2012 Seventh International Conference on Broadband, Wireless Computing, Communication and Applications

An Error Control Scheme for Delay Constrained Data Communication in a Chain-Based Wireless Sensor Network

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

***Abstract*—Delay sensitive 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 unreliable nature of WSNs, achieving real-time guarantees and providing reliable data are quite challenging. Reliable data dissemination is traditionally performed by applying error control protocols which are not suitable for time critical applications deployed in a packet burst loss WSN. In this paper, we propose READ+ which is an enhanced version of READ, i.e., a real time and energy efficient error controlling scheme to ensure reliability for delay constrained aggregated data over a bursty channel in a chain-based WSN. In READ+ link qualities are updated either locally by sensor nodes or globally by the Base Station (BS). The simulation results show READ+ performs better considering both hit ratio and energy efficiency compared with the XOR based Forward Error Correction (FEC) and the stop-and-wait (S-W) ARQ specially when duty cycling is low and when either average packet loss is low or packet TTL is short.**

I. INTRODUCTION

Wireless sensor networks are one of the most promising technologies for applications such as structural health monitoring. Monitoring operational performance of large civil engineering (infra)structures such as bridges, tunnels, and highways require deployment of long linear arrays of sensor nodes. As the length of these (infra)structures is often much greater than their width, their topologies resemble a long chain. Long linear chain-type sensor networks have often a large number of hop counts and 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 Outdated data is not only useless but may also be harmful as it may have negative impacts on the decisions made by providing invalid information.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 in other networks 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 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 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 knock out just a few bits at a time but it cannot guarantee full reliability in 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 propose an error control scheme. Unlike existing techniques, our proposed protocol combines real-time and reliability guarantees for each packet and increases hit ratio (the percentage of the packets received by the BS 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 be reached the BS within their TTL are dropped in order

978-0-7695-4842-5/12 $26.00 © 2012 IEEE
DOI 10.1109/BWCCA.2012.68

to save energy of the intermediate nodes.

The rest of this paper is organized as follows. First, we briefly discuss the FEC and ARQ schemes. Then some preliminaries of this study will be presented in section III, followed by detailed description of the proposed approach in Section IV. Performance evaluation will be presented in Section V, while finally we draw some conclusions in Section VI.

II. Error Control Schemes

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 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 schemes has a different complexity level and different error recovery efficiencies. The simplest way to generate parity is the use of exclusive OR (XOR) [3], 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) [3] 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 the 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 the 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, and (iii) maintain the reliability of ARQ. We aim to improve both reliability and energy efficiency parameters for delay bounded applications so that packets are received before their TTL expires.

III. Preliminaries

A. 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 chain. 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 BSs 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 these applications, sensor nodes are placed at known and fixed locations (for instance, at critical locations) in a long linear topology and send their data periodically or upon detection of abnormality via relaying nodes to the BS.

Every 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 BS aggregate the data received from the downstream nodes with their own (if any) and forward the local aggregated value towards the BS.

B. Policies regarding Reliability and Real-Timeness

To eliminate the delay and overheads introduced by acknowledgement 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 reach to the destination after sending some copies of a packet. QoS-ACA [4] which is an approach to guarantee reliability by sending several copies of one packet, estimates the optimal number of retransmission 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 realtimeness and only aims to ensure high reliability for a delay tolerable application. Thereby, we cannot utilize such equations 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 answer this question by proposing 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 use the packet TTLs on intermediate nodes in such a way that reliability gain and on-time end to end delivery ratio is maximized.

IV. Detailed Description of READ Protocol

READ [5] starts with chain construction using PEGASIS algorithm proposed in [6]. 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 must be relayed, in a packet and sends it toward the BS. BS after finding the last situation of the links quality, makes a packet conveys the new portion of each node based on the new link radiabilities and sends it to the leader who must send it as well to two side of the chain and inform sensor nodes from their new portion of the TTL. Each node receives this packet picks its portion up 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 influences 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 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 (nS) by having transmission time (TT) of one packet and awake time period (AwT) using $nS = \frac{AwT}{TT}$. It is worth noting that having duty cycle (DC) and toggle period (TP), the (AwT) can be calculated easily as $AwT = TP \times DC$.

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 packet copies. Therefore, having required time slots for a given TTL is enough to know the number of packet 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 packet copies through different nodes, using (1) while (AsT) represents the time when the node is in sleep mode.

$$nRc = \frac{TTL}{ns \times TT + AsT} \tag{1}$$

$$\text{where} \qquad AsT = TP \times (1 - DC) \tag{2}$$

Each time slot for a given node represents one receipt/transmission for that node. Leveraging (1) and (3), required time slots (rS) for the given packet is calculated. Actually, source node using (4) describes the TTL of a packet in terms of time slots.

$$rT = TTL - (nS \times TT + AsT) \times nRc \tag{3}$$

$$rS = \frac{rT}{TT} + nRc \times nS \tag{4}$$

Where (rT) denotes remaining time of the packet after using nRc awake cycles to transmit packet 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 (5). The first term of the right part of (5) represents the portion (Ptn_j) of (S_j) from TTL remaining of the packet. The second term of (5), 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.

$$n_j = \min(n'_j, \log^{1-RqRL}_{PL(S_j,S_{j+1})}) \tag{5}$$

$$n'_j = \frac{PL(S_j, S_{j+1})}{PL(S_{LID}, BS) + \sum\limits_{i=j}^{LID-1} PL(S_i, S_{i+1})} \times lS_j \tag{6}$$

$$\text{where} \qquad \begin{cases} lS_{SourceNode} = rS \\ lS_j = lS_{j-1} - C_{j-1} \\ 0 < C_{j-1} \leq n_{j-1} \end{cases} \tag{7}$$

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 leaderID that here is the closest node to the BS, 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 (7), 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 packet copy is received first, upstream node by looking at the copy number of the packet can easily recognize C.

A. Policies regarding updating link reliability in READ+

To deal with inherently non-deterministic quality of the 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 link quality and in fact do not have any idea about the packet loss rate (PLR) of other links, BS may seem to be the best place to update PLRs. On the other hand, it is quite possible that BS does not have recent 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 computed by stamping source data packets with a sequence number and assigning each copy of a packet with a copy number. An approximate method to calculate the new portion of each node from available TTL of a packet is $n_j^{new} = n_j^{old} \times \frac{PL^{new}(S_j,S_{j+1})}{PL^{old}(S_j,S_{j+1})}$. 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 information 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 recent quality of the links, BS assigns a new portion of TTL for each node to use for sensing multiple copies and sends it to the leader to inform sensor nodes about their new portion of the TTL. Upon receipt of this information, each node receives 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. But due to lack of a global view, it is possible that when equation (8) is true, TTL of some of the received packets has expired. These situations need to be prevented as they have significant impacts on lowering down the hit ratio and energy efficiency. In this equation N is number of nodes.

$$\sum_{j=0}^{N-2} PL^{old}(S_j, S_{j+1}) < \sum_{j=0}^{N-2} PL^{new}(S_j, S_{j+1}) \qquad (8)$$

V. Performance Evaluation

In this section, we compare our 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 [7].

We used Java JDK6 to perform simulations for different TTLs, average link raliabilities and duty cycle values. Each simulation was executed 100 times.

A. Comparison Metrics

We consider hit ratio μ and energy efficiency as two performance metrics. Hit Ratio is a metric that described 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 another metric we consider expressing the amount of useful energy (E_{eff}) spent to disseminate packets received by BS before their TTL expires to the total energy (E_{total}) spent to sent 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.

B. Description of scenarios

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. The results of two duty cycles, i.e., 1 (radio is always ON) and 0.04 (radio is rarely ON) are represented to judge about DC impacts. Other simulation parameters are listed in Table I.

TABLE I
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

C. Results Discussion

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.

Figure 1 and Figure 2 illustrate attained hit ratio and energy efficiency versus packets TTL for the given chain for three average PLRs in the network when DC=1 and DC=0.04. We have chosen three PLRs: 0.02, 0.15 and 0.45 to study the impact of different levels of packet loss. It can be seen that READ+

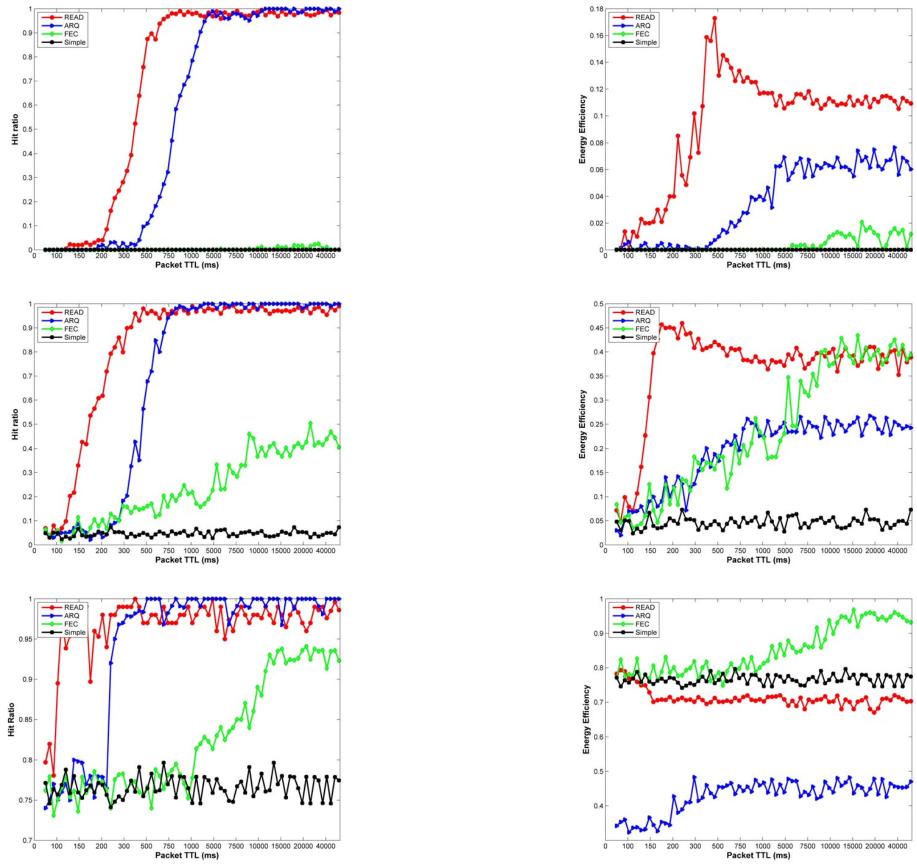

Fig. 1. Hit ratio (left-side graphs) and energy efficiency (right-side graphs) when DC=1 for PLR=0.45 (top), PLR=0.15 (middle), PLR=0.02 (bottom)

either outperforms S-W ARQ or has pretty much the same hit ratio as S-W ARQ. 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 under less reliable links, it is possible that one reconstructed packet undergoes more losses along the way which causes more delay. Secondly, as XOR-FEC is able to correct only one lost/erroneous packet, it can not manage losing more than one packet in a group. Lower duty cycles implies longer waiting time for the packets that are ready to send but they need to wait till nodes wakes up again. Lower duty cycles lead to send a smaller number copies or less retransmissions. As S-W ARQ wastes half of the awake time waiting for Acks, READ+ presents better hit ratio in presence of low duty cycles as it uses all awake time to send packet copies.

The right side graphs of Figure 1 and Figure 2 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 can not 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 as 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 packet. Figure 3 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.

VI. Conclusion

In this paper, we propose an improved version of READ, a reliable, real-time and energy efficient aggregation-aware data dissemination protocol designed for long chain-type WSNs to efficiently gather delay constrained data. Long chain-type

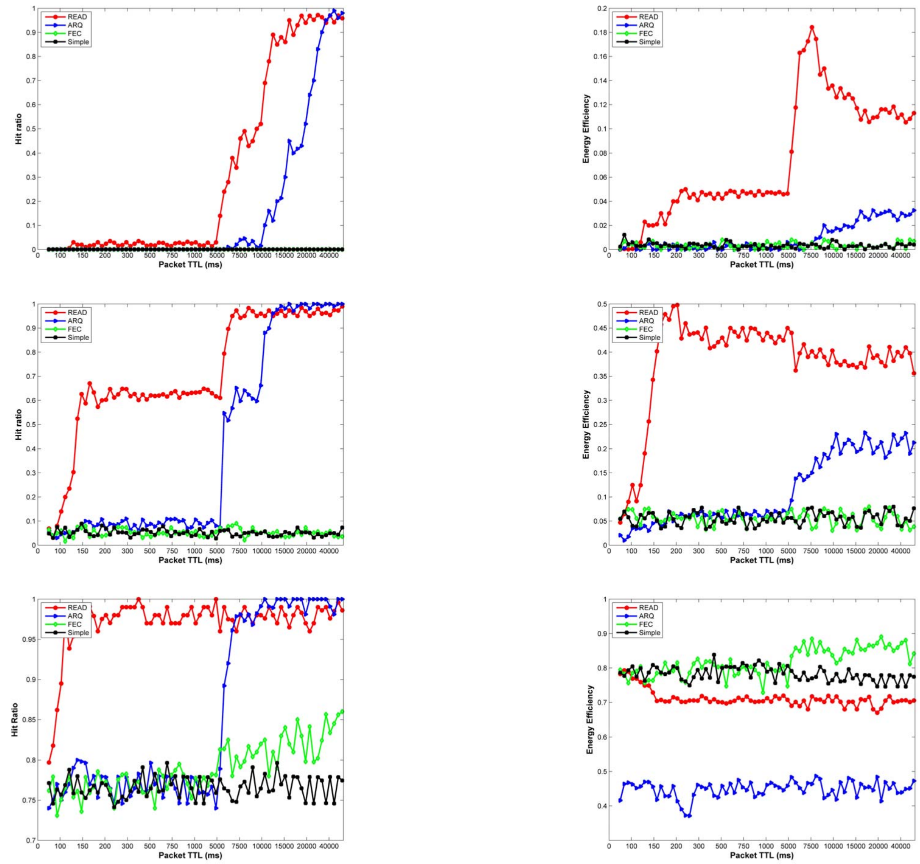

Fig. 2. Hit ratio (left-side graphs) and energy efficiency (right-side graphs) when DC=0.04 for PLR=0.45 (top), PLR=0.15 (middle), PLR=0.02 (bottom)

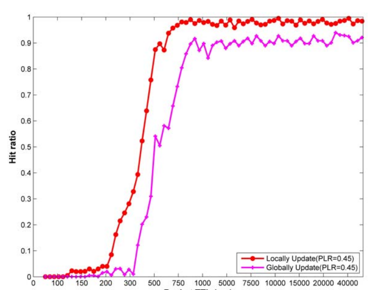

Fig. 3. Hit ratio of locally and globally updating PLRs

WSNs have often a large number of hop counts and to prolong lifetime, they usually need to work on a low duty cycle. READ+ uses a fair heuristic and hence allocates available packet TTLs proportionately to the packet loss probability of the links along the forwarding path to judiciously use the packet TTLs on intermediate nodes so that reliability gain and on-time end to end delivery ratio is maximized. Comparing with two well-known error controlling approaches, i.e., S-W ARQ and FEC, READ+ yields better performance in terms of both hit ratio and energy efficiency particularly in low duty cycling and when either average packet loss rate is low or TTL is short. We also propose and compare two schemes to update links quality and our experimental results show locally updating by nodes rather than the BS provides higher hit ratio.

ACKNOWLEDGEMENT

This work is supported by IST FP7 STREP GENESI.

REFERENCES

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

[4] 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, In press (2012).

[5] 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 (2012).

[6] Lindsey, S., Raghavendra, C.S.: *Power-efficient gathering in sensor information systems*. In: IEEE Aerospace Conference, Montana (2002).

[7] Lamoriniere, C.,Nafaa A., and Murphy L.: *Dynamic switching between adaptive fec protocols for reliable multi-source streaming*. In IEEE GLOBECOM, (2009).