A Reliable and Real-Time AggregationAaware Data Dissemination 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*—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. To ensure data reliability, traditionally various retransmission mechanisms have been used, which in turn introduce extra delay. In this paper, we propose READ, i.e., a reliable and real-time aggregation-aware data dissemination to ensure reliable and fast data delivery in a chain-based WSN. We also investigate the relatively unexplored topic of impact analysis of Time To Live (TTL) and link reliability parameters on network performance in terms of attained hit ratio for three different approaches, i.e., READ, QoS-ACA, and the stop-and-wait (S-W) ARQ to assess the appropriateness of each method facing different conditions. The simulation results show READ performs better in terms of hit ratio compared with QoS-ACA and S-W ARQ when link reliability is low and packet's TTL is short. Although not being the primary goal of READ, energy consumption of the protocol is also much lower than the other two approaches.**

Keywords- Chain-based wireless sensor network, reliable/real-time data dissemination.

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, highways, and water pipes 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 wireless sensor networks, while the low duty cycle introduces extra delays.

Time-critical applications such as disaster management and structural health monitoring highly depend on the availability of real-time data as in these applications data is neither useful nor valuable if it is received after its Time To Live (TTL). Outdated data is not only be useless but also harmless 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 their poor link quality, providing real-time guarantees and data reliability in WSNs is quite challenging. Link quality can be easily affected, among others, by weather, temporary obstacles, and mobility. Most of existing real-time algorithms applied in other networks than WSNs assume network is reliable and packets are not lost because of low link quality. Therefore, they cannot be directly applied to WSNs. The higher the packet loss due to low quality links, the lower the performance of a real-time wireless sensor network. One of the mechanisms to provide data reliability is through introduction of redundant data by transmitting the same data multiple times, which results in high energy consumption. It is somewhat clear that ensuring reliability may not always go hand in hand with ensuring network lifetime. Depending on the application at hand, one can also argue that energy efficiency, real-timeness, and data reliability are not always equally important. Data reliability and real-timeness become significant for applications dealing with identification of out-of-ordinary situations as well as warning and notification systems, while continuous monitoring applications demand long network lifetime and can tolerate latency and data unreliability to some extent [1] by using local techniques such as filtering and anomaly detection. In the latter applications, data aggregation is considered as a significant primitive, which not only helps save energy and bandwidth by communicating less data but also provides meaningful information to the end-users.

The main problem addressed in this paper is the design of an aggregation-aware data dissemination protocol for a chain-based WSN suffering from low reliable communication links while satisfying the delay and reliability requirements of the packet. 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 base station before their deadline expire). To deal with the energy consumption and to enrich data, we utilize data aggregation on the intermediate nodes as far as it does not influence packet deadline. We also investigate the relatively unexplored relationship between the TTL and link reliability parameters and their impact on the hit ratio for three different approaches, i.e., READ, QoS-ACA [2] and an ARQ approach, to assess the appropriateness of each method facing different conditions.

The rest of this paper is organized as follows. First we briefly discuss state of the art and preliminaries of this study. Then a detailed description of our proposed approach will be provided, which will be followed by performance evaluation. Finally we draw some conclusions and future works.

II. RELATED WORK

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 [3] 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 the specified requirement is more likely to be chosen as the upstream node. MMSPEED [4], 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 path is in direct proportion to 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 [5] proposes a reliable and real-time data dissemination, in which reliability is satisfied by sending several copies of one packet through multipath such that sum of the reliability of the considered path is equal or higher than the requested reliability. This 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. [6] present a reliable data acquisition approach for time-critical application of WSNs. Reliability is provided similar to techniques of [4][5] leveraging multipath approach while real-time concern is supported by packet prioritization. This technique therefore deals with the priority scheduling to handle queuing delay, which is of the main causes of making end-to-end latency. Almost all of the aforementioned reliable 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 together in a chain-based WSN, in which only one path can be established between source and destination nodes. Moreover, since approaches of [4][5] 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 [2] aims to fast, reliably, and energy efficiently aggregate data in a chain-based WSN and send the aggregated value to a base station. To ensure reliability, it leverages the benefits of retransmission without using any acknowledgement (Ack). It utilizes the optimum number of retransmission to ensure the required reliability. It considers the residual and required energy of each sensor node and the distance between node and the base station as two main criteria to select a node as an aggregator. However, it does not guarantee delivery of a packet to the base station within its deadline.

III. PRELIMINARIES

The preliminaries of this study is presented here.

A. Quality of Service Parameters

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 WSN 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.

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.

B. Duty-cycling

Efficient energy consumption has one of the highest priorities in WSNs to ensure long network lifetime. As one of the most energy-expenditure operations is transmitting data, each sensor node must turn 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 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.

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

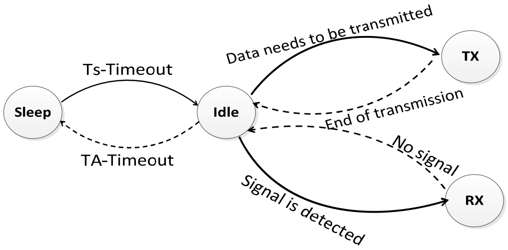

Figure 1. State diagram for radio states

C. *Network Model*

We make the following assumptions regarding the WSN. The WSN consists of *N* sensor nodes deployed in a linear topology and two base stations are located at two sides of the chain. We have described in [2] a mechanism using which a chain leader can be selected through which sensor data is forwarded to the base station. In case of not being a chain leader, sensor nodes can only communicate with their direct neighbors. The location of sensor nodes and the base stations 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 application, sensor nodes are placed at known and fixed locations (for instance, at critical locations) in a long linear array topology and send their data periodically or upon detection of abnormal situations via relaying nodes to a base station. It should be noted that we assume the packet loss probability of each link is almost fixed and does not change much. This is justified by the fact that we aim to find the relationship between TTL and link reliability with the network performance. We are aware that link reliability changes frequently in practice. In our ongoing work, therefore, we enhance READ by considering dynamic changes of links reliability. As far as this paper is concerned finding the relation between TTL and link reliability with the network performance requires a fixed link reliability to be assumed.

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 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 base station directly.

To motivate the need to address both data reliability and real-timeness in our protocol, let us consider the network illustrated in Figure 2, 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 base station is 1s. Clearly, this packet will be received by the base station 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.

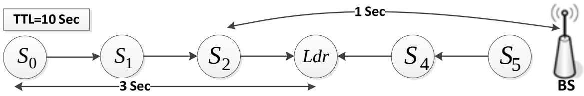

Figure 2. An example of a chain based network

D. *Policies regarding Reliability and Real-Timeness*

To cope with unreliability of the links, this paper leverages the benefit of retransmission approaches without using acknowledgement in order to support reliable transmission. Therefore, similar to QoS-ACA, we are going to estimate the optimal number of retransmissions for each link. Each sensor node sends multiple copies of the same packet to its upstream neighbor in order to improve transmission reliability. 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. Since the retransmission mechanism used in QoS-ACA imposes extra delay, we modify it to meet a given latency requirement by retransmitting as far as packet's deadline is not expired. 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 allocating the available packet's TTL proportionately to the packet loss probability of the links along the forwarding path to judiciously and fairly use the packet's TTL 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

Our algorithm starts with chain construction using PEGASIS algorithm proposed in [7]. In a given chain, one node must be selected as the leader in order to do the final aggregation and to send the aggregated result to the base station. Two QoS parameters, i.e., reliability and energy consumption as well as two assigned weights, are considered to make different criteria for electing a leader. To this end, we introduce the following formula:

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

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

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

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

Where S' represents a set of sensor nodes, which are able to directly communicate with one of the base stations and CL represents the candidate leader. The hop-by-hop reliability (HHR) between two sensor nodes are obtained using $HHR(S_i, S_{i+1}) = 1 - p_{pktloss}(S_i, S_{i+1})$. By having the hop-by-hop reliabilities, base station must evaluate the appropriateness of each member of S' to be an aggregator. To this end, base station first calculates the end-to-end reliability from each sensor node to the designated leader by employing (4). At the second step, base station 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 (3). This selection ensures the maximum reliability that this chain can provide. Base station also finds the benefit of each candidate leader in terms of prolonging lifetime (B^E) using (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, base station selects the sensor node, which provides the maximum benefit as the leader for a given chain using (1). The higher the benefit value of (1), the higher the probability of being selected as a leader. One should note that aggregation takes place at different locations of the network as the leader selection process results in selecting an aggregator in a dynamic way based on the energy and reliability parameters. Due to application specific nature of WSN, different applications have different requirements. Therefore, assigned weights (*w*) to each QoS parameter of (1) can be changed in order to satisfy the application requirements. As we have two base stations, two chain leaders can be selected such that they can communicate with one of the base stations directly. Sensor nodes must select one of these chain leaders to send their data to. This selection is done by considering distance between sensor nodes and the chain leaders.

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

Each sensor node must update packet TTL employing (5) where TT (Transmission Time) denotes the time required to transmit one packet to the upstream node.

$$\begin{cases} TTL_{Sourcenode} = PacketTTL \\ TTL_j = TTL_{j-1} - C_{j-1} \times TT & j < LID \\ TTL_j = TTL_{j+1} - C_{j+1} \times TT & j > LID \end{cases} \qquad (5)$$

Where: $0 < C_j \le n_j$

Using (5), required time to send C copies of a packet from one node to its upstream node is subtracted from the TTL of the packet where LID represents leader ID. 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. In the next step, the chain leader assigns a portion of the remaining TTL of the packet to each node by dividing the packet loss probability of the link adjacent to a given node by sum of the packet loss probabilities of the links located between the given node and leader. Equation 6 calculates optimal number of packet copies for each node to meet deadline requirement of the packet. The second term of (6), put 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.

$$n_j = \min(n'_j, \log {}^{1-RqRl}_{PL(S_j, S_{j+1})}) \qquad (6)$$

$$n'_j = \begin{cases} \dfrac{PL(S_j, S_{j+1})}{PL(S_{LID}, BS) + \sum_{i=j}^{LID-1} PL(S_i, S_{i+1})} \times \dfrac{TTL_j^{new}}{TT} & j \ne LID \\ \dfrac{PL(S_j, BS)}{PL(S_{LID}, BS) + \sum_{i=j}^{LID-1} PL(S_i, S_{i+1})} \times \dfrac{TTL_j^{new}}{TT} & j = LID \end{cases} \qquad (7)$$

Where S_{i+1} represents the upstream node of S_i in the chain and $PL(S_j, S_{j+1})$ denotes the packet loss between S_j and S_{j+1}. Equations 5 and 7 can be used if radios of all nodes are never turned off. As we also consider duty cycling in order to save energy, (7) requires significant revisions to include sleeping times which greatly influences remaining TTL of the packet. Therefore, the way we calculate the optimal number of packet copies changes. 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, it 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 base station. 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 (8) while AsT represents the time the node is in sleep mode.

$$nRc = \frac{TTL}{nS \times TT + AsT} \tag{8}$$

Where: $AsT = TP \times (1 - DC)$ (9)

Each time slot for a given node represents one receipt or one transmission for that node. Leveraging (8), (9) and (10), required time slots (rS) for the given packet is calculated. Actually, source node using (10) describes the TTL of a packet in terms of time slots.

$$rT = TTL - (nS \times TT + AsT) \times nRc \tag{10}$$

$$rS = \frac{rT}{TT} + nRc \times nS \tag{11}$$

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 (12). The first term of the right part of (12) represents the portion (Ptn_j) of S_j from TTL remaining of the packet.

$$n_j = \min(n'_j, \log {}^{1-RqRl}_{PL(S_j,S_{j+1})}) \tag{12}$$

$$n'_j = \begin{cases} \dfrac{PL(S_j, S_{j+1})}{PL(S_{LID}, BS) + \sum_{i=j}^{LID-1} PL(S_i, S_{i+1})} \times lS_j & j \neq LID \\ \dfrac{PL(S_j, BS)}{PL(S_{LID}, BS) + \sum_{i=j}^{LID-1} PL(S_i, S_{i+1})} \times lS_j & j = LID \end{cases} \tag{13}$$

Where: $\begin{cases} lS_{SourceNode} = rS, \quad lS_j = lS_{j-1} - C_{j-1} \\ 0 < C_{j-1} \leq n_{j-1} \end{cases}$

Here n_j represents the number of copies of a given packet which should be transmitted by the node S_j. Each sensor node upon receiving a packet must also update remaining or left time slots (lS_j) of the packet employing (13), 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.

Figure 3 shows the psuedocode of READ protocol.

V. PERFORMANCE EVALUATION

We used Java JDK 6 to implement all algorithms and the simulation environment. We perform simulations for different TTL, link reliability and duty cycle values. Each simulation is executed 100 times. In this section we aim to compare READ, which employs retransmission mechanism without any Ack while keeping an eye on packet's TTL remaining time, with (i) QoS-ACA, which is also a retransmission mechanism without Ack while ignoring TTL parameter and (ii) S-W ARQ, which is a retransmission mechanism with Ack. Traditional acknowledgement protocols, namely, stop-and-wait (S-W), go-back-n (GBN), and selective repeat (SR) [8][9][10], try to retransmit one erroneous frame regardless of the link reliability state. We compare our method with a hop by hop S-W ARQ which is a well-known ARQ scheme. We consider hit ratio and energy consumption as performance metrics. Hit Ratio is defined as the percentage of the packets received by the base station before their deadline expire. We aim to find out the relationship between different packet loss probabilities and various TTL values with the gained network performance for these three approaches in order to know in which condition which method should be employed to provide reliability and real-time concerns simultaneously.

A. *Description of scenarios*

For simulation, a chain of sensor nodes is formed consisting of 51 sensor nodes randomly distributed in a linear topology. Two base stations are located one hop away from the rightmost and the leftmost nodes of the chain. The output power of our radio model (TICC2420) is programmable in eight levels (from approximately –25 to 0 dBm). Therefore, every sensor node in case of being a leader utilizes the highest power level to provide the longest transmission range, otherwise the minimum power level which is required to reach the closest neighbor is employed. We change the packet loss probability ($p_{pktloss}$) on the links from 0.01 to 0.9. In all simulations, the first main source node is the middle node of the chain (S_{25}), which must select either the left side or right side leader towards which it transmits its data. This selection is done by looking at provided delay which is in direct relation with the link reliability and distance. The second and the third source nodes are the leftmost and the rightmost nodes in the chain. The toggle period (TP) is 1500 ms and energy threshold (θ) is 0.1.

```
Initialization
1. Construct chain using PEGASIS
2. Find $S' = \{S_i \mid S_i\ is\ in\ Commuicati\ on\ range\ of\ BS\}$
3. $Leader = \{S_i \mid S_i \in S' \&\ best\ satisfy\ Equation 1\}$
4. Duty cycling schedule;
5. $Ptn = \{\bigcup_i ptn_i \mid ptn_i\ is\ portion\ of\ S_i\ from\ TTL\}$
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 $(Ptn, LID)$ )
        $i = i - 1$ and go to step 9}
   until ( $\forall S_i$ receives $(Ptn, LID)$ )

READ Protocol
1. if (event detected by $S_i$ )
        $S_i$ calculates $n_i$ using equation 12
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 ( $hE == \varnothing$ ){ for (each $S_i \in S'$ )
            { $IniEg_i = RsdEg_i$ }
    d. Go to 3.b}
    e. else { BS finds another leader based on
        Equation 1}
4. if ( $S_{i+1}$ receives $Data_i$ )
5. { $AggData_{i+1} = Aggregate(AggData_i, Data_{i+1})$
   $Data_{i+1} = AggData_{i+1}$; $i = i - 1$ and go to step 2}
```

Figure 3. Pseudocode of READ

The results of two duty cycles, 0.99 (radio is almost always ON) and 0.1 (radio is almost always Off) are represented in this paper to better judge about duty cycling impacts. The other simulation parameters are listed in Table I.

TABLE I. SIMULATION PARAMETERS

No. of nodes	51
Area size	1m x 260m
Mac layer	IEEE 802.15.4
Transmit bit rate	250 kbps
Operation frequency	2.4 GHz
Packet size	128 bytes
Radio model	TI CC2420
Transmit current at 0dBm	17.4 mA
Transmit current at -25dBm	8.5 mA
Receive current	18.8 mA
Supply voltage	(1.6 – 2.0 V)
Idle current	0.426 mA
Transmission range	10-90 m
Receiver sensitivity threshold	-95 dBm

B. *Performance Evaluations*

1) *Hit Ratio*

The achieved hit ratio is plotted for these three methods when the Link Reliability (LR=1-PacketlossProbability) can be selected randomly from a set of intervals shown in Figure 4 and duty cycling is 0.1. Figure 4 illustrates attained hit ratio as the packet TTL increases from 80 to 3200 ms. It can be seen that hit ratio of READ is higher than S-W ARQ when the link reliability in the chain changes randomly between 0.1 and 1 or between 0.4 and 1. READ also outperforms QoS-ACA when TTL of the packet is small (smaller than 1500ms). It can be seen from Figure 4 (middle and bottom) that when the lower bounds of link reliability and TTL are increased, performance of S-W ARQ improves. We can conclude that if both link reliability and TTL of the packet are quite high, performance of all three techniques in terms of hit ratio is the same. Otherwise, READ outperforms S-W ARQ and outperform QoS-ACA in case of having short TTL. To have a better judgment about the exact relation between TTL of the packet and link reliability with attained hit ratio, in the following graphs the lower bound of the link reliability interval is increased from 0.1~0.97 that means the reliability of a link should be higher than the given lower bound. Figure 5 illustrates the hit ratio graphs of two different duty cycles for these three approaches. The left side graphs show impact of duty cycle 0.99 on hit ratio while the right side graphs are for duty cycle 0.1. From Figure 5, one can see that when either TTL is short or link reliability is low, READ has better hit ratio. But when TTL is long and link reliability is quite high (TTL>1500, LR>0.8), S-W ARQ outperforms READ because it has enough time to utilize acknowledgment and also because of high reliable link, packets are almost never lost. Although, the hit ratio of READ in these conditions (TTL>1500, LR>0.8) are almost 1 but it has a little fluctuation between 0.97 and 1. It is worth noting, the sharp changes seen in right side graphs of Figure 5 when TTL is about 1500ms are because of using duty cycling for sensor nodes. In case of S-W ARQ, one node requires to frequently switch between sending and receiving mode to be able to handle sending a packet in one time slot and receiving (or waiting to receive) corresponding acknowledgement in the next time slot. Therefore, half of time slots in one awake time period are used for the acknowledgement. This is not the case for READ or QoS-ACA approaches. READ and QoS-ACA utilize all time slots for sending several copies of the packet. The higher TTL, the greater number of awake cycles every node is allowed to utilize to send packet and receive acknowledgement in order to ensure reliability while packet TTL has not yet been expired. READ and QoS-ACA also undergo these sharp changes when duty cycle is too small (i.e. 0.1) as they cannot send all packet copies in one awake cycle that is about 150ms and they have to wait for another awake cycle(s) to be able to send rest copies. When TTL raised to 1500ms (which is start point of another awake cycle), the rest copies can also be sent and the hit ratio suddenly improved especially in case of high reliable links. Also, compared with QoS-ACA, READ has better hit ratio when TTL parameter is short (shorter than 500 ms). In this case, QoS-ACA sends several copies of a packet, especially when LR is low. QoS-ACA satisfies the required reliability for the given packet only for the first few hops, in which the packet has not yet been dropped due to TTL expiration.

Also, QoS-ACA outperforms RRDA when TTL is greater than 500ms and link reliability is lower than 0.3. This is due to the fact that RRDA has to be deterministic for supporting real-timeness and hence always ponders the worst case (longest delay) which means every packet may reach (if it could reach) its upstream node on the last retransmission. Therefore, downstream nodes cannot delay one packet more than the time is assigned to them. But it is also likely that the packet is received by an upstream node before *n* retransmissions. Therefore, the downstream nodes could spent this extra time for their own benefit and do more retransmission while meeting the packet deadline. QoS-ACA exploit this fact in order to increase hit ratio in case of large TTL and low link reliability.

2) *Energy Consumption*

Figure 6 provides a comparison between useful energy consumption (energy spent for packets received before their expiration) and total energy consumption for these three reliable approaches and for two duty cycles 0.99 (right graphs) and 0.1 (left graphs). It is clear that there is almost no difference between useful and total energy consumption for READ, as it drops packets which are more likely not to reach the base station on time. As illustrated in Figure 6, READ is much more energy efficient than QoS-ACA and S-W ARQ particularly in case of low reliable links and short TTL. The reason for this is that in case of low reliable links, data packets or acknowledgement packets are much more likely to get lost. In addition, in case of short TTL, intermediate nodes in QoS-ACA and S-W ARQ approaches will still relay expired packets towards the base station which comes in the expense of energy consumption.

VI. A HYBRID APPROACH

Figure 5 shows each of these three approaches outperforms the other two under some conditions. Therefore, it is more efficient to leverage the benefit from each in a hybrid approach to achieve maximum performance in terms of hit ratio. The idea behind this hybrid approach is to make a selection among these three approaches by looking at the TTL of the packet and link reliability interval related to a given chain. The performance of the hybrid approach is plotted in Figure 7. One can see, this hit ratio graph inherits advantages of the associated graphs of Figure 5.

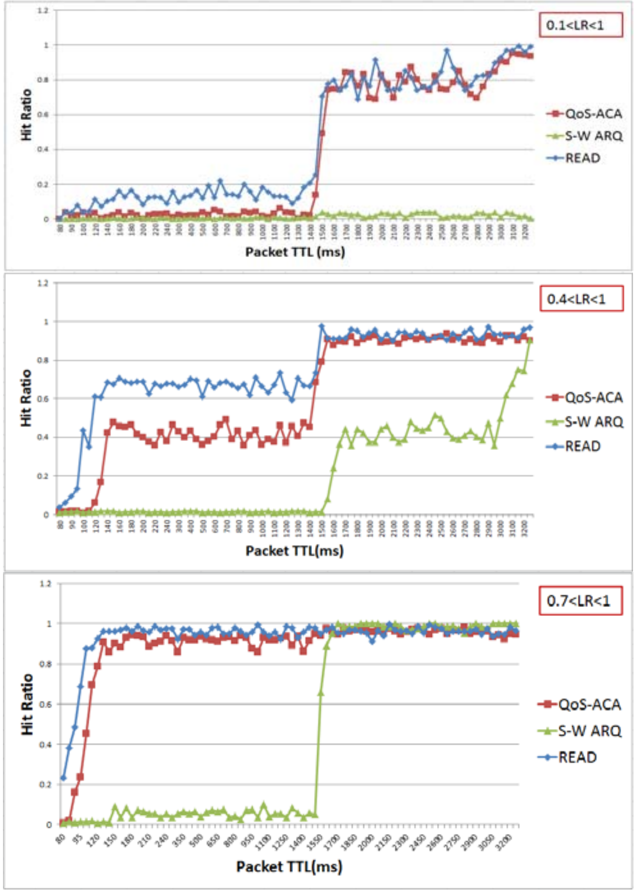

Figure 4. Hit ratio vs. Packet deadline for 0.1<LR<1(top),0.4<LR<1(middle),0.7<LR<1 (bottom)

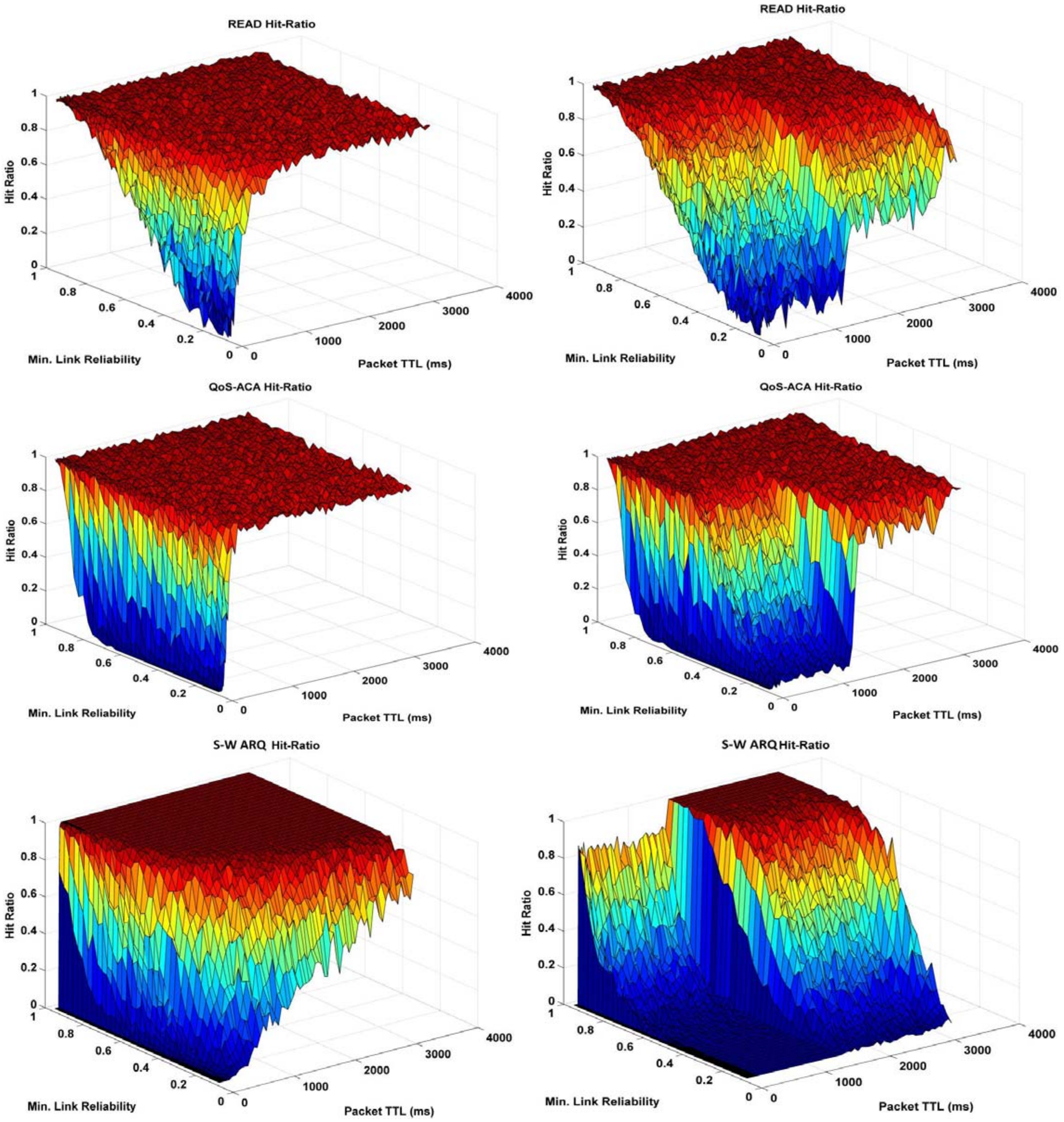

Figure 5. Hit ratio vs. Link reliability vs. Packet deadline for READ (top) QoS-ACA(middle) and S-W ARQ(bottom) while duty cycle of the left graphs is 0.99 and right graphs is 0.1

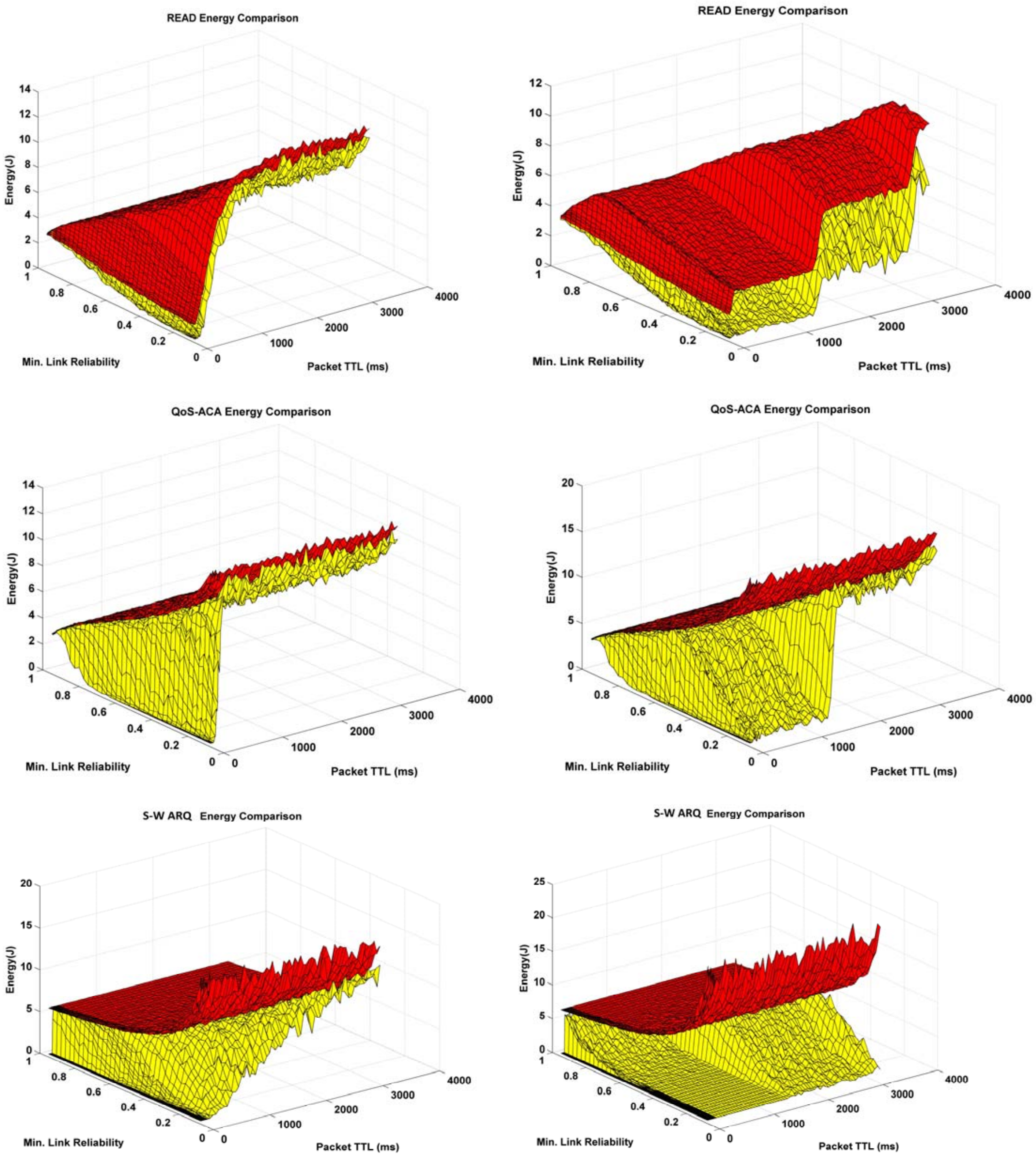

Figure 6. Comparison Between Total and Useful Energy Consumption for READ (top) QoS-ACA(middle) and S-W ARQ(bottom) while duty cycle of the left graphs is 0.99 and right graphs is 0.1

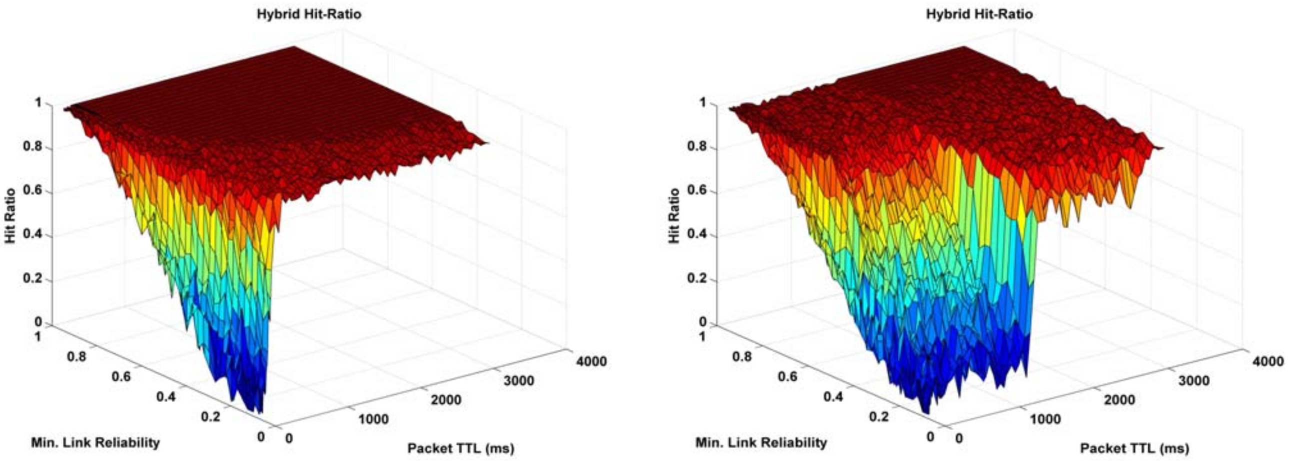

Figure 7. Hit ratio vs. Link reliability vs. Packet deadline for Hybrid approach while duty cycle of the left graphs is 0.99 and right graphs is 0.1

VII. CONCLUSION AND FUTURE WORK

In this paper, we propose READ, a reliable and real-time aggregation-aware data dissemination protocol designs for long chain-type wireless sensor networks, to cope with the problem of efficient data gathering of delay constrained sensor data. 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. We investigate the relatively unexplored relationship between TTL and link reliability with the attained hit ratio for time-critical WSNs.

READ allocates available packet’s TTL proportionately to the packet loss probability of the links along the forwarding path in order to judiciously and fairly use the packet’s TTL on intermediate nodes in such a way that reliability gain and on-time end to end delivery ratio is maximized.

READ assumes the packet loss of each link is fixed for each simulation and therefore does not update the packet loss of the links dynamically based on the last status of the links. This is due to that fact that in this paper we focus on finding the relationship between TTL and link reliability with the attained hit ratio. In our ongoing work, we enhance READ by considering dynamic changes of links reliability and will modify READ in such a way to be able to adaptively change number of copies each node is allowed to send based on the last status of the link reliabilities.

We also consider comparing READ with forward error correction mechanisms in on future work to know how well READ functions .

VIII. ACKNOWLEDGEMENT

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

REFERENCE

[1] Dezfouli B., Radi M., Nematbakhsh M., and Abd Razak S.,: A Medium Access Control Protocol with Adaptive Parent Selection Mechanism for Large-Scale Sensor Networks. In IEEE 24th International Conference on Advanced Information Networking and Applications, pp 402-408, Singapore (2011).

[2] Taghikhaki, Z., Meratnia, N., Zhang, Y., and Havinga, P.: QoS-aware Chain-based aggregation in cooperating VCN and WSN. In the book of Roadside Networks for Vehicular Communications: Architectures, Applications and Test Fields, In press (2012).

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

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

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

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

[7] Lindsey, S., Raghavendra, C.S.: PEGASIS: Power-efficient gathering in sensor information systems. In IEEE Aerospace Conference, pp 1125-1130, Montana (2002).

[8] J. Walrand, Communication Networks: A First Course. Boston, MA: Irwin and Aksen, (1991).

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

[10] M. Schwartz, Telecommunication Networks: Protocols, Modeling and Analysis. CA: Addison-Wesley, (1987).