Reliable Reporting of Delay-Sensitive Events in Wireless Sensor-Actuator Networks

Edith C. H. Ngai   Yangfan Zhou   Michael R. Lyu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong, China
Email: {chngai, yfzhou, lyu}@cse.cuhk.edu.hk

Jiangchuan Liu
School of Computing Science
Simon Fraser University
Vancouver, BC, Canada
Email: jcliu@cs.sfu.ca

***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.

I. INTRODUCTION

The advances of hardware and software technologies for embedded systems have turned micro sensors with radio transceivers into reality [1][2][3][4]. 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 [5]. 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 algorithm, 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

1-4244-0507-6/06/$20.00 ©2006 IEEE

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.

II. RELATED WORK

Wireless sensor networks (WSNs) have been extensively studied recently; see surveys in [1][2][3]. Efficient and reliable event reporting is also an important issue in WSANs. He et. al. [6] proposed a real-time communication protocol SPEED, which combines feedback control and non-deterministic QoS-aware geographic forwarding. Lu et. al. [7] described a packet scheduling policy, called Velocity Monotonic Scheduling, which inherently accounts for both time and distance constraints. Felemban et. al. [8] 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. [9] 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. [10] suggested that the failure of any node in a path can be detected and recovered using backup routes. Assayad et al. [11] proposed a bi-criteria scheduling heuristic in data-flow graphs to maximize the reliability and minimize the runtime. S. Jain et al. [12] 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.*, [13][14][15], but they do not cope with the special features of actuators. For WSAN, Hu et. al. [16] 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. [17] 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. [18] 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.

III. NETWORK MODEL AND OBJECTIVE

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

A. 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. 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 [19] or various localization techniques [20][21][22].

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 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.

B. 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

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

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

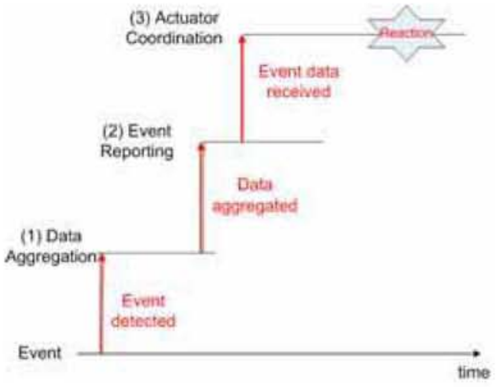

Fig. 1. Workflow of the Framework.

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.

IV. 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 1 illustrates the workflow of our framework. We now offer detailed descriptions of the three modules.

A. 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 2):

For each grid, there is an aggregating node that first collects the event data, $<x_1, x_2, ..., x_n>$, and finds their

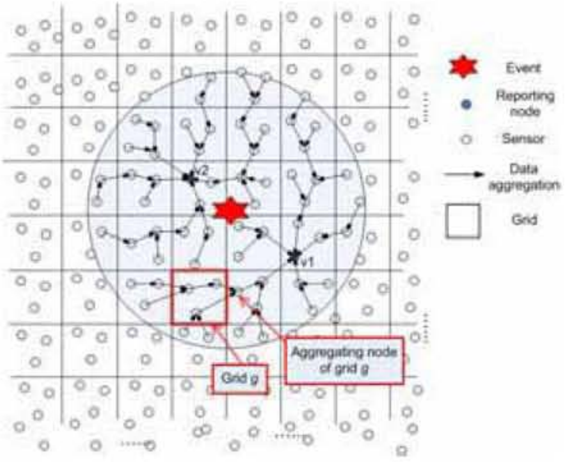

Fig. 2. Grid-based Data Aggregation.

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

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

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 2). 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 [23].

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
```

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

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, $\lambda_{high} = \sum_{k,imp(data_k)\quad imp(data_e)} \lambda_k$ is the sum of all packet arrival rates λ_k of the data that are equal or more important than $data_e$, and $\lambda_{low} = \sum_{k,imp(data_k)<imp(data_e)} \lambda_k$ 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, $delay_{i,j}$.

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

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 [24]:

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

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.

C. 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.

We provide an allocation algorithm that balances the load of the actuators as well as minimizes the anycast distances. 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. Detailed algorithm can be found in [24].

Figures 3 demonstrates our actuator allocation results with 6 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.

V. PERFORMANCE EVALUATION

We have conducted *ns-2* [25] simulations for our proposed reliable event reporting framework. The simulation settings are mainly drawn from [6], which are summarized in Table I.

A. 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

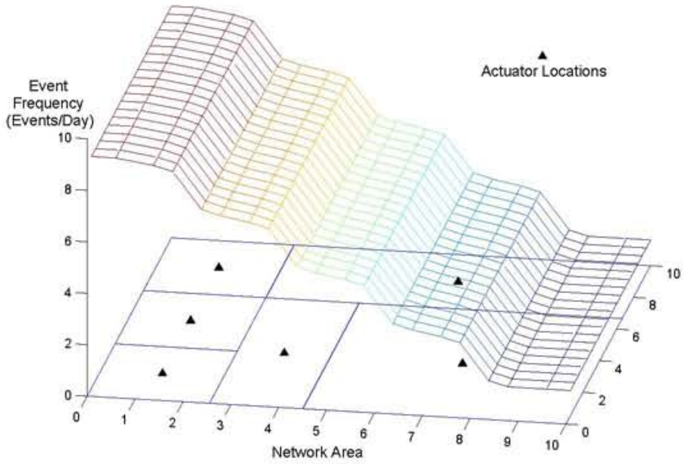

Fig. 3. Actuator Allocation with 6 Actuators.

TABLE I
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

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 4 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) [26], 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.

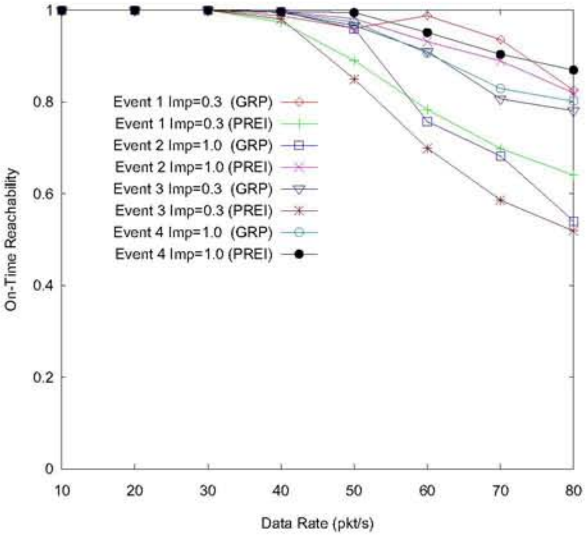

Fig. 4. On-Time Reachability.

Figure 5 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 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 6 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.

B. 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 7 and 8.

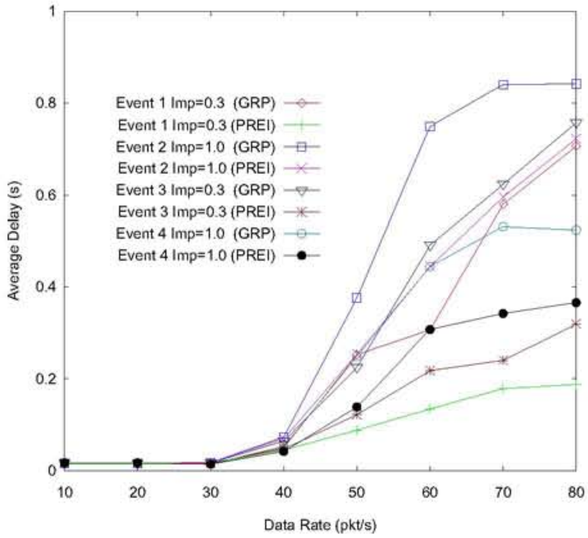

Fig. 5. Average Delay.

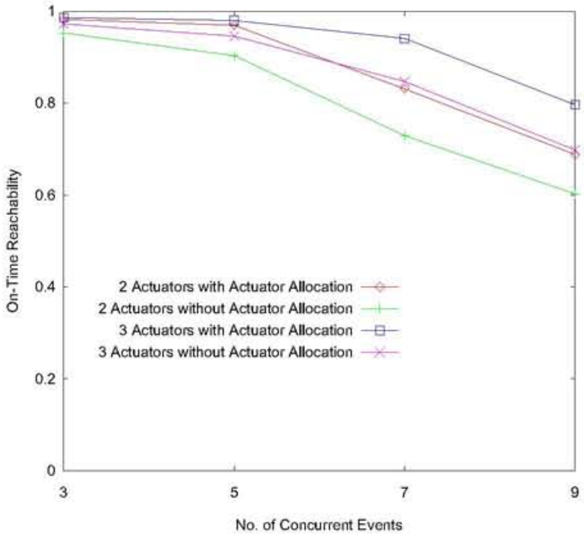

Fig. 7. On-Time Reachability with Actuator Allocation.

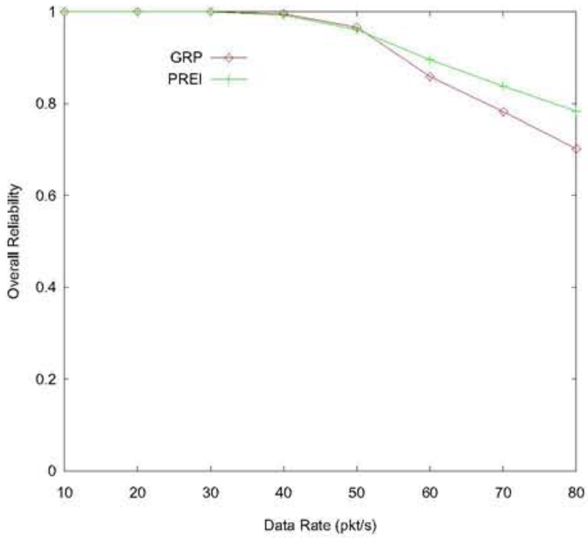

Fig. 6. Overall Reliability.

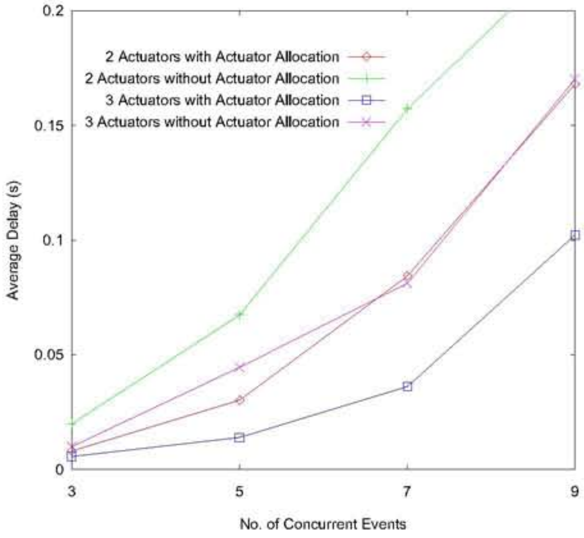

Fig. 8. Average Delay with Actuator Allocation.

Figure 7 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 7 that the reliability with actuator allocation outperforms that without 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 8 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.

VI. 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.

ACKNOWLEDGMENT

The work described in this paper was substantially supported by grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4205/04E).
J. Liu work was supported in part by a Canadian NSERC Discovery Grant 288325, an NSERC Research Tools and Instruments Grant, a Canada Foundation for Innovation (CFI) New Opportunities Grant, a a BCKDF Matching Grant, and an SFU President Research Grant.

REFERENCES

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

[2] 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.

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

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] I. Assayad, A. Girault, and H. Kalla, "A bi-criteria scheduling heuristic for distributed embedded systems under reliability and real-time constraints," in *Proc. of the IEEE DSN*, Florence, Italy, Jun 28 - Jul 1, 2004.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

[20] 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.

[21] 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.

[22] 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.

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

[24] E. C.-H. Ngai, Y. Zhou, M. R. Lyu, and J. Liu, "A delay-aware reliable event reporting framework for wireless sensor-actuator networks," *CSE Technical Report CS-TR-2006-04, The Chinese University of Hong Kong*, Mar 2006.

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

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