A Delay-Aware Reliable Event Reporting Framework for Wireless Sensor-Actuator Networks

PhD Term 4 paper

Department of Computer Science and Engineering

The Chinese University of Hong Kong

written by Edith C.H. Ngai

supervised by Prof. Michael R. Lyu

Spring 2006

Abstract

Wireless sensor-actuator networks, or WSANs, greatly enhance the existing wireless sensor network architecture by introducing powerful and even mobile actuators. The actuators work with the sensor nodes, but can perform much richer application-specific actions. To act responsively and accurately, an efficient and reliable reporting scheme is crucial for the sensors to inform the actuators about the environmental events. Unfortunately, the low-power multi-hop communications in a WSAN are inherently unreliable; the frequent sensor failures and the excessive delays due to congestion or in-network data aggregation further aggravate the problem.

In this paper, we propose a general reliability-centric framework for event reporting in WSANs. We argue that the reliability in such a real-time system depends not only on the accuracy, but also the importance and freshness of the reported data. Our design follows this argument and seamlessly integrates three key modules that process the event data, namely, an efficient and fault-tolerant event data aggregation algorithm, a delay-aware data transmission protocol, and an adaptive actuator allocation algorithm for unevenly distributed events. Our transmission protocol also adopts smart priority scheduling that differentiates the event data of non-uniform importance. We evaluate our framework through extensive simulations, and the results demonstrate that it achieves desirable reliability with minimized delay.

1 Background

1.1 Wireless Sensor Networks

Wireless sensor network (WSN) is a rapidly emerging area as an important research area. The variety and number of applications are growing on wireless sensor networks. They range from general engineering, environment science, health service, military, etc. Wireless sensor network requires large number of sensor collection data from the environments. They are tiny devices with limited energy, memory, transmission range, and computation power. WSN is self-organized with collaboration among the nodes. Base station is present in the network, which receives the aggregated data from the sensors. It is usually a powerful computer with more computational power, energy, memory, and connected to the Internet.

1.1.1 Characteristics

The following discuss the characteristics of wireless sensor networks:

A wireless sensor network (WSN) is self-organizing. It consists of a large number of sensor nodes. They are deployed over an area and form a wireless network. The position of sensor nodes need not be engineered or pre-determined. This allows random deployment in inaccessible terrains or disaster relief operations. On the other hand, this also means that sensor network protocols and algorithms must possess self-organizing capabilities.

A unique feature of sensor networks is the cooperative effort of sensor nodes. Sensor nodes are fitted with an on-board processor. Instead of sending the raw data to the nodes responsible for the fusion, sensor nodes use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data.

Sensor nodes exchange messages through short-range communication and multihop routing. Since large number of sensor nodes are densely deployed and they are having short communication range. Hence, multihop communication in sensor networks is expected to consume less

power than the traditional single hop communication. Furthermore, the transmission power levels can be kept low, which is highly desired in covert operations. Multihop communication can also effectively overcome some of the signal propagation effects experienced in long-distance wireless communication [1].

There are limitations on energy and computation power. The sensor nodes are autonomous devices with limited battery, computational power, and memory. Also, dynamic environmental conditions require the system to adapt over time to changing connectivity and system stimuli.

1.2 Wireless Sensor-Actuator Networks

Wireless sensor and actor networks (WSANs) can be viewed as an extension to wireless sensor networks. It refers to a group of sensors and actors linked by wireless medium to perform distributed sensing and actuation tasks. In a WSAN, sensors gather information about the physical world, while actors take decisions and then perform appropriate actions upon the environment, which allows a user to effectively sense and act at a distance. The differences between wireless sensor networks (WSNs) and wireless sensor and actor networks (WSANs) are mainly due to the existence of actuators [2]:

First of all, sensors and actuators are two kinds of devices. Sensor nodes are small, cheap devices with limited sensing, computation and wireless communication capabilities (see Figure 2 and 2), while actors are resource-rich nodes equipped with better processing capabilities, stronger transmission powers and longer battery life (see Figure 3 and 4. Since acting phenomenon is more complicated and energy consuming activity than sensing phenomenon, it is normally performed by the actuators. Also, the number of actors is much usually less than that of sensors. The number of sensor nodes deployed in studying a phenomenon may be in the order of hundreds or thousands. In the contrary, the number of actuators may be in the order of tens.

Many applications in WSANs requires a rapid respond from actuators to sensor input. For example, in the case of a fire, sensors relay the exact origin and intensity of the fire to water sprin-

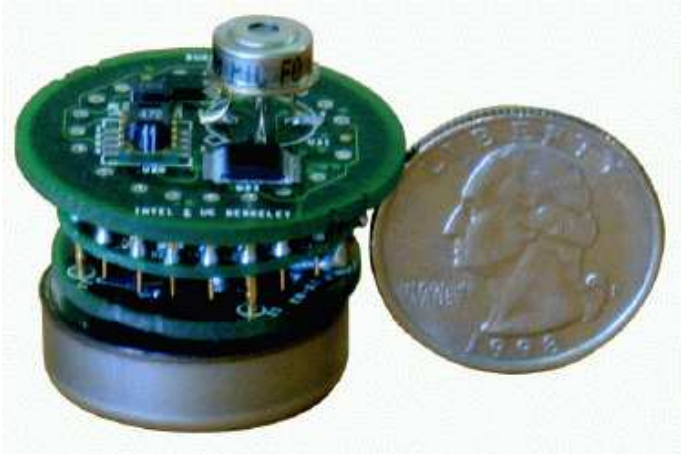

Figure 1: An Intel-Berkeley ”mote”

kler actors so that the fire can easily be extinguished before it becomes uncontrollable. Similarly, motion and light sensors in a room can detect the presence of people and then command the appropriate actors to execute actions based on the pre-specified user preferences [3]. Moreover, so as to provide right actions, sensor data must still be valid at the time of acting. Therefore, the issue of real-time communication is very important in WSANs as actions are performed on the environment after the sensing occurs. However, such a dense deployment is not necessary for actor nodes due to the different coverage requirements and physical interaction methods of acting task. In order to provide effective sensing and acting, a distributed local coordination mechanism is necessary among sensors and actors.

1.2.1 Operations

After sensors in the WSAN detect an event, they either transmit their readings to the resource-rich actuators which can process all incoming data and initiate appropriate actions, or route data back to the sink which issues action commands to actors. The former case as called Automated Architecture due to the nonexistence of central controller (human interaction). The latter one is called as Semi-Automated Architecture since the sink (central controller) collects data and coordinates

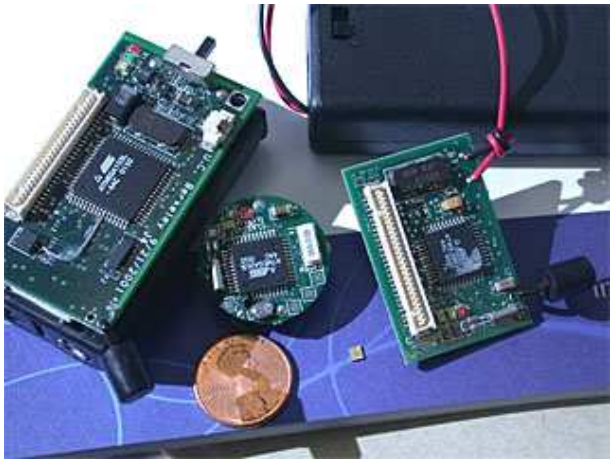

Figure 2: A sensor group

Figure 3: Robotic Mule

the acting process. These two architectures are shown in Figure 5. Depending on the types of applications, one of these architectures may be used. The advantage of Automated Architecture is that the information sensed is conveyed quickly from sensors to actors, since they are close to each other. Moreover, since event information is only transmitted locally through sensor nodes, only sensors around the event area are involved in the communication process which results in energy and bandwidth savings in WSANs [2].

Figure 4: Intelligent tele-robots

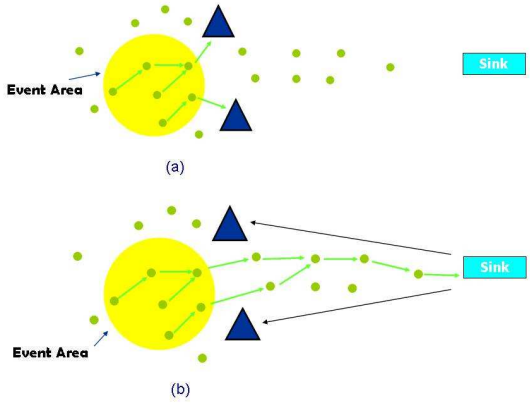

Figure 5: (a)Automated and (b)Semi-Automated Architecture.

1.2.2 Research Challenges in Routing

There are both multiple sensors and multiple actors which can communicate with each other in WSAN . When sensors detect an event, there is no specific actor to which a message will be sent. It means that anycast routing is applicable in WSAN. Anycast allows a node to send a message to at least one, and preferably only one, of the members in a group. This uncertainty occurring due to the existence of multiple actors causes challenges in terms of the routing issues. Moreover,

another challenging problem which routing protocols should deal with in WSANs is to provide reliable event transmission as well as end-to-end real-time guarantees.

Since there are multiple actors, selecting an actor node is another problem. The sensor, which detects an event, should select an actor node and establish a path toward it. The selection can be made on the basis of local information such as its own available energy, the available energy of the neighboring sensors, or on the basis of metrics related to the distance from the actors.

Also, multi-hop routing is common in sensor networks. There will be multiple possible paths between the source and the selected actor node. Thus, there is a need to develop a routing protocol which provides path selection, data delivery and path maintenance. Moreover, the routing protocol must be self-organizing and adaptive, such that it can be responsive to actors joining and leaving dynamically. It may avoid unpredictable congestion and holes in the network.

Furthermore, WSANs have timing constraints in the form of end-to-end deadlines. Therefore, the routing protocol is necessary to support real-time communication by considering that data in a system may have different deadlines due to different validity intervals. Therefore, routing protocol should also consider the issue of prioritization and provide data with small delay bound to arrive at the actor on time.

Apart from the sensor-actuator communication, there are actuator-actuator communication. If the actuators are equipped with strong transmission facilities, they can exchange messages with a different channel that provides long range communications. The use of flooding algorithms usually may not be efficient since flooding causes all resource-constrained sensors to receive multiple copies of the same packet which are irrelevant to them. Actually, routing protocols developed for ad-hoc networks such as DSR, AODV, OLSR can be used for actor-actor communication as long as communication overhead occurring at sensor nodes due to actor-actor communication is low. In addition, in order to provide timely actions and adaptability to different applications, ad-hoc routing protocols should be improved to get unified routing protocol which considers real-time restrictions and supports all types of decision processes as well as all types of task types [4].

2 Introduction

The advances of hardware and software technologies for embedded systems have turned micro sensors with radio transceivers into reality [3][5][6][7]. Wireless sensor networks (WSNs), constructed by a group of sensors, have been suggested for numerous novel applications, such as monitoring for harsh environments and protecting the national borders. Recently, actuator nodes, which have much stronger computation and communication power than uni-purpose micro-sensors, have also been introduced [4]. An actuator can perform diverse tasks, such as processing the data reported from the sensors and accordingly interacting with the environment; a mobile actuator (*e.g.*, a robot) could even change its location periodically to serve the application better.

The sensors and actuators can form a powerful and yet cost-effective hybrid network, that is, the Wireless Sensor-Actuator Network (WSAN). While the functionalities of the actuators are application-specific, a well-designed communication module between the two types of nodes is crucial to a WSAN. In particular, given that the actuators need accurate event data from the sensors to perform corresponding actions, reliability is an important concern in the sensor-actuator communication. Unfortunately, the low-power multi-hop communications in a WSAN are inherently unreliable; the frequent sensor failures and the excessive delays due to congestion or in-network data aggregation further aggravate the problem.

In this paper, we focus on the design of a generic framework for reliable event reporting in WSANs. We argue that the reliability in this context is closely related to the delay, or the freshness of the events, and they should be jointly optimized. We also suggest that the non-uniform importance of the events can be explored in the optimization. We therefore present an delay- and importance-aware reliability index for the WSANs. Our framework seamlessly integrates three key modules to maximize the reliability index: 1) A multi-level data aggregation scheme, which is fault-tolerant with error-prone sensors; 2) A priority-based transmission protocol, which accounts for both the importance and delay requirements of the events; and 3) an actuator allocation algo-

rithm, which smartly distributes the actuators to match the demands from the sensors.

Our framework is fully distributed, and is generally applicable for diverse WSANs. Within this generic framework, we present optimized design for each of the modules, and also discuss their interactions. The performance of our framework is evaluated through extensive simulations. The results demonstrate that our framework can significantly enhance the reliability in event reporting; it also makes more effective use of the expensive actuators.

The remainder of this paper is organized as follows: Section II presents the related work. In Section III, we outline our network model and the problem to be solved. The reliable event reporting framework is presented in Section IV, together with detailed descriptions of each module. In Section V, we provide simulation results for our framework. Finally, we conclude the paper in Section VI.

3 Related Work

Wireless sensor networks (WSNs) have been extensively studied recently; see surveys in [4][5][6]. Efficient and reliable event reporting is also an important issue in WSANs. He et. al. [8] proposed a real-time communication protocol SPEED, which combines feedback control and non-deterministic QoS-aware geographic forwarding. Lu et. al. [9] described a packet scheduling policy, called Velocity Monotonic Scheduling, which inherently accounts for both time and distance constraints. Felemban et. al. [10] proposed Multi-path and Multi-Speed Routing Protocol (MMSPEED) for probabilistic QoS guarantee in WSNs. Multiple QoS levels are provided in the timeliness domain by using different delivery speeds, while various requirements are supported by probabilistic multipath forwarding in the reliability domain. For reliable transmission with error-prone sensors, Aidemark et al. [11] presented a framework for achieving node-level fault tolerance (NLFT). It describes a lightweight NLFT approach that masks transient faults locally by using time-redundant task scheduling in the nodes. There are also related works in the general embedded or delay-tolerant network settings. For example, Khanna et al. [12] suggested that the failure of any node in a path can be detected and recovered using backup routes. S. Jain et al. [13] considered the problem of routing in a delay tolerant network in the presence of path failures. It improves the probability of successful message delivery by applying a combination of erasure coding and data replication.

Our work is motivated by the above studies. The key difference is that we focus on the interactions between sensors and actuators, while not uniform network nodes. In this context, additional considerations are needed to address the heterogeneous characteristics and the unique interactions.

There have been studies exploring the heterogenous sensor networks, *e.g.*, [14][15][16], but they do not cope with the special features of actuators. For WSAN, Hu et. al. [17] proposed an anycast communication paradigm. It constructs an anycast tree rooted at each event source and updates the tree dynamically according to the join and leave of the sinks. E. Cayirci et. al. [18]

offered a power-aware many-to-many routing protocol. Actuators register the data types of interest by broadcasting a task registration message; The sensors then build their routing tables accordingly. Melodia et. al. [19] further presented a distributed coordination framework for WSANs based on an event-driven clustering paradigm. All sensors in the event area forward their readings to the appropriate actors by the data aggregation trees. While these works have explored the potentials of WSANs, the reliability issues, in particular, that for event reporting from sensors to actuators, have yet to be addressed.

Transmission failures are frequently happened in wireless communications [20][21]. It may be due to link failures, buffer overruns, path selection errors, unscheduled delays, or other problems [22]. Reliable transport protocols have been investigated using acknowledgements, retransmissions, and replications. It is an common approach to deliver identical copies of a message by multiple paths to mitigate the effect of link failures. Dubois-Ferriere [23] introduced a scheme for error-correction that exploits temporal and spatial diversity through packet combining. Ganesan et. al. [24] described the use of multipath routing for energy-efficient recovery from node failures in wireless sensor networks. It proposes and evaluates the classical node-disjoint multipath and the braided multipath designs. Yu et. al. [25] studied scalable data delivery algorithms in mobile ad hoc networks with node and link failures. It proposes a data delivery algorithm for distributed data fusion in mobile ad hoc networks, where each node controls its data flows and learns routing decisions solely based on their local knowledge.

Erasure code is a replication technique, which can cope with partial data loss efficiently [26]. Several erasure coding techniques such as Reed-Soloman codes and Tornade codes [27]. There are some works in which erasure codes are used to cope with packet transmission failures. S. Kim et. al. [28] study a diverse options for achieving reliable data transfer in WSN with link-level retransmission, erasure code, and route fix. S. Jain et. al. [22] considers the routing problem in a delay tolerant network (DTN) in presence of path failures and show how to split and replicate erasure code message fragment over multiple delivery paths to optimize the probability of successful mes-

sage delivery. Y. Wang et. al. [29] proposed a forwarding algorithm based on the idea of erasure codes for opportunistic networks. Dulman et. al. [30] proposed a multiple transmission solution, which splits the data packets into k parts through erasure codes and send these subpackets instead of the whole packets.

Correlated losses due to obstacles, interference, can lead to consecutive losses, decreasing the effectiveness of erasure code. Weak correlation between quality and distance, hidden problems, and dynamic change of connectivity complicates the situation further [28]. Although the potentials of erasure codes to cope with link failures have been studied, it lacks a comprehensive solution for considering both link utilization and reliability. Moreover, the routing algorithm for event reporting in WSAN with different reliability requirements, have yet to be further investigated.

There are some more related work to our research. Gong et. al. [31] analyzed the anycast semantics for delay tolerant network (DTN) with three new models and presented a anycast routing protocol based on a new routing metric named EMDDA (Expected Multi-Destination Delay for Anycast). Fault tolerance and reliability issues have been studied by Bein et. al. [32]. They explored the reliability issues in multifusion sensor networks, presented and compared Markov models in terms of reliability, cost, and MTTF (Minimum-Time-To-Fail). Sun et. al. [33] presented a distributed technique, Confidence Weighted Voting (CWV), to improve the data reliability and fault tolerance of sensor networks.

4 Network Model and Objective

In this section, we present an WSAN model and list our design objectives of the reliable event reporting framework.

4.1 Network Model

We considered a wireless sensor-actuator network (WSAN) that consists of a collection of sensor nodes s and actuator nodes a. The field covered by this network is divided into virtual grids for event monitoring, as illustrated in Figure 6. We assume that the sensors and actuators are aware of their locations, and hence, the associated grids. The location information can be obtained either through GPS [34] or various localization techniques [35][36][37].

Each sensor is responsible for collecting event data in its associated grid. Since malfunctioned sensors may give inconsistent readings, the data in the same grid will be aggregated to form a consistent mean value before reporting. A subset of the sensors in the field, referred to as *reporting nodes*, v, are responsible for forwarding the aggregated event data to the actuators for further actions. As we will show later, the aggregation occurs in a distributed manner, along with the data flow toward the reporting node v. Also note that the communications from the sensors to the actuators follow an anycast paradigm, that is, an event reporting is successful if any of the actuators receives the report.

We focus on the reliable event data transmission from the sensors to actuators. The corresponding actions that the actuators should perform are out of the scope of this paper, and is really application specific. It is however worth noting that, for most of such applications, perfect reliability as in TCP is often not necessary and even impossible given the error/distortions in aggregation and transmission; on the other hand, timely delivering not only enables short response time for the actuators, but also implies more accurate decisions given the fresher data.

We thus propose a reliability index, which measures the probability that the event data are

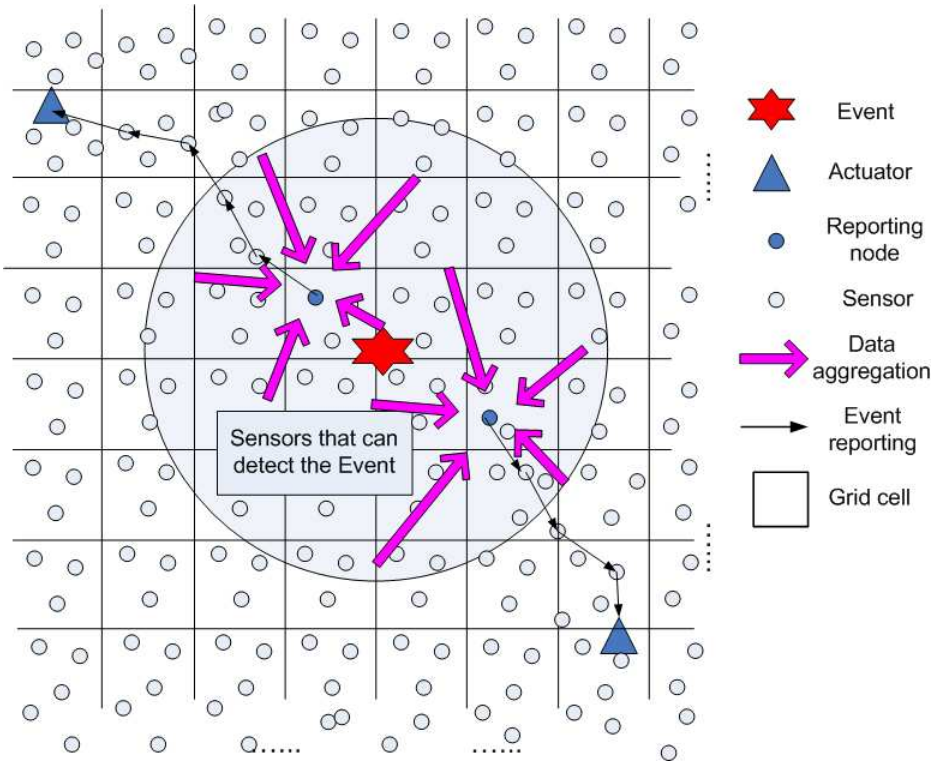

Figure 6: An Illustration of the WSAN Model and Event Reporting from Sensors to Actuators.

aggregated and received accurately within pre-defined latency bounds. Since the events may have different importance, depending on their types, urgency, and seriousness, our index and reporting framework also accommodates such differences. To realize this, each sensor in our framework maintains a priority queue, and, during transmission, important event data are scheduled with higher priorities. Beyond this differentiation in individual nodes, the queue utilization also serves as a criterion for next-hop selection in routing toward actuators.

4.2 Design Objective

We now give a formal description of the system parameters, and our objective is to maximize the overall reliability index, $\mathbb{R}$, across all the events, as follows:

System Parameters

e : Event

q_e : Data report of event e

Q_e : Set of data reports of event e that satisfy the end-to-end latency constraint

$Imp(e)$: Importance of event e

B_e: Latency bound for sensor-actuator reporting of event e

D_{q_e}: End-to-end delay of data report q_e

N_e: Number of data reports for event e

f: Probability of failures in data aggregation

Objective

Maximize

$$\mathbb{R} = \sum_{\forall e} Imp(e) * r_e, \tag{1}$$

where $r_e = \frac{|Q_e|(1-f)}{N_e}$.

Subject to

$$D_{q_e} \leq B_e \tag{2}$$

Clearly, the overall reliability of the system, $\mathbb{R}$, depends on the importance of the events and their respective reliability, r_e. The latter further depends on the reports reaching an actuator within the delay bound and without failure in aggregation. The aggregation failure happens only if malfunctioned sensors dominate a grid.

5 The Reliable Event Reporting Framework

Our framework addresses the whole process for event reporting, and integrates three generic modules to achieve the above reliability objective. Specifically, when an event (*e.g.*, a fire) occurs, the sensors located close to the event will detect it. After aggregation, which removes redundancy and inconsistent readings, the reporting nodes will forward the reports to the actuators. Such forwarding is delay- and importance-aware, implemented through prioritized scheduling and routing in each sensor. We also provide an actuator allocation module that determines the locations of the actuators. It ensures a balanced and delay-minimized allocation of actuators to process the unevenly distributed events in the network.

Figure 7 illustrates the workflow of our framework. We now offer detailed descriptions of the three modules.

5.1 Grid-based Data Aggregation

In a densely deployed sensor network, multiple sensors may sense the same event with similar readings. Hence, it is preferably to aggregate them before reporting to the actuators. Our grid-based aggregation algorithm works as follows (see Figure 8):

For each grid, there is an aggregating node that first collects the event data, $<x_1, x_2, ..., x_n>$, and finds their median med. It will compare each data x_i with med and filter out those with significant difference (*e.g.*, greater than a predefined threshold Δd). These data could be from malfunctioned sensors, which will then be blacklisted. Then, the aggregating node will calculate the mean value $\overline{x_g}$ from the remaining data in grid g (Algorithm 1). We consider the aggregated data to be reliable if more than half of the sensors in the grid are normal. The reliability for the aggregated data from grid g thus can be evaluated as

$$1 - f_g = 1 - \sum_{i=\lceil N_x/2 \rceil}^{N_x} \binom{N_x}{i} (f_s)^i (1 - f_s)^{N_x - i},$$

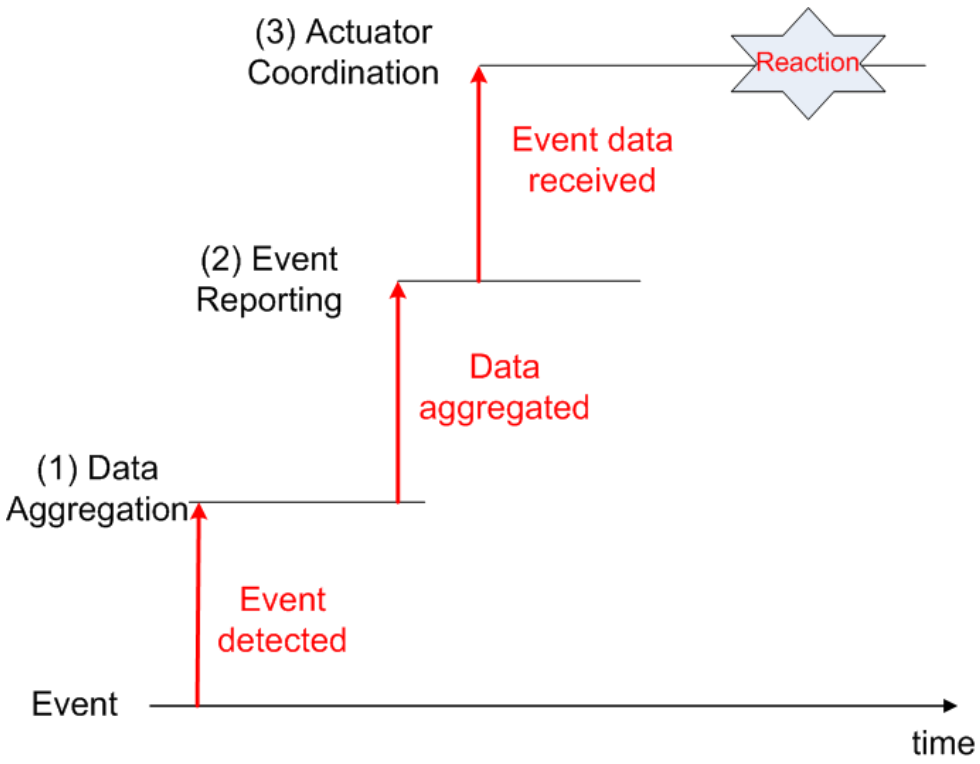

Figure 7: Workflow of the Framework.

where f_g is the failure probability of grid g on data aggregation, N_x is the number of nodes in grid g, and f_s is ratio of the malfunctioned sensors.

The aggregating node may serve as the reporting node to forward the aggregated data to actuators. The aggregation however can be easily extended to multiple levels, where a reporting node is responsible for further collecting and aggregating the data from the aggregating nodes in surrounding grids, as shown in v (Figure 8). For the 2-level case, each sensor independently decides whether it will serve as a reporting node according to probability p_v. Here, $p_v = \frac{1}{N_g * N_x}$, where N_g is the number of data reports to be transmitted by a reporting node. Notice that each grid has only one summarized mean data value, so N_g is also equal to the number of grids to be reported by one reporting node. Other bidding algorithms for reporting nodes selection could be used as well in our framework, *e.g.*, those in [38].

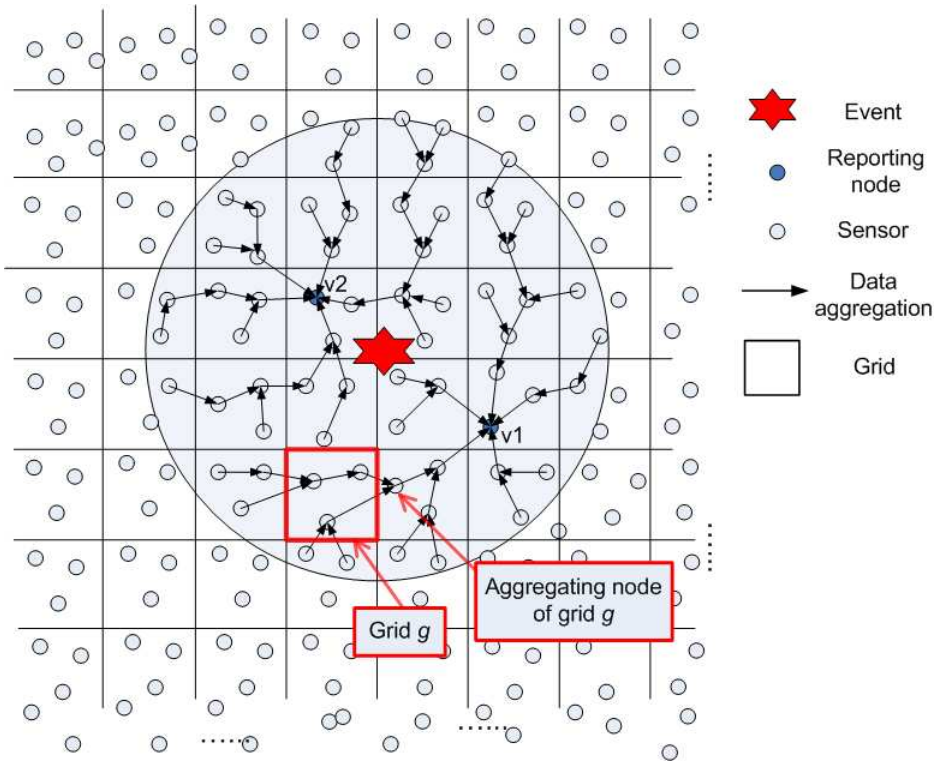

Figure 8: Grid-based Data Aggregation.

5.2 Priority-based Event Reporting

The routing and transmission protocol for event reporting from the reporting nodes to the actuators is the core module in our framework. The key design objective here is to maximize the number of reports reaching the destination within their latency bound, and, for different event types, give preference to important events. To this end, we adopt a priority queue in each sensor, which plays two important roles: 1) prioritized scheduling to speed up important event data transmission; and 2) queue utilization as an index for route selection to meet the latency bounds.

In our preemptive priority queue, the packets for the event data are placed according to their data importance, and each priority is served in a first-in-first-out (FIFO) discipline. Since a light-weighted sensor network with few event occurrences seldom suffers from excessive transmission delays, we focus on the network with frequent event occurrences. In such a network, queuing delay

Algorithm 1 Data Aggregation

Define: $\overline{x_g}$ as aggregated data mean of grid g;
for each sensor s receive data x_i **do**
  if multiple $x_i \in g$ and s is the aggregating node **then**
    find the median med among data $<x_1, x_2, ..., x_n>$;
    for each data $x_i \in g$ **do**
      if x_i - $med > \Delta d$ **then**
        blacklist node i
      end if
    end for
    $\overline{x_g}$ = mean of the un-blacklisted data $x_i \in g$
  end if
end for

can be the dominating factor over the processing and propagation delays.

The queueing delay of the highest priority queue is $\overline{d_{q_1}} = \overline{R} + \overline{SN_{q_1}}$, where $\overline{R} = \frac{1}{2}\sum_{k=1}^{K} \lambda_k \overline{S^2}$ is the mean residual service time in the node, $\overline{N_{q_1}}$ is the mean number of packets in first queue, K is the number of priority queues, λ_k is the arrival rate of the packets in priority queue k, $\overline{S}$, and $\overline{S^2}$ and are the expectation and second moment of the service time of the sensor. We assume the packet arrival is Poisson. $\overline{S}$ can be obtained in each individual sensor by observing the time it takes to serve a packet.

By Little's theorem, $\overline{N_{q_1}} = \lambda_1 \overline{d_{q_1}}$, and the load of priority k is $\rho_k = \lambda_k \overline{S}$, the waiting time of packet in the first priority queue is:

$$\overline{d_{q_1}} = \frac{\overline{R}}{1 - \rho_1}$$

Similarly, the waiting time of packet in the second priority is:

$$\overline{d_{q_2}} = \overline{R} + \overline{SN_{q_1}} + \overline{SN_{q_2}} + \overline{S}\lambda_1 \overline{d_{q_2}} = \frac{\overline{R} + \rho_1 \overline{d_{q_1}}}{1 - \rho_1 - \rho_2}$$

The mean waiting time $\overline{d_{q_k}}$ of packet in the k^{th} priority is:

$$\overline{d_{q_k}} = \frac{\overline{R}}{(1 - \rho_1 - ... - \rho_{k-1})(1 - \rho_1 - ... - \rho_k)}$$

Sensors periodically exchange control information with neighboring nodes through beacon messages or piggyback messages. A control message contains such information as waiting time and rate to the actuators. When routing the event data packets, a sensor should not select a next hop that is busy in forwarding important data. On the contrary, it selects a next hop that has a smaller queueing time for the corresponding priority, or it may select a next hop that it can preempt the data packets with lower importance.

More formally, consider node i that receives a new event data $data_e$. Given the control message it received from neighbor j, node i can obtain $<a, \overline{S}, \lambda_{high}, \lambda_{low}>$, where a is the target actuator, $\overline{S}$ is the expected service time of node j, $\lambda_{high} = \sum_{\forall k, imp(data_k) \geq imp(data_e)} \lambda_k$ is the sum of all λ_k of the data that are equal or more important than $data_e$, and $\lambda_{low} = \sum_{\forall k, imp(data_k) < imp(data_e)} \lambda_k$ is the sum of all λ_k of the data that are less important than $data_e$.

Node i needs to ensure that the end-to-end latency for $data_e$ is no more than the latency bound B_e. To this end, it first estimates the advancement $h_{i,j}$ towards the actuator a from i to j, and then the maximum hop-to-hop delay from i to j, $delay_{i,j}$.

$$h_{i,j} = \frac{\parallel a, i \parallel - \parallel a, j \parallel}{\parallel a, i \parallel}$$

So,

$$delay_{i,j} \leq B_e * h_{i,j}$$

Since $delay_{i,j} = d_q + d_{tran} + d_{prop} + d_{proc}$, the maximum queueing delay $d_{q_{max}}$ is:

$$d_{q_{max}} = B_e * h_{i,j} - (d_{tran} + d_{prop} + d_{proc})$$

Only neighbors with $d_{q_{max}} > 0$ will be considered as the next hop; otherwise the latency bound cannot be met. Among these candidates, node i starts inspecting the neighbors with both $\lambda_{low} = 0$ and $\lambda_{high} = 0$, followed by the remaining neighbors. Here, $\lambda_{low} = 0$ implies that it is

not forwarding any event data with importance lower than that considering by node i; if node i forwards the data to this node, it will not affect the transmission time for the existing packets in that node; Similarly, $\lambda_{high} = 0$ means that it is not transmitting any data with higher importance, so the data from node i, if forwarded, can be served with the highest priority. For each candidate above, node i calculates the maximum data rate λ_i that it can forward while satisfying the latency bound:

$$d_{q_{max}} > \frac{\overline{R}}{(1 - \lambda_{high}\overline{S})(1 - \lambda_{high}\overline{S} - \rho_i)},$$

and

$$\rho_{i,j} < 1 - \lambda_{high}\overline{S} - \frac{\overline{R}}{(1 - \lambda_{high}\overline{S})d_{q_{max}}},$$

where $\rho_{i,j} = \lambda_{i,j}\overline{S}$ is the maximum affordable load of j for handling data from i on event e.

Then the event data packets are forwarded to the neighbor with the highest $h_{i,j}$ and $\lambda_{i,j}$, which is the closest to the destination with enough capacity for transmission. Each intermediate node updates the latency bound B_e before forwarding the packet to next hop by this equation:

$$B_e = B_e - (t_{depart} - t_{arrive}) - d_{tran} - d_{prop},$$

where $(t_{depart} - t_{arrive})$ is the elapse time of the packet in a node, d_{tran} can be computed using the transmission rate and the length of the frame containing the packets, and d_{prop} is the propagation time, which is in the order of several microseconds in wireless transmission.

After the transmission starts, the sensor will update its $\overline{S}$ and the routes regularly to make sure the transmission can be completed within the latency bound. If the latency bound is not met, the sensor has to forward the packets to another route. In the worst case, if no alternative can be found, the sensor may inform the previous node to select another route in the future.

5.3 Resist to Link Failures

As mentioned before, messages will be dropped if they expire before reaching the actuators. Apart from that, data may be loss due to link failures, like link transmission errors, buffer overflow, or node failures along the path. There exist multiple destination (actuators) and multiple paths for anycast event reporting in WSAN. Different levels of reliability can be obtained based on the requirements of various event data. We adopt the erasure coding techniques to handle link failures and provide QoS in terms of reliability. In this Section, we extend the above routing algorithm to cope with link failures in event reporting.

In erasure coding, a message is converted into a larger set of code blocks such that any sufficiently large subset of the generated code blocks can be used to reconstruct the original message. Existing converting algorithms include Vandermonde Matrix, Reed-Solomon codes, etc. can be applied in our routing algorithm. The key point is that when using erasure coding with a replication factor of r_p, only $1/r_p$ of the code blocks are required to decode the message. Our routing algorithm will choose the appropriate value of r_p according to the event reliability requirement, distance to the actuators, and network conditions. More specifically, each node i maintain the recent average packet loss rate $L_{i,j}$ to each immediate neighbor j.

For simplicity, we consider the event reliability requirement R_{req} is proportional to its event importance. Say, an event with important level of 0.8 will have the reliability requirement of 0.8. The reliability represents the probability of the event data to be reported to the actuator successfully. Instead of forwarding a message to one next hop with the highest $h_{i,j}$ and $\lambda_{i,j}$, an reporting node v decides the replication factor r_p as follow:

It selects the top k next hops with the highest $h_{i,j}$ and $\lambda_{i,j}$. They have the corresponding packet loss rate of $L_{i,j}$. Then, v estimates the path reliability R_{req_j} via each neighbor j.

$$R_{req_j} = (1 - L_{i,j})^{1/h_{i,j}}$$

Reporting node v will allocate the code blocks to the neighbors according to their $\lambda_{i,j}$. The neighbor with higher $\lambda_{i,j}$ will be allocated with more code blocks. The probability that the code block can be delivered successfully R_i to the actuator by the k neighbors can be estimated as:

$$R_i = \sum_{j=1}^{k} (\frac{\lambda_j}{\sum_{j=1}^{k} \lambda_j} * R_{req_j})$$

Then, v determines the replication factor r_p with the following equation:

$$R_i * r_p \geq R_{req}$$

It produces $N_b = M * r_p/b$ number of code blocks, where M is the data size and b is the size of code blocks. Each of the above neighbor will be allocated with $\frac{\lambda_j}{\sum_{j=1}^{k} \lambda_j} * N_b$ code blocks.

The corresponding R_{req_j} becomes the required reliability of that particular path from j to the actuator. Each node j, which received the code blocks, selects the next hop among the neighbors k with high $h_{i,j}$ and $\lambda_{i,j}$. It then estimates the reliability obtained forwarding the block to neighbor k. Similarly, the reliability obtained must be greater than the R_{req_j}, such that the selected neighbor k' must the link loss rate:

$$L_{j,k'} \leq 1 - (R_{req_j})^{h_{j,k'}}$$

Since the reliability of a path is composed by the a series of links in the path:

$$(1 - L_1)(1 - L_2)(1 - L_3)...(1 - L_n) > R_{req_j},$$

where the $L_1, L_2, ..., L_n$ are the packet loss rate of the links on the path.

and

$$(1 - L_2)(1 - L_3)...(1 - L_n) > R_{req_j}/(1 - L_1),$$

Node j updates the reliability R_{req_j} and forwards it with the code blocks to the selected neighbor k'.

$$R'_{req_j} = R_{req_j}/(1 - L_{j,k'})$$

5.4 Actuator Allocation

Once an actuator receives the event report, it will perform application-specific actions. Meanwhile, it will inform other actuators to suppress their potential actions in case some of them receive the same report later. Such coordination can be achieved through direct one-hop communications with another wireless channel given that the actuators are much more powerful.

In this anycast paradigm, reducing the distances from the sensors to their closest actuators clearly decreases the reporting delay. Since the reports are triggered by events, we suggest that an actuator allocation be performed according to the event occurrence frequency. Intuitively, the locations with more events should be allocated more actuators, so as to reduce the reporting distances. Such an allocation can be performed in the initial stage based on pre-estimated frequencies, or, with mobile actuators, performed periodically to accommodate event dynamics.

Algorithm 2 gives an allocation that balances the load of the actuators as well as minimizes the anycast distances. In this algorithm, first, the event frequency $freq_g$ of every grid g will be summed up. Then, the field A will be equally divided into two, denoted by $A1$ and $A2$, according to the frequency distribution. That is, $A1$ and $A2$ have the same event occurrence frequency and each is allocated half of the actuators. The process repeats recursively for $A1$ and $A2$, until each subfield contains only one actuator.

Figures 9 and 10 demonstrate our actuator allocation results with 6 and 10 actuators, respectively. In practice, the algorithm can be executed by one designated actuator after collecting the event frequency information. It then informs the allocation result to other actuators, which may then move to the corresponding locations.

Algorithm 2 Actuator Allocation

$ActuatorAllocation$(Field A, int $ActuatorNum$)
$TotalFreq \leftarrow \sum_{\forall g_i \in A} freq_{g_i}$
$TmpFreq \leftarrow 0$;
$i \leftarrow 0$;
while TmpFreq $<$ TotalFreq/2 **do**
  $TmpFreq \leftarrow TmpFreq + freq_{g_i}$;
  $i++$;
end while
$A1 \leftarrow \bigcup_{k=0}^{i} g_i$;
$A2 \leftarrow A - A1$;
$ActuatorAllocation(A1, ActuatorNum/2)$;
$ActuatorAllocation(A2, ActuatorNum-$
$ActuatorNum/2)$;
end $ActuatorAllocation$

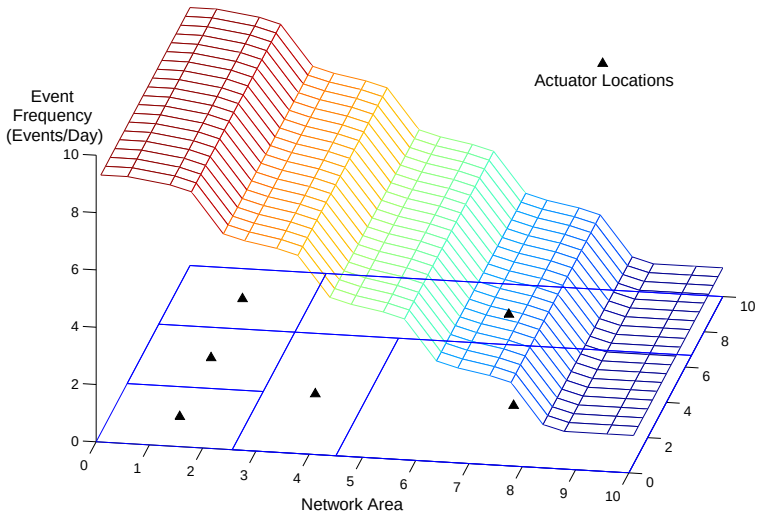

Figure 9: Actuator Allocation with 6 Actuators.

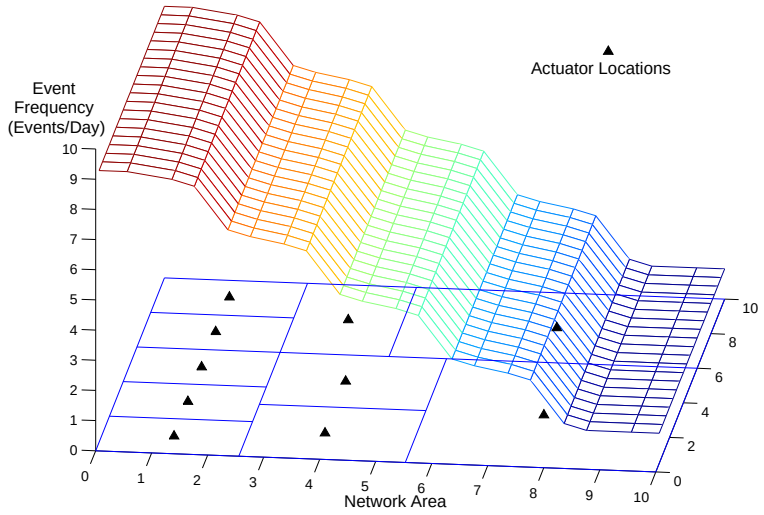

Figure 10: Actuator Allocation with 10 Actuators.

6 Performance Evaluation

We have conducted *ns-2* [39] simulations for our proposed reliable event reporting framework. The simulation settings are mainly drawn from [8], which are summarized in Table 1.

Table 1: Simulation Parameters

Network size	200m x 200m
No. of sensors	100
Node placement	Uniform
Radio range	40m
MAC layer	IEEE 802.11
Bandwidth	2Mbps
Packet size	32 bytes
No. of actuators	1-6
No. of concurrent events	3-10
B_e	2sec

6.1 Reliability on Event Reporting

In the first set of experiments, we evaluate the reliability of our event reporting algorithm. To this end, we generate 4 events randomly in the network and vary their data rate from 10pkt/sec to 80pkt/sec. Two of the four events are high priority events with importance 1.0 (events 2 and 4), while the two are low priority events with importance 0.3 (events 1 and 3). Each packet should be reported to the actuator within the latency bound of 2 sec.

We first assume that all the reports are destined to the same actuator.

We fix the locations of the events and change the seed to generate different sensor locations. Figure 11 shows the on-time reachability of the four events with our priority-based event reporting with event importance (PREI). For comparison, we also show the result with the geographic routing protocol (GRP) [40], where greedy forwarding is employed and there is no differentiation

regarding the event types. We can see that our PREI achieves much higher on-time reachability for the important events (event 2 and 4). The reachability for the low important events however is lower than that in GRP. This follows our design objective that important events will be served with higher priority and better quality routes.

Note that, even the two different events are of the same importance, their reachabilities could be different, depending on their locations. This also happens when we compare the average delay. However, our PREI generally performs better for the same event.

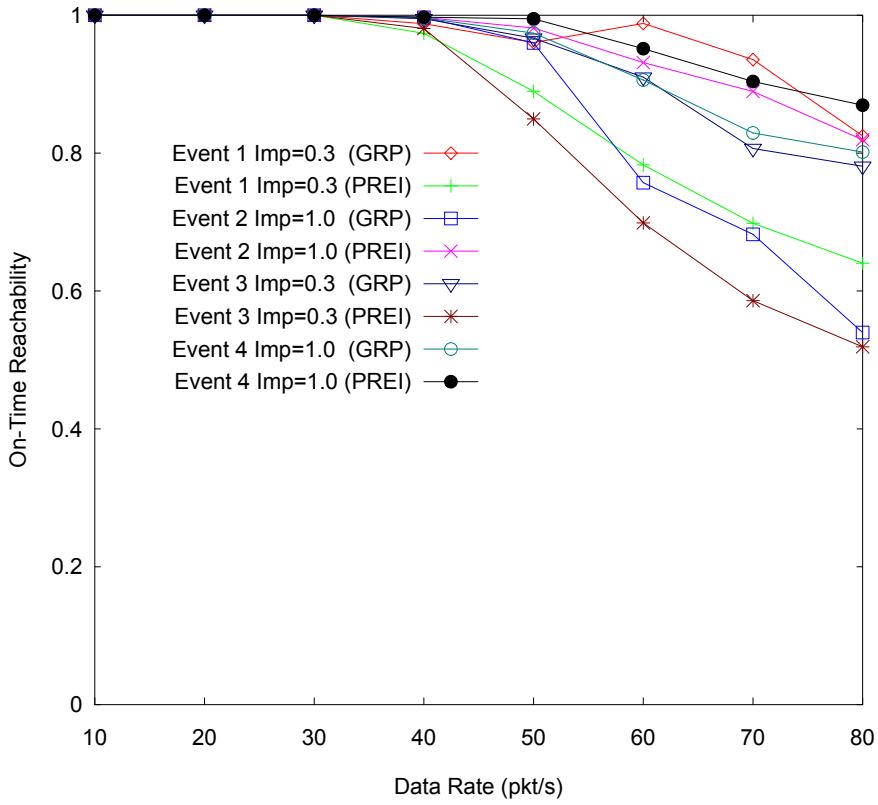

Figure 11: On-Time Reachability.

Figure 12 further shows the average delays in the PREI and GRP. It is clear that the delay in PREI is generally lower than that in GRP. This is because the PREI considers the workload of the neighbors when selecting the route. An interesting observation is that, in PREI, the average delays

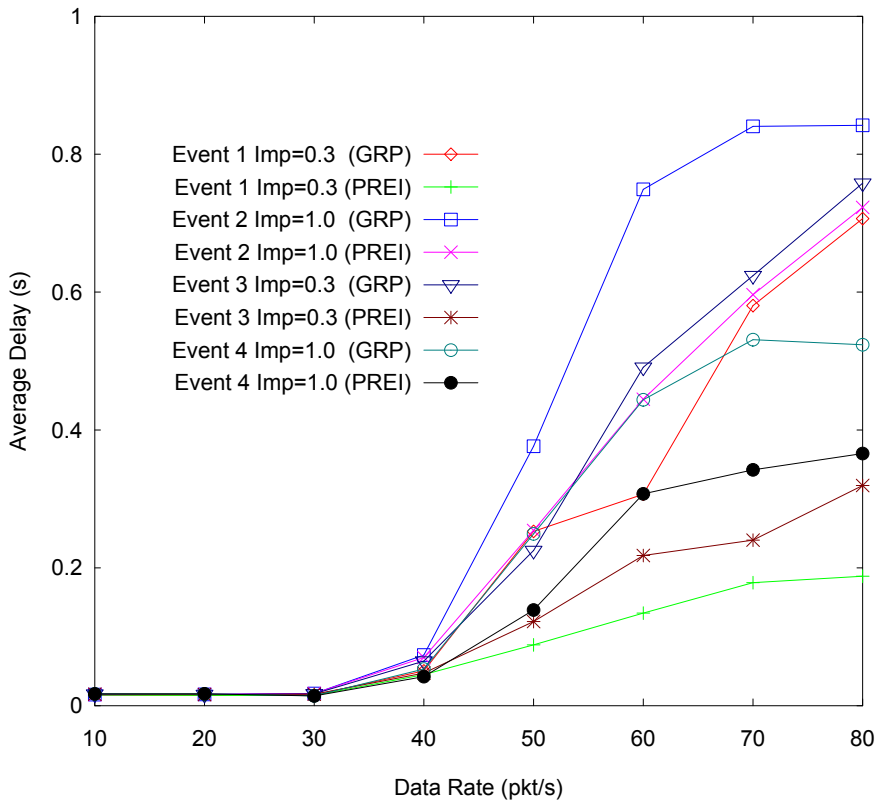

Figure 12: Average Delay.

of the more important events are not necessarily lower than the less important events; e.g., the delay for Event 1 is lower than all others, though its importance is not high. The reason is that this event is closer to the actuator than others. We have calculate the average per-hop delays, which we find are generally lower for important events. Also note that the actuator allocation algorithm can mitigate this problem, as will be examined later.

Finally, Figure 13 shows the overall reliability index, $\mathbb{R}$, of the two protocols. Again, it demonstrates that the PREI outperforms GRP, and the gap increases when the data rate becomes higher.

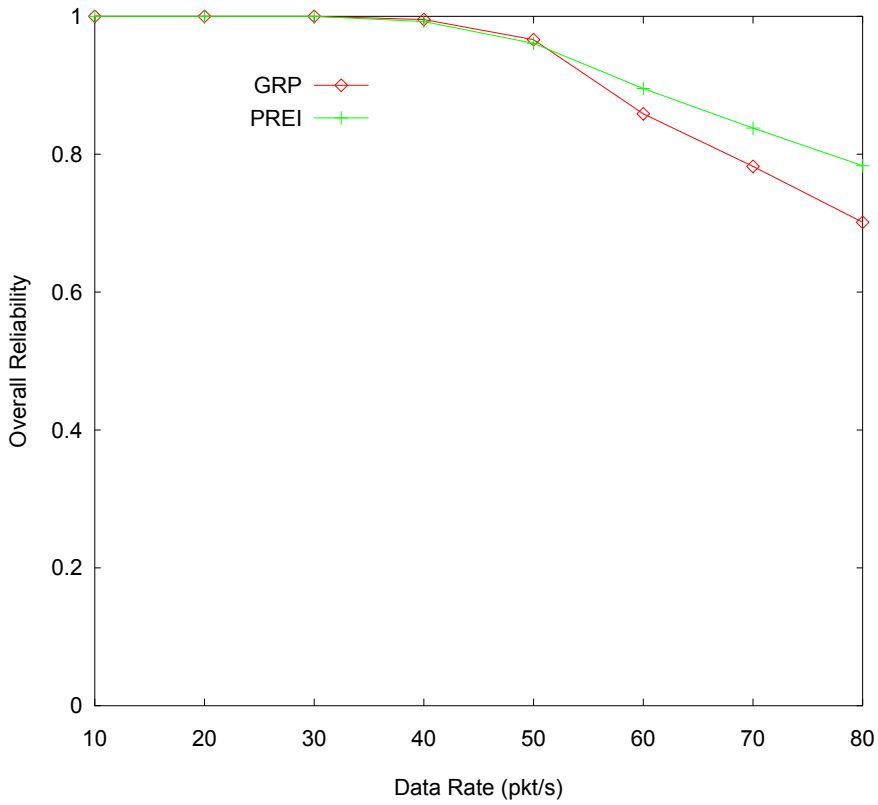

Figure 13: Overall Reliability.

6.2 Actuator Allocation

In this experiment, we show the effectiveness of our actuator allocation algorithm. To emulate the nonuniform event occurrences, we divide the whole field into three, with the event occurrence probability 0.6, 0.333, and 0.067, respectively.

Our simulator generates events according to the above probability with data rate 60pkt/s, and it allows different number of concurrent events in the network as represented in the x-axis of Figures 14 and 15.

Figure 14 gives the on-time reachability with different number of concurrent events. We first focus on 2 and 3 actuators only, and will investigate the impact of using more actuators later. We can see from Figure 14 that the reliability with actuator allocation outperforms that without

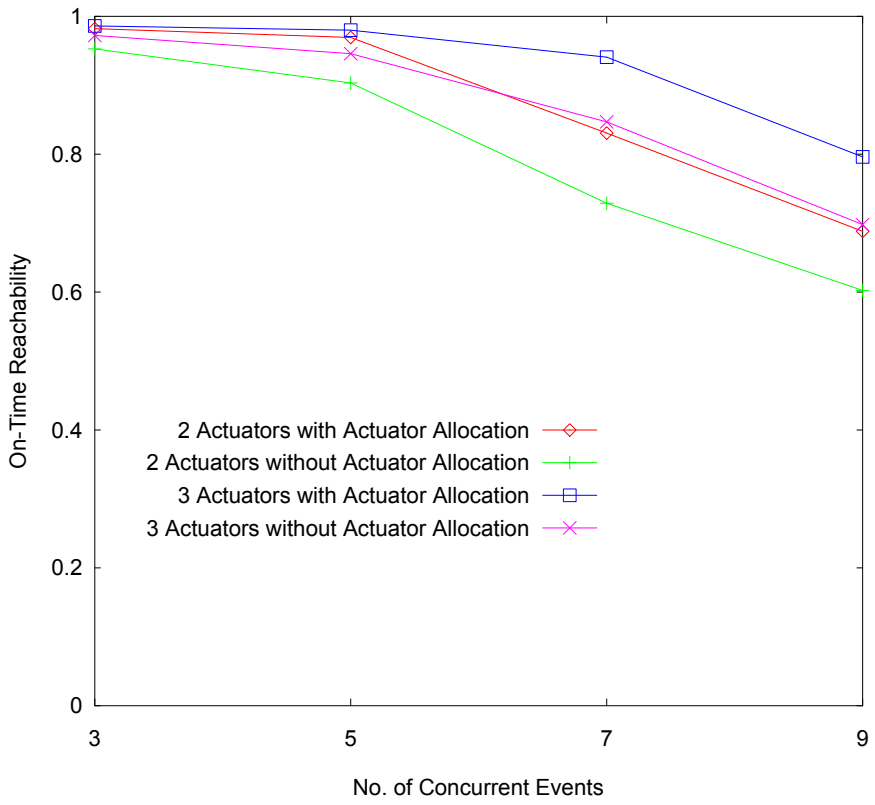

Figure 14: On-Time Reachability with Actuator Allocation.

allocation (i.e., random distribution). While the more actuators, the better performance we can expect, we notice that the effect of allocation is remarkable.In fact, the performance of 2-actuator with allocation is very close to that of 3-actuator without allocation, and even outperforms it for less concurrent events.

Figure 15 shows the corresponding average delay. Not surprisingly, 3-actuator with allocation achieves the lowest delay. Similar to the on-time reachability, the delay for the 2-actuator with allocation is close to the 3-actuator without allocation case. The results suggest that actuator allocation is an effective tool to improve the efficiency of event reporting.

To further investigate the impact of the number of actuators, we fix the number of concurrent events to 10 and vary the number of actuators from 1 to 6. Figure 16 shows the on-time-reachability

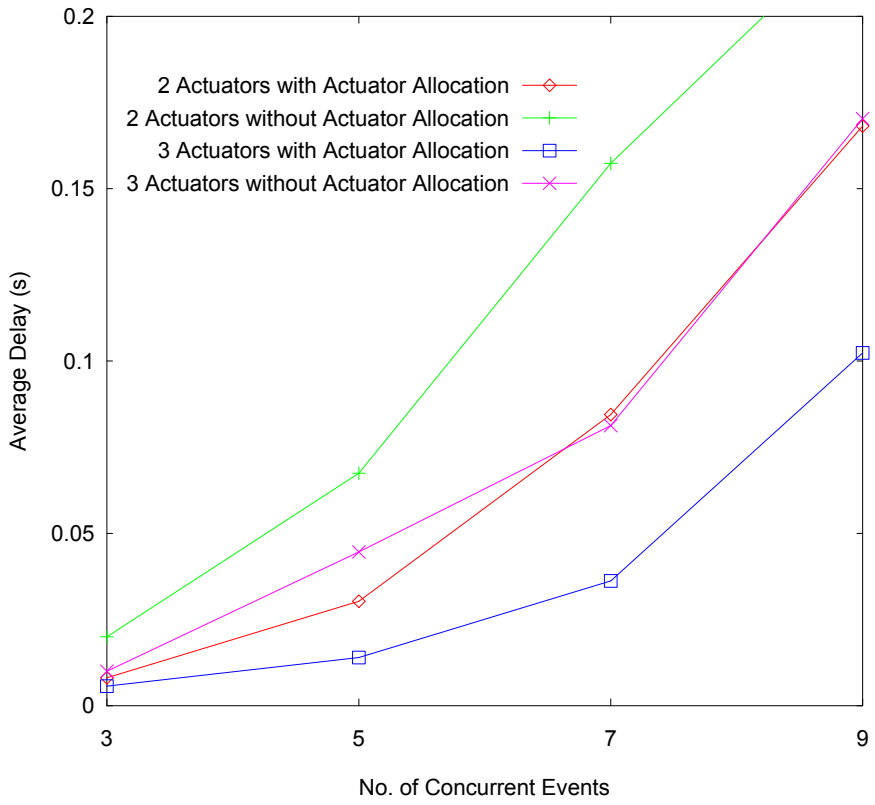

Figure 15: Average Delay with Actuator Allocation.

as a function of the number of actuators with and without actuator allocation. Again, event reporting with actuator allocation achieves higher on-time reachability than that without actuator allocation with the same number of actuators. Intuitively, given more actuators, we can generally expect better performance, even if they are randomly deployed. This can be verified from the figure. We can see that the on-time reachability monotonically increases with more actuators, while the difference between the two schemes (with/without allocation) becomes smaller. Similar trends can also be found in Figure 17, which shows the average delay of event reporting as a function of the number of actuators.

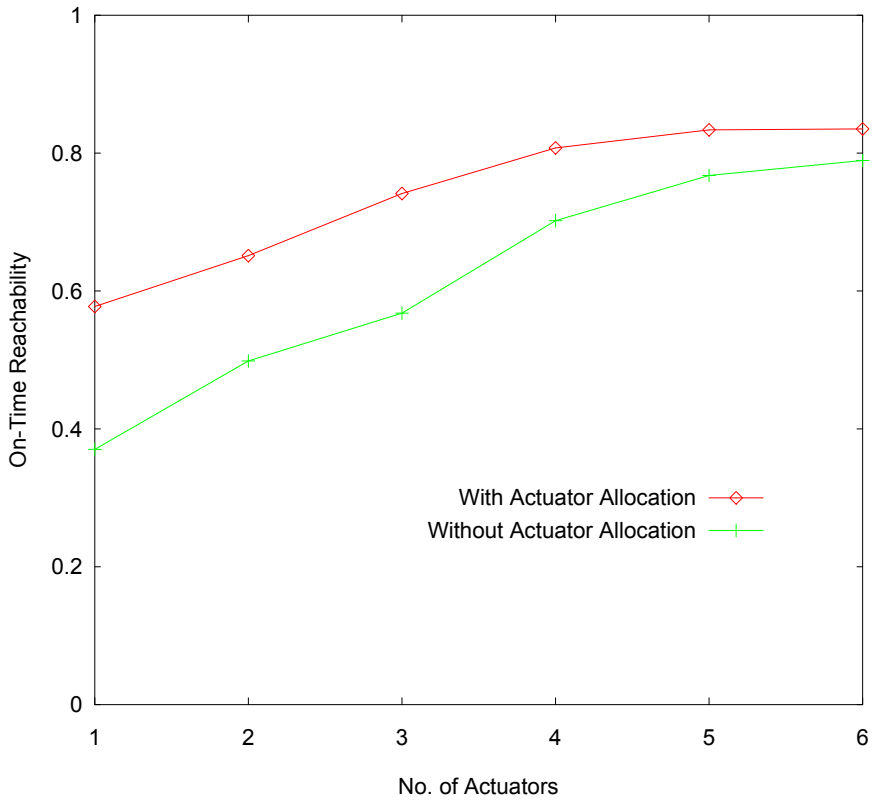

Figure 16: On-Time Reachability vs. No. of Actuators.

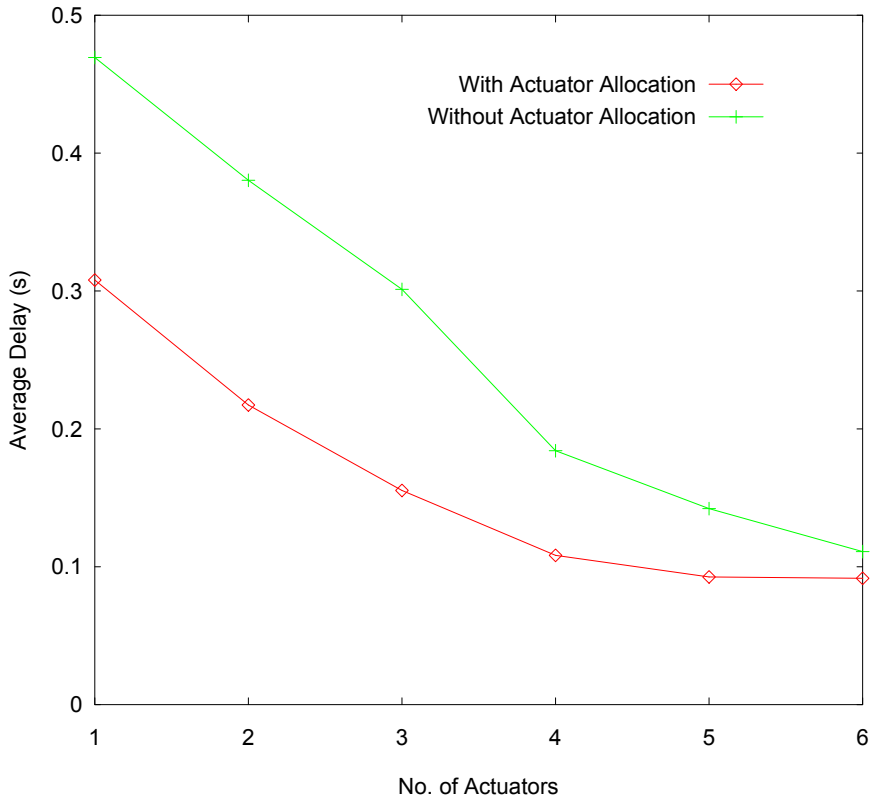

Figure 17: Average Delay vs. No. of Actuators.

7 Conclusion

In this paper, we focused on reliable event reporting from sensors to actuators in a wireless sensor-actuator network (WSAN). We argued that the reliability in this context is closely related to the delay, or the freshness of the events, and they should be jointly optimized. We also suggested that the non-uniform importance of the events can be explored in the optimization. Following this argument, we proposed a general delay- and importance-aware event reporting framework. Our framework seamlessly integrates three key modules to maximize the reliability index: 1) A multi-level data aggregation scheme, which is fault-tolerant with error-prone sensors; 2) A priority-based transmission protocol, which accounts for both the importance and delay requirements of the events; and 3) an actuator allocation algorithm, which smartly distributes the actuators to match the demands from the sensors.

Within this generic framework, we presented optimized design for each of the modules, and also discussed their interactions. We also evaluated the performance of our framework through simulations. The results demonstrated that our framework makes effective use of the actuators, and can significantly enhance the reliability in event reporting.

References

[1] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *Elsevier Ad Hoc Networks Journal*, pp. 325–349, 2005.

[2] I. Akyildiz, I. H. Kasimoglu, D. Pompili, and T. Melodia, *Actor (Actuator) and Sensor Networks*, Apr 2004, http://users.ece.gatech.edu/ ismailhk/actors/index.html.

[3] I. F. Akyildiz, W. Su, and T. Sandarasubramaniam, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 5, pp. 393–422, 2002.

[4] I. F. Akyldiz and I. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Elsevier Ad Hoc Networks Journal*, Octocber 2004.

[5] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proc. of ACM MobiCom*, Seattle, Washington, U.S., 1999.

[6] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications ACM*, vol. 43, no. 5, pp. 551–558, 2000.

[7] E. C.-H. Ngai, M. R. Lyu, and J. Liu, "A real-time communication framework for wireless sensor-actuator networks," in *Proc. of the IEEE Aerospace Conference*, Big Sky, Montana, U.S., Mar 2006.

[8] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: a real-time routing protocol for sensor networks," in *Proc. of the IEEE ICDCS*, Providence, RI, U.S., May 2003, pp. 46–55.

[9] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, "RAP: a real-time communication architecture for large-scale wireless sensor networks," in *Proc. of the IEEE RTAS*, San Jose, CA, U.S., Sep 2002.

[10] E. Felemban, C.-G. Lee, E. Ekici, R. Boder, and S. Vural, "Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks," in *Proc. of the IEEE Infocom*, Miami, FL, U.S., Mar 2005.

[11] J. Aidemark, P. Folkesson, and J. Karlsson, "A framework for node-level fault tolerance in distributed real-time systems," in *Proc. of the IEEE DSN*, Yokohama, Japan, Jun 28 - Jul 1, 2005.

[12] G. Khanna, S. Bagchi, and Y.-S. Wu, "Fault tolerant energy aware data dissemination protocol in sensor networks," in *Proc. of the IEEE DSN*, Florence, Italy, Jun 28 - Jul 1, 2004.

[13] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in delay tolerant network," in *Proc. of the ACM SIGCOMM*, Pennsylvania, U.S., Aug 2005.

[14] R. K. Sahoo, A. Sivasubramaniam, M. S. Squillante, and Y. Zhang, "Failure data analysis of a large-scale heterogeneous server environment," in *Proc. of the IEEE DSN*, Florence, Italy, Jun 28 - Jul 1, 2004.

[15] V. P. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, and N. Shroff, "A minimum cost heterogeneous sensor network with a lifetime constraint," *IEEE Transaction on Mobile Computing*, vol. 4, no. 1, Jan/Feb 2005.

[16] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh, "Exploiting heterogeneity in sensor networks," in *Proc. of the IEEE Infocom*, Miami, FL, U.S., Mar 2005.

[17] W. Hu, S. Jha, and N. Bulusu, "A communication paradigm for hybrid sensor/actuator networks," in *Proc. of the 15th IEEE Intl. Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Bacelona, Spain, Sep 2004.

[18] E. Cayirci, T. Coplu, and O. Emiroglu, "Power aware many to many routing in wireless sensor and actuator networks," in *Proc. of the 2nd European Workshop on Wireless Sensor Networks (EWSN)*, Istanbul, Turkey, 31 Jan - 2 Feb 2005, pp. 236–245.

[19] T. Melodia, D. Pompili, V. C. Gungor, and I. F. Akyildiz, "A distributed coordination framework for wireless sensor and actor networks," in *Proc. of ACM Mobihoc*, Urbana-Champaign, IL, U.S., 2005, pp. 99–110.

[20] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *ACM Sensys*, Nov 2003.

[21] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of relable multiple routing in sensor networks," in *ACM Sensys*, Nov 2003.

[22] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in delay tolerant network," in *SIGCOMM*, Philadelphia, Pennsylvania, U.S., Aug 2005.

[23] H. Dubois-Ferriere, D. Estrin, and M. Vetterli, "Packet combining in sensor networks," in *ACM Sensys*, San Diego, California, U.S., Nov 2005.

[24] D. G. amd R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *Mobile computing and communication review*, vol. 1, no. 2, 2001.

[25] B. Yu, P. Scerri, K. Sycara, Y. Xu, and M. Lewis, "Scalable and reliable data delivery in mobile ad hoc sensor networks," in *ACM AAMAS*, Hakodate, Hokkaido, Japan, May 2006.

[26] J. W. Byers, M. Luby, and M. Mitzenmacher, "A digital fountain approach to asynchronous reliable multicast," *IEEE J-SAC*, vol. 20, no. 8, 2002.

[27] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Transactions on Information Theory*, 2001.

[28] S. Kim, R. Fonseca, and D. Culler, "Reliable transfer on wireless sensor networks," in *SECON*, 2004.

[29] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-coding based routing for opportunistic networks," in *SIGCOMM Workshop*, Philadelphia, Pennsylvania, U.S., Aug 2005.

[30] S. Dulman, T. Nieberg, J. Wu, and P. Havinga, "Trade-off between traffic overhead and reliability in multiple routing for wireless sensor networks," in *WCNC*, 2003.

[31] ——, "Anycast routing in delay tolerant network," in *Microsoft Technical Report MSR-TR-2006-04*, Jan 2006.

[32] D. Bein, V. Jolly, B. Kumar, and S. Latifi, "Reliability modeling in wireless sensor networks," *International Journal of Information Technology*, vol. 11, no. 2, 2005.

[33] T. Sun, L.-Y. Chen, C.-C. Han, and M. Gerla, "Reliable sensor networks for planet exploration," in *ICNSC*, 2005.

[34] J. G. McNeff, "The global positioning system," *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, pp. 645–652, Mar 2002.

[35] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proc. of ACM Mobicom*, Philadelphia, PA, U.S., 26 Sep - 1 Oct 2004, pp. 99–110.

[36] A. Savvides, C. C. Han, and M. B. Srivastava, "Dynamic fine-grained location in ad hoc networks of sensors," in *Proc. of ACM Mobicom*, Philadelphia, PA, U.S., 2001, pp. 166–179.

[37] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proc. of ACM Mobicom*, San Diego, CA, U.S., 2003, pp. 81–95.

[38] B. Krishnamachari, D. Estrin, and S. Wicker, "Modelling datacentric routing in wireless sensor networks," in *Proc. of IEEE Infocom*, 2002.

[39] K. Fall and K. Varadhan, *The ns manual*, Dec 2003, http://www.isi.edu/nsnam/ns.

[40] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. of ACM MobiCom*, Boston, Massachusetts, U.S., 2000.