END-TO-END RELIABLE EVENT TRANSFER IN WIRELESS SENSOR NETWORKS

Nurcan Tezcan[1], Erdal Cayirci[2], and M. Ufuk Caglayan[1]

[1] Computer Engineering Dept., Bogazici University, 34342 Bebek, Istanbul, Turkey, {ntezcan,caglayan}@boun.edu.tr
[2] Computer Engineering Dept., Istanbul Technical University, 34111 Maslak, Istanbul, Turkey, cayirci@cs.itu.edu.tr

Abstract - In this paper, a new group of end-to-end reliable event transfer schemes is introduced for sensor networks. In these schemes, reliable event delivery is considered rather than reliable delivery of data packets, since the ultimate goal is the detection of events in sensor networks. Reliable event transfer is critical in many applications. Therefore, the need for transferring the events in a reliable way coerced us to introduce a new group of end-to-end event transfer schemes. In sensor networks end-to-end reliable event transfer schemes can be categorized into two broad classes, as acknowledgement based and non-acknowledgement based. Our new schemes introduced in this paper are in acknowledgement based class. The performance of the proposed schemes is also evaluated for various application areas by simulation.

Keywords - Sensor networks, end-to-end, reliability, event transfer, acknowledgement.

I. INTRODUCTION

Wireless sensor networks (WSN) are based on the collaborative effort of large number of sensor nodes [1]. The ultimate goal of a sensor network is the detection of specified events of interest in a sensor field. Since the detection range of sensor nodes often overlaps, the same event is usually reported by multiple sensor nodes. However, the sheer number of sensor nodes, the environmental characteristics of sensor fields and power limitation of the nodes may pose frequent unexpected loss of data packets. In some cases, all packets that report the same event may be lost. Therefore, an event may completely be lost although it is reported by multiple sensor nodes. To overcome this problem, new end-to-end event transfer schemes that fit the characteristics of sensor networks are needed. In this paper, we introduce a new group of end-to-end reliable event transfer schemes for WSN.

To the best of our knowledge, there has been limited number of works on the design of an efficient reliable transport protocol. The Reliable Multi-Segment Transport (RMST) [4] scheme is one of these, and it is designed to provide end-to-end reliable data packet transfer for directed diffusion [10]. It is a selective negative acknowledgement based protocol that has two modes: caching mode and non-caching mode. In caching mode, a number of nodes along a reinforced path, i.e., nodes along a path that directed diffusion protocol uses to convey the data to the sink, are assigned as RMST nodes. Each RMST node caches the fragments of a flow. Watchdog timers are maintained for each flow. When a fragment is not received before the timer expires, a negative acknowledgement is sent backward in the reinforced path. The first RMST node that has the required fragment along the path retransmits the fragment. Sink acts as the last RMST node and it becomes the only RMST node in the non-caching mode.

The Pump Slowly Fetch Quickly (PSFQ) scheme [3] is similar to RMST [4]. PSFQ comprises three functions: message relaying (pump operation), relay initiated error recovery (fetch operation) and selective status reporting (report operation). Every intermediate node maintains a data cache in PSFQ. A node that receives a packet checks its content against its local cache and discards any duplicates. If the received packet is new, the TTL field in the packet is decremented. Forwarding the packet is scheduled if the TTL field is greater than 0 after it is decremented and there exists no gap in the packet sequence numbers. The packets are delayed a random period between T_{min} and T_{max}, and then relayed. A node goes to fetch mode once a sequence number gap is detected. The node in fetch mode requests the retransmission of lost packets from neighboring nodes.

PSFQ and RMST schemes are designed to enhance end-to-end data packet transfer reliability. Event-to-Sink Reliable Transport (ESRT) [5] protocol is the first transport layer protocol that focuses on end-to-end reliable event transfer in WSN. In ESRT, reliable event transfer is not guaranteed but increased by controlling the event reporting frequencies of sensor nodes.

The main design issues of our schemes are collective/cooperative paradigm and energy efficiency. Proposed schemes do not incur additional overhead on the protocols in the lower layers and aim to increase reliability of event delivery with minimum energy expenditure. Characteristics of our new schemes are summarized as follows:

- They fit the factors influencing the WSN design such as scalability, hardware constraints and power consumption [1].
- They are simple and lightweight.
- They provide trade off mechanisms for various levels of reliability requirements.
- They are independent from the underlying network protocols.
- They comply with the applications known to us.

We categorize end-to-end reliable event transfer schemes

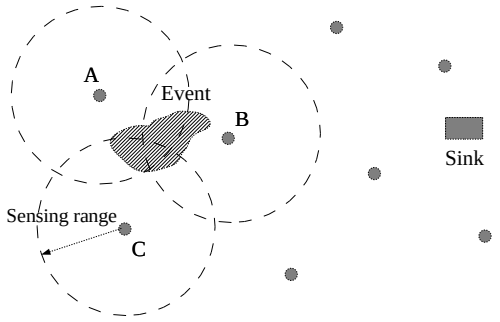

Fig. 1
Event transfer.

into two groups: *Non-Acknowledgement (NoACK) Based Schemes* and *Acknowledgement (ACK) Based Schemes*. In each group, three different schemes are presented for different application scenarios. In NoACK based schemes, we discuss three methods namely *implicit acknowledgement* [1], *event reporting frequency* [5] and *node density* [2] [7]. For the second group, we present three new schemes, which are *selective acknowledgement*, *enforced acknowledgement* and *blanket acknowledgement*. In these schemes, an acknowledgement mechanism is triggered only when an event is detected, so unnecessary acknowledgement traffic and thus energy expenditure are precluded.

The remainder of this paper is organized as follows. In Section II, end-to-end reliable event transfer schemes are introduced. We describe every scheme in detail in subsections II-A and II-B. In Section III, we evaluate the performance of our schemes and conclude our paper in Section IV.

II. End-to-End Reliable Event Transfer Schemes for WSN

In reliable event transfer approach, event is defined as the critical data generated by sensor nodes. The decision of whether the reported data are critical or not is determined by the application. Reliable event transfer schemes are designed for the reliable delivery of such critical data packets. In most cases, the same critical data is generated by more than one sensor node because sensor nodes are usually densely deployed in WSN. We emphasize that an event is successfully transferred to the sink, when the sink node receives at least one packet reporting the event. New schemes are designed to accomplish end-to-end reliability based on this approach.

For example, nodes A, B and C detect the same event in Figure 1. Shaded area shows the event region, i.e. the area covered by the event. Since all three nodes can sense this event in their ranges, all of them generate data packets reporting the same event. The end-to-end transfer of the event succeeds even if the sink receives one of these data packets.

In our schemes, we choose to implement reliability in transport layer, which traditionally aims to solve end-to-end issues. Another alternative is using MAC layer reliability where intermediate nodes take the responsibility for loss detection and recovery. Since the aim is to transfer an event to the sink node successfully rather than transferring each data packet, hop-by-hop error recovery incurs extra unacceptable cost due to the large number of sensor nodes. Moreover, hop-by-hop reliability cannot guarantee the end-to-end transfer of an event. Also note that the schemes that we propose are not connection-oriented and therefore they are different from conventional end-to-end protocols.

In next two sections, we explain the NoACK based schemes and ACK based schemes. NoACK based schemes are collection of alternative methods of increasing reliability without waiting end-to-end acknowledgement. In contrast, ACK based schemes make use of acknowledgements.

A. Non-Acknowledgement Based Schemes

In this section, the alternative schemes where sensor nodes do not wait an end-to-end acknowledgement are presented. Three different schemes are introduced: *implicit acknowledgement* [1], *event reporting frequency* [5] and *node density* [2] [7] based end-to-end reliable event transfer schemes.

1) Implicit Acknowledgement: One method for reliable event transfer in sensor networks is *implicit acknowledgement*. Implicit acknowledgement makes use of the broadcast characteristic of the wireless channel. Sensor nodes monitor packets sent by neighbors. When a data packet sent by them is repeated by their gradient, i.e., the gradient nodes are supposed to repeat the packets to convey them to the sink, this can be accepted as a hop-by-hop acknowledgement of the sent packet. Since the method does not need a separate acknowledgement packet, its only overhead is the additional energy consumption due to listening to gradient nodes.

One may argue that, this is not an end-to-end reliability scheme but a hop-by-hop technique. However this technique increases the level of end-to-end event transfer reliability, therefore we will also examine its impact on the end-to-end event reliability in this paper.

2) Event Reporting Frequency: This method is used in the ESRT (Event to Sink Reliable Transport) protocol [5] which is based on the event-to-sink reliability model. Level of reliable end-to-end delivery is controlled by increasing or decreasing the event reporting frequency. As the reporting frequency is higher, the number of packets generated by a sensor node increases. This approach decreases the probability that the reported event is lost.

One other point is that reporting frequency can be increased until a certain point, beyond which the reliability drops. This is because the network is unable to handle the

increased injection of data packets and data packets are dropped due to congestion. The details about the scheme can be found in [5].

3) Node Density: In sensor networks, there are usually multiple nodes that have overlapping sensing regions. Hence, it is possible that multiple nodes collaborate to detect the same event. The number of nodes that report the same event has an impact on the end-to-end event transfer reliability. If the number of sensor nodes in critical regions or the number of nodes involved in reporting an event is managed by a network management protocol, the end-to-end event transfer reliability rate can also be controlled.

In many applications sensor nodes are randomly deployed in inaccessible terrains [1]. Due to this characteristic of WSN, required node density may not be obtained by physically adding new nodes. Instead, higher end-to-end reliable event transfer rate can be achieved by increasing the number of nodes involved in a sensing task. Task set concept, introduced in [7], can be used to manage the number of nodes involved in a task. A task set [7] consists of a group of sensor nodes, which are queried by the same task specification. A practical distributed algorithm is proposed for creating task sets in [7]. The higher number of nodes in a task set indicates higher accuracy and reliability.

Task sets can be managed by sensor network management protocols such as Data Aggregation and Dilution by Modulus Addressing (DADMA) [6] and sensor field queries by using quad-tree based dynamic clusters and task set scheme [7]. Therefore, we already have some practical techniques to control the number of nodes involved in a query. It is possible to tune the level of end-to-end event reliability based on our requirements by using these techniques.

B. Acknowledgement Based Schemes

Although the acknowledgement mechanism is a traditional way of achieving end-to-end reliability, it may not be viable for many WSN applications due to the following reasons:

- Most of the WSN applications have very stringent energy constraints. Therefore, overhead of the acknowledgement packets may not be justifiable.
- Since some of the events reported by sensor nodes may not be as critical as others, generating acknowledgement for all packets received may incur unnecessary costs.
- Since many sensor nodes may report the same event, acknowledging all of them by a single acknowledgement may be more effective.

In this section we introduce our new schemes based on the acknowledgement mechanism: *selective acknowledgement, enforced acknowledgement, and blanket acknowledgement.*

1) Selective Acknowledgement: Since WSN consists of thousands of densely deployed sensor nodes, waiting an acknowledgement for each data packet may not be viable. Instead, each sensor node activates the acknowledgement mechanism when it detects critical data.

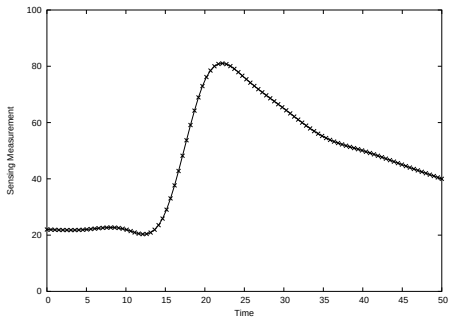

Fig. 2
Example measurements of a sensor node

There are various ways to determine whether a data packet carries critical data or not. One approach is using a threshold value. Sensor nodes and the sink node can come to an agreement on a threshold value before deployment. Threshold value depends on the application. Then, sensor node decides whether the measurement is critical or not by using the agreed threshold value.

A data packet that carries event information is called critical. An example for critical data is given in Figure 2, where a time series of measurements made by a temperature sensor is illustrated. In Figure 2 the sensed data are almost the same up to t=20. Unless the reported temperature does not change over the threshold value, the reported data may be accepted as non-critical. In our example, even if one of the data packets reporting the temperature is lost until t=20, the sink can figure out that the temperature does not change. Assume that the threshold value is 45 for this example. At t=20, the difference between the reported temperature and the temperature reported in the previous packet exceeds the threshold and therefore the data packet at t=20 is accepted as critical.

The acknowledgement mechanism is triggered for the critical data packets. Each data packet received by the sink node is compared with the threshold value and categorized as critical or not. If a critical data packet is received, an acknowledgement packet is sent to the sensor node immediately. If the source sensor node does not receive an acknowledgement packet within a predetermined timeout period, i.e., retransmission time, it retransmits the packet. Thus, the acknowledgement mechanism controls the event transfer reliability with a minimum overhead. We explain how to determine an appropriate time-out period later in this section.

2) Enforced Acknowledgement: In enforced acknowledgement, the idea is almost the same as the selective acknowledgement. Not all of the packets but the data packets that carry critical data need to be acknowledged. However, sink node does not need to figure out whether a data packet is critical or not. Instead, the source sensor node decides if

a data packet is critical or not and marks the packets that carry critical data. The sink node is supposed to acknowledge the marked packets. The source nodes retransmit unacknowledged marked packets after a predetermined retransmission period.

Comparing to the enforced acknowledgement, the number of retransmissions is slightly lower in the selective acknowledgement because if the packet that follows the critical packet also exceeds the threshold, the sink node detects the second packet as the critical data and acknowledges it when the first data packet is lost. For example, if the packet at t=20 is lost, the sink acknowledges the packet at t=25 in the example given in Figure 2. However, memory and computational requirements for the sink is higher in selective acknowledgement since the last packet of every node should be stored in the sink. On the other hand, there is no need for any additional memory for the enforced acknowledgement.

The selective acknowledgement may better fit one WSN application, and the enforced acknowledgement may better perform on another application. When the memory space available in the sink is enough to store the last sent measurements by every node and the energy constraints at nodes are very stringent, selective acknowledgement can be preferred. If the memory space is not enough to store the last measurements, then the enforced acknowledgement should be used. The enforced acknowledgement also better catch the time that the critical data, i.e., the event, is detected by the sensor node because the first critical data packet needs to be acknowledged in the enforced acknowledgement scheme.

3) Blanket Acknowledgement: Multiple sensor nodes reporting the same event may be acknowledged by a single acknowledgement packet. In *blanket acknowledgement*, a single acknowledgement packet is broadcast for an event. A sensor node that receives an acknowledgement packet for the event, accepts that the data packet generated for the same event has been received by the sink successfully. Since the reception of the event is more important than the reception of every packet, this is an efficient acknowledgement mechanism for WSN.

An application for blanket acknowledgement is the sensor networks for disaster relief operations where sensor nodes are responsible to report life-signs from humans trapped under rubble. When the sink acknowledges the presence of a life-sign, sensor nodes do not need to worry whether their report is received or not. The important point here is that the sink must know that there is a live human under rubble.

One practical way to implement blanket acknowledgement is to use sectoral sweepers [8] where the sink broadcasts all of the nodes in a region at a single hop. Therefore, the acknowledgements are not flooded but overheard by every node when broadcasted by the sink. Blanket acknowledgement can be used also in conjunction with selective and enforced acknowledgements to broadcast acknowledgement packets.

4) Timeout Mechanism for Acknowledgement Based Schemes: Since topology of sensor networks changes frequently and sink does not acknowledge every packet but only the selected packets, it is not feasible to determine timeout periods dynamically based on the time elapsed between sensed data packet transmissions and acknowledgement packet receptions. The timeout period is based on applications, event types, event frequencies and memory available in sensor nodes.

For our acknowledgement based schemes, it may be preferred waiting for the maximum timeout period t_{max} that can be tolerated by the application before retransmission because sink may acknowledge the same event when it is reported by another node or in another time series as explained in Section II-B. Therefore, even if the packet that carries an event is lost, its retransmission may not be needed. The maximum timeout period t_{max} depends on the application. It can be as high as few minutes in applications such as sensor networks for disaster relief operations management, and as low as few seconds in applications such as intrusion detection. Event frequencies and memory available in sensor nodes also have an impact on the timeout period because the events that need to be acknowledged must be stored in the source nodes until they are acknowledged. If t_{max} is too long to store all of the events transfered during t_{max}, then some events may be lost before acknowledgement. Therefore, we introduce the following algorithm for determining retransmission time for an event:

```
if(numberofeventsinthelist > listsize-n)
 for (all events in the list)
  if(eventtime >= t_max ||eventtime >= t_avg)
   retransmit(event)
```

When a node transmits an event, it inserts the transmitted event into the unacknowledged event list, and starts a timer for the event. If there is enough space only for less than *n* more events in the list after this insertion, then all the nodes that have timer values larger than t_{max} or t_{avg} are retransmitted. As long as there is enough space for more than *n* events, only the events that have timer values larger than t_{max} are retransmitted. All acknowledged events are removed from the list. *n* is a parameter determined based on the expected event frequency, the maximum timeout period t_{max} and the list size. The value for the parameter t_{avg} is determined by using an approach similar to TCP timeout mechanism:

$$t_{avg} = \alpha t_{avg} + (1 - \alpha) t_{ack} \quad (1)$$

where α is the weight and t_{ack} is the acknowledgement time for the last event. t_{avg} can be initially assigned an expected value or t_{max}.

III. Performance Evaluation

In this section we present the simulation platform and the results from our experiments. Our new acknowledgement schemes were implemented in NESLsim [14], which is a discrete event sensor networks simulation tool based on

Table 1
Simulation Parameters

Number of sensor nodes	500
MAC	TDMA
Routing protocol	Directed diffusion
Directed diffusion reinforcement parameter	1
Total simulation time	3600 sec
Simulation time unit	10e-6
Data packet size	400 bit
ACK packet size	100 bit
MAC/PHY header	8 bit
Radio bitrate	20 Kbps
Bit error rate (BER)	1e-5

PARSEC (Parallel Simulation Environment for Complex systems) [15].

In our simulations we randomly deploy our sensors in a square shape sensor field according to uniform distribution. The size of the sensor field is determined according to the node density, ρ, given as:

$$\rho = \frac{\pi r^2}{a^2} \tag{2}$$

where r is the wireless transmission range of the node, and a^2 is the area of the sensor field. Our node density parameter ρ indicates the average number of nodes in the sensing range of a sensor node. For the base scenario, *500* sensor nodes are placed to a terrain of *168 m * 168 m*. Sensor nodes are not mobile. The other parameters related to our simulation are given in Table 1.

An event based random traffic pattern is generated for the base scenario. Number of events that will be observed during the simulation period is taken as a parameter. Each event has the following attributes: *event time, radius, duration, and location*. The event time and the location are randomly distributed according to uniform distribution. Thus, critical data packets are sent during the event duration which is also a parameter. In our simulation, events take place in circular regions so that more than one sensor detects the same event in most cases. Each node has a transmission range of *30m (1̃00 feet)* and a sensing period of *20s*. All results are averaged over 100 simulation runs.

Our performance metrics are *Successful Event Delivery Ratio (SEDR), Acknowledgement Overhead Ratio (AOR)* and *Retransmission Overhead Ratio (ROR)*. *SEDR* is a frequently used metric that gives the ratio of the events that are reported to the sink node successfully over all generated events. *AOR* is a classical metric that shows the portion of the consumed bandwidth to make the proposed schemes work. It is determined by calculating number of sent, received and relayed acknowledgement packets and the ACK packet size. *ROR* is another performance metric that gives the proportion of the amount of overhead for retransmitted packets.

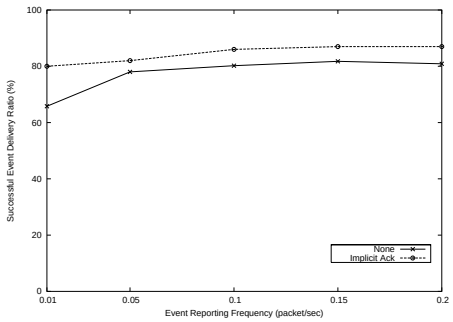

Fig. 3
Event Reporting Frequency vs successful event delivery ratio

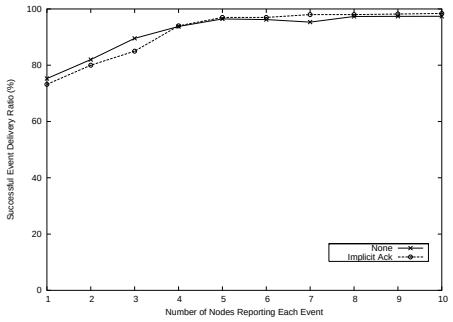

Fig. 4
Number of nodes reporting an event vs successful event delivery ratio

Figure 3 shows the performance of the *event reporting frequency based scheme* also for implicit acknowledgement. Event reporting frequency is the number of packets sent in one second. It is adjusted by increasing or decreasing the sensing period of the sensor node. When we increase the sensing period, the number of data packets sent for an event increases. Thus, as shown in the Figure 3, SEDR increases up to a certain ratio, after which increasing the frequency does make any impact in both scenarios.

In Figure 4, the impact of the node density based scheme on SEDR is depicted. Node density is controlled by defining task sets for each event. As shown in Figure 4, SEDR increases by increasing the number of nodes in the task set. In our experiments, we observe that the impact of increasing node density is the same as the impact of event reporting frequency. After a certain point, increasing node density does not affect the end-to-end event transfer reliability.

Table 2 shows the SEDR of the schemes derived for the same scenario. In the scenario, node density is chosen as 10 and the event radius is 10m. In order to emphasize the role of our schemes, we take *20s* as event duration and sensing

Table 2
Successful Event Delivery Ratio

Scheme	Successful EDR %
None	76.22
Selective Acknowledgement	94.18
Enforced Acknowledgement	93.33

Table 3
Acknowledgement Overhead Ratio

	ACKs. Sent	ACKs. Relayed	ACKs. Received
Selective	9.21	9.22	9.55
Enforced	8.40	8.01	8.8
Blanket	1.40	0	1.40

Table 4
Retransmission Overhead Ratio

Scheme	Retransmitted Packets %
Selective Acknowledgement	6.0
Enforced Acknowledgement	13.10
Blanket Acknowledgement	11.02

period. Therefore, each event can be observed only once by every node that can detect it. As shown in Table 2, selective acknowledgement and enforced acknowledgement increase our successful end-to-end event delivery ratio more than 20% in the average.

In Table 3, performance of the schemes is evaluated for AOR where we use the same scenario. The results show the ratio of total number of bits transferred for acknowledgement packets. The overhead for selective and enforced acknowledgements are approximately 10%. This indicates that the overhead for more than 20% increase in the end-to-end reliability is around 10%. Additionally, it is obviously seen that blanket acknowledgement is more advantageous than others. The major reason is because of

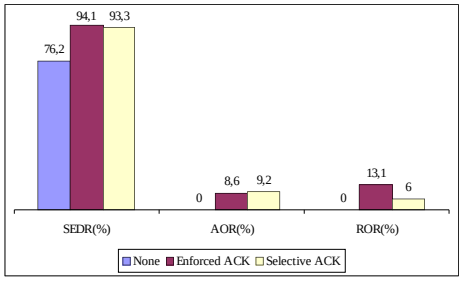

Fig. 5
Enforced and Selective Ack. Performances

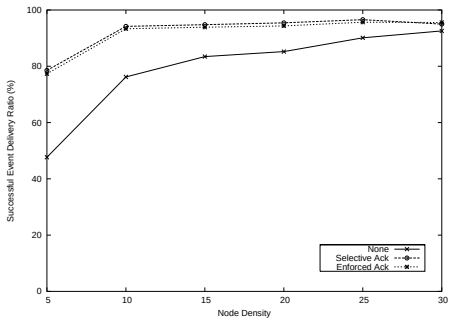

Fig. 6
Node density vs successful event delivery ratio

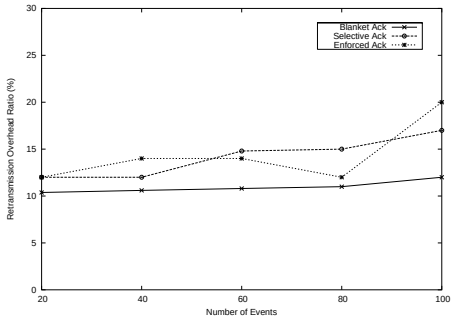

Fig. 7
Number of Events vs Retransmission Overhead Ratio

the single hop acknowledgement packet broadcast. Please note that selective and enforced acknowledgements can be used in conjunction with blanket acknowledgement which implies further decrease in the cost of our acknowledgement schemes.

In order to come up with a comparable evaluation of performance of the proposed schemes, retransmission overheads are listed in Table 4. The analysis is made for the same scenario.

In Figure 5 we depict the results that we obtained from experiments in Table 2, 3, 4 all together. As shown selective acknowledgement performs better. The reason for this is explained in Section II-B in detail. Please note that there are some additional memory and computational requirements in the sink for selective acknowledgement, but there is no such additional needs for enforced acknowledgement.

In Figure 6, SEDR with respect to node density, which is the number of neighboring nodes in the sensing range of a sensor node, is shown. The SEDR increases as the node density gets higher. This behavior is related to the fact that more nodes detect the same event. Therefore, event lost probability decreases because more packets are sent for the same event. The results are close to each other for our ACK

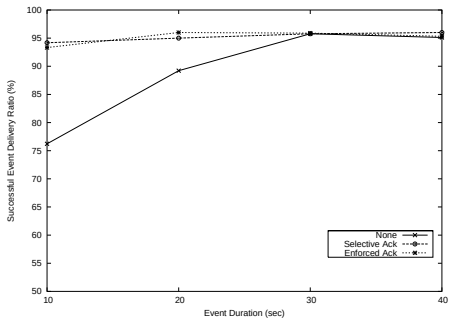

Fig. 8
Event duration vs successful event delivery ratio

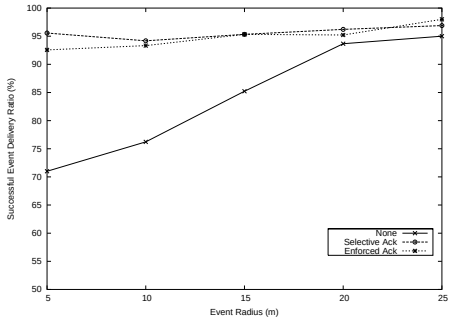

Fig. 9
Event radius vs successful event delivery ratio

based schemes, which perform in the orders of magnitude better than the case where acknowledgements are not used.

The ratio of retransmitted packets can be seen for various number of events in Figure 7. Retransmission ratio does not change rapidly by the event rate. RORs are slightly higher than the results in Table 4. This indicates that proposed acknowledgement schemes do not bring higher retransmission overhead as the event rate increases.

Figure 8 and 9 show SEDR for varying event duration and event radius. SEDR is sensitive to event duration and event radius. When event duration and radius are increased, the number of nodes that detects the same event increases, which makes the successful end-to-end event delivery ratio higher.

IV. Conclusion

In this paper, a group of end-to-end reliable event transfer schemes is presented for WSN. These schemes comply with existing network layer protocols such as [10]. Based on some reference applications various schemes are proposed to satisfy the end-to-end reliability requirements of the applications. We also evaluate the performance of our schemes in the paper.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey", *Elsevier Computer Networks*, vol. 38, no. 4, pp. 393–422, March 2002.

[2] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "Infrastructure Tradeoffs for Sensor Networks", *In Proc. of WSNA 2002*, Atlanta, GA, USA, September 2002.

[3] C.Y. Wan, A.T. Campbell, L. Krishnamurthy, "PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks", *Proc. ACM Int.Workshop on Wireless Sensor Networks and Applications*, pp. 1–11, Georgia, 2002.

[4] F. Stann, J. Heideman, "RMST: Reliable Data Transport in Sensor Networks", *Proc. First IEEE Int. Workshop on Sensor Network Protocols and Applications*, pp. 102–113, Anchorage, USA, May 2003.

[5] Y. Sankarasubramaniam, O.B. Akan, and I.F. Akyildiz, "ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks", *to appear in ACM MobiHoc Conference*, June 2003.

[6] E. Cayirci, "Data Aggregation and Dilution by Using Modulus Addressing in Wireless Sensor Networks", *IEEE Communications Letters*, August 2003.

[7] C. Cimen, E. Cayirci, and V. Coskun, "Querying Sensor Field By Using Quadtree Based Dynamic Clusters And Task Sets", *Proc. MilCom*, 2003.

[8] A. Erdogan, E. Cayirci, and V. Coskun, "Sectoral Sweepers for Task Dissemination and Location Estimation in AdHoc Sensor Networks", *Proc. MilCom*, 2003.

[9] W.R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks", *Proc.ACM MobiCom '99*, pp. 174–85, Seattle, WA, 1999.

[10] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", *Proc. IEEE/ACM MOBICOM*, pp. 56–67, Boston, USA, August 2000.

[11] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy Efficient Communication Protocols for Wireless Micro-Sensor Networks", *Proc. Hawaian Int. Conf. on Systems Science*, vol. 8, January 2000.

[12] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie, "Protocols for Self-Organization of a Wireless Sensor Network", *IEEE Pers. Communications*, pp. 16–27, October 2000.

[13] S. Ganeriwal, V. Tsiatsis, and C. Schurgers, "NESLsim: A parsec based simulation platform for sensor networks", *http://www.ee.ucla.edu/saurabh/NESLsim*, 2002.

[14] R. Bagrodia, R. Meyer, M. Takai, Y.A. Chan, X.Zeng, J. Marting, H.Y. Song, "Parsec:A Parallel Simulation Environment for Complex Systems", *Computer*, Vol.31, No.10, pp.77-85, October 1998.