Event-to-Sink Reliable Transport in Wireless Sensor Networks

Özgür B. Akan, *Member, IEEE,* and Ian F. Akyildiz, *Fellow, IEEE*

***Abstract*—Wireless sensor networks (WSNs) are event-based systems that rely on the collective effort of several microsensor nodes. Reliable event detection at the sink is based on collective information provided by source nodes and not on any individual report. However, conventional end-to-end reliability definitions and solutions are inapplicable in the WSN regime and would only lead to a waste of scarce sensor resources. Hence, the WSN paradigm necessitates a collective *event-to-sink* reliability notion rather than the traditional end-to-end notion. To the best of our knowledge, reliable transport in WSN has not been studied from this perspective before.**

In order to address this need, a new reliable transport scheme for WSN, the event-to-sink reliable transport (ESRT) protocol, is presented in this paper. ESRT is a novel transport solution developed to achieve reliable event detection in WSN with minimum energy expenditure. It includes a congestion control component that serves the dual purpose of achieving reliability and conserving energy. Importantly, the algorithms of ESRT mainly run on the sink, with minimal functionality required at resource constrained sensor nodes. ESRT protocol operation is determined by the current network state based on the reliability achieved and congestion condition in the network. This self-configuring nature of ESRT makes it robust to random, dynamic topology in WSN. Furthermore, ESRT can also accommodate multiple concurrent event occurrences in a wireless sensor field. Analytical performance evaluation and simulation results show that ESRT converges to the desired reliability with minimum energy expenditure, starting from any initial network state.

***Index Terms*—Congestion control, energy conservation, event-to-sink reliability, reliable transport protocols, wireless sensor networks.**

I. Introduction

THE Wireless Sensor Network (WSN) is an event-driven paradigm that relies on the collective effort of numerous microsensor nodes. This has several advantages over traditional sensing including greater accuracy, larger coverage area and extraction of localized features. In order to realize these potential gains, it is imperative that desired event features are reliably communicated to the sink.

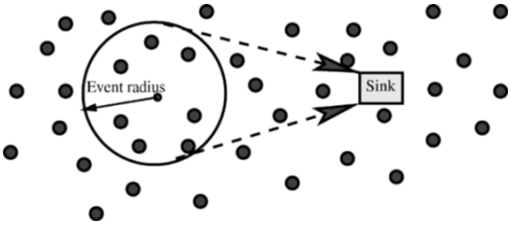

Fig. 1. Typical sensor network topology with event and sink. The sink is only interested in collective information of sensor nodes within the event radius and not in their individual data.

To accomplish this, a reliable transport mechanism is required in addition to robust modulation and media access, link error control and fault tolerant routing. The functionalities and design of a suitable transport solution for WSN are the main issues addressed in this paper.

The need for a transport layer for data delivery in WSN was questioned in a recent work [12] under the premise that data flows from source to sink are generally loss tolerant. While the need for end-to-end reliability may not exist due to the sheer amount of correlated data flows, an event in the sensor field needs to be tracked with a certain accuracy at the sink. Hence, unlike traditional communication networks, the sensor network paradigm necessitates an *event-to-sink* reliability notion at the transport layer. This is a truly novel aspect of our work and is the main theme of the proposed Event-To-Sink Reliable Transport (ESRT) protocol for WSN. Such a notion of collective identification of data flows from the event to the sink is illustrated in Fig. 1.

ESRT is a novel transport solution that seeks to *achieve reliable event detection with minimum energy expenditure and congestion resolution*. It has been tailored to match the unique requirements of WSN. We emphasize that ESRT has been designed for use in typical WSN applications involving event detection and signal estimation/tracking, and not for guaranteed end-to-end data delivery services. Our work is motivated by the fact that the sink is only interested in reliable detection of event features from the collective information provided by numerous sensor nodes and not in their individual reports. This notion of event-to-sink reliabidescribility distinguishes ESRT from other existing transport layer models that focus on end-to-end reliability. To the best of our knowledge, reliable transport in WSN has not been studied from this perspective before.

In this paper, we have also extended our work in [6] by enhancing ESRT protocol in order to accommodate the scenarios where multiple concurrent events occur in the wireless sensor field. Such enhancement is significant since the data flows generated by the multiple events occurring simultaneously may not

Manuscript received August 20, 2003; revised June 17, 2004, and October 12, 2004; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor N. Shroff. This work was supported by the National Science Foundation under Contract ECS-0225497. An earlier version of this paper appeared in the *Proceedings of the ACM MOBIHOC 2003*, Annapolis, MD, June 2003.

Ö. B. Akan was with the Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA. He is now with the Department of Electrical and Electronics Engineering, Middle East Technical University, 06531 Ankara, Turkey (e-mail: akan@eee.metu.edu.tr).

I. F. Akyildiz is with the Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: ian@ece.gatech.edu).

Digital Object Identifier 10.1109/TNET.2005.857076

be always isolated in the WSN. Thus, uncoordinated protocol actions may fail to achieve required event-to-sink transport reliability and to resolve congestion for individual event flows because of the interaction between these flows in the network. Therefore, it is necessary to accurately capture the event occurrence situation in the network and accordingly act to assure the event-to-sink reliability with minimum energy expenditure for all of the multiple concurrent events in the sensor field.

The remainder of the paper is organized as follows. In Section II, we present a review of related work in transport protocols, both in WSN and other communication networks, and point out their inadequacies. We formally define the transport problem in WSN in Section III. The operation of ESRT is described in detail in Section IV and a pseudo-algorithm is also presented. In Section V, we explain how the default ESRT protocol operation is extended to accommodate the scenarios where multiple concurrent events occur in the wireless sensor field. ESRT performance analysis and simulation results are presented in Section VI. Finally, the paper is concluded in Section VII.

II. Related Work

In [12], the PSFQ (Pump Slowly, Fetch Quickly) mechanism is proposed for reliable retasking/reprogramming in WSN. PSFQ is based on slowly injecting packets into the network, but performing aggressive hop-by-hop recovery in case of packet losses. The pump operation in PSFQ simply performs controlled flooding and requires each intermediate node to create and maintain a data cache to be used for local loss recovery and in-sequence data delivery. Although this is an important transport layer solution for WSN, it is applicable only for strict sensor-to-sensor reliability and for purposes of control and management in the reverse direction from the sink to sensor nodes. Hence, the use of PSFQ for the forward direction can lead to a waste of valuable resources. In addition to this, PSFQ does not address packet losses due to congestion.

In [10], the Reliable Multi-Segment Transport (RMST) protocol is proposed to address the requirements of reliable data transport in WSN. RMST is mainly based on the functionalities provided by *directed diffusion* [2]. Furthermore, RMST utilizes in-network caching and provides guaranteed delivery of the data packets generated by the event flows. However, event detection/tracking does not require guaranteed end-to-end data delivery since the individual data flows are correlated loss tolerant. Moreover, such guaranteed reliability via in-network caching may bring significant overhead for the sensor networks with power and processing limitations.

In contrast, ESRT is based on an event-to-sink reliability model and provides reliable event detection without any intermediate caching requirements. ESRT also seeks to achieve the required event detection accuracy using minimum energy expenditure and has a congestion control component.

On the other hand, transport solutions in other wireless networks mainly focus on reliable data transport following end-to-end TCP semantics and are proposed to address the challenges posed by wireless link errors and mobility. The primary reason for their inapplicability in WSN is their notion of end-to-end reliability. Furthermore, all these protocols bring considerable memory requirements to buffer transmitted packets until they are ACKed by the receiver. In contrast, sensor nodes have limited buffering space (<4 KB in MICA motes [5]) and processing capabilities. Hence, there is a need for a novel transport mechanism in WSN that emphasizes on collective reliability, resource efficiency and simplicity.

III. The Reliable Transport Problem in WSN

In the preceding discussions, we introduced the notion of event-to-sink reliability in WSN and pointed out the inapplicability of existing transport solutions. Before proceeding to discuss our proposed Event-To-Sink Reliable Transport (ESRT) protocol, we formally define the reliable transport problem in WSN in this section. We also introduce the evaluation environment used in our studies and set the stage for ESRT by defining five characteristic reliability regions.

A. Problem Definition

Consider typical WSN applications involving the reliable detection and/or estimation of event features based on the collective reports of several sensor nodes observing the event. Let us assume that for reliable temporal tracking, the sink must decide on the event features every τ time units. Here, τ represents the duration of a decision interval and is fixed by the application. At the end of each decision interval, the sink decides based on reports received from sensor nodes during that interval. The specifics of such a decision making process are application dependent and beyond the scope of our paper.

The least we can assume is that the sink derives an event reliability indicator r_i at the end of the decision interval i. Note that r_i must be calculated only using parameters available at the sink. Hence, notions of throughput/goodput, which are based on the number of source packets sent out are inappropriate in our case.

We measure the reliable transport of event features from source nodes to the sink in terms of the number of received data packets. Regardless of any application-specific metric that may actually be used, the number of received data packets is closely related to the amount of information acquired by the sink for the detection and extraction of event features. Hence, this serves as a simple but adequate event reliability measure at the transport level. The observed and desired event reliabilities are now defined as follows:

Definition 1: The *observed event reliability*, r_i, is the number of received data packets in decision interval i at the sink.

Definition 2: The *desired event reliability*, R, is the number of data packets required for reliable event detection. This is determined by the application.

If the observed event reliability, r_i, is greater than the desired event reliability, R, then the event is deemed to be reliably detected. Else, appropriate action needs to be taken to achieve the desired event reliability, R.

Note also that we assume that as long as sensor nodes are within the coverage area and hence have readings of the event features, they packetize their readings and send them to the sink.

While the information read and packetized by each sensor may differ based on their relative locations to the event center, all of the packets received at the sink are used to calculate the observed event transport reliability, r. Any possible inaccuracy in sensor readings is assumed to be addressed by the sensor application while the actual decision on the event features is made using the data received at the sink.

With the above definition, r_i can be computed by stamping source data packets with an event ID and incrementing the received packet count at the sink each time the ID is detected in decision interval i. Note that this does not require individual identification of sensor nodes. Further, we model any increase in source information about the event features as a corresponding increase in the reporting rate, f, of sensor nodes.

Definition 3: The *reporting frequency rate* f of a sensor node is the number of packets sent out per unit time by that node.

Definition 4: The *transport problem in WSN* is to configure the reporting rate, f, of source nodes so as to achieve the required event detection reliability, R, at the sink with minimum resource utilization.

The main rationale behind such *event-to-sink reliability* notion is that the data generated by the sensors are temporally correlated which tolerates individual packets to be lost to the extent where the distortion, D_i, observed when the event features are estimated at the sink does not exceed a certain distortion bound, i.e., $D_{\max}$. The reporting frequency f can be attributed to the sampling rate, the number of quantization levels, the number of sensing modalities, etc. Hence, the reporting frequency rate f controls the amount of traffic injected to the sensor field while regulating the number of correlated samples taken from the phenomenon. This, in turn, affects the observed event distortion, i.e., event detection reliability.

In fact, the observed event estimation distortion D at the sink in a decision interval of τ has been derived as a function of reporting frequency rate f in [11]. Here, an event signal $s(t)$ is assumed to be a Gaussian random process with $\mathcal{N}(0, \sigma_s^2)$, and the sink is interested in finding the expectation of the signal $s(t)$ over the decision interval τ, i.e., $S(\tau)$. Assuming the observed signal $s(t)$ is wide-sense stationary (WSS) and with the following definitions: $E\{S[n]\} = 0$; $E\{(S[n])^2\} = \sigma_S^2$; $E\{S[n]S[m]\} = \sigma_S^2 \hat{\rho}_S(n,m)$; $E\{s(t)s(t+\delta)\} = \sigma_S^2 \rho_S(\delta)$ where $\hat{\rho}_S(n,m) = \rho_S(|m-n|/f) = e^{-|m-n|/\theta}$ is the covariance function that depends on the time difference between signal samples, i.e., n and m, and the covariance coefficient θ; the distortion function is obtained as [11]

$$\begin{aligned} D(f) = \sigma_S^2 + \frac{\sigma_S^4}{\tau f(\sigma_S^2+\sigma_N^2)} &+ \frac{\sigma_S^6}{\tau^2 f^2 (\sigma_S^2+\sigma_N^2)^2} \sum_{k=1}^{\tau f} \sum_{l \neq k} e^{-\frac{\left(\frac{|k-l|}{f}\right)}{\theta}} \\ &- \frac{2\sigma_S^4 \theta}{\tau^2 f(\sigma_S^2+\sigma_N^2)} \sum_{k=1}^{\tau f} \left(2 - e^{-\frac{k}{f\theta}} - e^{-\frac{\left(\tau - \frac{k}{f}\right)}{\theta}} \right). \end{aligned} \tag{1}$$

As observed from (1), the distortion D observed in the estimation of the signal S being tracked depends on the reporting frequency rate f used by the sensor nodes sending their readings to the sink in the decision interval i. The variation of the observed event distortion D at the sink is shown by plotting (1) for varying reporting frequency rate f. It is observed from (1) and Fig. 2 that D decreases with increasing f. This is because the number of samples received in a decision interval i increases with increasing f conveying more information to the sink from the event area. Note that after a certain reporting frequency rate f, D cannot be further reduced. Therefore, a significant energy saving can be achieved by selecting small enough f which achieves a certain event distortion bound, i.e., the desired event reliability objective R, and does not lead to an overutilization of the scarce sensor resources. This is one of the main motivations behind the ESRT protocol which aims the reliable event transport with minimum energy expenditure as will be discussed in Section IV.

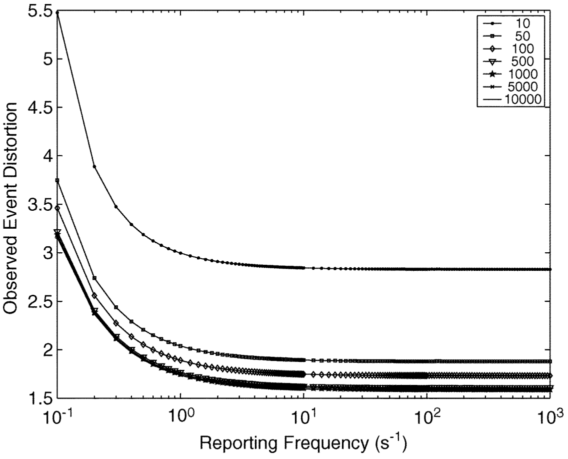

Fig. 2. Observed event distortion for varying reporting frequency f (for different covariance coefficient values, i.e., $10 \leq \theta \leq 10\,000$) [11].

On the other hand, any f chosen arbitrarily small to achieve a certain distortion bound may not necessarily achieve the desired distortion level and hence assure the event transport reliability. This is mainly because all of the sensor samples generated with this chosen reporting frequency may not be received because of packet losses in the sensor network due to link errors and network disconnectivity. Similarly, as very high values of f do not bring any additional gain in terms of observed event distortion as shown in Fig. 2; on the contrary, it may endanger the event transport reliability by leading to congestion in the sensor network. Therefore, it is imperative to efficiently control the reporting frequency rate f so that the event features are reliably transported without leading to congestion and hence with minimum energy consumption. This is the main problem that ESRT addresses for reliable event transport in wireless sensor networks as explained in Section IV.

B. Evaluation Environment

In order to study the relationship between the observed event reliability r at the sink and the reporting frequency rate f of sensor nodes, we developed an evaluation environment using *ns*-2. The parameters used in our study are listed in Table I.

Two hundred sensor nodes were randomly positioned in a 100×100 sensor field and the randomly created topology does not vary. However, note that the sensor nodes may die due to energy depletion leading to variations in the overall topology.

TABLE I
NS-2 SIMULATION PARAMETERS

Area of sensor field	100x100 m^2
Number of sensor nodes	200
Radio range of a sensor node	40 m
Network density	100 *neighbor/node*
Packet length	30 bytes
IFQ length	65 packets
Transmit Power	0.660 W
Receive Power	0.395 W
Decision interval (τ)	10 sec

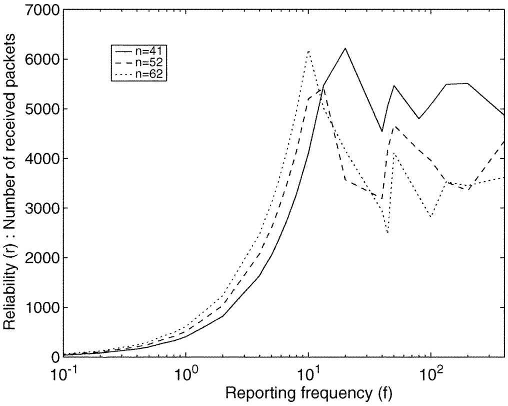

Fig. 3. Effect of varying the reporting rate, f, of source nodes on the event reliability, r, observed at the sink. The number of source nodes is denoted by n.

TABLE II
EVENT CENTERS FOR THE THREE CURVES WITH n = 41, 52, 62 IN FIG. 3

Number of source nodes	**Event Center** (X_{ev},Y_{ev})
41	(88.2,62.8)
52	(32.6,79.3)
62	(39.2,58.1)

Node parameters such as radio range and IFQ (buffer) length were carefully chosen to mirror typical sensor mote values [5]. One of these nodes was chosen as the sink to which all source data were sent. Event centers (X_{ev}, Y_{ev}) were randomly chosen and all sensor nodes within the event radius behave as sources for that event. In order to communicate source data to the sink, we employed a simple CSMA/CA based MAC protocol and Dynamic Source Routing (DSR) [3]. The impact of using other routing protocols on the achieved goodput behavior with reporting period was shown to be insignificant. Hence, it is reasonable to assume that the r versus f behavior and ESRT performance are insensitive to the underlying routing protocol.

The results of our study are shown in Fig. 3 for the number of source nodes n = 41, 52, 62. Note that each of these curves was obtained by varying the reporting rate f for a certain event center (X_{ev}, Y_{ev}) and the corresponding number of senders n. These values are tabulated in Table II. For each value of the reporting frequency rate f, we run five simulations and take the average of the measured event reliability values, i.e., η. The event radius was fixed throughout at 30 m.

We make the following observations from Fig. 3:

1) The event reliability, r, shows a linear increase (note the log scale) with source reporting frequency rate, f, until a certain $f = f_{\max}$, beyond which the event reliability drops. This is because the network is unable to handle the increased injection of data packets and packets are dropped due to congestion.
2) Such an initial increase and subsequent decrease in event reliability is observed regardless of the number of source nodes, n.
3) $f_{\max}$ decreases with increasing n, i.e., congestion occurs at lower reporting frequencies with greater number of sources.
4) For $f > f_{\max}$, the behavior is rather wavy and not smooth. An intuitive explanation for such a behavior is as follows. The number of received packets, which is our event reliability, r, is the difference between the total number of source data packets, s, and the number of packets dropped by the network, d. While s simply scales linearly with f, the relationship between d and f is nonlinear. In some cases, the difference $s - d$ is seen to increase even though the network is congested. The important point to note however, is that this wavy behavior always stays well below the maximum event reliability at $f = f_{\max}$.

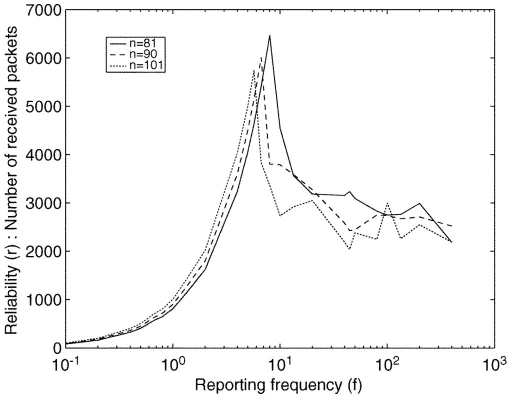

Fig. 4. Effect of varying the reporting rate, f, of source nodes on the event reliability, r, observed at the sink. The number of source nodes is denoted by n.

TABLE III
EVENT CENTERS FOR THE THREE CURVES WITH n = 81, 90, 101 IN FIG. 4

Number of source nodes	**Event Center** (X_{ev},Y_{ev})
81	(32.6,79.3)
90	(61.1,31.5)
101	(60.0,63.6)

Fig. 4 shows a similar trend between r and f with further increase in n (n = 81, 90, 101). As before, we tabulate the event centers in Table III. The event radius was fixed at 40 m for this set of experiments.

The wavy behavior for $f > f_{\max}$ observed in Fig. 3 persists in Fig. 4, but appears rather subdued because of much steeper drops due to congestion. All the other trends observed earlier are confirmed in Fig. 4.

Note also that the traditional metrics such as the number of packets sent and successfully received during the experiments can also be implicitly observed in Figs. 4 and 5. Recall that Figs. 4 and 5 show the event reliability in terms of the number of

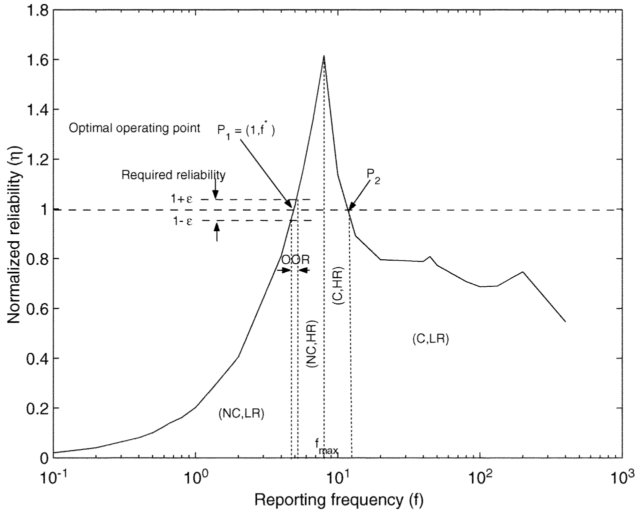

Fig. 5. The five characteristic regions in the normalized event reliability η versus reporting frequency f behavior.

packets received within a decision interval of τ when n sensor nodes in the event coverage send their readings with the reporting frequency of f. The values of τ, n, and f are given in Table I and on Fig. 4 and 5. Hence, the number of packets sent with the reporting frequency of f in each decision interval of τ can be calculated by $f * \tau * n$. Therefore, the ratio of the number of packets sent to that of received is $f * \tau * n/r$. For example, in Fig. 5, for $f \approx 6.67$ packets/s, $\tau = 10$ s, $n = 101$, the number of packets received is $r = 5746$ and the number of packets sent is $6.67 * 10 * 101 = 6736$.

In addition, the evaluation scenarios explored here represent densely deployment cases where congestion is more likely to occur. As it is observed from Fig. 2 and 3, as the number of source nodes sending data packets increases, the maximum reporting frequency that the network can accommodate, i.e., $f_{\max}$, decreases. However, note that the general $r - f$ behavior remains the same. Hence, for the cases where the density is not that high, congestion occurs at higher values of reporting frequency $f_{\max}$. Note that the discussions in this section are directly on the general $r - f$ behavior. Consequently, the results obtained here apply to the cases with lower densities as well.

C. Characteristic Regions

We now take a closer look at the r versus f characteristics and identify five characteristic regions, which are important for the operation of ESRT.

Consider a representative curve from Fig. 4 for $n = 81$ senders. This is replicated for convenience in Fig. 5. All our subsequent discussions use this particular case for illustration. However, it was verified that the r versus f behavior shows the general trend of initial increase and subsequent decrease due to congestion regardless of the parameter values. This is indeed observed in Figs. 3 and 4 for varying values of n. Hence, our discussions and results in this paper apply to the general r versus f behavior in WSN with any set of parameter values, with the specific case $(n = 81)$ used only for illustration purposes.

Let the desired event reliability determined by the application be R. Hence, a measure of event reliability is $\eta = r/R$. Here, η_i denotes the normalized event reliability at the end of each decision interval i.

Our aim is to operate as close to $\eta = 1$ as possible, while utilizing minimum network resources (f close to f^* in Fig. 5). We call this the *optimal operating point*, marked as P_1 in Fig. 5. For practical purposes, we define a tolerance zone of width 2ϵ around P_1, as shown in Fig. 5. Here, ϵ is a protocol parameter. The suitable choice of ϵ and its impact on ESRT protocol operation is dealt with in Section VI-C.

Note that the $\eta = 1$ line intersects the event reliability curve at two distinct points P_1 and P_2 in Fig. 5. Though the event is reliably detected at P_2, the network is congested and some source data packets are lost. Event reliability is achieved only because the high reporting frequency of source nodes compensates for this congestion loss. However, this is a waste of limited energy reserves and hence is not the optimal operating point. Similar reasoning holds for $\eta > 1 + \epsilon$.

From Fig. 5, we identify five characteristic regions (bounded by dotted lines) using the following decision boundaries:

- $(\mathbf{NC}, \mathbf{LR})$: $f < f_{\max}$ and $\eta < 1 - \epsilon$ (No Congestion, Low Reliability);
- $(\mathbf{NC}, \mathbf{HR})$: $f \leq f_{\max}$ and $\eta > 1 + \epsilon$ (No Congestion, High Reliability);
- $(\mathbf{C}, \mathbf{HR})$: $f > f_{\max}$ and $\eta > 1$ (Congestion, High Reliability);
- $(\mathbf{C}, \mathbf{LR})$: $f > f_{\max}$ and $\eta \leq 1$ (Congestion, Low Reliability);
- $\mathbf{OOR}$: $f < f_{\max}$ and $1 - \epsilon \leq \eta \leq 1 + \epsilon$ (Optimal Operating Region).

As seen earlier, the sink derives a reliability indicator η_i at the end of decision interval i. Coupled with a congestion detection mechanism (to determine $f \gtreqless f_{\max}$), this can help the sink determine in which of the above regions the network currently resides. Hence, these characteristic regions identify the state of the network. Let $\mathbf{S}_i$ denote the network state variable at the end of decision interval i. Then

$$\mathbf{S}_i \epsilon \{(\mathbf{NC}, \mathbf{LR}), (\mathbf{NC}, \mathbf{HR}), (\mathbf{C}, \mathbf{HR}), (\mathbf{C}, \mathbf{LR}), \mathbf{OOR}\}.$$

The operation of ESRT is closely tied to the current network state $\mathbf{S}_i$. The ESRT protocol state model and transitions are shown in Fig. 6.

IV. ESRT: Event-to-Sink Reliable Transport Protocol

The primary motive of ESRT is to achieve and maintain the network operation in state $\mathbf{OOR}$. Hence, the aim is to configure the reporting frequency rate f to achieve the desired event detection accuracy with minimum energy expenditure. To help accomplish this, ESRT uses a congestion control mechanism that serves the dual purpose of reliable detection and energy conservation.

Recall that the r versus f characteristic shown in Fig. 5 can change with dynamic topology resulting from either the failure or temporary power-down of sensor nodes. Hence, an efficient transport protocol should keep track of the reliability observed at the sink and accordingly configure the operating point. If η_i

is within the desired reliability limits $(1-\epsilon \leq \eta_i \leq 1+\epsilon)$ and no congestion notification alert is received, then state $\mathbf{OOR}$ has been reached and the sink informs source nodes to maintain the current reporting frequency f_i. Here, we make the reasonable assumption that the sink is powerful enough to reach all source nodes by broadcasting.

In general, the network can reside in any one of the five states $\mathbf{S}_i \in \{(\mathbf{NC},\mathbf{LR}),(\mathbf{NC},\mathbf{HR}),(\mathbf{C},\mathbf{HR}),(\mathbf{C},\mathbf{LR}),\mathbf{OOR}\}$. Depending on the current state $\mathbf{S}_i$, ESRT calculates an updated reporting frequency rate f_{i+1}, which is then broadcast to the source nodes. For example, if $\mathbf{S}_i \in \{(\mathbf{NC},\mathbf{LR}),(\mathbf{C},\mathbf{LR})\}$, the observed reliability levels are inadequate to detect the desired event features. In such a case, ESRT aggressively updates the reporting frequency rate to reliably track the event as soon as possible.

This self-configuring nature of ESRT helps it adapt to dynamic topology and random deployment, both typical for WSN. Another important feature of ESRT is its inclination to conserve scarce energy resources when reliability levels exceed those required for event detection. This is the case when $\mathbf{S}_i \in \{(\mathbf{NC},\mathbf{HR}),(\mathbf{C},\mathbf{HR})\}$. The motivation to reduce the reporting frequency rate in this case comes from energy conservation. However, our primary motive of reliable event detection must not be compromised. Hence, ESRT takes a conservative approach in this case and decreases f in a controlled manner.

The algorithms of ESRT mainly run on the sink, with minimal functionality at the source nodes. More precisely, sensor nodes only need the following two additional functionalities:

- Sensor nodes must listen to the sink broadcast at the end of each decision interval and update their reporting rates.
- Sensor nodes must deploy a simple and overhead-free local congestion detection support mechanism.

While the former is an implementation issue and is not within the scope of this work, the details of a congestion detection mechanism are provided in Section IV-B. Such a graceful transfer of complexity from sensor nodes to the sink node reduces the management costs and saves on valuable sensor resources. ESRT uses sink broadcast to communicate the updated reporting frequency rate to the sensor nodes in order to avoid any feedback latency problem as well as to save scarce sensor energy resources. Furthermore, ESRT works on the collective identification principle and does not require unique source IDs.

A. ESRT Protocol Operation

ESRT identifies the current state $\mathbf{S}_i$ from:

- reliability indicator η_i computed by the sink for decision interval i;
- a congestion detection mechanism;

using the decision boundaries defined in Section III-C. Depending on the current state $\mathbf{S}_i$, and the values of f_i and η_i, ESRT then calculates the updated reporting frequency f_{i+1} to be broadcast to the source nodes. At the end of the next decision interval, the sink derives a new reliability indicator η_{i+1} corresponding to the updated reporting frequency f_{i+1} of source nodes. In conjunction with any congestion reports, ESRT then determines the new network state $\mathbf{S}_{i+1}$. This process is repeated until the optimal operating region (state $\mathbf{OOR}$) is reached. As also shown in Fig. 6, note that not all transitions between states are possible, as explained in Section VI-A. This is due to the frequency update policies adopted by ESRT, which are described in detail for each of the five states.

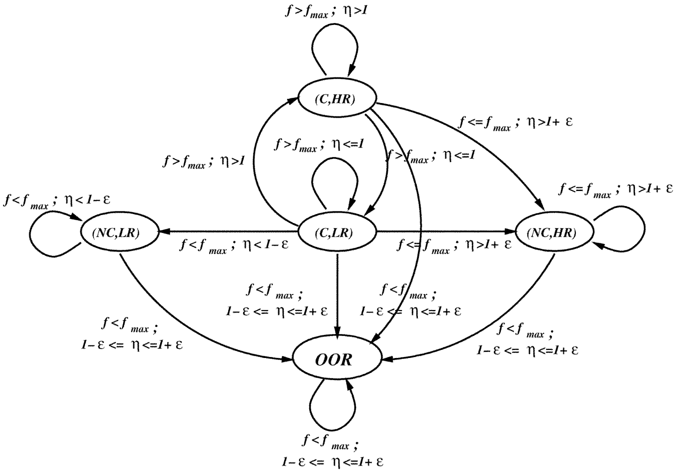

Fig. 6. ESRT protocol state model and transitions.

1) $(\mathbf{NC},\mathbf{LR})$ (*No Congestion, Low Reliability*): In this state, no congestion is experienced and the achieved reliability is lower than that required, i.e., $\eta < 1-\epsilon$ and $f < f_{\max}$. This can be the result of one/more of the following:
   - failure/power-down of intermediate routing nodes;
   - packet loss due to link errors;
   - inadequate information sent by source nodes.

   When intermediate nodes fail/power-down, packets that need to be routed through these nodes are dropped. This can cause a decrease in reliability even if enough source information is sent out. However, fault-tolerant routing/re-routing in WSN is provided by several existing algorithms [2], [7]. ESRT can work with any of these schemes.

   Packet loss due to link errors may be fairly significant in WSN due to the energy inefficiency of powerful error correction [8] and retransmission techniques. However, regardless of the packet error rate, the total number of packets lost due to link errors is expected to scale proportionally with the reporting frequency rate f. Here, we make the assumption that the net effect of channel conditions on packet losses does not deviate considerably in successive decision intervals. This is reasonable with static sensor nodes, slowly time-varying [8], [9], [13] and spatially separated channels for communication from event-to-sink in WSN applications. Hence, even in the presence of packet losses due to link errors, the initial reliability increase (Observation 1, Section III-B) is expected to be linear.

   It is now clear that in order to improve the reliability to acceptable levels, we need to increase the source information. Since the primary objective of ESRT is to achieve event-to-sink reliability, the reporting frequency rate f is aggressively increased to attain the required reliability as soon as possible. We can achieve such an aggressive increase by invoking the fact that the r versus f relationship in the absence of congestion, i.e., for $f < f_{\max}$, is linear. This prompts the use of the following multiplicative increase strategy to calculate reporting frequency rate update f_{i+1}

$$f_{i+1} = \frac{f_i}{\eta_i} \tag{2}$$

where η_i is the reliability observed at the sink at the end of decision interval i.

2) $(\mathbf{NC}, \mathbf{HR})$ (*No Congestion, High Reliability*): In this state, the required reliability level is exceeded, and there is no congestion in the network, i.e., $\eta > 1 + \epsilon$ and $f \leq f_{\max}$. This is because source nodes report more frequently than required. The most important consequence of this condition is excessive energy consumption by sensor nodes. Therefore the reporting frequency rate should be reduced in order to conserve energy. However, this reduction must be performed cautiously so that the event-to-sink reliability is always maintained. Hence, the sink reduces reporting frequency rate f in a controlled manner with half the slope, as opposed to the aggressive approach in the previous case. Intuitively, we are striking a balance here between saving the maximum amount of energy and losing reliable event detection. Thus the updated reporting frequency rate can be expressed as

$$f_{i+1} = \frac{f_i}{2}\left(1 + \frac{1}{\eta_i}\right). \tag{3}$$

It is shown in Section VI that such an update policy reduces the energy consumption in the network and does not compromise on event reliability.

3) $(\mathbf{C}, \mathbf{HR})$ (*Congestion, High Reliability*): In this state, the reliability is higher than required, and congestion is experienced, i.e., $\eta >$ and $f > f_{\max}$. This is due to the unique feature of WSN where the required event detection reliability can be attained even when some of the source data packets are lost. In this case, ESRT decreases the reporting frequency in order to avoid congestion and conserve energy in sensor nodes. As before, this decrease should be performed carefully such that the event-to-sink reliability is always maintained. However, the network operating in state $(\mathbf{C}, \mathbf{HR})$ is farther from the optimal operating point than in state $(\mathbf{NC}, \mathbf{HR})$. Therefore, we need to take a more aggressive approach so as to relieve congestion and enter state $(\mathbf{NC}, \mathbf{HR})$ as soon as possible. This is achieved by emulating the linear behavior of state $(\mathbf{NC}, \mathbf{HR})$ with the use of multiplicative decrease as follows:

$$f_{i+1} = \frac{f_i}{\eta_i}. \tag{4}$$

It can be shown that such a multiplicative decrease achieves all objectives (see Section VI).

4) $(\mathbf{C}, \mathbf{LR})$ (*Congestion, Low Reliability*): In this state the observed reliability is inadequate and congestion is experienced, i.e., $\eta \leq 1$ and $f > f_{\max}$. This is the worst possible state since reliability is low, congestion is experienced and energy is wasted. Therefore, ESRT reduces reporting frequency aggressively in order to bring the network to state $\mathbf{OOR}$ as soon as possible. Note that the reliability is a nonlinear function of reporting frequency in state $(\mathbf{C}, \mathbf{LR})$ as shown in Fig. 5. Hence in order to assure sufficient decrease in the reporting frequency rate, it is exponentially decreased and the new reporting frequency rate is expressed by

$$f_{i+1} = f_i^{\left(\frac{\eta_i}{k}\right)} \tag{5}$$

where k denotes the number of successive decision intervals for which the network has remained in state $(\mathbf{C}, \mathbf{LR})$ including the current decision interval, i.e., $k \geq 1$. The aim is to decrease f with greater aggression if a state transition is not detected. Such a policy also ensures convergence for $\eta = 1$ in state $(\mathbf{C}, \mathbf{LR})$.

5) $\mathbf{OOR}$ (*Optimal Operating Region*): In this state, the network is operating within ϵ tolerance of the optimal point, where the required reliability is attained with minimum energy expenditure. Hence, the reporting frequency rate is left unchanged for the next decision interval.

$$f_{i+1} = f_i. \tag{6}$$

```
k = 1;
ESRT()
  If (CONGESTION)
    If ($\eta < 1$)
    /* State=(C,LR) */
    /* Decrease Reporting Frequency Aggressively */
      $f = f^{\eta/k}$;
      $k = k + 1$;
    else if ($\eta > 1$)
    /* State=(C,HR) */
    /* Decrease Reporting Frequency to Relieve Congestion; No Compromise on Reliability */
      $k = 1$;
      $f = f/\eta$;
    end;
  else if (NO_CONGESTION)
    $k = 1$;
    If ($\eta < 1 - \epsilon$)
    /* State=(NC,LR) */
    /* Increase Reporting Frequency Aggressively */
      $f = f/\eta$;
    else if ($\eta > 1 + \epsilon$)
    /* State=(NC,HR) */
    /* Decrease Reporting Frequency Cautiously */
      $f = \frac{f}{2}\left(1 + \frac{1}{\eta}\right)$;
    else if ($1 - \epsilon \leq \eta \leq 1 + \epsilon$)
    /* Optimal Operating Region */
    /* Hold Reporting Frequency */
      $f = f$;
    end;
end;
```

Fig. 7. Algorithm of the ESRT protocol operation.

The entire ESRT protocol operation is summarized in the pseudo-algorithm given in Fig. 7.

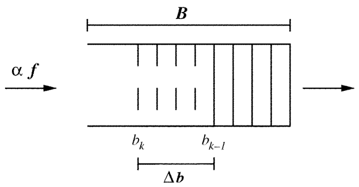

Fig. 8. Illustration of buffer level monitoring in sensor nodes.

B. Congestion Detection

In order to determine the current network state $\mathbf{S}_i$ in ESRT, the sink must be able to detect congestion in the network. However, the conventional ACK/NACK-based detection methods for end-to-end congestion control purposes cannot be applied here. The reason once again lies in the notion of event-to-sink reliability rather than end-to-end reliability. Only the sink, and not any of the sensor nodes, can determine the reliability indicator η_i and act accordingly. Moreover, end-to-end retransmissions and ACK/NACK overheads are a waste of limited sensor resources. Hence, ESRT uses a congestion detection mechanism based on local buffer level monitoring in sensor nodes. Any sensor node whose routing buffer overflows due to excessive incoming packets is said to be congested and it informs the sink of the same. The details of this mechanism are as follows.

In our event-to-sink model, the traffic generated during each reporting period, i.e., $1/f$, mainly depends on the reporting frequency rate f and the number of source nodes n. The reporting frequency rate f does not change within one reporting period since it is controlled periodically by the sink at the end of each decision interval with period of $\tau > 1/f$. Assuming n does not significantly change within one reporting period, the traffic generated during the next reporting period will have negligible variation. Therefore, the amount of incoming traffic to any sensor node in consecutive reporting intervals is assumed to stay constant. This, in turn, signifies that the increment in the buffer fullness level at the end of each reporting interval is expected to be constant.

Let b_k and b_{k-1} be the buffer fullness levels at the end of kth and $(k-1)$th reporting intervals, respectively, and B be the buffer size as in Fig. 8. For a given sensor node, let Δb be the buffer length increment observed at the end of last reporting period, i.e.,

$$\Delta b = b_k - b_{k-1}. \tag{7}$$

Thus, if the sum of current buffer level at the end of kth reporting interval and the last experienced buffer length increment exceeds the buffer size, i.e., $b_k + \Delta b > B$, the sensor node infers that it is going to experience congestion in the next reporting interval. Hence, it sets the CN (Congestion Notification) bit in the header of the packets it transmits as shown in Fig. 9. This notifies sink for the upcoming congestion condition to be experienced in next reporting interval.

Hence, if the sink receives packets whose CN bit is marked, it infers that congestion is experienced in the last decision interval. In conjunction with the reliability indicator η_i, the sink determines the current network state $\mathbf{S}_i$ at the end of decision interval i and acts according to the rules in Section IV-A.

Event ID	*CN* (1 bit)	Destination	Time Stamp	Payload	FEC

Fig. 9. Typical data packet with congestion notification field, which is marked to alert the sink for congestion.

V. Multiple Event Occurrences

The ESRT protocol operation defined in Section IV directly applies to the scenarios where a single event occurs in the wireless sensor field. In Section V-A, we explain how ESRT mechanisms can accurately detect multiple event occurrences and extract the required information for the protocol operation. Then, we present the ESRT protocol operation in multiple event scenarios in Section V-B.

A. Multiple Event Detection

In order to address the scenarios where multiple events occur simultaneously, it is necessary to accurately obtain the following information:

1) Is there a single event or multiple concurrent events in the sensor field?
2) If there are multiple events, are the generated data flows from sensor nodes to the sink passing through any common node?

In order to accurately capture the answers to these two questions, the sink utilizes the *Event ID* field of a data packet shown in Fig. 9. Note that this field accurately provides the answer to the first question above. If all of the data packets received by the sink carry the same *Event ID*, then there is a single event occurrence in the wireless sensor field as shown in Fig. 1. In this case, the sink achieves the desired event-to-sink reliability with minimum energy expenditure using the ESRT protocol operation shown in Fig. 7 as explained in Section IV.

If the sink receives data packets carrying different event IDs in their *Event ID* fields as shown in Fig. 9, it infers that multiple concurrent events occurred in the sensor field.

Note that we have implicitly assumed that the *Event ID*s can be obtained or distributed by using any existing high level network information collection mechanisms such as the existing in-network data aggregation method or location-aware routing for data aggregation or using the cluster-based event identification method. One simple conceivable Event ID assignment methodology is the *dynamically random Event ID assignment strategy* that is initiated at the time when the event is first detected. In this case, the sensor node that is the first in detecting the event chooses a random *Event ID* with a length of 16 bits. Since it first detects the event, generates the data packet conveying the event information and captures the wireless communication channel; it sends its data packet with this randomly selected *Event ID*. Any neighboring node hearing this local broadcast uses this *Event ID* to stamp its packet headers. Therefore, this randomly selected *Event ID* is dynamically propagated within the event coverage area. Note that this dynamic event ID distribution terminates at the boundary of the event coverage area. Thus, the forwarding sensor nodes do not need to perform any modification on the *Event ID* field of the data packets being routed. Note also that the random selection of *Event IDs* with a length of 16 bits

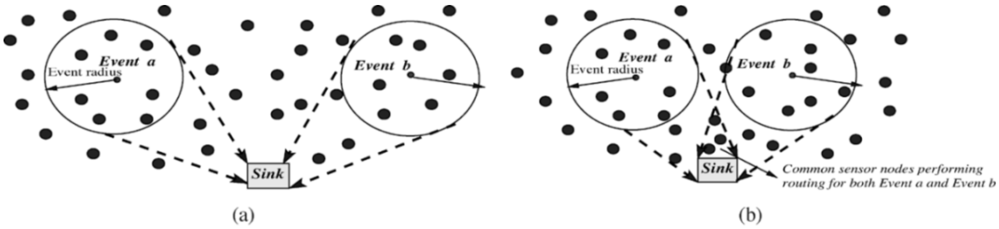
Event a
Event radius
Event b
Sink
(a)
Event a
Event radius
Event b
Sink
Common sensor nodes performing routing for both Event a and Event b
(b)

Fig. 10. Multiple event occurrences in the same wireless sensor field. (a) The flows generated by two events, i.e., *Event a* and *Event b*, are isolated. (b) The flows pass through some common sensor nodes.

corresponds to the probability of an ID conflict of less than 10^{-5}, which can be practically assumed to be negligible. On the other hand, when the event is first sensed by a sensor node which randomly assigns an *Event ID* and broadcasts its packets with it, the other sensor nodes may also sense the event and attempt to assign an ID to the same event. However, since the medium is not idle due to the local broadcast of the sensor node which was the first in sensing the event, they defer their broadcast at the MAC level. Hence, the other sensor nodes hear this first broadcast, and use this ID in the *Event ID* field of their packet headers. Therefore, it is also highly unlikely to generate two different *Event IDs* for the same event. Consequently, this dynamic random Event ID assignment strategy does not lead to ID conflict problem and can be safely used for this objective. However, note that the ESRT operation for multiple event occurrence scenarios[1] does not depend on a specific event ID assignment strategy, and hence other possible approaches for distributed ID assignment can be easily incorporated into the ESRT protocol operation.

In the scenarios where multiple concurrent events occur in the sensor field, it is necessary to find the answer to the second question above, i.e., if there are any common sensor nodes serving as a router for the flows generated by these multiple events. This information is detrimental to the selection of appropriate ESRT operation due to the reasons as follows. If there is no common wireless sensor node performing routing for these multiple events occurred simultaneously, then the flows generated by these multiple events are isolated, i.e., do not share any common path as shown in Fig. 10(a). Thus, in this case, ESRT protocol can address the event-to-sink reliability requirements of these multiple events individually with the default ESRT operation explained in Section IV.

If there exist common sensor nodes performing routing for the multiple events occurred simultaneously as shown in Fig. 10(b), then the flows generated by these events are not isolated. In this case, treating them individually may not always lead to the best possible solution. This is because any action taken by the sink on any of these flows may alter the reliability level and the congestion situation of the other event flows. Therefore, protocol actions need to be taken cautiously by and considering all of the concurrent event flows in the wireless sensor field. The updated ESRT protocol operation in order to accommodate these cases are explained in Section V-B.

Hence, in order to determine the necessary protocol operation, the sink must accurately detect whether the flows generated by these multiple events pass through any common sensor node functioning as a router. Furthermore, if indeed there exist such common router sensor nodes, it is necessary to learn which event flows share these common nodes. For this purpose, the sink utilizes the *Event ID* field of a data packet shown in Fig. 9. Here, we assume that *Event ID* field shown in Fig. 9 is a multidimensional field which can accommodate the Event IDs of several events occurring simultaneously. Therefore, the additional functionality required at the sensor nodes which perform routing can be stated as follows:

1) A sensor node keeps the *event-list*, i.e., the list of IDs of the events it serves as a router node in the wireless sensor field.
2) When the node receives a new data packet, it checks its event-list and the multidimensional *Event ID* field of this data packet.
   a) If there exists an ID in its *event-list*, which is not in the multidimensional *Event ID* field of this data packet, the sensor node:
      - adds this ID on top of the *Event ID* field of this data packet;
      - forwards the data packet.
   b) If there is not such an ID, then the sensor node checks whether its event-list includes the first element of the multidimensional *Event ID* field of this packet. If so, then the router sensor node leaves its event-list and the packet header intact and forward the packet. If not, it adds the first element of the multidimensional Event ID field of this packet into its event-list and forward the packet intact.

To illustrate the accurate detection of a multiple events case, assume that a sensor node performs routing for the data packets generated by Events with Event IDs a and b as shown in Fig. 10(b). Thus, this sensor node knows that it is indeed serving as a router node for the events a and b hence it has a and b in its *event-list*. Now, suppose that a data packet with only c in its *Event ID* field arrives at this sensor node. Hence, this sensor node adds a and b in the *Event ID* field of the data packet and then forward it. The sensor node also updates its event-list since now it received a data packet generated by the event c. Consequently, when the sink receives this data packet carrying c, a, and b in its *Event ID* field, it infers that the flows generated by the events a, b, and c are not isolated and pass through common nodes. Accordingly, it performs the necessary protocol actions as explained in Section V-B.

[1] Although the handling of multiple concurrent events at the software and signal processing levels is currently an active research area [1], [4], it is beyond the scope of our paper.

B. ESRT Operation in Multiple Event Scenarios

As described in Section V-A, the sink utilizes the *Event ID* field of a data packet in order to capture information about the multiple event occurrence in the sensor field.

If a single event occurs in the sensor field as shown in Fig. 1, i.e., all of the data packets received by the sink carry the same *Event ID*, then the sink brings the network state $\mathbf{S}$ to the optimal operating region $\mathbf{OOR}$ with the default ESRT protocol operation as explained in Section IV.

For the multiple event occurrence scenarios, the ESRT protocol operation varies based on whether the flows generated by these multiple events are isolated or not as explained in Section V-A. Hence, the detailed protocol operation for these two distinct cases are explained in the following sections.

1) Multiple Isolated Events: If there are multiple concurrent events in the sensor field, i.e., the sink receives data packets with different Event IDs, then the sink checks the *Event ID* fields of the data packets it received at the end of decision interval i. If all of the data packets have a single value in their multidimensional *Event ID* fields, it infers that the flows generated by these multiple events are isolated and do not share any common router sensor node as shown in Fig. 10(a).

In this case, let $\mathbf{S}_i^k$ and f_i^k be the current network state and the reporting frequency rate for the event k. Note that ESRT determines the current network state for event k, i.e., $\mathbf{S}_i^k$, from the reliability indicator η_i^k computed by the sink for decision interval i as explained in Section IV. Thus, the sink calculates the updated reporting frequency f_{i+1}^k based on $\mathbf{S}_i^k$, η_i^k, and f_i^k and broadcasts it to the sensor nodes in the event radius of event k in order to bring the network state to the optimal operating region $\mathbf{OOR}$ for the flows generated by event k. Consequently, the sink achieves the event-to-sink reliability requirements of these multiple events individually with the default ESRT operation explained in Section IV.

2) Multiple Events Passing Through Common Nodes: If there are data packets which carry multiple event IDs in their *Event ID* fields, then the sink infers that there exist common sensor nodes routing the flows generated by these different events as shown in Fig. 10(b). Therefore, the flows generated by these multiple events are not isolated. Hence, an action taken by the sink for any of these events may affect the reliability and congestion situation of the other events' flows.

In this case, instead of treating these event flows independently, it is better to take action cautiously and considering all of the concurrent event flows in the wireless sensor field. This is mainly because of the fact that the primary objective of ESRT is to achieve event-to-sink reliable transport. This leads to the fact that the event flows which are in different network states pose different levels of urgency in terms of protocol action. For example, while in state $(\mathbf{NC}, \mathbf{HR})$ no congestion is experienced and the observed reliability is higher than required, it is completely opposite in state $(\mathbf{C}, \mathbf{LR})$ where there is a congestion in the network and the event-to-sink reliability is not achieved as shown in Fig. 5. Hence, the event flows whose current network state are $(\mathbf{C}, \mathbf{LR})$ have greater urgency and hence have higher priority in terms of action to be taken by the sink. Similarly, although there is no congestion in both of the states $(\mathbf{NC}, \mathbf{LR})$ and $(\mathbf{NC}, \mathbf{HR})$, the event flows which are currently in state $(\mathbf{NC}, \mathbf{LR})$ do not receive their desired reliability levels and have higher priority than the ones in state $(\mathbf{NC}, \mathbf{HR})$. With this respect, we group the network states $\{(\mathbf{C}, \mathbf{LR}), (\mathbf{NC}, \mathbf{LR}), (\mathbf{C}, \mathbf{HR}), (\mathbf{NC}, \mathbf{HR})\}$ into *high priority states*, i.e., $(\mathbf{C}, \mathbf{LR})$, $(\mathbf{NC}, \mathbf{LR})$, and *low priority states*, i.e., $(\mathbf{C}, \mathbf{HR})$, $(\mathbf{NC}, \mathbf{HR})$, based on the observed reliability level associated with each of these network states.

Consequently, the sink takes the required action based on the priority of the network states of the multiple concurrent events sharing the same router sensor nodes. Let N_e be the number of concurrent events whose flows are passing through common router sensor nodes. The IDs of these events are obtained from the multidimensional Event ID field of the received data packets as explained in Section V-A. Let $\mathbf{S}_i^k$ and f_i^k be the current network state and the reporting frequency rate for the event k for $k \in N_e$.

1) The sink determines the network state $\mathbf{S}_i^k$ for each of the flows generated by the event $k \in N_e$ at the end of decision interval i as described in Section IV.
2) If there are events whose network state are high priority, i.e., $\exists j \in N_e$ such that $\mathbf{S}_i^j = (\mathbf{C}, \mathbf{LR})$ or $\mathbf{S}_i^j = (\mathbf{NC}, \mathbf{LR})$:
   a) The sink immediately performs the default ESRT operation described in Section IV for these events. That is, the sink calculates and broadcasts the updated reporting frequency f_{i+1}^j to the sensor nodes which are in the radius of event j, i.e., $\forall j$ with $\mathbf{S}_i^j = (\mathbf{C}, \mathbf{LR})$ or $\mathbf{S}_i^j = (\mathbf{NC}, \mathbf{LR})$.

      This action is more urgent to take because these events are not reliably communicated to the sink hence the first priority action is to make these events reach their desired reliability levels.

   b) The sink does not update the reporting frequencies for the other event flows whose network states are low priority, i.e., $f_{i+1}^j = f_i^j \; \forall j$ with $\mathbf{S}_i^j = (\mathbf{C}, \mathbf{HR})$ or $\mathbf{S}_i^j = (\mathbf{NC}, \mathbf{HR})$.

      This is because the actions taken for the event flows whose network states are high priority (step 2.(a)) may affect these events which already have higher reliability. Therefore, any further simultaneous actions to minimize energy expenditure of these flows is avoided in order not to compromise their reliability levels. Note that this is also consistent with the primary objective of ESRT protocol operation which is to achieve event-to-sink reliability.
3) If there are no events whose network state are high priority, i.e., $\mathbf{S}_i^j = (\mathbf{C}, \mathbf{HR})$ or $\mathbf{S}_i^j = (\mathbf{NC}, \mathbf{HR}) \; \forall j \in N_e$, then the sink follows the default ESRT operation described in Section IV for these events, i.e., calculates and broadcast the updated reporting frequency rate f_{i+1}^j to sensor nodes which are in the radius of the event $j \; \forall j \in N_e$.

The sink repeats these steps until all of the event flows reach the optimal operating region $\mathbf{OOR}$ as described in Section IV. As a result, the ESRT protocol operation described in Section IV can accommodate the scenarios where multiple events occur simultaneously in the sensor field.

On the other hand, if the events themselves overlap, i.e., they occur within the same vicinity and the associated event coverage areas intersects, ESRT resumes its normal operation discussed in Section IV treating those overlapping events as a single unified event. Note also that the same Event ID is used by the nodes within the unified coverage area of these overlapping events as the dynamic random Event ID distribution terminates at the boundaries of the unified event coverage area as discussed in Section V-A.

VI. ESRT Performance

In this section, we present both analytical and simulation results on the performance of ESRT protocol. Our results show that ESRT converges to state $\mathbf{OOR}$ starting from any of the other four initial network states $\mathbf{S}_i \in \{(\mathbf{NC}, \mathbf{LR}), (\mathbf{NC}, \mathbf{HR}), (\mathbf{C}, \mathbf{HR}), (\mathbf{C}, \mathbf{LR})\}$. Furthermore, the convergence times presented in this section are derived under the assumption that the r versus f characteristic does not change considerably within this duration. They can hence be interpreted as achievable lower bounds.

A. Analytical Results

We first present some analytical results on ESRT performance depending on the initial network state $\mathbf{S}_0$. Note that these results are obtained for the cases where a single event occurs in the sensor field although they may still apply for most of the multiple events cases. Recall that ESRT aims to reach state $\mathbf{OOR}$ starting from any initial state $\mathbf{S}_0$.

Lemma 1: Starting from $\mathbf{S}_0 = (\mathbf{NC}, \mathbf{HR})$, and with linear reliability (η) behavior when the network is not congested, the network state remains unchanged until ESRT converges to state $\mathbf{OOR}$.

Proof: The linear reliability (η) behavior for $f < f_{\max}$ can be expressed as $f = \alpha\eta$, where α denotes the slope. ESRT conservatively decrements f as follows [(3)]:

$$f_{i+1} = \frac{f_i}{2}\left(1 + \frac{1}{\eta_i}\right). \tag{8}$$

Hence,

$$\eta_{i+1} = \frac{1 + \eta_i}{2}. \tag{9}$$

Since $f_{i+1} < f_i$ from (8), it follows that $\mathbf{S}_i \in \{(\mathbf{NC}, \mathbf{HR}), (\mathbf{NC}, \mathbf{LR}), \mathbf{OOR}\}$, $\forall i \geq 0$ until ESRT converges. If possible, let $\mathbf{S}_{i+1} = (\mathbf{NC}, \mathbf{LR})$ when $\mathbf{S}_i = (\mathbf{NC}, \mathbf{HR})$ for some $i \geq 0$ before ESRT converges. Then

$$\eta_{i+1} = \frac{1 + \eta_i}{2} < 1 - \epsilon. \tag{10}$$

This implies that $\eta_i < 1 - 2\epsilon$, but $\eta_i > 1 + \epsilon$ since $\mathbf{S}_i = (\mathbf{NC}, \mathbf{HR})$. Hence, $\mathbf{S}_i \neq (\mathbf{NC}, \mathbf{LR})$ for any $i \geq 0$ until ESRT converges. In conjunction with our earlier inference, we conclude that $\mathbf{S}_i = (\mathbf{NC}, \mathbf{HR})$ $\forall i \geq 0$, until ESRT converges to state $\mathbf{OOR}$. ■

Lemma 2: Starting from $\mathbf{S}_0 = (\mathbf{NC}, \mathbf{HR})$, and with linear reliability (η) behavior when the network is not congested, ESRT converges to state $\mathbf{OOR}$ in $\tau\lceil\log_2(\eta_0 - 1/\epsilon)\rceil$ time units, where τ is the duration of the decision interval.

Proof: To establish the convergence time, we proceed as follows. Let the jth decision interval be the first one where $\mathbf{S}_j = \mathbf{OOR}$. It follows from Lemma 1 that j is the least index such that $\eta_j < 1 + \epsilon$. Using (9)

$$\begin{aligned} \eta_j &= \frac{\eta_{j-1} + 1}{2} < 1 + \epsilon \\ \eta_{j-1} &= \frac{\eta_{j-2} + 1}{2} < 1 + 2\epsilon \\ &\vdots \\ \eta_1 &= \frac{\eta_0 + 1}{2} < 1 + 2^{j-1}\epsilon. \end{aligned} \tag{11}$$

Hence, $j > \log_2(\eta_0 - 1/\epsilon)$ and the result follows. Note that this represents the time required to reach state $\mathbf{OOR}$ in order to conserve maximum energy. Our primary objective of reliable event detection is maintained all along by virtue of the conservative decrease (8). ■

Lemma 3: With linear reliability (η) behavior when the network is not congested, the network state transition $\mathbf{S}_i = (\mathbf{C}, \mathbf{HR}) \rightarrow \mathbf{S}_{i+1} = (\mathbf{NC}, \mathbf{LR})$ is not possible for any $i \geq 0$.

Proof: The linear reliability (η) behavior for $f < f_{\max}$ can be expressed as $f = \alpha\eta$, where α denotes the slope. It is seen from the r versus f characteristics in Figs. 3, 4, and 5, that for every $f > f_{\max}$ in state $(\mathbf{C}, \mathbf{HR})$, there exists one $f' < f_{\max}$ (in linear region) such that $\eta(f) = \eta(f')$.

The proof now proceeds by contradiction. Let us assume that $\mathbf{S}_{i+1} = (\mathbf{NC}, \mathbf{LR})$ when $\mathbf{S}_i = (\mathbf{C}, \mathbf{HR})$, for some $i \geq 0$. From the state definitions in Section III-C and update policy in Section IV-A, it follows that

$$f'_i \frac{(1 - \epsilon)}{\eta_i} > \frac{f_i}{\eta_i}. \tag{12}$$

Hence, a necessary condition is

$$f'_i > \frac{f_i}{1 - \epsilon} > f_i \tag{13}$$

but this is not true since $f_i > f_{\max} > f'_i$. This completes the proof. In accordance with this result, there is no transition from state $(\mathbf{C}, \mathbf{HR})$ to $(\mathbf{NC}, \mathbf{LR})$ in the state diagram shown in Fig. 6. This achieves our objective of relieving congestion and reducing energy consumption while not compromising on the event reliability (see Section IV-A). ■

In order to determine the convergence times of the ESRT protocol starting from $\mathbf{S}_0 \in \{(\mathbf{C}, \mathbf{HR}), (\mathbf{C}, \mathbf{LR})\}$, the nonlinear r versus f behavior needs to be tracked analytically. Due to space constraint, we demonstrate the convergence in these two cases using simulations.

B. Simulation Results

In order to study the convergence of ESRT using simulations, we once again developed an evaluation environment using *ns*-2. We first run the simulation experiments for the scenario where a single event occurs in the wireless sensor field. We run five experiments for each simulation configuration. We use the same sensor node and simulation configurations provided in

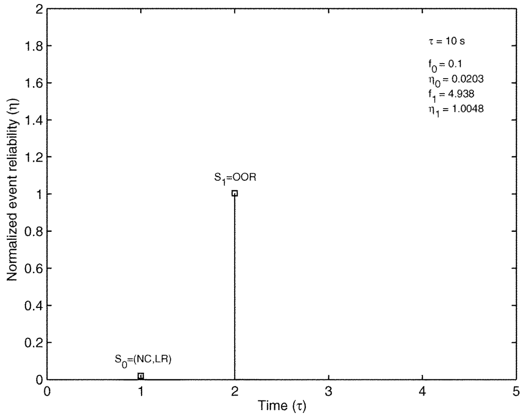

Fig. 11. The ESRT protocol trace for $\mathbf{S}_0 = (\mathbf{NC}, \mathbf{LR})$. Convergence is attained in a total of two decision intervals.

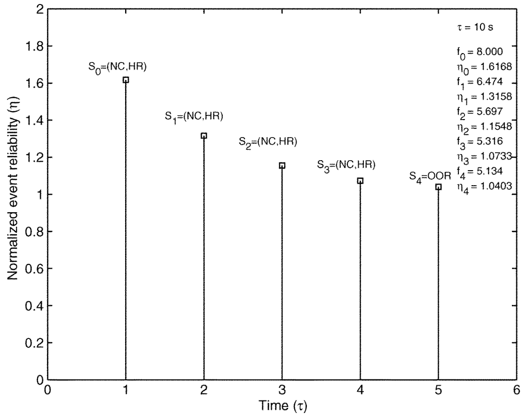

Fig. 12. The ESRT protocol trace for $\mathbf{S}_0 = (\mathbf{NC}, \mathbf{HR})$. Convergence is attained in a total of five decision intervals.

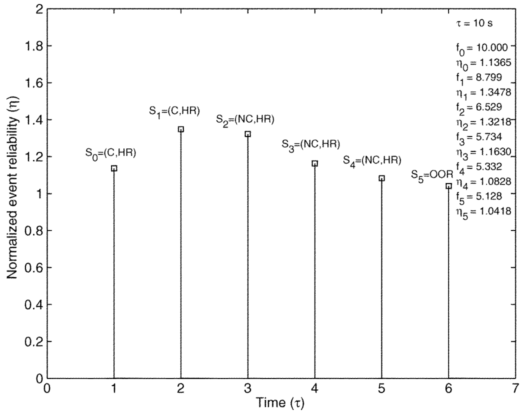

Fig. 13. The ESRT protocol trace for $\mathbf{S}_0 = (\mathbf{C}, \mathbf{HR})$. Convergence is attained in a total of six decision intervals in this case.

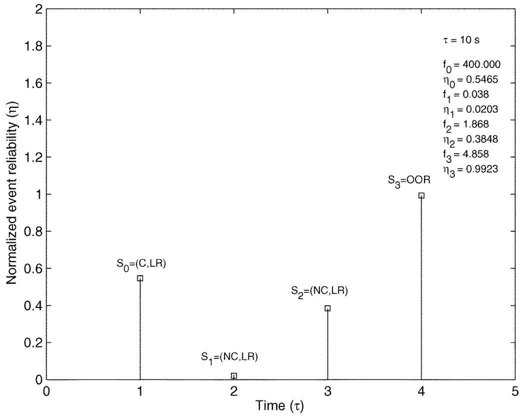

Fig. 14. The ESRT protocol trace for $\mathbf{S}_0 = (\mathbf{C}, \mathbf{LR})$. Convergence is attained in a total of four decision intervals in this case.

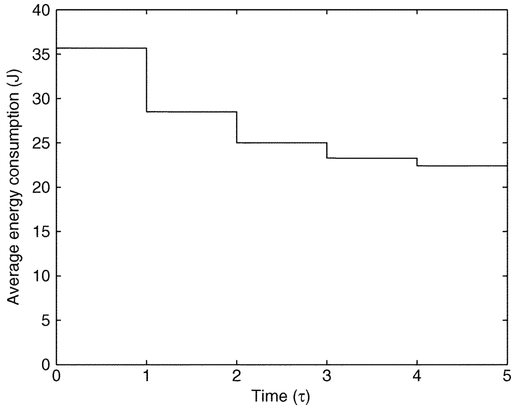

Fig. 15. Average power consumption of sensor nodes in each decision interval for $\mathbf{S}_0 = (\mathbf{NC}, \mathbf{HR})$.

Table I. Our convergence results are shown in Figs. 11–14 for initial network states $\mathbf{S}_0 = (\mathbf{NC}, \mathbf{LR}), (\mathbf{NC}, \mathbf{HR}), (\mathbf{C}, \mathbf{HR})$, and $(\mathbf{C}, \mathbf{LR})$, respectively. For these state convergence experiments of which results are shown in Figs. 10–14, all of the experiments for each initial state, i.e., $\mathbf{S}_0$, showed the same convergence pattern in terms of the number of decision intervals and the state transitions; and only the values of f_i and η_i varied very slightly. Therefore, we show one graph for each initial state. The corresponding trace values (f_i, η_i) and states are listed within each figure. The energy conservation property of ESRT for $\mathbf{S}_0 = (\mathbf{NC}, \mathbf{HR})$ is illustrated in Fig. 15 by taking the average of the experiment results. For all our simulation results presented here, the number of senders $n = 81$ and tolerance $\epsilon = 5\%$. The event radius was fixed at 40 m. Other simulation parameters are the same as those listed in Table I in Section III-B.

It is seen from Fig. 11 that the ESRT protocol for $\mathbf{S}_0 = (\mathbf{NC}, \mathbf{LR})$ converges in two decision intervals $(2\tau = 20\text{ s})$. This is expected from the aggressive multiplicative policy employed. Lemmas 1, 2 and 3 in Section VI-A can be verified from the trace values (f_i, η_i) and states listed within Fig. 12 and 13.

We also run simulation experiments to assess the ESRT performance in multiple events scenarios. We use the same sensor node and simulation configurations provided in Table I. We run five experiments for each simulation configuration. Events occur at random points in the sensor field. Fig. 16–19 show the average of five simulation experiment results for each graph. We first observe the number of intervals it takes for all of the event flows to converge to state $\mathbf{OOR}$. We also observe the average power consumption of the sensor nodes. Simulation experiments are performed for varying number of multiple concurrent events.

In the first scenario, we perform simulation experiments for the cases where the flows generated by the multiple events are isolated and do not share any common router sensor node. As

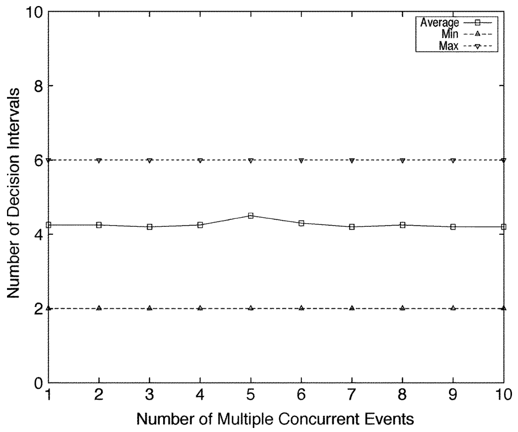

Fig. 16. Number of decision intervals for all of the event flows to converge to state $\mathbf{OOR}$ for varying number of multiple concurrent events. In this set of experiments, the multiple concurrent events are isolated and their flows do not pass through any common router sensor node.

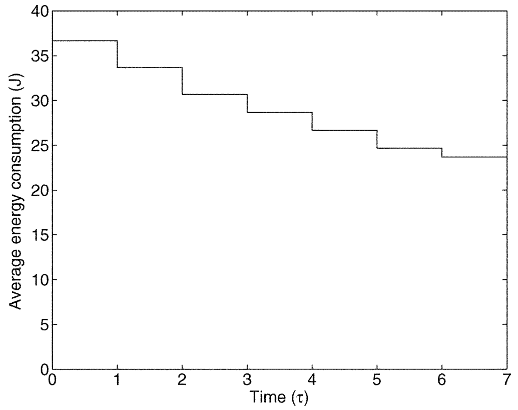

Fig. 17. Average power consumption of sensor nodes in each decision interval for the case where five concurrent events occur in the wireless sensor field. In this case, the flows generated by these events are isolated.

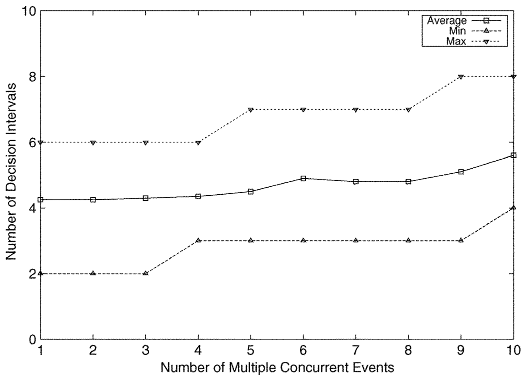

Fig. 18. Number of decision intervals for all of the event flows to converge to state $\mathbf{OOR}$ for varying number of multiple concurrent events. In this set of experiments, the multiple concurrent events are not isolated.

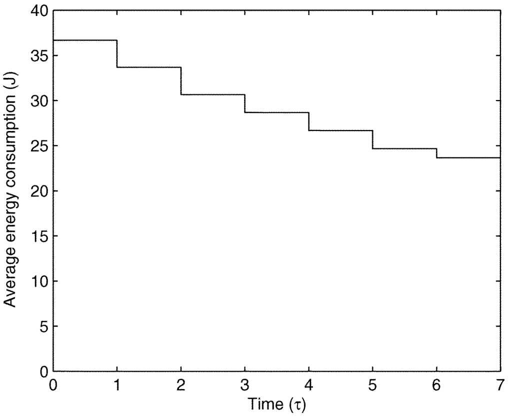

Fig. 19. Average power consumption of sensor nodes in each decision interval for the case where five concurrent events occur in the wireless sensor field. In this case, the flows generated by these events are not isolated.

shown in Fig. 16, the average number of decision intervals it takes for all of the event flows to converge to the state $\mathbf{OOR}$ does not vary significantly for varying number of multiple concurrent events. This is mainly because the flows generated by these multiple events are isolated and hence ESRT brings the network state of these flows to $\mathbf{OOR}$ individually as explained in Section V-B. Note also that the minimum and maximum number of decision intervals required for convergence are 2 and 6, which are equal to the case where a single event occurs. Hence, the convergence to the $\mathbf{OOR}$ state is not delayed in the case of multiple isolated events.

Moreover, as shown in Fig. 17, the average power consumed by the sensor nodes also show the same pattern we observed for a single event scenario as shown in Fig. 15. This is also because of the fact that the sink takes action for the flows generated by the multiple isolated events independently. Therefore, the average power consumption decreases with time as the ESRT protocol works to minimize the energy expenditure while maintaining the event-to-sink reliability.

In the second scenario, we perform simulation experiments for the cases where the flows generated by the multiple events are not isolated and there are common router sensor nodes routing these multiple flows in the sensor field. As shown in Fig. 18, the average number of decision intervals it takes for all of the event flows to converge state $\mathbf{OOR}$ slightly increases with the number of concurrent events. This is mainly because the flows generated by these multiple events are not isolated and hence ESRT considers the priority of the current network states of these flows as explained in Section V-B. Therefore, the sensor nodes which are in the radius of the events that already have adequate reliability may not experience reporting frequency update at the end of each decision interval. Thus, the number of decision intervals it takes for those events to converge increases. Note also that the minimum and maximum number of decision intervals required for convergence also vary with the number of multiple concurrent events due to the same reason. However, as shown in Fig. 18, the increase in the convergence time is very small even in case of 10 nonisolated concurrent events. Hence, the ESRT protocol can effectively address the cases where multiple events occur simultaneously.

Furthermore, as shown in Fig. 19, the average power consumed by the sensor nodes also shows the same pattern we observed for the previous case in Fig. 17. However, the decrease in the average consumed power is slightly slower in this case. This is because the fact that the sink may not take any action for some of the flows which already have adequate reliability levels. Note that this result is also consistent with the average convergence time results in Fig. 18.

C. Suitable Choice of ϵ

For practical purposes, ESRT uses a tolerance zone of ϵ around the optimal operating point P_1 in Fig. 5. If at the end of decision interval i, the reliability η_i is within $[1-\epsilon, 1+\epsilon]$ and if no congestion is detected in the network, then the network is in state $\mathbf{OOR}$. The event is deemed to be reliably detected at the sink and the reporting frequency remains unchanged. Greater proximity to the optimal operating point can hence be achieved with small ϵ. However, as seen from Lemma 2 in Section IV-A, the smaller is ϵ, the greater is the convergence time. Hence, a good choice of ϵ is one that balances the tolerance and convergence requirements. For example, a 1% tolerance requirement can offset the convergence time by as much as 7τ time units when $\mathbf{S}_0 = (\mathbf{NC}, \mathbf{HR})$. Note however that reliable event detection is maintained all along (Lemma 2 in Section IV-A) due to the conservative decrease.

VII. Conclusion

The notion of event-to-sink reliability is necessary for reliable transport of event features in WSN. Based on such a collective reliability notion, a new reliable transport scheme for WSN, the event-sink reliable transport (ESRT) protocol, is presented in this paper. To the best of our knowledge, this is the first study of reliable transport in WSN from the event-to-sink perspective. ESRT has a congestion control component that serves the dual purpose of achieving reliability and conserving energy. The algorithms of ESRT mainly run on sink and require minimal functionality at resource constrained sensor nodes. The primary objective of ESRT is to configure the network as close as possible to the optimal operating point, where the required reliability is achieved with minimum energy consumption and without network congestion. We have also extended ESRT protocol operations to accommodate the scenarios where multiple events concurrently occur in the sensor field. Analytical performance evaluation and simulation results show that ESRT converges to state $\mathbf{OOR}$ regardless of the initial network state $\mathbf{S}_0$. Furthermore, the simulation experiments show that ESRT can also achieve the required event-to-sink reliability in case of multiple concurrent events.

References

[1] R. Brooks, P. Ramanathan, and A. Sayeed, "Distributed target classification and tracking in sensor networks," *Proc. IEEE*, vol. 91, no. 8, pp. 1163–1171, Aug. 2003.

[2] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion for wireless sensor networking," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 2–16, Feb. 2003.

[3] D. Johnson, D. Maltz, Y. Hu, and J. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," IETF Internet draft, Feb. 2002.

[4] J. Liu, M. Chu, J. J. Liu, J. E. Reich, and F. Zhao, "State-centric programming for sensor and actuator network systems," *IEEE Pervasive Comput.*, vol. 2, no. 4, pp. 50–62, Oct./Dec. 2003.

[5] MICA Motes and Sensors [Online]. Available: http://www.xbow.com/

[6] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz, "ESRT: event-to-sink reliable transport for wireless sensor networks," in *Proc. ACM MOBIHOC*, Annapolis, MD, Jun. 2003, pp. 177–188.

[7] S. D. Servetto and G. Barrenechea, "Constrained random walks on random graphs: routing algorithms for large scale wireless sensor networks," in *Proc. ACM WSNA*, Atlanta, GA, Sep. 2002, pp. 12–21.

[8] E. Shih *et al.*, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," in *Proc. ACM MOBICOM*, Rome, Italy, Jul. 2001, pp. 272–286.

[9] K. Sohrabi, B. Manriquez, and G. Pottie, "Near-ground wideband channel measurements," in *Proc. IEEE Vehicular Technology Conf. (VTC)*, vol. 1, New York, 1999, pp. 571–574.

[10] F. Stann and J. Heidemann, "RMST: reliable data transport in sensor networks," in *Proc. IEEE SNPA*, May 2003, pp. 102–112.

[11] M. C. Vuran, O. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: theory and applications for wireless sensor networks," *Comput. Netw. J.*, vol. 45, no. 3, pp. 245–261, Jun. 2004.

[12] C. Y. Wan, A. T. Campbell, and L. Krishnamurthy, "PSFQ: a reliable transport protocol for wireless sensor networks," in *Proc. ACM WSNA*, Atlanta, GA, Sep. 2002, pp. 1–11.

[13] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proc. ACM SENSYS*, 2003, pp. 1–13.

Özgür B. Akan (M'00) received the B.S. and M.S. degrees in electrical and electronics engineering from Bilkent University and Middle East Technical University, Ankara, Turkey, in 1999 and 2001, respectively. He received the Ph.D. degree in electrical and computer engineering from the Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, in 2004.

He is currently an Assistant Professor with the Department of Electrical and Electronics Engineering, Middle East Technical University. His current research interests include sensor networks, next-generation wireless networks, and deep-space communication networks.

Ian F. Akyildiz (M'86–SM'89–F'96) received the B.S., M.S., and Ph.D. degrees in computer engineering from the University of Erlangen-Nuernberg, Germany, in 1978, 1981, and 1984, respectively.

Currently, he is the Ken Byers Distinguished Chair Professor of the School of Electrical and Computer Engineering, and Director of the Broadband and Wireless Networking Laboratory, Georgia Institute of Technology, Atlanta. His current research interests are in sensor networks and next-generation wireless networks. He is the Editor-in-Chief of *Computer Networks* (Elsevier) and of *Ad Hoc Networks Journal* (Elsevier).

Dr. Akyildiz has been a Fellow of the ACM since 1996. He was the technical program chair and general chair for several IEEE and ACM conferences including IEEE INFOCOM, ACM MOBICOM, and IEEE ICC. He received the Don Federico Santa Maria Medal for his services to the Universidad Federico Santa Maria in Chile in 1986. He served as a National Lecturer for ACM from 1989 until 1998 and received the ACM Outstanding Distinguished Lecturer Award for 1994. He received the 1997 IEEE Leonard G. Abraham Prize award (IEEE Communications Society) for his paper entitled "Multimedia Group Synchronization Protocols for Integrated Services Architectures" published in the IEEE Journal of Selected Areas in Communications (JSAC) in January 1996. He received the 2002 IEEE Harry M. Goode Memorial award (IEEE Computer Society) with the citation "for significant and pioneering contributions to advanced architectures and protocols for wireless and satellite networking." He received the 2003 IEEE Best Tutorial Award (IEEE Communications Society) for his paper entitled "A survey on sensor networks," published in *IEEE Communication Magazine*, in August 2002. He also received the 2003 ACM Sigmobile Outstanding Contribution Award with the citation "for pioneering contributions in the area of mobility and resource management for wireless communication networks."