As it was mentioned earlier, in case of monitoring (as a time-driven application), BS expects to have data of all nodes after a sampling interval. If the downstream node's data has not been received according to the schedule, the given node (node C in Fig. 5.b.) has to send its raw data to the upward neighbors (node D and E) and exploiting them makes its data reachable for the BS. One of the reason that downstream node's data does not reach the given node respecting the schedule is that, another copy of that node's data has been received by another neighbor (like as B) and hence, that neighbor is responsible to relay the given node's data. In the worst case, none of the copies are received by the upstream neighbors and hence, contribution of all nodes even if $\varphi = 1$ may not be 100% for the monitoring application.

In Fig. 12. we compare the energy consumption of the event detection and monitoring application. In case of monitoring, the bigger the φ, the more nodes may be bypassed and therefore their data needs to be relayed through another data transmission flow. Each data transmission flow exploits the intermediate nodes once and then it is quite possible one intermediate node is used several times for a given sampling interval.

Although, it is expected that energy consumption of the READ+ ($\varphi = 1$) for both monitoring and event detection remains the same, simulation results show different figures. In case of monitoring when $\varphi = 1$, even though several copies of a packet could be sent, it is quite possible that all copies sent through one unique path ($\varphi = 1$) are lost along the path and then the upstream node does not receive any packet. In this case, as the upstream node does not receive any packet, no energy is dissipated to receive the lost packet. This slightly affects total energy consumption as shown in the graph.

10. Conclusion

In this paper, we compare several error control approaches in term of different QoS parameters i.e., reliability, energy efficiency and real-timeness guarantee for a long chain-type WSNs. Long chain-type WSNs have often a large number of hop counts and to prolong lifetime, they usually need to work on a low duty cycle. Therefore, duty cycling is also considered as a determinative parameter in comparing these error control schemes.

Additionally, we introduce and evaluate an enhanced version of a real-time error control scheme, called READ-MN, in terms of its Quality of Service (QoS) guarantees. READ-MN studies on choosing between one or more paths through which source node sends its data copies toward destination so that reliability and realtimenes are ensured. Using multiple paths to send packet copies brings about some advantages and disadvantages which are elaborated on in this paper.

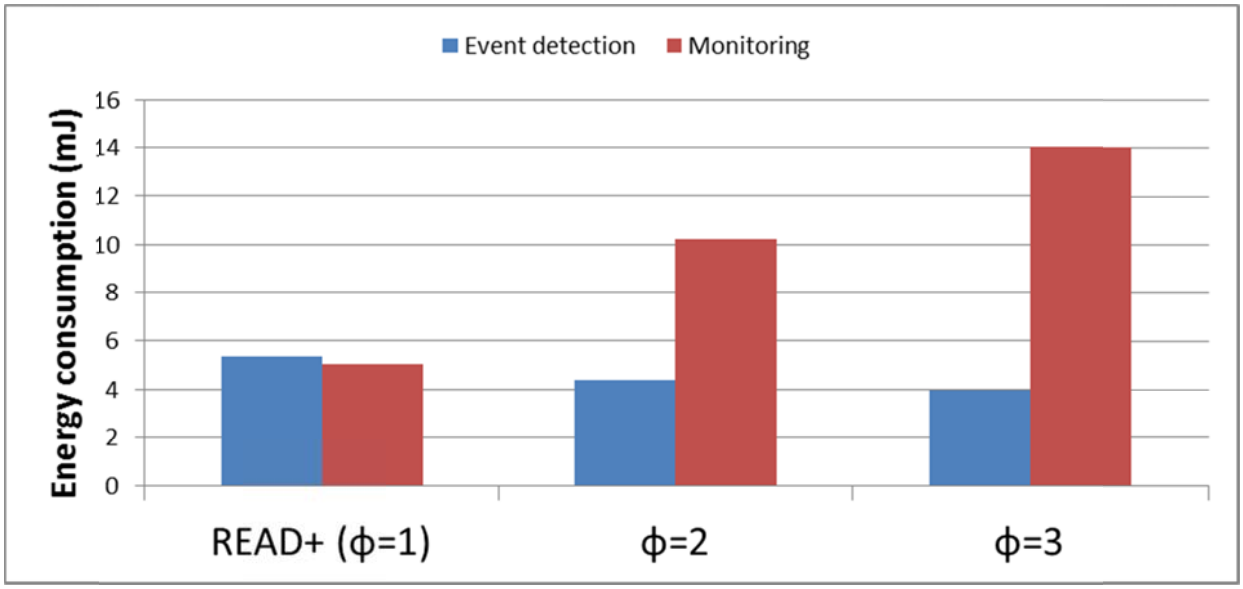

Fig. 12. Energy consumption vs. φ

Acknowledgements

This work is supported by IST FP7 STREP GENESI: Green sEnsor NEtworks for Structural monItoring project.

Reference

[1]. Halsall, F. Data Communications, Computer Networks and Open Systems. *U.K.: Addison-Wesley*, 1996.

[2]. Schwartz, M. Telecommunication Networks: Protocols, Modeling and Analysis. *CA: Addison-Wesley*, 1987.

[3]. Taghikhaki, Z., Meratnia, N., Zhang, Y., Havinga, P. QoS-aware Chain based aggregation in cooperating VCN and WSN, in: book of Roadside Networks for Vehicular Communications, IGI Global, Hershey, PA, USA, 2012.

[4]. He, T., Stankovic, J., Lu, C., Abdelzaher, T. SPEED: A stateless protocol for real-time communication in sensor networks, in 23rd International Conference on Distributed Computing Systems, RI , 2003, pp. 46-55.

[5]. Felemban, E., Lee, C., Ekici, E., Boder, R., Vural, S. Probabilistic QoS guarantee in reliability and timeliness domains in WSN, in INFOCOM, 2005, pp 2646-2657.

[6]. Kim, K., Park, S., Park H., Ham, Y. Reliable and real-time data dissemination in wireless sensor networks, in IEEE Military Communication, San Diego, 2008, pp 1-5.

[7]. Soyturk, M., Altilar, D. Reliable real-time data acquisition for rapidly deployable mission-critical Wireless Sensor Networks, in IEEE INFOCOM, Phoenix, 2008, pp 1-6.

[8]. Peltotalo, J., Peltotalo, S. and Roca V. Simple XOR, Reed-Solomon, and Parity Check Matrix-based FEC Schemes, IETF, draft-peltotalo-rmt-bbfec-supp-xor-pcm-rs-00.txt, 2004.

[9]. Taghikhaki, Z., Meratnia, N., Havinga, P. A Reliable and Real-Time Aggregation-aware Data Dissemination in a Chain-based Wireless Sensor Network, in IEEE SENSORCOMM, Shanghai, China, 2012.

[10]. Taghikhaki, Z. and Meratnia, N. and Havinga, P.J.M. An Error Control Scheme for Delay Constrained Data Communication in a Chain-Based Wireless Sensor Network, in the third international workshop on Advances in Sensor Technologies, Systems and Applications (ASTSA), Victoria, Canada, 2012.

[11]. Lindsey, S., Raghavendra, C.S. Power-efficient gathering in sensor information systems, in IEEE Aerospace Conference, Montana, 2002.

[12]. Lamoriniere, C.,Nafaa A., and Murphy L. Dynamic switching between adaptive fec protocols for reliable multi-source streaming, in IEEE GLOBECOM, 2009.