Trading Transport Timeliness and Reliability for Efficiency in Wireless Sensor Networks

Vinay Sachidananda, Abdelmajid Khelil, Dhananjay Umap, Matthias Majuntke and Neeraj Suri
TU Darmstadt, Hochschulstr. 10, 64289 Darmstadt, Germany
{vinay,khelil,umap,majuntke,suri}@informatik.tu-darmstadt.de

***Abstract*—A key task in wireless sensor networks is to deliver information from sensor nodes to the sink. Many applications require the delivery to be reliable and timely. However, increasing reliability/timeliness comes at the cost of higher energy consumption as in both cases additional messages have to be sent: Retransmissions to increase reliability and information delivery via a second, faster path to ensure timeliness. Existing transport protocols either over- or under-provide reliability and/or timeliness and lack optimized efficiency. This work aims in tuning reliability and timeliness in composition for a maximized efficiency. Our approach's takes the reliability/timeliness requirements as input and features a message efficiency that optimally meets user requirements. Information transport proceeds in two steps in a fully distributed way: (i) Finding the optimal number of retransmissions on a per hop basis with delay compensation, and (ii) path split and/or replication if reliability or timeliness requirements are violated. We validate the approach viability through extensive simulations for a wide range of requirements and network conditions.**

***Keywords*-Wireless Sensor Networks, Information Quality, Transport, Protocols, Timeliness, Reliability**

I. INTRODUCTION

In Wireless Sensor Networks (WSNs) delivering the gathered information with the user required quality is the main concern. To satisfy the user required quality, we should carefully design the functional blocks, such as (a) sampling schemes in order to accurately represent the physical phenomena, and (b) transport schemes in order to reliably and timely deliver the information to the sink. In our work, we focus on the key operations of information transport and their quality attributes, i.e., transport timeliness and reliability.

Achieving the best possible timeliness and reliability is related to a large overhead regarding resources, particularly because sensor nodes rely on batteries. A higher reliability usually is achieved through a higher number of retransmissions resulting in a higher energy/bandwidth overhead. Timeliness may require path splitting instead of simple retransmissions on the same path, thus, causing higher traffic related with higher energy/bandwidth overhead. Hence, besides attaining the required quality levels, it is indispensable to maximize energy/bandwidth efficiency.

Fortunately, varied applications may be satisfied with varied reliability and timeliness levels. To reduce deployment costs, WSNs are more and more required to serve multi-users for multi-purposes. For instance, the purpose of a WSN deployment may suddenly need to be changed. For instance upon a catastrophic event, the WSN should support rescue operation and stop unnecessary monitoring activities. In future smart cities and rural areas, public WSNs should deliver different information entities for varied authorities or users.

Here, multi-users may use multi-sinks with possibly different actuation plans to react on the information delivered by the WSN. In a future wireless automation scenario such as the smart grid, different sinks may rely on wireless sensor information to control wireless actuators such as valves and switches. Opposite to automation scenarios that usually require higher reliability and timeliness, biologists may tolerate the delayed and lossy delivery of forest temperature data. Lower application requirements represent an opportunity to increase the WSN efficiency.

Common to all these observations is that different information entities are generated and should be transported to their corresponding users/sinks. Typically, users have different requirements on transport timeliness and reliability. Timeliness requirements may range from strict and realtime to soft deadlines that can vary from seconds to minutes to hours. Varied WSN users usually require best effort reliability with different levels of efficiency. Best effort reliability requirements can be expressed in message delivery success rate or ratio of event detection false positives or false negatives. As WSN is more and more used for multi-purpose deployments, the WSN protocol suite such as information transport should provide for tunability in order to support these applications with varied/evolvable reliability and timeliness requirements while maximizing efficiency.

Available approaches usually optimize for best effort reliability or timeliness. As it is not always required to provide best effort reliability or timeliness, it is challenging to just provide the user required performance. Unfortunately, there are no efforts addressing the tunability of both reliability and timeliness in composition. In this work, we address this tradeoff by providing the user required evolvable reliability and timeliness levels while maximizing efficiency.

Achieving both transport reliability and timeliness while maximizing efficiency requires a sophisticated tradeoff technique, which is the main contribution of this paper. It is complex to tune timeliness and reliability in composition, therefore, we progress stepwise to master the complexity. First, we prioritize timeliness to reliability and provide user required

---

Research supported in part by DFG GRK 1362 (TUD GKmM).

timeliness with reliability as best effort. Second, we combine the tunability of both reliability and timeliness to provide the tradeoff solution by satisfying the user requirements and by maximizing the efficiency of the network. We provide two information transport protocols. The contributions in this paper are:

- The rT algorithm that provides tunable timeliness with best effort reliability. This algorithm finds the optimal number of retransmissions and implements a delay compensation on a per hop basis. If delay compensation is not effective, a path replication is conducted.
- The RT algorithm that provides tunable reliability and timeliness in composition. RT extends rT by path replication if either retransmissions or delay compensation at the same path are not effective, i.e., replicating the path if either timeliness or reliability requirements are violated.
- We show the performance of our algorithms against previous efforts through extensive simulations. To the best of our knowledge, we are not aware of any other algorithm that achieves the tuning of both reliability and timeliness satisfying the varying user evolvable requirements.

The structure of the paper is as follows. In Section II, we present the related work. Section III describes the preliminaries with system model and terminology. In Section IV, we detail our approach on designing tunable reliability and timeliness, i.e., adaptive techniques for retransmissions, delay compensation and if needed path splitting or replication. We provide the performance evaluation results in Section V.

II. RELATED WORK

To guarantee both reliability and timeliness in WSNs is not straight forward due to the dynamic environmental and network conditions. For WSN and ad hoc networks several Quality of Service (QoS) provisioning protocols have been proposed [3][11][22]. Similar to wired networks, these approaches consider node to node communication ignoring the sink-oriented communication in WSN. On the other hand, these approaches consider only a few coarse-grained classes of application requirements. In this work, we provide a fine-grained consideration of application requirements. The state-of-the-art in data transport in WSN focuses either on the reliability ([7][6][10][9][8][5][1]) or timeliness ([12][13][14][15][2][16]) or both ([17][4][24][25]). Moreover, none of these studies addresses the tunability of reliability and timeliness in composition.

In [7], the reliability of convergecast applications is addressed. In [6], diversity coding and the dissemination of packets over multiple disjoint paths are considered. [10] uses a simple time averaged estimator to model the reliability. Experiments in [9] provide some insight to the behavior of link reliability with regard to physical and Medium Access Control (MAC) layers. However, [7][6][10][9] do not provide for tuning the transport reliability. [8][5] propose leverage path redundancy in WSN for service differentiation in the reliability domain but require global network topology knowledge. GIT [1], aims at satisfying the end-to-end (e2e) reliability by dividing the reliability per hop. The proposed information transport protocol is tunable regarding the achievable reliability. However, GIT neglects the attribute timeliness.

In CFLOOD [12], the authors address the problem of flooding and improve it with a new concept of controled flooding. Due to the controlled flooding timely delivery of the packets is possible. However, the authors miss the important aspect of reliability and target maximum reliability as they focus on the detection of critical events. By means of a Time Division Multiple Access (TDMA) scheme at the expense of limiting the length of routing paths delay guarantees are provided in [13]. Traffic regulation mechanisms are explored as means to provide e2e guarantees with combination of queuing models and message schedulers in [14]. In [15], the packets are scheduled with high/low priority in the velocity-monotonic order without any guarantee in the e2e sense. In [2], to maximize the throughput for non-real-time traffic a scheme is proposed to minimize the delay in real-time traffic. In [16], e2e deadline guarantees for real-time packets in WSN are provided. However, the above protocols overlook to provide the user defined timeliness and consequently to tune reliability and timeliness in composition.

In [17], the authors propose probabilistic multi-path forwarding to ensure e2e delays. This approach is not adaptable to fluctuating network conditions and evolving user requirements. In MMSPEED [4], probabilistic techniques are applied for service differentiation. However, [4] aims at providing strict conditions for messages and unfortunately does not support tuning of both reliability and timeliness. As part of the performance evaluation, we compare our work with GIT [1], CFLOOD [12] and MMSPEED [4].

III. PRELIMINARIES

After discussing the considered system model, we precisely define the terminology we use in this paper.

A. System Model

We consider homogeneous WSNs with N sensor nodes and one sink. We allow both large-scale and small-scale WSNs, i.e., network sizes ranging from dozens to hundreds. Typically, each node is equipped with short range wireless communication, and shows limited processing, storage and energy capabilities. We allow the sink to be adequate in power, memory and processing capabilities. We assume that all nodes are static. Sensor nodes communicate with each other and the sink via bi-directional (multihop) wireless links. We assume a default Carrier Sense Multiple Access (CSMA)-based MAC and an underlying routing protocol, which provides a path for all nodes towards the sink. Each sensor node knows its direct neighbors, e.g., through beaconing. Typically, a sensor node sends local data (e.g., sensor readings) to the sink using the path determined by the routing protocol. We consider the communication disruptions constitute the most frequent failures hindering information transport in WSN, such as collisions, congestion constitute the major causes of message loss. At node level, we consider the message loss is caused by dropping messages from full buffers. Usually, the network conditions are dynamic.

Information is the collated and interpreted sensor data systematized by purposeful acumen and processing required for an application (e.g., event occurrence). The information is generated in the network from one sensor node (e.g., the sensor node detecting the event) and forwarded towards the sink. Without loss of generality, we assume the vicinity of the event detecting node will be congested due to activities such as event detection and aggregation. We consider multi-purpose WSNs, i.e., different applications are running simultaneously in the network. In addition, we allow that applications may change their requirements during the operation of the WSN. The requirements are disseminated to all nodes, e.g., through an efficient flooding protocol such as [21]. We assume that the most strict user requirements do not exceed the maximal capacity of the WSN [23].

B. Terminology

Consider Sensor Node S that is h hops from the sink and has an information to send to the sink with a user-specified reliability and timeliness. H_1, H_2,......H_h denote the h hops from S to the sink where H_i is the i^{th} hop from the sink. In the following, we define the reliability and timeliness as it is required by the user or achieved by a transport protocol.

- *Transport Reliability (R)* is the average success probability of the information to reach the sink.
- *User Desired Reliability (R_d)* is the average reliability as required by the user.
- *Link Reliability (R_{L_i})* is the achieved success probability of one single message transmission on Hop H_i.
- *Hop Reliability (R_{H_i})* is the achieved success probability after r transmissions of the same message on H_i.
- *Desired Hop Reliability ($R_{d_{hop}}$)* is the hop reliability to be maintained in order to achieve the overall user required reliability R_d.
- *Transport Latency (L)* is the time needed for the information to reach the sink.
- *User Tolerated Latency (L_{tol})* is the maximum delay allowed for the information to reach the sink.
- *Hop Latency (L_{H_i})* is the delay experienced on H_i.
- *Tolerated Link Latency ($L_{tol_{H_i}}$)* is the maximum delay allowed on H_i.

The transport reliability is the ability of the transport protocol to meet the desired reliability, i.e., $R = R_d$. The transport timeliness is the ability of the transport protocol to meet the tolerated e2e deadline, i.e., $L = L_{tol}$. The transport tunability is the ability of the transport protocol to just meet the required reliability without violating the tolerated e2e requirements, i.e., to ensure that $R = R_d$ *AND* $L = L_{tol}$. Being close to the requirements allows to maximize efficiency, which represents the key reasoning behind our approach.

IV. TRADING TIMELINESS AND RELIABILITY FOR EFFICIENCY

Before detailing our approach, we provide an overview on how our solution progress towards tuning both reliability and timeliness for information transport.

A. Guide Through the Approach

In Fig. 1, we illustrate three typical scenarios for information transport. These scenarios are the drivers to develop our algorithms.

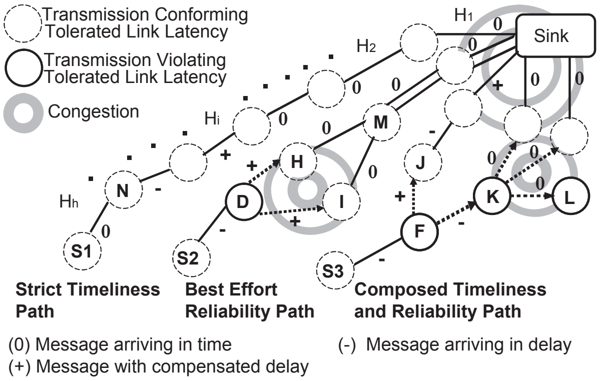

Fig. 1: Three illustrative scenarios for the proposed information transport.

In order to allow for a fully distributed solution, we propose to make per hop decisions. For instance, it has been proven that the per hop reliability in WSN outperforms the e2e acknowledgment and retransmissions [1]. Accordingly, hop-by-hop retransmissions towards the sink is the standard approach. To this end, the overall path reliability is equally divided among all hops on the path. Similarly, we design a timeliness strategy on a per hop basis. Our approach provides the desired application reliability despite evolving application requirements and dynamic network conditions by adopting the adaptive retransmission techniques for tunable reliability from [1]. We modify the tunable reliability scheme in [1] to couple the selection of appropriate retransmissions per hop to the allowed tolerated link latency.

In case all required retransmissions can be performed within the tolerated link latency on all hops along the path no modifications to [1] are required. If on a hop H_i the number of required transmissions are not possible without violating the $L_{tol_{H_i}}$, then appropriate countermeasures are needed. In the following, we briefly discuss these developed countermeasures, which represent our main contributions. As mentioned before and in order to master the complexity, we proceed progressively by considering the three basic scenarios illustrated in Fig. 1, i.e., the information entities sent by S_1, S_2 and S_3.

Delay Compensation: Consider $S1$ that generates an information and sends it to the sink. We assume that Node N requires a number of retransmissions which would violate the tolerated link latency. If the caused delay does not exceed a portion (say δ) of the tolerated link latency of the next hop, we propose a scheme for delay compensation. This strategy ensures strict timeliness notion while providing the best effort reliability.

Delay Compensation with Path Split: Consider $S2$ has made delay compensation, however, Node D can not conduct delay compensation anymore as the link latency would exceed the δ of next hop's tolerated latency. Accordingly, we propose a mechanism to split the path to ensure $R_{d_{hop}}$ within the required $L_{tol_{H_i}}$. We refer to path split by sending the same message to two neighboring nodes.

Delay Compensation with Path Replication: Consider the scenario of $S3$. Node F requires delay compensation and path split to two neighboring sensor nodes J and K. However, delay compensation and path split are not sufficient at Node K. Hence, Node K has to conduct path replication to three neighbors (the number three is based on the number of remaining retransmissions). We refer to path replication by the fact of sending the same message to three or more neighboring nodes.

In all the scenarios above, we briefly explained how our approach efficiently finds the tradeoff between provisioning the reliability and timeliness on one side and minimizing the number of retransmissions on the other side, through delay compensation, then path split, then path replication if required. We note that path split and path replication are local decisions and the paths may converge to the same path after a certain number of hops (this means a node may forward the same message more than once, e.g., Node M).

B. Mapping User Requirements

Our aim is to satisfy user required reliability and timeliness. As we follow a hop-by-hop reliability and timeliness assurance, we should carefully map the e2e user requirements to the single hops. Obviously, the hop-by-hop selection of requirements should satisfy $1 - \prod_{i=1}^{h}(1 - R_{d_{hop}}) \geq$ R$_d$ and $\sum_{i=1}^{h} L_{tol_{H_i}} \leq$ L$_{tol}$. For satisfying the user required reliability R_d we adopt the per-hop decisions which are equally distributed to every hop according to Eq. (3). Recall that h is the total number of hops from the information source to the sink.

To satisfy the required timeliness, we need a mechanism to perform per-hop decisions. Usually, the per-hop deadline computation can follow a constant, increasing or decreasing function. A constant function allocates the e2e deadline evenly to all the hops from the source to the sink, implicitly assuming that a packet would suffer the same delay at each hop.

Intuitively, in a convergecast network, the closer a node to the sink, the greater will be the traffic that the node has to forward towards the sink. Thus, longer will be the delay that a packet will suffer at nodes closer to the sink. Accordingly, a longer hop deadline should be assigned for the hops closer to the sink. Thus, the partitioning/mapping function should be increasing. This assumes that congestion occurs only in the surrounding of the sink (e.g., path from $S2$ in Fig. 1). The growth of deadlines can be then either linear, polynomial or exponential. Inspired by exponential back-off algorithms that double the retrial time upon an unsuccessful medium access, we propose to use an exponential growth for deadlines.

Similarly, the information source area usually undergoes high communication activities (event detection, aggregation, etc). In some scenarios, more than one node from the event area will report information to the sink. This increases the contention level in that area. Accordingly, an information source should select higher hop deadlines. Usually, the contention at the source node is lower than at the sink that would receive data from different information source areas simultaneously. Between the information source area and the sink shorter hop deadlines can be allocated as messages may select different disjoint less loaded paths. In the following, we introduce a novel deadline partition model.

Considering both contention effects above, the hop deadline allocation can be calculated as an exponential decrease with the distance from the source ($\epsilon * e^{-\alpha*(h-h_i)}$) and an exponential increase towards the sink ($e^{\alpha*h_i}$). Accordingly, we propose to compute the tolerable latency on hop H_i using Eq. (1)

$$L_{tol_{H_i}} = \frac{\epsilon * e^{\alpha*(-h_i+(h/2))} + e^{\alpha*(h_i-(h/2))}}{\tau} + \beta \qquad (1)$$

$\epsilon \in [0.5,1]$ is a constant to address the fact that deadlines at the sink should be higher than at the source; α is a constant to control the gradient of increase/decrease; β is the minimum deadline that should be allocated to a hop; τ is the time scale factor to be able to select deadlines so that $\sum_{i=1}^{h} L_{tol_{H_i}} = L_{tol}$.

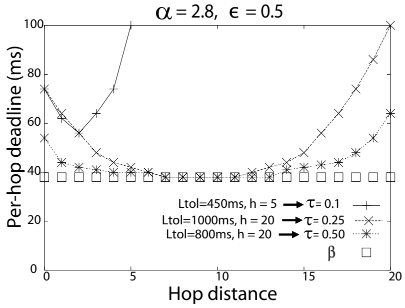

Fig. 2: Hop deadline distribution along a path

Fig. 2 exemplifies the deadline assignment for 3 paths. Given the constants ϵ, α and β, a source node that is h hops from the sink can compute an appropriate τ, so that $\sum_{i=1}^{h} L_{tol_{H_i}} = L_{tol}$ is valid. The source node forwards τ and h values along the information so that every node on the path to the sink can calculate its own deadline using Eq. (1) without violating the e2e timeliness requirement.

C. Tunability of Reliability

In this work, we adopt the tunable reliability concepts proposed in GIT [1]. Therefore, we briefly summarize the GIT approach on providing tunable reliability. To ensure the desired reliability on one hop, more than one transmission maybe required to overcome node and communication level pertubations. Given r the number of transmissions required, then the information transport reliability across Hop H_i is:

$$R_{H_i} = 1 - (1 - R_{L_i})^r \qquad (2)$$

Since r is the total number of transmissions, $\#ret_{max} = r-1$. For an R_d imposed by the application and known number of hops h from the sink, a source sensor node can calculate the desired reliability requirement across one hop as:

$$R_{d_{hop}} = R_d^{1/h} \qquad (3)$$

$R_{d_{hop}}$ is forwarded by the source node along the path to the relay nodes. Once the decision of sending the information is taken by the sensor node it calculates the maximum number of transmissions required to maintain the $R_{d_{hop}}$ using Eq. (2) [1] as follows:

$$r = \lceil \frac{\log(1-(R_{d_{hop}}))}{\log(1-R_{H_i})} \rceil \tag{4}$$

With Eq. (4) we can conclude that the achieved e2e reliability is a function of the path length and $\#ret$. The desired number of retransmissions required to satisfy the reliability R_d is without any time bound L_{tol}. Moreover, in the case of prioritizing reliability to timeliness either the timeliness is under- or over-provided. Prioritizing reliability to timeliness is appropriate for applications that are not sensitive to timeliness. Hence, in order to satisfy both reliability and timeliness, we need a supplemental mechanism.

D. Tunability of Timeliness

Now, we investigate prioritizing timeliness to reliability. The result is our first contribution, the rT algorithm, which is the first step towards a composite tunability, i.e., the RT algorithm. We discuss on how a possible delay can be compensated with and without path split. We first calculate the hop deadline distribution for satisfying the user specified requirements. Knowing, the value of L_{tol}, the source node divides the L_{tol} into tolerated hop latencies as according to Section IV-B. Though, we divide L_{tol} into $L_{tol_{H_i}}$, we take into consideration the retransmission probability on a relay node, where reliability $R_{d_{hop}}$ will be time bound by L_{tol}. We append the user messages with $R_{d_{hop}}$, L_{C_i}, τ and h, where L_{C_i} is the cumulative latency from the source to Hop H_i.

We compensate the delay when the tolerated hop latency for intermediate an hop H_i is not met, as explained below. Unfortunately, in the case of prioritizing timeliness to reliability the reliability maybe either under- or over-provided, i.e., $R_{H_i} < R_{d_{hop}}$ or $R_{H_i} > R_{d_{hop}}$. Thus, prioritizing timeliness to reliability, leads to best effort reliability.

We now present how delay compensation and path split can be achieved to better satisfy the e2e deadline. For this, we propose the rT algorithm. In any intermediate hop H_k, if $L_{C_k} > \sum_{i=h}^{k} \mathrm{L}_{tol_{H_i}}$, the process of compensating the delay and path split is conducted. On the other hand, the mechanism of path split to two neighboring nodes is also done. We split the path to two neighboring nodes because L_{tol} is without any bound R_d. However, it is not always true that we can compensate the delay and meet the e2e requirements of reliability and timeliness.

During tolerated link latency calculation, a node calculates the number of retransmissions that can be achieved for that current hop and tolerated link latency. Usually, we have sensor nodes meeting the tolerated link latency (i.e., $L_{C_k} < \sum_{i=h}^{k} \mathrm{L}_{tol_{H_i}}$) (Alg. 1, L. 15-16). However, if the link delay for a hop H_k is larger than the tolerated link latency, it is unfortunate to receive the ACK from hop H_{k-1} at H_k. Hence, when $L_{C_k} > \sum_{i=h}^{k} \mathrm{L}_{tol_{H_i}}$, our algorithm compensates the delay by borrowing the time from next hop (Alg. 1, L. 20-22). The compensation condition is that $\delta = L_{C_k} - \sum_{i=h}^{k} \mathrm{L}_{tol_{H_i}}$ varies from $0 < \delta < 0.3 * L_{tol_{H_{k-1}}}$.

If the δ condition is violated, compensating the delay at Hop H_k, may not allow to send the required number of retransmissions. Hence, the path split approach decides to forward the information to two neighboring nodes H_{k-1} and H'_{k-1} (Alg. 1, L. 24-27). Though, tolerated link latency $\sum_{i=h}^{k} L_{tol_{H_i}}$ for the intermediate hop H_k is not met, the reliability R_d across the hops is increased. After the message is forwarded to both next hops, the receivers will still check the condition $L_{C_k} > \sum_{i=h}^{k} \mathrm{L}_{tol_{H_i}}$, for compensating the delay. However, if the delay is higher than L_{tol}, the message is just dropped (Alg. 1, L. 29).

Though, we compensate the delay and split the path, we may still fail to ensure both the desired reliability and timeliness. Hence, in this case, we need a further mechanism.

Algorithm 1 Tunable Timeliness with Best Effort Reliability (rT Alg. at Hop H_k)

```
1:  Const: $\epsilon$, $\beta$, $\alpha$, $TO$
2:  Var: $L_{tol_{H_k}}$, $L_{C_k}$
3:  start timers T1, T2;
4:  if $k$==$h$ then
5:      /*Source node*/
6:      $L_{C_k}$=0; $\delta$=0; $h$= No. of hops to the sink;
7:  else
8:      Upon receiving a data $msg$
9:      extract $\tau$, $h$, $R_{d_{hop}}$, $L_{C_{k+1}}$
10:     $L_{C_k}$ = $L_{C_{k+1}}$;
11:     send ACK to $H_{k+1}$;
12:     $\delta$ = $L_{C_k} - \sum_{i=h}^{k} L_{tol_{H_k}}$;
13: end if
14: /*If message is not delayed*/
15: if $\delta < 0$ then
16:     $rT\text{-}Transport(msg,\ H_{k-1})$; exit();
17: else
18:     /*message is delayed*/
19:     if ($\delta \leq 0.3 * L_{tol-H_k}$) then
20:         /*Delay $\leq$ threshold $\rightarrow$ Delay Compensation Scheme*/
21:         $L_{tol_{H_k}} \mathrel{+}= 0.3 * L_{tol_{H_{k-1}}}$;
22:         $rT\text{-}Transport(msg, H_{k-1})$; exit();
23:     else
24:         if ($0.3 * L_{tol-H_k} < \delta \leq L_{tol-H_k}$) then
25:             Select a second next-hop $H'_{k-1}$
26:             $rT\text{-}Transport(msg, H_{k-1})$;
27:             $rT\text{-}Transport(msg, H'_{k-1})$; exit();
28:         else
29:             exit(); /* e2e deadline violated*/
30:         end if
31:     end if
32: end if
33: /*Upon receiving an ACK */
34: Stop timers T1, T2;
35:
36: /*Function $rT\text{-}Transport()$*/
37:
38: transport($msg$, $H_{k-1}$); {
39: /*Do $\#ret$ that are allowed in tolerated hop latency*/
40: while (T2.value() < $L_{tol_{Hk}}$) do
41:     for (i=0, i<$r$, i++) do
42:         $L_{C_k} += T.value()$;
43:         T1.reset();
44:         msg.append($\tau$, $h$, $R_{d_{hop}}$, $L_{C_k}$);
45:         send msg to $H_{k-1}$;
46:         wait for ACK or $TO$ expiration;
47:     end for
48: end while
49: }
```

E. Composite Tunability of Timeliness and Reliability

In this section we propose a solution which would provide the composite tunability of reliability and timeliness i.e., the RT algorithm.

1) Composite Timeliness and Reliability: A sensor node at Hop H_k includes $R_{d_{hop}}$, L_{C_k}, τ and h to the message when it forwards it to next hops. However, to achieve tunability and to reach a suitable tradeoff between reliability and timeliness, we need a holistic investigation of r, $\sum_{i=h}^{k} \mathrm{L}_{tol_{H_i}}$, R_d, $R_{d_{hop}}$, L_{tol} and L_{C_k}. To achieve the tradeoff between reliability and timeliness, the decision is based on nodes local network conditions and application requirements.

The path replication approach ensures for compensating the loss of reliability in any intermediate hop H_k when $L_{C_k} > \sum_{i=h}^{k} \mathrm{L}_{tol_{H_i}}$ and $R_{d_{hop}}$ is not satisfied. In order to maintain the required information transport reliability and timeliness, each node along the path dynamically adapts r according to its local timeliness and reliability requirements.

Algorithm 2 : Composite Tunability of Reliability and Timeliness (RT Alg. at Hop H_k)

```
1:  Const: $\epsilon$, $\beta$, $\alpha$, $TO$
2:  Var: $R_d$, $R_{d_{hop}}$, $L_{tol_{Hk}}$, $L_{C_k}$
3:  start timers T1, T2;
4:  **if** ($k$==$h$) **then**
5:      /*Source node*/
6:      $R_{d_{hop}}$ = $R_d^{1/h}$;
7:      calculate $r$ using Eq. (4);
8:      $L_{C_k}$=0; $T2T$ = $L_{tol_{Hk}}$; $h$= No. of hops to the sink;
9:  **else**
10:     /*Upon receiving a data message $msg$ */
11:     extract $\tau$, $h$, $R_{d_{hop}}$, and $L_{C\,k+1}$
12:     $L_{C_k}$ = $L_{C_{k+1}}$;
13:     send ACK to $H_{k+1}$;
14:     $T2T$ = $L_{tol_{Hk}}$ − ($L_{C_k}$ − $\sum_{i=h}^{k} L_{tol_{Hk}}$);
15:     $R_{d_{hop}}$ = $R_d^{1/h}$;
16:     calculate $r$ using Eq. (4);
17: **end if**
18: /*If desired hop reliability can be satisfied and msg is not delayed*/
19: **if** (S of possible trans in $T2T \geq r$) **then**
20:     $RT\text{-}Transport(msg,\ H_{k-1},\ r)$; exit();
21: **else**
22:     $\delta$ = $T4r\text{-}T2T$;
23:     **if** ($\delta \leq 0.3 * L_{tol-H_k}$) **then**
24:         $L_{tol_{H_k}} \mathrel{+}= 0.3 * L_{tol_{H_{k-1}}}$;
25:         $RT\text{-}Transport(msg,\ H_{k-1},\ r)$; exit();
26:         **if** ($0.3 * L_{tol-H_k} < \delta < L_{tol-H_k}$) **then**
27:             **if** ($r > \sigma$) **then**
28:                 /*Path Split*/
29:                 select a second next-hop $H'_{k+1}$;
30:                 compute $r1$ for $H_{k-1}$ and $r2$ $H'_{k+1}$; /*$r = r1 + r2$*/
31:                 $RT\text{-}Transport(msg,\ H_{k-1},\ r1)$;
32:                 $RT\text{-}Transport(msg,\ H'_{k+1},\ r2)$; exit();
33:             **end if**
34:         **else**
35:             **if** ($r < \sigma$) **then**
36:                 /*Path Replication*/
37:                 Compute remaining $r_k$ for $H_{k-1}$;
38:                 select $H_n$ neighbors;
39:                 compute $r_n$ for $H_{n-1}$;
40:                 $RT\text{-}Transport(msg,\ H_{n-1},\ r_n)$; exit();
41:                 **if** ($R_{d_{hop}} \geq R_d$) **then**
42:                     send Implicit ACK to $H_{k-1}$;
43:                     $RT\text{-}Transport(msg,\ H_{k-1},\ r)$; exit();
44:                 **end if**
45:             **else**
46:                 Exit();
47:             **end if**
48:         **end if**
49:     **end if**
50: **end if**
51: /*Upon receiving an ACK */
52: Stop timers T1, T2;
53: /*Function $RT\text{-}Transport()$*/
54:
55: $RT\text{-}Transport(msg, H_{k-1}, r)$ {
56: **while** ($T2.value() < L_{tol_{Hk}}$) **do**
57:     **for** (i=0, i<$r$, i++) **do**
58:         $L_{C_k} \mathrel{+}= T.value()$;
59:         T1.reset();
60:         msg.append($\tau$, $h$, $R_{d_{hop}}$, $L_{C_k}$);
61:         send msg to $H_{k-1}$;
62:         wait for ACK or $TO$ expiration;
63:     **end for**
64: **end while**
65: }
```

2) Trading off Timeliness and Reliability: Now, we provide the composite tunability of the optimal timeliness along with improving the reliability of the information reaching the sink. Fig. 1 illustrates the algorithm execution.

The sink spreads the user defined e2e R_d and L_{tol} to all nodes. The source nodes (e.g.,S_1, S_2 and S_3 in Fig. 1) accordingly calculate the tolerated link latency and per-hop reliability. Node S_1 retransmits the message until it receives an implicit ACK by listening to a forward of the same message.

Now, consider Node S_2 during its first hop to D. The transmission of S_2 meet the tolerated link latency and also the per-hop reliability. For Node D, $L_{C_k} > \sum_{i=h}^{k} \mathrm{L}_{tol_{H_i}}$ and the per-hop reliability is lower than that required by the user, therefore, Node D first compensates the delay and also starts path split (Alg. 2, L. 22-32). Nodes H and I send implicit ACKs to Node D after the information is correctly received and forwarded. Nodes H and I forward the information to the next hop meeting the tolerated link latency $\sum_{i=h}^{k} \mathrm{L}_{tol_{H_i}}$. The next hop nodes forward the received information to the sink, until the tolerated link latency expires or an ACK is received.

Considering Node S_3, the information is forwarded to Node F. However, $L_{C_k} > \sum_{i=h}^{k} \mathrm{L}_{tol_{H_i}}$ at Node F. Apart from delay compensation, Node F decides on path split (Alg. 2, L. 22-32). The information forwarded to Node J meets the next tolerated link latencies and information reaches the sink. However, Node K suffers from supplemental delay. Hence, Node K conducts path replication to three nodes to have the required tradeoff between reliability and timeliness. The number of neighboring nodes for path replication are decided based on the required number of retransmissions and L_{C_k}, and if $r < \sigma$, then the path replication is carried on with adapting the reliability and timeliness (Alg. 2, L. 35-40). Except Node L, the other two nodes which received the information from Node K send an implicit ACK to Node K. As Node L suffers from an additional delay, Node L sends a negative ACK to Node K or tolerated link latency expires before receiving ACK. If Node K receives positive ACK from other two neighboring nodes the retransmission to Node L is canceled while fulfilling the user timeliness and reliability requirements. Node K forwards the information to next hops meeting tolerated link latency and delivering the information to the sink.

V. PERFORMANCE EVALUATION

In order to evaluate our work, we first describe the simulation environment, simulation settings and the performance metrics. Next, we present our simulation results. We evaluate our approach based on simulations in TOSSIM [20].

A. Simulation Environment and Studies

We simulate between 20 to 200 sensor nodes in an area of 75×75 unit^2 which is partitioned in a grid topology. The distance between two neighboring nodes is 5 units. The sink is located at one corner. The information is generated from one corner and from the middle of the network and transported towards the sink. The e2e latency is measured in *ms*. We set α = 2.8, β = 2ms and ϵ = 0.5.

First, we perform simulations for fixed R_d and L_{tol}. To evaluate the tunability of the considered algorithms, we vary the desired reliability and keep the desired timeliness constant. Next, we keep the desired reliability constant and vary the desired timeliness. We select representative protocols from the existing literature as discussed in Section II and compare them with our work. The competitor protocols we have chosen are GIT [1], $CFLOOD$ [12] and $MMSPEED$ [4].

The performance of our protocol is measured in terms of reliability, timeliness and average number of transmissions (which includes all the transmissions and retransmissions from the source node to the sink). We also varied the traffic by varying the information rate, i.e., the number of messages sent from the source per second. In order to address various perturbation levels, we varied the Bit Error Rate (BER).

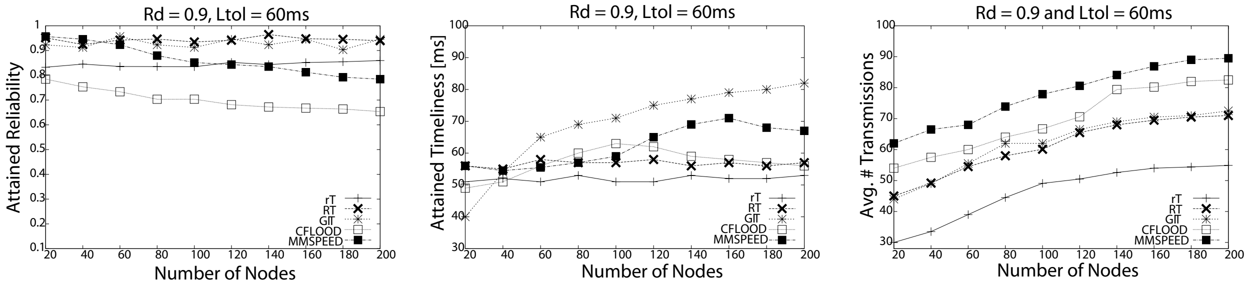

Fig. 3: Impact of network size for fixed desired reliability and timelinesses

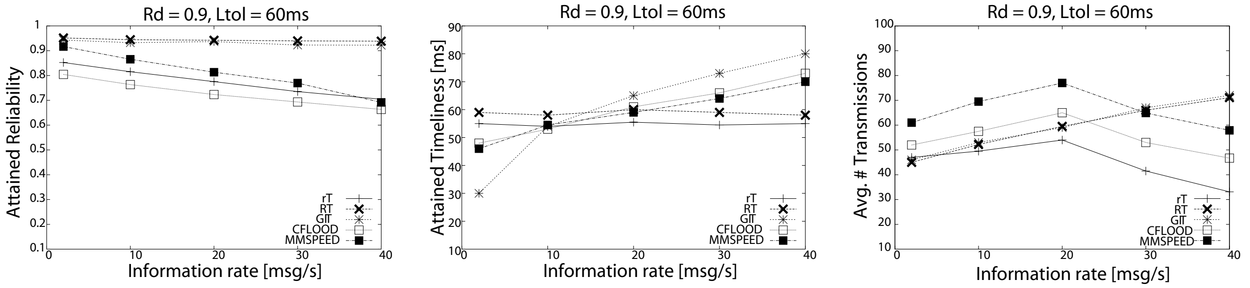

Fig. 4: Impact of network load for fixed desired reliability and timelinesses

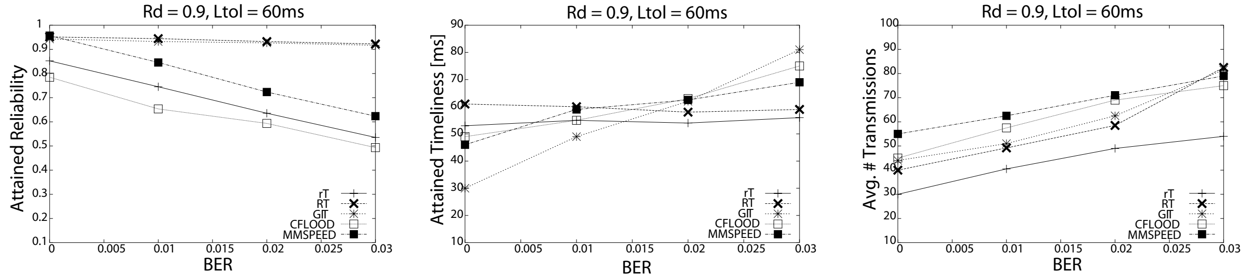

Fig. 5: Impact of BER for fixed desired reliability and timelinesses

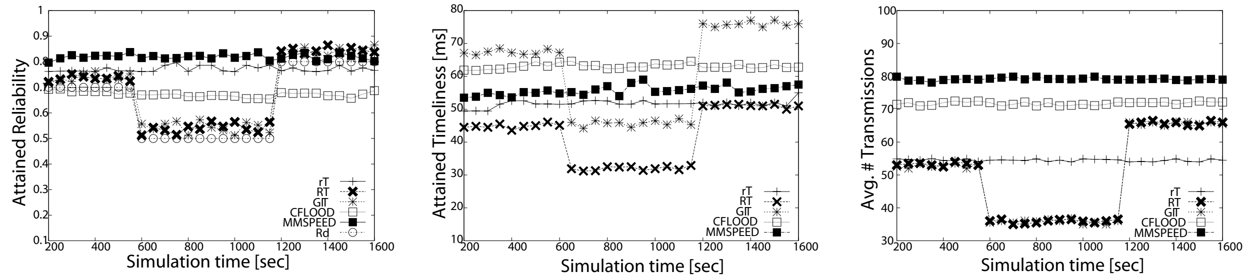

Fig. 6: Tunability for varying desired reliability

B. Simulation Results

1) Fixed User Requirements and Varying Network Properties: We fix the user requirements on information transport, i.e., R_d = 0.9 and L_{tol} = 60ms.

(a) Impact of Network Size: Fig. 3 shows the performance for different number of nodes. We observe that RT and GIT attain the desired reliability with a slight difference. The reliabilities attained by rT, $CFLOOD$ and $MMSPEED$ are independent of the desired reliability. Regarding timeliness, RT meets the tolerated latency independent from node density and outperforms competitor protocols. The number of transmissions for RT are the lowest except GIT.

(b) Impact of Network Load: Fig. 4 shows the performance for different information rates. We observe that RT attains desired reliability with varying information rate. Regarding timeliness, the RT outperforms competitor protocols. The number of transmissions for RT and GIT differs only slightly.

(c) Impact of Perturbation Levels: Fig. 5 shows the performance for varied BER. We observe that RT always attains the desired reliability. RT outperforms competitor protocols by meeting tolerated link latency. Fig. 5 shows the total number of transmissions required to attain the desired transport reliability.

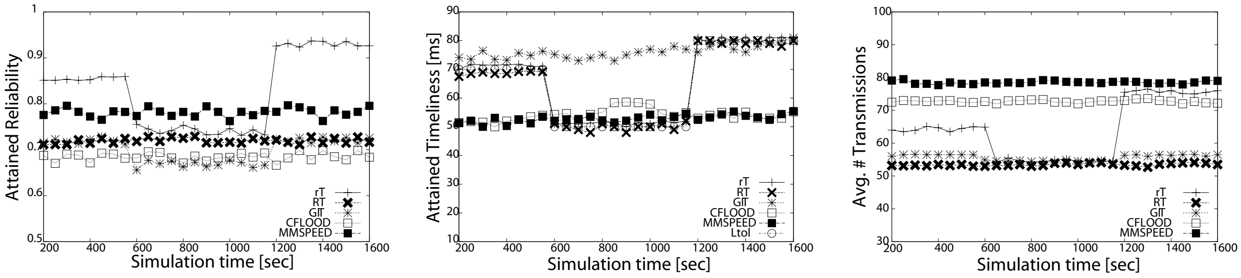

Fig. 7: Tunability for varying desired timelinesses

The number of transmissions for RT and GIT are very comparable. The number of transmissions varies with varying BER for rT, $CFLOOD$ and $MMSPEED$.

2) Varying User Requirements and Fixed Network Properties: Now we show the tunability by varying reliability and timeliness, we use 200 nodes for this study.

(a) Impact of Varying the Desired Reliability: Fig. 6 shows the adaptation to user requirements with varying reliability, i.e., R_d = 0.7, 0.5 and 0.8 and constant desired timeliness L_{tol} = 50ms (τ = 1.75). RT and GIT adapt according to the application requirements and provide tunable reliability. $CFLOOD$ and $MMSPEED$ do not adapt to the varying reliability requirements. RT outperforms other algorithms w.r.t timeliness, thanks to the tunability of RT. The benefit of RT can be observed in terms of #Transmissions (We are not showing the number of messages transmitted to disseminate the user requirements, which is a one time cost).

(b) Impact of Varying the Desired Timeliness: Fig. 7 shows the adaptation to application requirements with varying timeliness, i.e., constant desired reliability R_d = 0.7 and L_{tol} = 70ms, 50ms and 80ms and corresponding values of τ = 1.75, 2.65 and 1.65. RT adapts according to the application requirements with varying timeliness and provides the desired reliability (Fig. 7). The timeliness of RT is always satisfying and outperforms GIT, $CFLOOD$ and $MMSPEED$. RT performs better w.r.t #Transmissions with varying timeliness, because of its adaptability and path split mechanism.

VI. CONCLUSION

Through this paper, we have achieved the composite tuning of reliability and timeliness as per the application requirements. We have introduced the tunable timeliness which efficiently assigns the tolerable hop latencies on the path, compensates delays, and splits the path when needed. The optimized solution combines the re-transmission approach meeting the tolerable hop latencies and the path replication approach when the tolerable hop latency is violated. This is the first instance of tuning when the combination of both the reliability and timeliness is implemented. The present work is just focusing on the reliability and timeliness attributes and is further being extended for considering the sampling accuracy attribute. A higher sampling accuracy leads to higher energy/bandwidth overhead, and lower sampling accuracy might be out of scope to the user/application. Hence, all three attributes being interdependent make it is very challenging to find the optimal tradeoff satisfying transport reliability, transport timeliness and sampling accuracy requirements while maximizing efficiency.

REFERENCES

[1] F. K. Shaikh et al., Generic information transport for wireless sensor networks. In Proc. of SUTC, 2010, pp. 27 - 34.

[2] K. Akkaya et al., An energy-aware QoS routing protocol for wireless sensor networks. In Proc. of DCS Workshops, 2003, pp. 710 - 715.

[3] J. Martnez et al., QoS in wireless sensor networks: Survey and approach. In Proc. of EATIS, 2007, pp. 1 - 8.

[4] E. Felemban et al., MMSPEED: multipath Multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks. In Proc. IEEE Mobile Computing, 2006, pp. 738 - 754.

[5] B. Deb et al., ReInForM: Reliable information forwarding using multiple paths in sensor networks. In Proc. of LCN, 2003, pp. 406 - 415.

[6] A. Tsirigos and Z.J. Hass, Analysis of multipath routing, part 2: mitigation of the effects of frequently changing network topologies. In IEEE Trans. on Wireless Communications, 2004, pp. 500 - 511.

[7] Y. Sankarasubramaniam et al., ESRT: Event-to-sink reliable transport in wireless sensor networks. In Proc. of MobiHoc, 2003, pp. 1003 - 1016.

[8] S. Bhatnagar et al., service differentiation in sensor networks. In Proc. of Wireless Personal Multimedia Comm., 2001.

[9] J. Zhao and R. Govindan, Understanding packet delivery performance in dense wireless sensor networks. In Proc. of SenSys, 2003, pp. 1 - 13.

[10] A. Woo et al., Taming the underlying challenges of reliable multihop routing in sensor networks. In Proc. of SenSys, 2003, pp. 14 - 27.

[11] M. Ash et al., Quality of Service in mobile ad hoc networks: a survey. In Proc. of IJAHUC, 2010, pp. 75 - 98.

[12] B. Jiang et al., CFlood:A constrained flooding protocol for real-time data delivery in WSNs. In Proc. of SSS, 2009, pp. 413 - 427.

[13] A. Sahoo et al., An energy efficient MAC in wireless sensor networks to provide delay guarantee. In Proc. of LANMAN, 2007, pp. 25 - 30.

[14] K. Karenos and V. Kalogeraki, Real-time traffic management in sensor networks. In Proc. of RTSS, 2006, pp. 422 - 434.

[15] C. Lu et al., RAP:A real-time communication architecture for large-scale wireless sensor networks. In Proc. of RTAS, 2002, pp. 55 - 66.

[16] T. He et al., SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks. In Proc. of DCS, 2003, pp. 46 - 55.

[17] Huang, X et al., Multiconstrained QOS multipath routing in wireless sensor networks. In Proc. of Wirel. Netw, 2008, pp. 465 - 478

[18] Liu K et al., JITS: just-in-time scheduling for real-time sensor data dissemination. In Proc. of PerCom, 2006, pp. 5 - 46

[19] IETF: RFC 2681 - A Round-trip delay metric for ippm, http://www.ietf.org/rfc/rfc2681.txt?number=2681

[20] P. Levis et al., TOSSIM: Accurate and scalable simulation of entire tinyos applications. In Proc. of SenSys, 2003, pp. 126 - 137.

[21] A. Khelil et al., Gossiping: adaptive and reliable broadcasting in MANETs. In Proc. of LADC, 2007, pp. 123 - 141.

[22] P. Suriyachai et al., A Survey of MAC protocols for mission-critical applications in WSNs. In Proc. of IEEE CST, 2011, pp. 1 - 25.

[23] P. Gupta et al., The capacity of wireless networks. In Proc. of IEEE Transactions on Information Theory, 2000, pp. 388 - 404.

[24] P. Park et al., Breath: An Adaptive Protocol for Industrial Control Applications Using Wireless Sensor Networks. In Proc. of IEEE Transactions on Mobile Computing, 2011, pp. 821 - 838.

[25] V. Sachidananda et al., STC: Sampling and Transport Co-design in Wireless Sensor Networks. TR-TUD-DEEDS-04-01-2012.