Generic Information Transport for Wireless Sensor Networks

Faisal Karim Shaikh, Abdelmajid Khelil, Brahim Ayari, Piotr Szczytowski and Neeraj Suri
Department of CS, TU Darmstadt, Germany.
{fkarim|khelil|brahim|piotr|suri}@cs.tu-darmstadt.de

***Abstract*—A primary functionality of wireless sensor networks (WSNs) is transporting the information acquired by the sensors as per the desired application requirements. The diverse applications supported by WSNs also stipulate a diverse range of reliability requirements for the transport of various information types. The continuous variation of application requirements and dynamic operational perturbations complicates the design of a generic solution for information transport in WSNs. In this paper, we present a new framework for generic information transport (GIT), which considers varied application requirements and evolvable network conditions in WSNs. GIT manages the information and utilizes a probabilistic approach to ensure tunable reliability of information transport. The GIT framework is distributed in nature and performs its operations locally. The simulation results validate the tunability of the GIT framework. In some setups GIT achieves up to 4-5 times reduction in number of transmissions compared to existing approaches.**

***Keywords*-Wireless Sensor Network; Transport Protocol; Information Modeling; Tunable Reliability; Congestion Control**

I. INTRODUCTION

The applications running on wireless sensor networks (WSNs) stipulate their specific information (event/status) needs from the network. As the discrete WSN nodes are limited in battery, communication bandwidth and processing capabilities to meaningfully process the monitored raw data, a commonly advocated approach is to utilize distributed in-network pre-processing. Accordingly, the resultant information is composed of in-network pre-processing operations of filtering and selective aggregation on the raw data. The transport of this information across the network to the application via a gateway node termed as the *sink* is the core operation of WSNs. Given the diverse application driven nature of WSNs, it is typical to see customized solutions designed for information transport [1].

Normally, WSNs utilize intrinsic sensor node redundancy for assuring the desired information delivery reliability. However, the redundancy of sensor nodes comes at a cost for information transport, since the delivery of redundant information also depletes the limited node energy. Furthermore, different applications running on WSNs demand various types of information with diverse reliability requirements. For example, a critical event detection application may require high reliability of event delivery. Alternatively, a monitoring application can tolerate some loss of information. The varying application requirements impose consequent reliability obligations for the base information transport in a WSN. In addition, the same WSN application may change its requirements over time. Being an ad-hoc environment, WSN is subjected to a wide range of operational perturbations affecting both the nodes and their communication links. These perturbations naturally lead to deviation between the attained and the desired reliability of information transport. If the attained reliability is higher than the required, the information transport wastes valuable resources inside the network. Conversely, if the attained reliability is lower than desired, the information usefulness for the application can get compromised. The existing approaches have little or limited capabilities to cope with varying application requirements and evolving network conditions. Since, they are designed for specific applications and their main design driver is to efficiently maximize the attained reliability [1]. To the best of our knowledge, there exists no generalized solution that considers generic information, adapts to the varying application reliability requirements and copes with a wide range of dynamic operational conditions.

This paper targets a comprehensive solution for information transport in WSNs and accordingly proposes a Generic Information Transport (GIT) framework. Our approach is to design an adaptable solution which provides necessary tools to support generalized applications based on abstract system, perturbation, information and reliability models. GIT conducts its functionality in a decentralized manner. The proposed modular architecture keeps the generality of GIT intact by allowing different modules to adapt/reuse existing mechanisms. GIT provides best-effort latency, while contributing to reduction of the end-to-end delay by managing redundancy, mitigating perturbations and reducing the number of transmissions. The simulation results show that the GIT framework outperforms the existing solutions in terms of tunable reliability, resulting in significant decrease of transmissions.

In particular, this paper makes the following contributions.

- We design the GIT framework to provide tunable reliability of information transport for various information types despite evolving network conditions.
- We classify the different information types required by the users/applications and present an abstract generalized information model for WSNs. We develop mechanisms to efficiently consider the properties of the information of interest (type and level of redundancy).
- We adapt appropriate existing techniques in order to allow a localized/efficient and on-the-fly tunability of reliability.
- We develop a modular architecture in order to seamlessly integrate and tune the building blocks of GIT.

This research is supported in part by MUET, DFG GRK 1362 (TUD GKMM), EU INSPIRE, EU CoMiFin, and CASED (www.cased.de)

The paper is organized as follows. We present related work in Section II. Section III details our generalized underlying models. Section IV describes the proposed GIT framework. We evaluate the performance of GIT in Section V and present overall conclusions in Section VI.

II. RELATED WORK

There is no single protocol supporting the varied applications, providing tunable reliability and coping with evolving network conditions. We first provide a brief survey of existing information transport protocols that deal with reliability and cope with perturbations to some extent. To achieve information transport reliability in non-congested networks, approaches for temporal redundancy (e.g., retransmissions [2], [3]), information redundancy (e.g., error codes [4]), spatial redundancy (e.g., number of sources [5], [6] or paths [7], [8]) or some combinations of these [9] are typically utilized. To mitigate congestion a number of back-pressure schemes have been proposed in the literature [10], [11]. These schemes implicitly assume that whenever congestion is detected, it is network wide and long lasting. To avoid local congestion recent works suggest increasing the outgoing information rate either by increasing the resources around the sensor nodes [12] or by using multiple paths [13]. There are few efforts to provide reliability for both congested and non-congested scenarios [14]–[17]. In [14], [15], a centralized approach is utilized for reliability and congestion control, which is not efficient in WSNs. In [17], we developed the basics for tunable reliability along with congestion control to fulfill application requirements.

The lack of an integrated approach that provides tunable reliability combined with congestion control and managing the redundancy of information motivated us to design the GIT framework. Furthermore, recent surveys [18], [19] also emphasize the need for such an integrated solution.

III. GENERALIZED MODELS

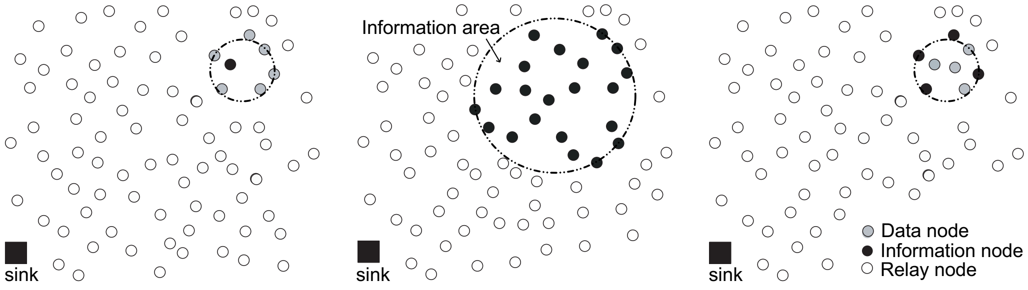
In order to keep the GIT framework generic, we consider generalized models of WSNs. We first present a simple yet comprehensive system model to capture generic WSN properties. Next, we detail the underlying abstract information and reliability models to consider generic applications.

A. System Model

We consider the conventional model of a WSN having $\mathbf{N}$ sensor nodes and a sink. Typically, each node is equipped with one or more sensing devices, short range transceivers with limited processing, memory and energy capabilities. We consider the sink to be adequate in power, memory and higher processing capabilities as compared to the sensor nodes. We assume that all nodes are static (including the sink) and are placed in a finite area. The sensor nodes communicate with each other over bi-directional multihop wireless links. For any two nodes X, Y we define their link quality $LQ = p_{(X,Y)} \cdot p_{(Y,X)}$, where $p_{(X,Y)}$ and $p_{(Y,X)}$ indicates the probability that a packet sent by node X is received correctly by node Y and vice versa. X, Y are defined to be neighbors, if $LQ \neq 0$. All sensor nodes know their hop distance h_X from the sink and their 1-hop neighbors. Based on hop distances, the neighbors of a node can be classified as upstream neighbors, downstream neighbors and equal neighbors. We denote the set $N_u = \{Y : \{X,Y\} \in \mathbf{N} \wedge h_Y = h_X + 1\}$ as the upstream neighbors, the set $N_d = \{Y : \{X,Y\} \in \mathbf{N} \wedge h_Y = h_X - 1\}$ as the downstream neighbors and the set $N_e = \{Y : \{X,Y\} \in \mathbf{N} \wedge h_Y = h_X\}$ as the equal neighbors of node X. We assume an underlying routing protocol, which provides a path for all nodes towards the sink. The information transport perturbations are categorized with respect to *message loss* due communication failures and buffer overflows.

B. Information Model

Due to the increasing number of applications running on WSNs, we propose an abstraction for applications corresponding to the information they expect from a WSN. Such an abstraction supports application independence and transparency for the information transport. We refer to an *information area* as the geographic area where raw data is generated and the information of interest is extracted through in-network processing as depicted in Fig. 1. An established example of information area is an event area. The sensor nodes in an information area are classified as *data* and *information nodes*. The data nodes generate raw data while information nodes generate information entities. We refer to an *information entity* as the processed raw data which is required by the application. We assume that information entity is realized through a single message. The data nodes do not send data directly to the sink. On the other hand, information entity generated by an information node is required to be transported to the sink via relay nodes. The information entity can be generated on a single node (cluster head, fusion center, aggregation node etc.) or in a distributed manner by multiple nodes (spatially correlated). The information entities can also be grouped for higher semantics, e.g., the event perimeter. Accordingly, we classify any information required by the applications into two broader classes: *Atomic information* and *Composite information*. Atomic information is composed of a single information entity (Fig. 1(a)), which cannot be sub-divided and is complete in nature, e.g., aggregated value of many sensor nodes. We further classify the atomic information as redundant atomic information. In redundant atomic information the information entities are generated densely, i.e., all sensor nodes within the information area generate similar information (Fig. 1(b)). For example, in event detection applications more than one sensor node detects the event and transports this information towards the sink [9], [14]. In this case information is used in a non-standard way and all information nodes send information entities towards the sink. Alternately, the composite information is composed of a set of unique information entities from different information nodes (Fig. 1(c)). For example, the tracking or event perimeter applications focus on boundaries of the event to better understand its progression. The boundaries may have different shapes and may be reconstructed to track the event if application have sufficient information entities.



(a) Atomic information (b) Redundant atomic information (c) Composite information

Fig. 1. Information classification

C. Reliability Model

The application reliability requirements are typically statistical in nature. For example, monitoring applications do not require reliability of a single information entity but they need a certain number of entities to be available at the sink. Similarly, event detection applications require that a certain number of events to be reported over the lifetime of WSNs. This entails providing x% (probabilistically-guaranteed) reliable information transport instead of best effort or transporting all information entities. Accordingly, the application level end-to-end desired reliability for information transport (R_d ($0 < R_d \leq 1$)) is described by the probability of successfully transporting the information entity to the sink. Consequently, we define the atomic information transport reliability as the degree of tolerating the information loss over time. For composite information, the transport reliability is defined as the degree of tolerating loss of information entities by the application without loosing the semantic of the composite information. We assume that the nodes know R_d with which information is to be transported. Many existing techniques [20], [21] can be employed to distribute R_d to sensor nodes.

IV. GIT: THE PROPOSED FRAMEWORK

We now present GIT framework that dynamically and autonomously adapts to maintain the desired information transport reliability. First, we provide a conceptual overview of the GIT framework. Next, we show how it adaptively integrates and controls spatio-temporal redundancy techniques in order to provide tunable reliability.

A. Overview - The Core of our Approach

In order to appropriately tune the reliability of information transport it is necessary to (1) manage different types of information and their level of redundancy, (2) cope with perturbations, and (3) select and tune the suitable reliability assurance techniques. To fulfill these objectives GIT proposes four modules that reside on each sensor node as shown in Fig. 2. The modular approach allows for the easy integration of different mechanisms with GIT modules. Once the information of interest is generated inside the network the information module (IM) identifies and removes redundancies from the information before transporting it to the sink. We develop efficient and distributed techniques for information management that are utilized by IM. In WSNs (given the power depletions of the sensor nodes and the lossy nature of their communications) perturbations are the norm rather than the exception which hinders in the delivery of information to the applications. To handle perturbations GIT provides a reliability module (RM), which incorporates multiple techniques to locally detect information loss. The wireless medium and limited memory capabilities of the sensor nodes are major causes for information loss. To appropriately detect these losses, GIT integrates message loss detection module (MLDM) and congestion control module (CCM). In WSNs information often has to travel multiple hops to reach the sink. To maintain application reliability along the entire path, RM also includes a reliability allocation module (RAM) which allocates reliability across the hops. To recover information loss, GIT provides a tuning and adaptation module (TAM). TAM exploits existing approaches for utilizing in-network spatio-temporal redundancies to overcome the information loss. The GIT framework uses network monitoring module (NMM) to monitor local conditions around the sensor nodes and provide network health indicators to RM for maintaining desired reliability. The GIT modules are individually detailed in the subsequent sections.

B. IM - Managing the Information

IM is responsible for managing the information in the network and accordingly selects the subset of information nodes (amongst all possible information sources) in a distributed way for further transport. IM selects information nodes based on application requirements and type of information. The sensor nodes locally identify the type of generated information based on criteria specified by the application, e.g., a clustering algorithm selects the cluster head for data collection to generate the atomic information. Such criteria specification is beyond the scope of this paper and we assume that the type of information is specified by the application over design or deployment time. We now describe how IM selects the information nodes for different types of information.

1) Node Selection for Atomic Information: If an application specifies non redundant atomic information, the sensor node becomes an information node after filtering or pre-processing of the raw data. The information entity is then handed over to RM for transportation towards the sink.

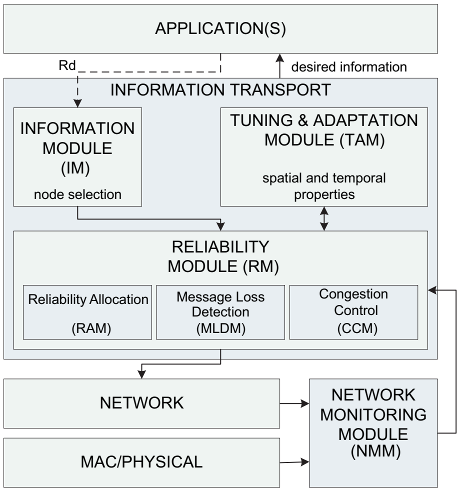


Fig. 2. The GIT framework

If the application stipulates redundant atomic information to be transported, it is not efficient to let all the information nodes deliver the same information. We distinguish between the redundant raw data required to generate atomic information and redundant information. In this case a single node is sufficient to transport the redundant atomic information. In order to select a single information node clustering algorithms can be utilized. The major drawback of clustering is the overhead required for creating and maintaining the clusters [22], since the information generation and the corresponding information area is dynamic in nature. Furthermore, the cluster head itself may possibly be selected farthest from the sink, which requires more transmissions for information transport. We propose an efficient solution for selecting a single information node to report the redundant atomic information, thus eliminating the redundancy. The main objective is to minimize the number of transmissions required to transport the information to the sink. Therefore, the basic idea is to select an information node within the information area which is closest to the sink. In order to efficiently transport the redundant atomic information, IM identifies the information periphery, i.e., the perimeter that spatially covers the information area, in a distributed manner. Once the periphery is available, IM categorizes the information frontier, defined as the periphery nodes closest to the sink in terms of number of hops. The sensor nodes on the frontier select a single information node for the information transport. Once the redundant atomic information is generated, the information nodes broadcast (with some delay to avoid collisions) an information verification message to its 1-hop neighbors in order to verify the generation of information. The verification message also serves the purpose of identifying the neighbors. Based on neighborhood knowledge and using approaches like Isolines [23] information nodes locally discover whether they lie inside or on the periphery of information area. If a node lies inside the information periphery it will not select itself as an information node since it is farther from the sink as compared to nodes on the frontier. As nodes belonging to the information frontier are closest to the sink, the remaining periphery nodes suppress their information transport. In one message exchange the majority of information nodes are suppressed from being selected for information transport. Next, we have to allocate a node on the information frontier that is closest to the sink for the information transport. To ensure that the closest node is selected, each node on the frontier sends a traversal message along the frontier to suppress other nodes from sending redundant information. Each frontier node starts its traversal timer according to its hop distance to the sink. The shorter the distance to the sink, the earlier a frontier node starts its traversal. If a frontier node receives traversal message before it starts its own traversal, it will discard its traversal and forward the received traversal message to other frontier nodes. All frontier nodes suppress themselves after receiving a traversal message from the node closest to the sink on the frontier. The last node on the frontier replies back to the closest node for its selection as information node.

2) Node Selection for Composite Information: For composite information, the main challenge is to select the information nodes according to the desired application requirements. For sensor node selection, solutions based on game theory [24] are available, but such schemes select the nodes after a certain number of iterations. These approaches cannot be utilized by GIT as the information may last for only a short time inside a WSN. Other solutions such as [25] are very application specific. The GIT framework implements a simple heuristic to randomly select k information nodes in order to meet the desired application reliability. The information nodes can autonomously decide whether to be selected or not according to their probability of selection, i.e., R_d. The property of uniform random numbers assures that statistically $(R_d \times 100)\%$ information nodes are selected for the composite information transport.

We now describe the efficient integration of our developed mechanisms [17] such as hybrid acknowledgment, adaptive retransmissions and proactive congestion control with GIT framework. Furthermore, we present how the developed mechanisms enhance the interactions among the modules of GIT in order to ensure desired application reliability.

C. RM - Detecting Unreliability

To achieve and maintain tunable reliability, RM manages reliability allocation along the path and keeps track of information loss. If RM examines that information entity can achieve the desired reliability it passes the information to the network layer for transporting it to the next hop along the path. If the desired reliability is not attainable, RM notifies TAM for tuning and adaptation of temporal and/or spatial properties in order to maintain the desired reliability. We now discuss how the sub-modules of RM realize the task of tunable reliability.

1) RAM - Allocating the Reliability: The multihop communication is commonly used in WSNs. Due to the nature of wireless medium and perturbations inside the network, the reliability across the hops varies. The RAM is responsible for allocation of application reliability across hops to reach the destination. For optimal allocation the information node must have a global knowledge of all intermediate hop reliabilities, which is hard to achieve in WSNs due to high communication overhead. The RAM makes use of a simple heuristic to allocate the reliability across each hop along the path according to hop distance to the sink. For known R_d and number of hops from the sink, an information node locally calculates the desired reliability requirement (R_{h_d}) at each hop as: $R_{h_d} = (R_d)^{1/h_{inf}}$ where h_{inf} represents number of hops from the information node to the sink. R_{h_d} considers a uniform reliability requirement across all hops. Each information node calculates R_{h_d} and the relay nodes along the path ensure the allocated reliability.

2) MLDM - Detecting Link Losses: The main objective of MLDM is to efficiently detect the information loss due to communication perturbations. Several message loss detection techniques can be adopted to provide reliability such as acknowledgment (ACK), negative ACK, implicit ACK and timers. GIT utilizes hybrid ACK, a unique combination of implicit ACK and explicit ACK along with probabilistic suppression of the information. When a node sends a message, it waits for an implicit ACK to ensure the message delivery to the receiver node. After a predetermined time, if implicit ACK is not received, the node retransmits the message. GIT also employs local retransmission timers to efficiently maximize the benefit of hybrid ACK scheme by observing the neighbor node's buffer status. As the information entity is comprised of a single message, the choice of hybrid ACK is beneficial. However, MLDM can easily integrate other strategies, e.g., negative ACK scheme can be utilized if information consists of more than one information entities [2].

3) CCM - Congestion Control: Another common perturbation in WSNs is congestion due to buffer overflow, which leads to the information loss. The commonly used congestion detection schemes in the WSN literature rely on monitoring (i) channel utilization, (ii) buffer utilization, and (iii) average message queuing time. These schemes are reactive in nature and lead to message loss until they stabilize. CCM comprises of a pro-active congestion detection mechanism by monitoring information flow across each sensor node. CCM identifies three types of congestions inside WSNs, i.e., link level congestion, short lived congestion and long lived congestion. CCM triggers link level congestion avoidance using application aware scheduling of information transport. As soon as CCM detects a higher incoming information rate than the outgoing information rate, it suspects a short lived congestion. Consequently, to alleviate short lived congestion, CCM sends a request to TAM for adapting appropriate parameters. If CCM observes that the buffers of its neighbor nodes are also full (by coordinating with NMM), it concludes that long lived congestion is prevailing and indicates TAM to adjust the information flow accordingly.

D. TAM - Tuning and Adaptation of Parameters

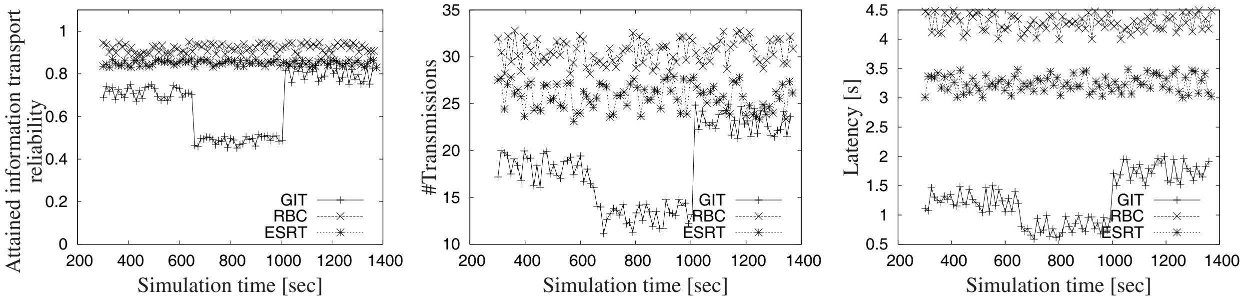
Once the information loss is detected by RM, TAM has the responsibility to recover the desired information by tuning and adapting the reliability parameters. The reliability parameters exploit spatial and temporal properties of the network. We categorize spatial parameters as number of sources ($\#src$), number of paths ($\#path$) and number of cache points ($\#CP$)). On the other hand, temporal parameters include maximum number of retransmissions ($\#ret$) and information rate. $\#CP$ is related to the storage of messages along the path such that in case of message loss the recovery can be initiated. We assume that the information is cached at each hop along the path until an implicit ACK is received. GIT efficiently controls the $\#src$ for atomic and composite information and avoids unnecessary transmissions using IM. When message loss is indicated by MLDM, TAM adapts $\#ret$ on-the-fly along the path if there is no congestion inside the network. When the congestion is observed by CCM, TAM adapts $\#paths$ and reduces the information rate.

1) Adapting Temporal Parameters: To ensure reliability across a hop (X, Y) and to tolerate perturbations more than one transmissions are required. Let $r_{(X,Y)}$ be the maximum number of transmissions required than, $r_{(X,Y)} = \frac{\log(1-R_{h_d})}{\log(1-R_{hop})}$, where R_{hop} is a hop reliability across X and Y. Each node along the path dynamically adapts r according to its local hop reliability and application requirements. If a sensor node receives an implicit ACK it stops retransmitting.

2) Adapting Spatial Parameters: TAM distinguishes between short lived and long lived congestion and accordingly reacts to the situation. If CCM indicates short lived congestion, TAM utilizes split-path and sends information to neighbor nodes. If the neighbor node is from the set N_d, it will not change the reliability allocation of RAM as the path length is not changed. If the selected neighbor is from N_u or N_e, the path length is changed and TAM indicates RAM to recalculate the reliability allocation. When CCM indicates a long lived congestion, TAM integrates multiplicative decrease policy to reduce the information rate. Once the congestion is over TAM uses additive increase policy to increase the information rate.

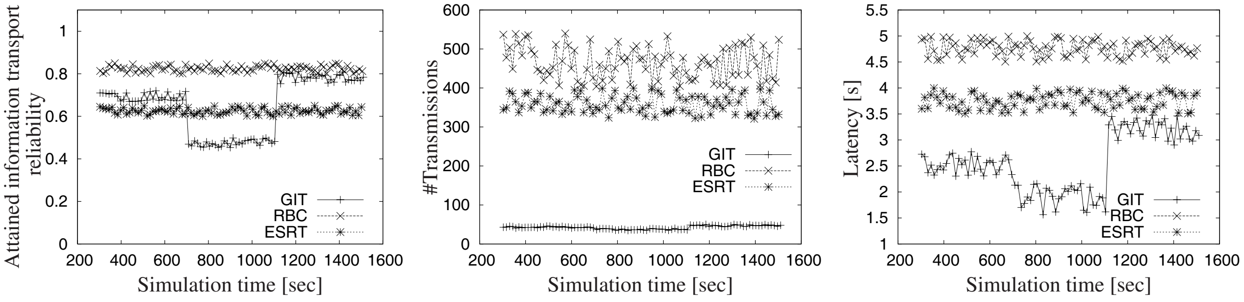
E. NMM - Monitoring the Network

NMM is responsible for observing the local network conditions. There is a broad research in link quality estimators [26]. We emphasize two indicators which are readily available and provide good link estimations, i.e., link quality indicator (LQI) and bit error probabilities (BEP). LQI is on-chip indicator for link quality and is available on current mote platforms such as Micaz and TelosB. LQI is shown to be an acceptable indicator for link quality estimation [3], [27]. By contrast, BEP is available in simulation environments and also shown to be effective indicator [28]. BEP provides local conditions around the node and represents the elementary indicator for other aggregated indicators such as packet error rate [29]. BEP reflects a wide range of cases, i.e., network congestion, collisions and contention, since they tend to corrupt the message which is similar to BEP. The modular approach of GIT allows other link quality estimators such as received signal strength indicator or expected number of transmissions to be utilized by NMM.



(a) Reliability (b) Efficiency (c) Timeliness

Fig. 3. Tunability of atomic information



(a) Reliability (b) Efficiency (c) Timeliness

Fig. 4. Tunability of redundant atomic information

V. PERFORMANCE EVALUATION

In order to evaluate the GIT framework we first describe the evaluation scenarios, simulation settings and the performance metrics. Next, we present our simulation results.

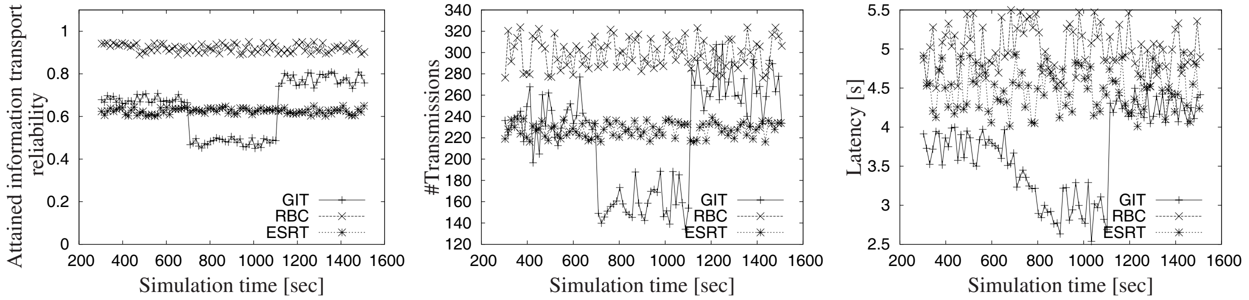
A. Evaluation Methodology

To observe the tunability of GIT and how it manages different types of information, we simulated the following scenario. We considered atomic (AI), redundant atomic (RAI) and composite information (CI) for transportation. The application requires $R_d = 0.7$ in the first phase. Each type of information is generated every 10 seconds and transported towards the sink. After 5 mins the application tune its requirement, i.e., $R_d = 0.5$. To simulate the evolving application requirement scenario, again after 5 mins the application requirements evolve to $R_d = 0.8$. The above scenario is representative for various situations inside the network. First, it represents evolving application requirements. Second, when many nodes send information it leads to evolving network conditions, i.e., increased collisions and contention. In another simulation setting we kept the application requirements unvarying ($R_d = 0.8$) and change the information rate to monitor GIT performance.

We evaluate our approach based on simulations using TinyOS 1.1.15 (TOSSIM [29]). In order to simulate the scenario, we deployed 225 sensor nodes in an area of 75×75 $unit^2$ with 15×15 grid topology. The distance between two neighboring nodes is 5 units. We have used the empirical radio model [30] provided by TOSSIM. In this model, a sensor node sends and receives messages using an error distribution based on empirical data, where bit errors depend on distances from sender to receiver and background noise. The sink is located at one corner of the area. For atomic information one node is randomly chosen from the opposite corner of the sink. For redundant atomic information, 20 nodes are selected randomly close to each other within the radius of 15 units. For composite information, nodes on the periphery of redundant atomic information choose themselves for information transport. Both redundant atomic information and composite information is generated in the opposite corner of sink. The size of the message is 29 bytes and each sensor node has buffer size of 36 messages.

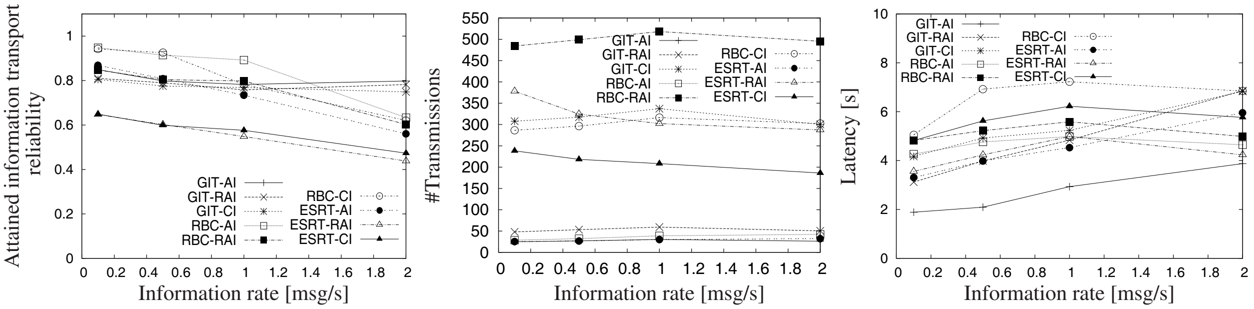
We used representative protocols from the existing literature as discussed in Section II and compared them with the GIT framework. We selected reliable bursty convergecast (RBC) [9], which provides only reliability and event to sink reliable transport (ESRT) [14], which provides both reliability and congestion. The code of RBC is available for the mica2 mote platform, consequently we ported the RBC code to run under the TOSSIM environment. The code for ESRT is not available thus, we implement ESRT in TOSSIM. As for routing the messages, RBC uses by default logical grid routing (LGR) [31] protocol, we have chosen LGR for routing the messages for fair comparison.

The performance of GIT is measured in terms of reliability, timeliness and efficiency. The information transport reliability is the ratio between the amount of information entities received by the sink and the total amount of the information entities



(a) Reliability (b) Efficiency (c) Timeliness

Fig. 5. Tunability of composite information



(a) Reliability (b) Efficiency (c) Timeliness

Fig. 6. Impact of information rate (R_d = 0.8)

generated over time. The timeliness is defined as the time elapsed from the generation of the first information entity to the delivery of the first information entity at the sink. We define the efficiency as the total number of transmissions including the retransmissions.

B. Simulation Results

First, we consider the varying application requirements scenario. Fig. 3 depicts the tunability of atomic information transport. The reliability attained by the GIT and other protocols is shown in Fig. 3(a). As RBC and ESRT do not provide tunability aspects, their attained reliability is steady over time. GIT supports tunable behavior and copes with the evolving application requirements. We observe that GIT always fulfills the application requirements ($\pm 3\%$). The reliability attained by the ESRT is lower than RBC since it does not utilize any message loss recovery mechanism. RBC reliability remains higher which is in contradiction to application requirements and waste critical resources by having more transmissions. Fig. 3(b) shows that GIT is efficient and uses fewer transmissions and adapts them according to application requirements owing to adaptive transmissions mechanism by TAM. ESRT shows less number of transmissions compared to RBC because ESRT do not employ retransmissions to recover information loss, resulting in low reliability as well. Fig. 3(c) shows the timeliness tradeoff of different protocols. GIT latency remains lower than RBC and ESRT due to efficient transport mechanisms which lower the probability of retransmissions thus, reducing the latency. For ESRT and RBC as they always send information without caring of application requirements, information is sent across the network every time resulting in contention leading to high latency.

Fig. 4 shows the adaptation to application requirements for redundant atomic information. GIT adapts according to the application requirements and provides tunable reliability (Fig. 4(a)). The reliability attained by ESRT is lower than the GIT and RBC because of message loss due to sudden bursts of information. Similar is in the case with RBC but it survives better than ESRT due to its fixed number of retransmissions. We observe a tremendous benefit of GIT in terms of efficiency (Fig. 4(b)). The number of transmissions is reduced 4-5 times. This is achieved by GIT due to information awareness and the mechanisms utilized to reduce the redundant information. As ESRT and RBC cannot distinguish between the transported information, they result in higher number of transmissions. Fig. 4(c) shows the latency of GIT remains lower than ESRT and RBC because when the burst of information is generated by 20 nodes, RBC and ESRT have to cope with contention and retransmission. Whereas, GIT utilize efficient techniques as discussed in Section IV-B, which reduces number of nodes to contend, thus, resulting in low timeliness.

Fig. 5 depicts the tunability of composite information. For composite information GIT also provides tunable reliability (Fig. 5(a)) compared to other protocols. In Fig. 5(b) we observe that at higher application requirement (i.e., 0.7 and 0.8) number of transmissions of GIT is comparable to ESRT and RBC. The number of transmissions is directly proportional to the higher reliability requirement of the application. Furthermore, as multiple information flows start to flow towards the sink, congestion results. Here GIT adapts to split paths resulting in slightly higher number of transmissions. This is also because the sensor nodes selection is directly related to the application requirements. Therefore, at higher reliability requirements more nodes are selected resulting in short lived congestion. The timeliness of GIT also increases with higher application requirements (Fig. 5(c)) due to split paths where information follows slightly longer paths.

We now consider a second scenario, where we assume $R_d = 0.8$ and a varying information rate. Fig. 6(a) depicts the GIT tunability to fulfill desired application requirements for varying information rate of different types of information. Fig. 6(a) depicts that GIT always provide reliability close to the application requirements. With high information rate (2 msgs/sec) RBC and ESRT are not able to cope due to collisions and congestion. While at low information rate RBC and ESRT provide high reliability. With increasing information rate congestion start to build and GIT efficiently handles the situation as shown in Fig. 6(b). We observe that for composite information at 1 msg/sec GIT utilizes more transmissions owing to split path mechanism used to avoid short lived congestion. When the information rate is further increased, the number of transmissions is less due to decreased information rate as done by GIT to handle long lived congestion. For other types of information GIT also efficiently adapts transmissions. It is noteworthy that the number of transmissions for RBC and ESRT decreases with the increasing information rate. This is due to the fact that the protocols start dropping the information due to congestion. Similar effect can be viewed for timeliness in Fig. 6(c), where for RBC and ESRT latency decreases which is directly proportional to reliability and dropped information. GIT follows congestion control mechanisms and local timer management which results in slightly increased latency as information rate increases.

VI. CONCLUSIONS AND FUTURE WORK

We propose a framework for generic information transport (GIT) in wireless sensor networks. The GIT approach provides the desired application reliability despite evolving application requirements and dynamic network conditions. GIT reduces application dependency by utilizing generic information abstraction and its ability to tune itself according to application requirements. In particular, we designed techniques to restrict redundant information as close as possible to the information area. Our solution efficiently reduces the information redundancy and for certain types of information it shows 4-5 times less number of transmissions compared to state of the art solutions. GIT copes with a wide range of network conditions ranging from basic wireless links to network wide congestion by adapting between basic temporal and spatial reliability mechanisms. The simulation results confirm the capability of GIT to tune according to the application requirements and maintain the desired reliability. Due to modular architecture of GIT it is straightforward to include other functionalities such as duty cycling. The duty cycling in WSNs can be viewed by GIT as node perturbation, where underlying routing will provide alternate routes via active sensor nodes. In future, we would like to further optimize and study the behavior of GIT for duty cycling. Furthermore, we are also planning to implement GIT on a testbed to verify its operations in real environment.

REFERENCES

[1] F. K. Shaikh et al. A comparative study for data transport protocols for WSN. In *Proc. of WOWMOM*, pages 1 – 9, 2008.

[2] F. Stann and J. Heidemann. RMST: reliable data transport in sensor networks. In *Proc. of SNPA*, pages 102–112, 2003.

[3] T. Le et al. ERTP: energy-efficient and reliable transport protocol for data streaming in wsn. *Comp. Comm.*, 7:1154–1171, 2009.

[4] B. Marchi et al. DTSN: distributed transport for sensor networks. In *Proc. of ISCC*, pages 165–172, 2007.

[5] M. C. Vuran et al. Spatio-temporal correlation: Theory and applications for wireless sensor networks. *J. Comp. Net*, 45(3):245–259, 2004.

[6] F. K. Shaikh et al. AReIT: adaptable reliable information transport for service availability in wsn. In *Proc. of ICWN*, pages 75–81, 2009.

[7] E. Felemban et al. MMSPEED: multipath multi-speed protocol for qos guarantee of reliability and timeliness in wireless sensor networks. *IEEE Trans. on Mobile Computing*, 5(6):738–754, 2006.

[8] J.-Y. Teo et al. Interference-minimized multipath routing with congestion control in wireless sensor network for high-rate streaming. *IEEE Trans. on Mob. Comp.*, 7(9):1124–1137, 2008.

[9] H. Zhang et al. Reliable bursty convergecast in wireless sensor networks. In *Proc. of MobiHoc*, pages 266–276, 2005.

[10] C. Wang et al. Upstream congestion control in wireless sensor networks through cross-layer optimization. *IEEE Journal on Selected Areas in Comm.*, 25(4):786–798, 2007.

[11] S. Chen and N. Yang. Congestion avoidance based on lightweight buffer management in sensor networks. *IEEE Trans. Parallel Distrib. Syst.*, 17(9):934–946, 2006.

[12] T. He et al. Alleviating congestion using traffic-aware dynamic routing in wireless sensor networks. In *Proc. of SECON*, pages 233–241, 2008.

[13] L. Popa et al. Reducing congestion effects in wireless networks by multipath routing. In *Proc. of ICNP*, pages 96–105, 2006.

[14] Y. Sankarasubramaniam et al. ESRT: event-to-sink reliable transport in wireless sensor networks. In *Proc. of MobiHoc*, pages 177–188, 2003.

[15] J. Paek and R. Govindan. RCRT: rate-controlled reliable transport for wireless sensor networks. In *Proc. of SenSys*, pages 305–319, 2007.

[16] R. Kumar et al. Mitigating performance degradation in congested sensor networks. *IEEE Trans. on Mobile Computing*, 7(6):682–697, 2008.

[17] F. K. Shaikh et al. TRCCIT: Tunable reliability with congestion control for information transport in wsns. In *Proc. of WICON*, 2010.

[18] C. Wang et al. A survey of transport protocols for wireless sensor networks. *IEEE Network Magazine*, 20(3):34–40, 2006.

[19] M. A. Rahman et al. Wireless sensor network transport layer: State of the art. In *Sensors: Adv. In Model., Design Issues, Fabr. And Pract. App.*, volume 21, pages 221–245, 2008.

[20] C. M. Liang et al. Typhoon: A reliable data dissemination protocol for wireless sensor networks. In *Proc. of EWSN*, pages 268–285, 2008.

[21] S. Park et al. A scalable approach for reliable downstream data delivery in wireless sensor networks. In *Proc. of MobiHoc*, pages 78–89, 2004.

[22] A. Abbasi and M. Younis. A survey on clustering algorithms for wireless sensor networks. *Comput. Commun.*, 30(14-15):2826–2841, 2007.

[23] I. Solis and K. Obraczka. Isolines: energy-efficient mapping in sensor networks. In *Proc. of ISCC*, pages 379–385, 2005.

[24] A. Willig and H. Karl. Data transport reliability in wireless sensor networks – a survey of issues and solutions. *PIK*, 28:86–92, 2005.

[25] W. Choi and S. K. Das. CROSS: A probabilistic constrained random sensor selection scheme in wireless sensor networks. *Perf. Eval.*, 2009.

[26] N. Baccour et al. A comparative simulation study of link quality estimators in wireless sensor networks. In *Proc. of MASCOTS*, 2009.

[27] J. Polastre et al. Telos: enabling ultra-low power wireless research. In *Proc. of IPSN*, pages 364–369, 2005.

[28] W. Dong et al. Performance of bulk data dissemination in wireless sensor networks. In *Proc. of DCOSS*, pages 356–369, 2009.

[29] P. Levis et al. TOSSIM: accurate and scalable simulation of entire tinyos applications. In *Proc. of SenSys*, pages 126–137, 2003.

[30] A. Woo et al. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proc. of SenSys*, pages 14–27, 2003.

[31] Y. Choi et al. Stabilization of grid routing in sensor networks. *J. of Aerospace Computing, Inf. and Comm.*, 3:214–233, 2006.