Event-to-Sink Reliable Transport in Wireless Sensor Networks

Özgür B. Akan Ian F. Akyildiz Yogesh Sankarasubramaniam

Broadband & Wireless Networking Laboratory
School of Electrical & Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332
Tel: (404) 894-5141 Fax: (404) 894-7883
Email:{akan,ian,yogi}@ece.gatech.edu

***Abstract*— Wireless sensor networks (WSN) 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. Hence, 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. However, the absence of reliable transport altogether can seriously impair event detection. 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. If the event-to-sink reliability is lower than required, ESRT adjusts the reporting frequency of source nodes aggressively in order to reach the target reliability level as soon as possible. If the reliability is higher than required, then ESRT reduces the reporting frequency conservatively in order to conserve energy while still maintaining reliability. 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*— Wireless Sensor Networks, Reliable Transport Protocols, Event-to-Sink Reliability, Congestion Control, Energy Conservation.**

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.

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 [13] 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.

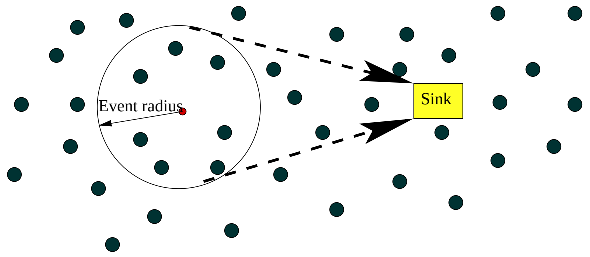

Fig. 1. Typical sensor network topology with event and sink. The sink is only interested in collective information of sensor nodes within the event radius and not in their individual data.

Our work is also motivated by the results in [12], which emphasize the need for congestion control in WSN. It was shown in [12] that exceeding network capacity can be detrimental to the observed goodput. However, the authors stopped short of providing a solution to this problem.

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. Some of its salient features are

1) *Self-configuration* - Reliable event detection must be

An earlier version of this work was published in [6].
This work is supported by the National Science Foundation under contract ECS-0225497.

established and maintained in the face of dynamic topology in WSN. Topology dynamics can result from either the failure or temporary power-down of energy constrained sensor nodes. Spatial variation of events and random node deployment only exacerbate the above problem. ESRT is self-configuring and achieves flexibility under dynamic topologies by self-adjusting the operating point (see Section IV).

2) *Energy awareness* - Although the primary goal of ESRT is reliable event detection, it aims to accomplish this with minimum possible energy expenditure. For instance, if reliability levels at the sink are found to be in excess of that required, the source nodes can conserve energy by reducing their reporting rate (see Section IV).
3) *Congestion Control* - Packet loss due to congestion can impair event detection at the sink even when enough information is sent out by the sources. Hence, congestion control is an important component for reliable event detection in WSN. An important feature of ESRT is that congestion control is also used to reduce energy consumption. Correlated data flows are loss tolerant to the extent that event features are reliably communicated to the sink. Due to this unique characteristic of WSN, required event detection accuracy may be attained even in the presence of packet loss due to network congestion. In such cases however, a suitable congestion control mechanism can help conserve energy while maintaining desired accuracy levels at the sink. This is done by conservatively reducing the reporting rate. Details of such a mechanism are presented in Section IV.
4) *Collective identification* - In typical WSN applications, the sink is only interested in the collective information provided by numerous sensor nodes and not in their individual reports. In accordance with this, ESRT does not require individual node IDs for operation. This is also in tune with our proposed event-to-sink model rather than the traditional end-to-end model. More importantly, this can ease implementation costs and reduce overhead.
5) *Biased Implementation* - The algorithms of ESRT mainly run on the sink with minimum functionalities required at sensor nodes. This helps conserve limited sensor resources and shifts the burden to the high-powered sink. Such a graceful transfer of complexity is possible only due to the event-to-sink reliability notion.

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 reliability 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.

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 and identify five characteristic reliability regions. These regions determine the appropriate actions taken by ESRT. 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

Despite the considerable amount of research on several aspects of sensor networking, the problems of reliable transport and congestion control are yet to be efficiently studied and addressed. The urgent need for congestion control is pointed out within the discussion of infrastructure tradeoffs for WSN in [12]. However, the authors do not propose any solution for the problem they identify.

In another recent work [13], 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 loss. 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. Event detection/tracking in the forward direction does not require guaranteed end-to-end data delivery as in PSFQ. Individual data flows are correlated and loss tolerant to the extent that desired event features are collectively and reliably informed to the sink. 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 loss due to congestion.

In [10], the RMST (Reliable Multi-Segment Transport) protocol is proposed to address the requirements of reliable data transport in wireless sensor networks. RMST is mainly based on the functionalities provided by *directed diffusion* [3]. Furthermore, RMST utilizes in-network caching and provides guaranteed delivery of the data packets generated by the event flows. However, as discussed above, 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.

A novel transmission control scheme for use at the MAC layer in WSN is proposed in [14] with the main objective of per-node fair bandwidth share. Energy efficiency is maintained by controlling the rate at which MAC layer injects packets into the channel. Although such an approach can control the transmission rate of a sensor node, it neither considers congestion control nor addresses reliable event detection. For similar reasons, the use of other MAC protocols like the IEEE 802.11 DCF or S-MAC [15] that provide some form of hop reliability is inadequate for reliable event detection in WSN.

Next, we briefly examine transport solutions in other wireless networks and point out their inadequacies when applied to WSN. These studies 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 [1]. 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 ($<$4KB 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.

The multi-hop and many-to-one nature of data flows in WSN prompts a review of reliable multicast solutions proposed in other wired/wireless networks. There exist many such schemes that address the reliable transport and congestion control for the case of single sender and multiple receivers [2]. Although the communication structure of the reverse path, i.e., from sink to sources in WSN, is an example of multicast, it is not valid for the forward channel where multiple correlated reports are sent to a single destination. Similar transport problems with multiple senders and a single receiver in other wired/wireless networks simply corresponds to a multiple unicast. However, the WSN paradigm requires the notion of collective reliability. Hence, neither the reliable multicast nor unicast transport solutions can be applied in our case.

III. The Reliable Transport Problem in WSN

In 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 makes an informed decision based on reports received from sensor nodes during that interval. The specifics of such a decision making process are application dependent and beyond our present scope.

The least we can assume is that the sink derives a reliability indicator r_i at the end of decision interval i. Note that r_i must be calculated only using parameters available at the sink. Hence, notions of throughput/goodput (as in [12]), 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 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 reliability, R, then the event is deemed to be reliably detected. Else, appropriate action needs to be taken to achieve the desired reliability, R.

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[1]. 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. The reporting rate of a sensor node is defined as the number of packets sent out per unit time by that node. 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.*

B. Evaluation Environment

In order to study the relationship between the observed reliability at the sink, r, and the reporting frequency, f, of sensor nodes, we developed an evaluation environment using ns-2 [11]. The parameters used in our study are listed in Table 1.

200 sensor nodes were randomly positioned in a 100x100 sensor field. 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 was 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) [4].

[1]With in-network data aggregation, one must account for data packets that were aggregated en route to the sink

The impact of using other routing protocols on the achieved goodput behavior with reporting period was shown to be insignificant in [12]. Hence, it is reasonable to assume that the r vs. f behavior and ESRT performance are insensitive to the underlying routing protocol.

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
Packet length	30 bytes
IFQ length	65 packets
Transmit Power	0.660 W
Receive Power	0.395 W
Decision interval (τ)	10 sec

The results of our study are shown in Fig. 2 for 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 corresponding number of senders n. These values are tabulated in Table 2. The event radius was fixed throughout at 30m.

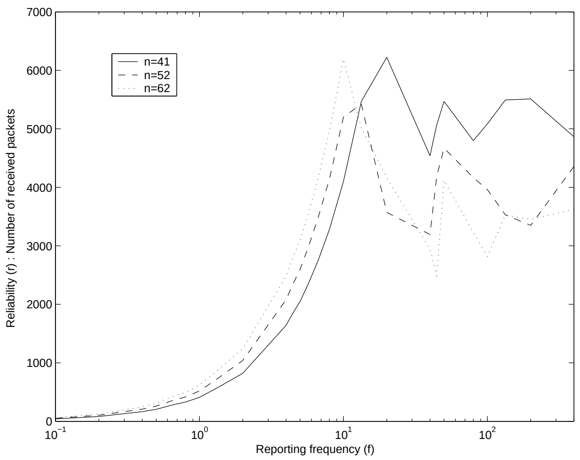

Fig. 2. The effect of varying the reporting rate, f, of source nodes on the event reliability, r, observed at the sink. The number of source nodes is denoted by n.

We make the following observations from Fig. 2

1) The reliability, r, shows a linear increase (note the log scale) with source reporting rate, f, until a certain $f = f_{max}$, beyond which the 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 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 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 non-linear. In some cases, the difference $s - d$ is seen to increase eventhough the network is congested. The important point to note however, is that this wavy behavior always stays well below the maximum reliability at $f = f_{max}$
5) The drop in reliability due to network congestion is more significant with increasing n.

TABLE II
EVENT CENTERS FOR THE THREE CURVES WITH N=41,52,62 IN FIG. 2

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)

Fig. 3 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 3. The event radius was fixed at 40m for this set of experiments.

The wavy behavior for $f > f_{max}$ observed in Fig. 2 persists in Fig. 3, but appears rather subdued because of much steeper drops due to congestion (see observation 5 earlier). All the other trends observed earlier are confirmed in Fig. 3.

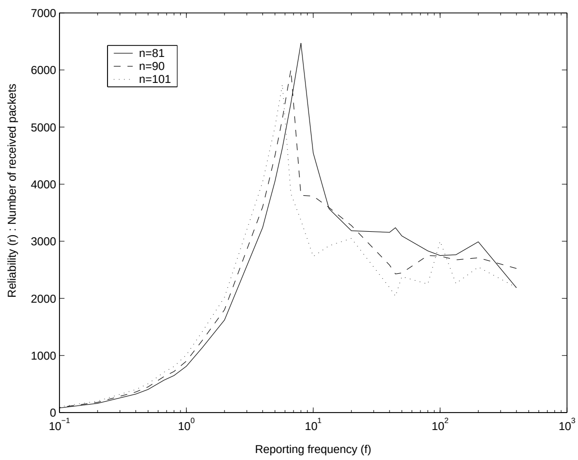

Fig. 3. The effect of varying the reporting rate, f, of source nodes on the event reliability, r, observed at the sink. The number of source nodes is denoted by n.

TABLE III
EVENT CENTERS FOR THE THREE CURVES WITH N=81,90,101 IN FIG. 3

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)

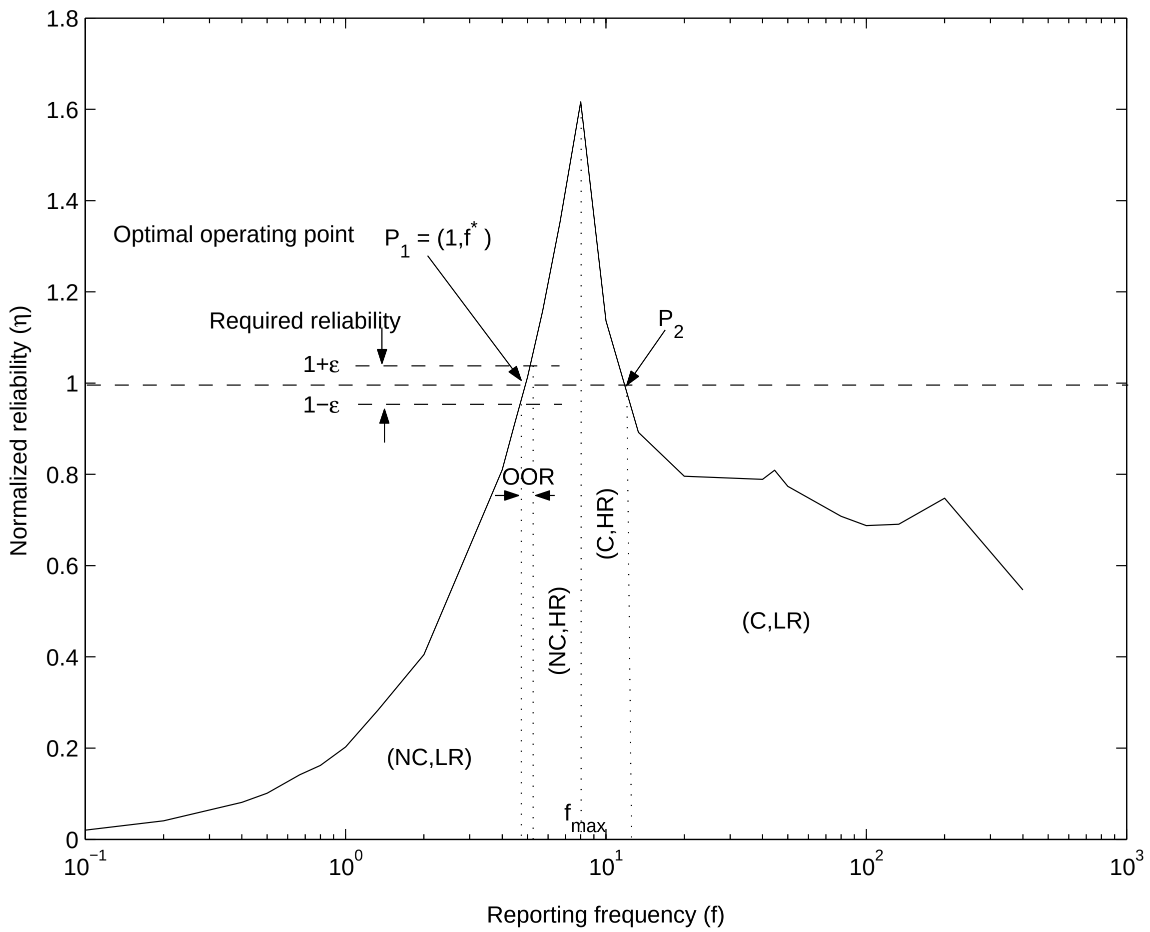

Fig. 4. The five characteristic regions in the normalized reliability, η, vs. reporting frequency, f, behavior.

C. Characteristic Regions

A general trend of initial reliability, r, increase with reporting frequency, f, and subsequent decrease due to congestion loss is evident from our preliminary studies in Fig. 2 and Fig. 3. This confirms the urgent need for an event-to-sink reliable transport solution with a congestion control mechanism in WSN. We now take a closer look at the r vs. f characteristics and identify five characteristic regions. As will be seen shortly, these regions are important for the operation of ESRT.

Consider a representative curve from Fig. 3 for $n = 81$ senders. This is replicated for convenience in Fig. 4. All our subsequent discussions use this particular case for illustration. However, it was verified that the r vs. 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. 2 and 3 for varying values of n. Hence, our discussions and results in this paper apply to a general r vs. f behavior in WSN with any set of parameter values, with the specific case ($n = 81$) used only for illustration purposes.

Let the desired reliability as laid down by the application be R. Hence, a normalized measure of reliability is $\eta = \frac{r}{R}$. As before, η_i denotes the normalized reliability at the end of 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. 4). We call this the *optimal operating point*, marked as P_1 in Fig. 4. For practical purposes, we define a tolerance zone of width 2ϵ around P_1, as shown in Fig. 4. 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 reliability curve at two distinct points P_1 and P_2 in Fig. 4. 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. 4, we identify five characteristic regions (bounded by dotted lines) using the following decision boundaries

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

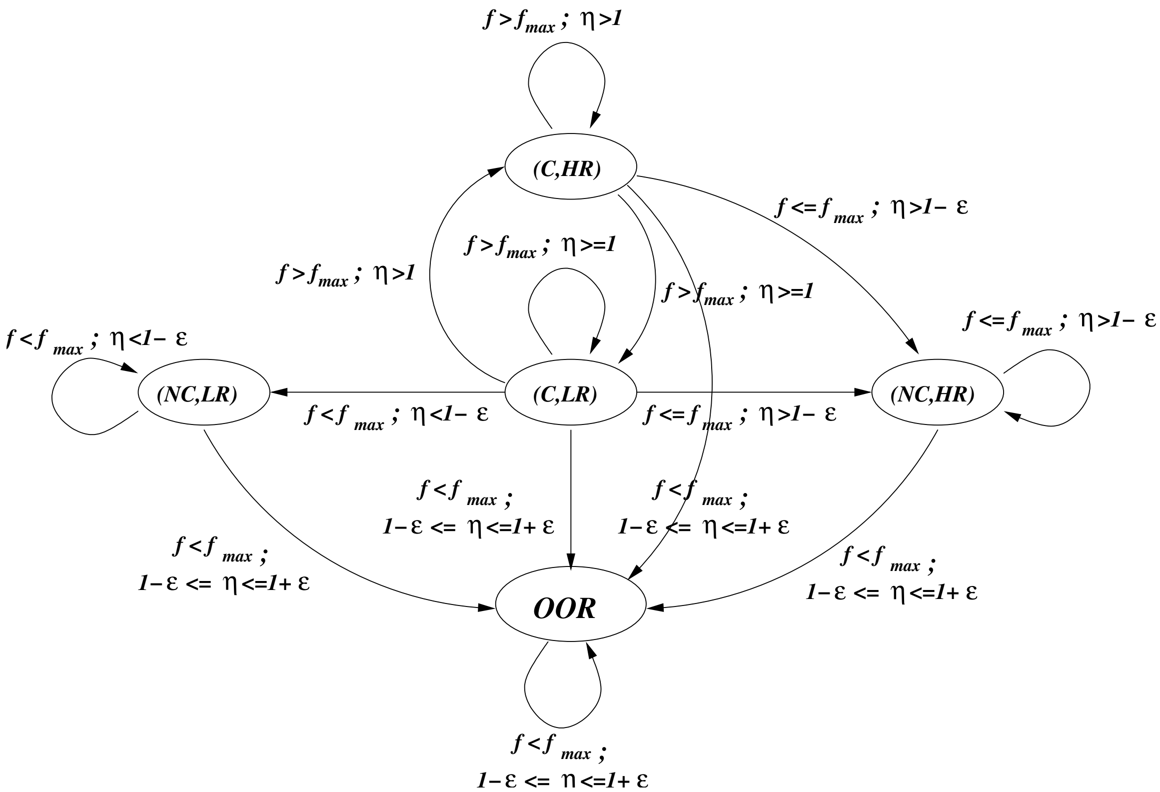

Fig. 5. ESRT protocol state model and transitions.

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 $f \gtrless f_{max}$), 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,

$$\mathbf{S}_i \in \{\textbf{(NC,LR),(NC,HR),(C,HR),(C,LR),OOR}\}$$

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. 5. We now proceed to discuss the specifics of ESRT and its operation in each of these states in detail.

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

ESRT is a novel solution that is proposed to address the transport problem in WSN. The primary motive of ESRT is to achieve and maintain operation in state **OOR**. Hence, the aim is to configure the reporting frequency 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 vs. f characteristic shown in Fig. 4 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 **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 broadcast.

In general, the network can reside in any one of the five states $\mathbf{S}_i \in$ {**(NC,LR),(NC,HR),(C,HR),(C,LR),OOR**}. Depending on the current state $\mathbf{S}_i$, ESRT calculates an updated reporting frequency f_{i+1}, which is then broadcast to the source nodes. For example, if $\mathbf{S}_i \in$ {**(NC,LR),(C,LR)**}, the observed reliability levels are inadequate to detect the desired event features. In such a case, ESRT aggressively updates the reporting frequency 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 of 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 $\mathbf{S}_i \in$ {**(NC,HR),(C,HR)**}. The motivation to reduce the reporting frequency 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 management costs and saves on valuable sensor resources. Further simplifying implementation is the fact that ESRT works on the collective identification principle and does not require unique source IDs.

In the following subsection, we discuss the operation of ESRT in each network state and also present a pseudo-algorithm for its implementation.

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 **OOR**) is reached. The state model of the ESRT protocol and state transitions are shown in Fig. 5. 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 now described in detail for each of the five states.

1) **(NC,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 drop in reliability even if enough source information is sent out. However, fault-tolerant routing/re-routing in WSN is provided by several existing routing algorithms [3], [7]. ESRT can work with any of these routing 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 f. Here, we make the assumption that the net effect of channel conditions on packet loss does not deviate appreciably in successive decision intervals. This is reasonable with static sensor nodes, slowly time-varying ([8], [9]) and spatially separated channels for communication from event-to-sink in WSN applications. Hence, even in the presence of packet loss 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 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 vs. 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 rate update f_{i+1}

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

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

2) **(NC,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 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 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 can be expressed as

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

   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) **(C,HR)** *(Congestion, High Reliability)* : In this state, the reliability is higher than required, and congestion is experienced, i.e., $\eta > 1$ and $f > f_{max}$. This is due to the unique feature of WSN where 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 **(C,HR)** is farther from the optimal operating point than in state **(NC,HR)**. Therefore, we need to take a more aggressive approach

so as to relieve congestion and enter state **(NC,HR)** as soon as possible. This is achieved by emulating the linear behavior of state **(NC,HR)** with the use of multiplicative decrease as follows

$$f_{i+1} = \frac{f_i}{\eta_i} \tag{3}$$

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

4) **(C,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 **OOR** as soon as possible. Note that reliability is a non-linear function of reporting frequency in state **(C,LR)** as shown in Fig. 4. Hence in order to assure sufficient decrease in the reporting frequency, it is exponentially decreased and the new frequency is expressed by

$$f_{i+1} = f_i^{(\eta_i/k)} \tag{4}$$

where k denotes the number of successive decision intervals for which the network has remained in state **(C,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 **(C,LR)**.

5) **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 of source nodes is left unchanged for the next decision interval.

$$f_{i+1} = f_i \tag{5}$$

The entire ESRT protocol operation is summarized in the pseudo-algorithm given in Fig. 6

```
k = 1;
ESRT()
  If (CONGESTION)
    If (η < 1)
    /* State=(C,LR) */
    /* Decrease Reporting Frequency
Aggressively */
      f = f^(η/k);
      k = k + 1;
    else if (η > 1)
    /* State=(C,HR) */
    /* Decrease Reporting Frequency
to Relieve Congestion; No Compromise on
Reliability */
      k = 1;
      f = f/η;
    end;
  else if (NO_CONGESTION)
    k = 1;
    If (η < 1 − ε)
    /* State=(NC,LR) */
    /* Increase Reporting Frequency
Aggressively */
      f = f/η;
    else if (η > 1 + ε)
    /* State=(NC,HR) */
    /* Decrease Reporting Frequency
Cautiously */
      f = (f/2)(1 + 1/η);
    end;
    else if (1 − ε ≤ η ≤ 1 + ε)
    /* Optimal Operating Region */
    /* Hold Reporting Frequency */
      f = f;
    end;
end;
```

Fig. 6. Algorithm of the ESRT protocol operation.

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 f and the number of source nodes n. The reporting frequency 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.

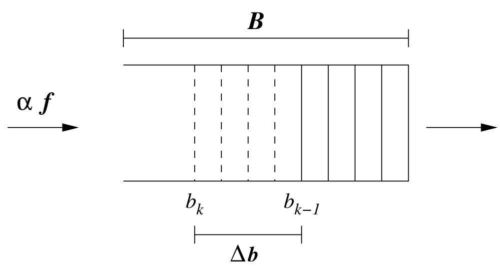

Fig. 7. An illustration of buffer level monitoring in sensor nodes.

Let b_k and b_{k-1} be the buffer fullness levels at the end of k^{th} and $(k-1)^{th}$ reporting intervals respectively and B be the buffer size as in Fig. 7. For a given sensor node, let Δb be the buffer length increment observed at the end of last reporting period, i.e.,

$$\Delta b = b_k - b_{k-1} \tag{6}$$

Thus if the sum of current buffer level at the end of k^{th} 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. 8. This notifies the sink for the upcoming congestion condition to be experienced in next reporting interval.

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

Fig. 8. A typical data packet with congestion notification field, which is marked to alert the sink for congestion.

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

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 this section, we extend ESRT protocol in order to accommodate the cases where multiple events concurrently occur in the same 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 wireless 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. 8. 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. 6 as explained in Section IV.

If the sink receives data packets carrying different event IDs in their Event ID fiels as shown in Fig. 8, it infers that multiple concurrent events occured in the sensor field.

In this case, 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. 9(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. 9(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 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. 8. Here, we assume that Event ID field shown in Fig. 8 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 packet.
   b) If there is not such ID, then the sensor node checks if 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 forwards the packet. If not, it adds the first element of the multidimensional Event ID field of this packet

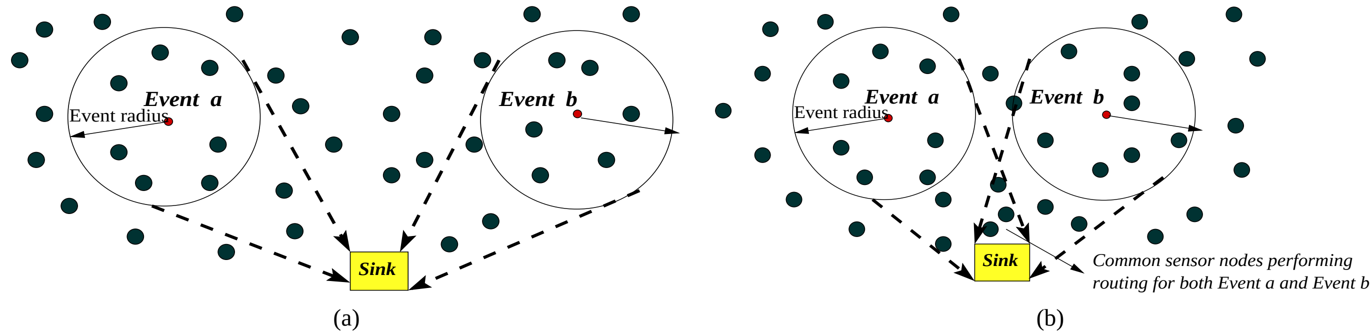

Fig. 9. The multiple event occurrences in the same wireless sensor field (a) the flows generated by two events, i.e., *Event a* and *Event b*, are isolated (b) the flows pass through some common sensor nodes.

into its event-list and leaves the packet intact and forwards it.

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. 9(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 nodes adds a and b in the event ID field of the data packet and then forwards 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.

Note that the multiple event detection mechanism described above does not affect the accurate identification of the events by using the Event ID field in the packet headers. In fact, the first element of the multidimensional Event ID field is the ID of the event which originally generated the data packet. Hence, as the sink receives a data packet whose Event ID field carry multiple event IDs, it uses the first element of the multidimensional Event ID field to associate this data packet to the event in order to accurately calculate the observed event reliability as described in Section IV. Furthermore, note also that the mechanism described above requires only a simple lookup function as the additional functionality at the sensor nodes, and exploits the collective identification of the sensor nodes and avoids the need for individual sensor IDs.

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 wireless sensor field.

If a single event occurs in the wireless 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 **S** to the optimal operating region **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.

B.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. 9(a).

In this case, let $\mathbf{S}_i^k$ and f_i^k be the current network state and the reporting frequency 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 **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.

B.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. 9(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 **(NC,HR)** no congestion is experienced and the observed reliability is higher than required, it is completely opposite in state **(C,LR)** where there is a congestion in the network and the event-to-sink reliability is not achieved as shown in Fig. 4. Hence, the event flows whose current network state are **(C,LR)** have greater urgency and hence high priority in terms of action to be taken by the sink. Similarly, although there is no congestion in both of the states **(NC,LR)** and **(NC,HR)**, the event flows which are currently in state **(NC,LR)** do not receive their desired reliability levels and has higher priority than the ones in state **(NC,HR)**. With this respect, we group the network states {**(C,LR), (NC,LR), (C,HR), (NC,HR)**} into *high priority states*, i.e., **(C,LR), (NC,LR)**, and *low priority states*, i.e., **(C,HR), (NC,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 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 $\mathbf{S}_i^j$=**(C,LR)** or $\mathbf{S}_i^j$=**(NC,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 $\mathbf{S}_i^j$=**(C,LR)** or $\mathbf{S}_i^j$=**(NC,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 $\mathbf{S}_i^j$=**(C,HR)** or $\mathbf{S}_i^j$=**(NC,HR)**.
   This is because the actions taken for the events flows whose network states are high priority (step 2.(a)) may affect these events which already have higher reliability. Therefore, any further simultaneous action to minimize energy expenditure of these flows is avoided to 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., $\mathbf{S}_i^j$=**(C,HR)** or $\mathbf{S}_i^j$=**(NC,HR)** $\forall j \in N_e$, then the sink follows the default ESRT operation described in Section IV for these events. That is, it calculates the updated reporting frequency f_{i+1}^j and broadcasts it to the sensor nodes which are in the event radius of event $j \ \forall j \in N_e$.

The sink repeats these steps until all of the event flows reach to the optimal operating region **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 wireless sensor field.

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 **OOR** starting from any of the other four initial network states $\mathbf{S}_i \in${**(NC,LR), (NC,HR),(C,HR),(C,LR)**}. ESRT is self-configuring in this sense and can hence perform efficiently under random, dynamic topology frequently encountered in WSN applications.

The convergence times presented in this section are derived under the assumption that the r vs.f characteristic does not change appreciably 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 event cases. Recall that ESRT aims to reach state **OOR** starting from any initial state $\mathbf{S}_0$.

Lemma 1: Starting from $\mathbf{S}_0$=**(NC,HR)**, and with linear reliability (η) behavior when the network is not congested, the network state remains unchanged until ESRT converges to state **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 (equation (2))

$$f_{i+1} = \frac{f_i}{2}\left(1 + \frac{1}{\eta_i}\right) \tag{7}$$

Hence,

$$\eta_{i+1} = \frac{1+\eta_i}{2} \tag{8}$$

Since $f_{i+1} < f_i$ from (7), it follows that $\mathbf{S}_i \in$ {**(NC,HR), (NC,LR),OOR**}, $\forall i \geq 0$ until ESRT converges. If possible, let $\mathbf{S}_{i+1}$=**(NC,LR)** when $\mathbf{S}_i$=**(NC,HR)** for some $i \geq 0$ before ESRT converges. Then,

$$\eta_{i+1} = \frac{1+\eta_i}{2} < 1 - \epsilon \tag{9}$$

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

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

Proof: To establish the convergence time, we proceed as follows. Let the j^{th} decision interval be the first one where

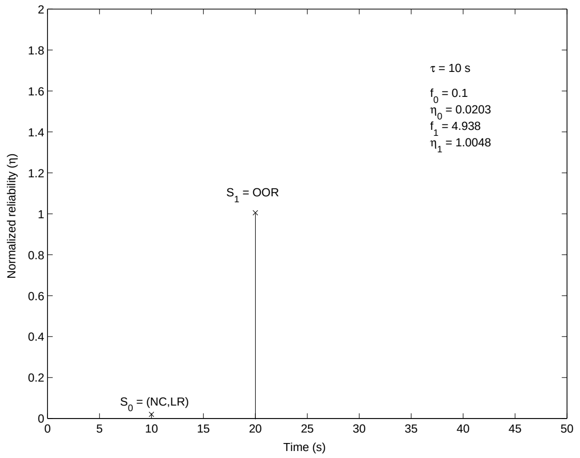

Fig. 10. The ESRT protocol trace for **S**$_0$**=(NC,LR)**. Convergence is attained in a total of two decision intervals. The trace values and states are also shown in the figure.

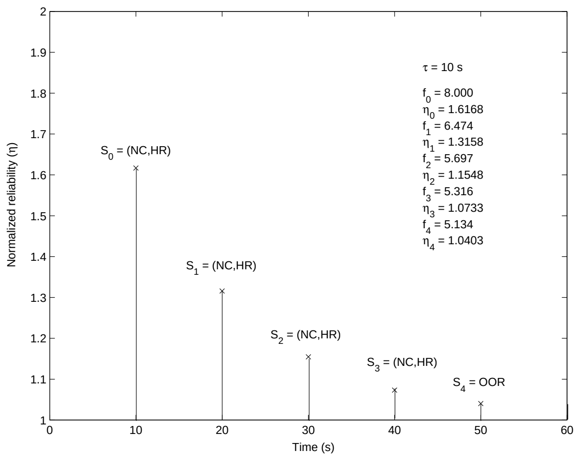

Fig. 11. The ESRT protocol trace for **S**$_0$**=(NC,HR)**. Convergence is attained in a total of five decision intervals. The trace values and states are also shown in the figure.

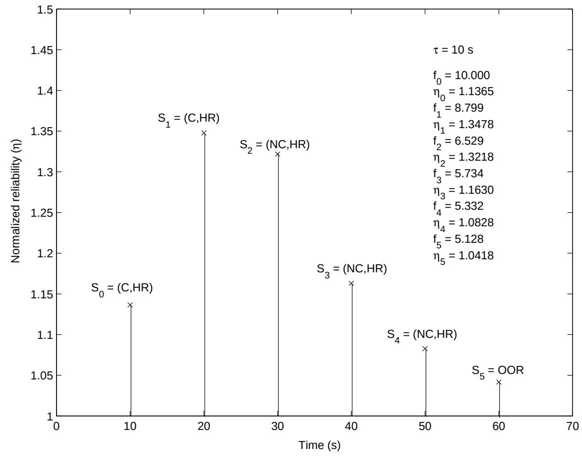

Fig. 12. The ESRT protocol trace for **S**$_0$**=(C,HR)**. Convergence is attained in a total of six decision intervals in this case. The trace values and states are also shown in the figure.

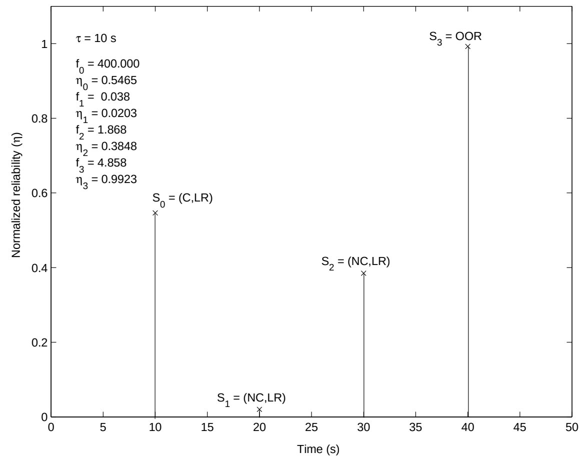

Fig. 13. The ESRT protocol trace for **S**$_0$**=(C,LR)**. Convergence is attained in a total of four decision intervals in this case. The trace values and states are also shown in the figure.

S$_j$**=OOR**. It follows from Lemma 1 that j is the least index such that $\eta_j < 1+\epsilon$. Using equation (8),

$$\begin{array}{rclcl} \eta_j & = & \frac{\eta_{j-1}+1}{2} & < & 1+\epsilon \\ \eta_{j-1} & = & \frac{\eta_{j-2}+1}{2} & < & 1+2\epsilon \\ \vdots & & & & \\ \eta_1 & = & \frac{\eta_0+1}{2} & < & 1+2^{j-1}\epsilon \end{array} \qquad (10)$$

Hence, $j > log_2\left(\frac{\eta_0-1}{\epsilon}\right)$ and the result follows. Note that this represents the time required to reach state **OOR** in order to conserve maximum energy. Our primary objective of reliable event detection is maintained all along by virtue of the conservative decrease (equation (7)). ∎

Lemma 3: With linear reliability (η) behavior when the network is not congested, the network state transition **S**$_i$**= (C,HR)**→**S**$_{i+1}$**=(NC,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 vs. f characteristics in Figs. 2, 3, and 4, that for every $f > f_{max}$ in state **(C,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 **S**$_{i+1}$**=(NC,LR)** when **S**$_i$**=(C,HR)**, for some $i \geq 0$. From the state definitions in Section III-C and update policy in Section IV-A, it follows that

$$f'_i \frac{(1-\epsilon)}{\eta_i} > \frac{f_i}{\eta_i} \qquad (11)$$

Hence, a necessary condition is

$$f'_i > \frac{f_i}{1-\epsilon} > f_i, \qquad (12)$$

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 **(C,HR)** to **(NC,LR)** in the state diagram shown in Fig. 5. 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 **S**$_0 \in \{$**(C,HR),(C,LR)**$\}$, the non-linear r vs. f behavior needs to be tracked analytically. However, this

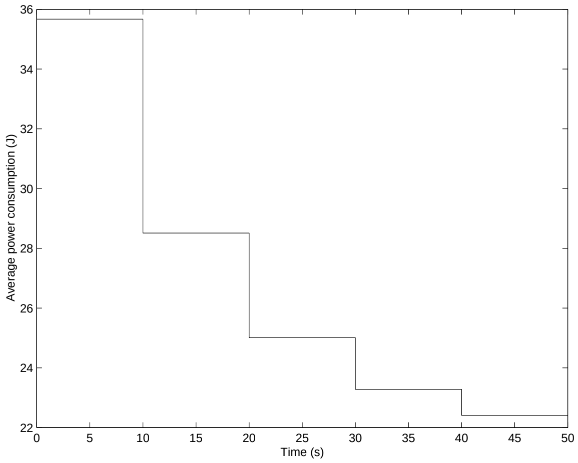

Fig. 14. The average power consumption of sensor nodes in each decision interval for $\mathbf{S}_0$=**(NC,HR)**.

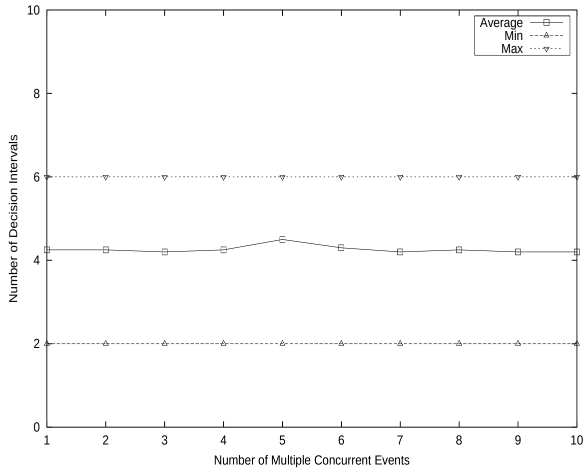

Fig. 15. The number of decision intervals for all of the event flows to converge to state **OOR** for varying number of multiple concurrent events. In this set of experiments, the multiple concurrent events are isolated and their flows do not pass through any common router sensor node.

is beyond our present scope. Hence, we study 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 [11]. We first run the simulation experiments for the scenario where a single event occurs in the wireless sensor field. Our convergence results are shown in Figs. 10 through 13 for initial network states $\mathbf{S}_0$=**(NC,LR)**,**(NC, HR)**,**(C,HR)**, and **(C,LR)**, respectively. The corresponding trace values (f_i, η_i) and states are listed within each figure. The energy conservation property of ESRT for $\mathbf{S}_0$=**(NC, HR)** is illustrated in Fig. 14. For all our simulation results presented here, number of senders $n = 81$ and tolerance $\epsilon = 5\%$. The event radius was fixed at $40m$. Other simulation parameters are the same as those listed in Table 1 in Section III-B.

It is seen from Fig. 10 that the ESRT protocol for $\mathbf{S}_0$=**(NC, LR)** converges in a total of two decision intervals (2τ=20s). 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 Figs. 11 and 12.

Furthermore, we run simulation experiments to assess the ESRT performance in the cases where multiple events occur simultaneously in the sensor field. Here, we observe the number of intervals it takes for all of the event flows to converge to state **OOR**. We also observe the average power consumption of the sensor nodes. We perform the simulation experiments 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 shown in Fig. 15, the average number of decision intervals it takes for all of the event flows to converge to the state **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 **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 **OOR** state is not delayed in the case of multiple isolated events.

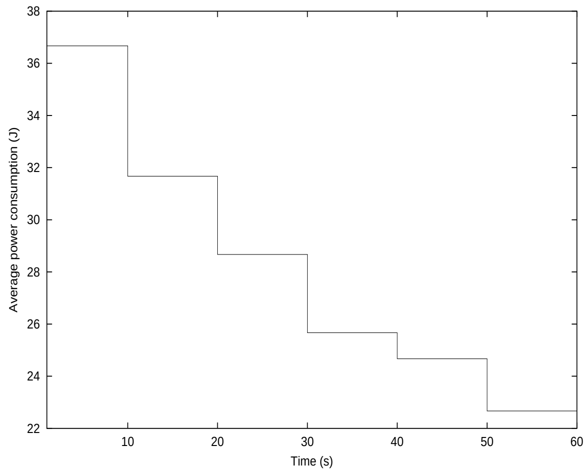

Fig. 16. The average power consumption of sensor nodes in each decision interval for the case where 5 concurrent events occur in the wireless sensor field. In this case, the flows generated by these events are isolated.

Moreover, as shown in Fig. 16, the average power consumed by the sensor nodes also show the same pattern we observed for a single event scenario as shown in Fig. 14. 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. 17, the average number of decision intervals it takes

TABLE IV
SUMMARY OF ESRT PROTOCOL OPERATION IN EACH OF THE FIVE STATES

Network State (S_i)	Description	ESRT Action
(NC,LR)	No Congestion, Low Reliability	Multiplicatively increase f Achieve required reliability as soon as possible
(NC,HR)	No Congestion, High Reliability	Decrease f conservatively Cautiously reduce energy consumption so as not compromise on reliability
(C,HR)	Congestion, High Reliability	Decrease f aggressively to state **(NC,HR)** to relieve congestion Then follow action in **(NC,HR)**
(C,LR)	Congestion, Low/equal Reliability	Decrease f exponentially Relieve congestion as soon as possible
OOR	Optimal Operating Region	f remains unchanged

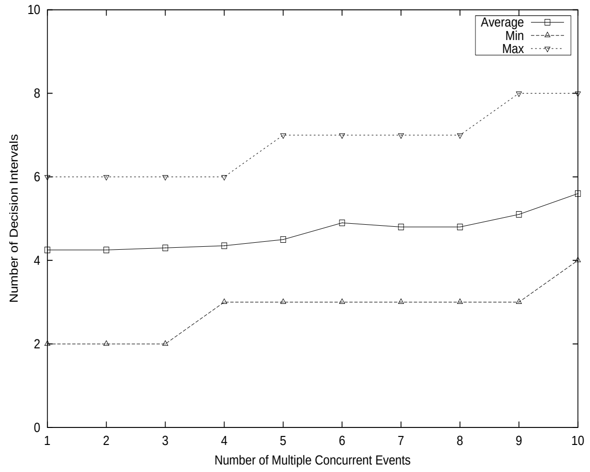

Fig. 17. The number of decision intervals for all of the event flows to converge to state **OOR** for varying number of multiple concurrent events. In this set of experiments, the multiple concurrent events are not isolated.

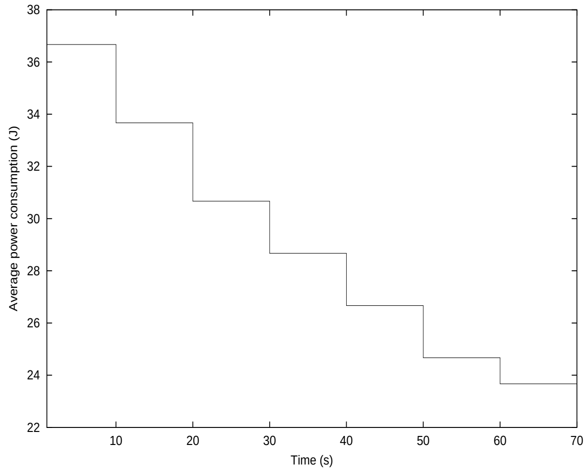

Fig. 18. The average power consumption of sensor nodes in each decision interval for the case where 5 concurrent events occur in the wireless sensor field. In this case, the flows generated by these events are not isolated.

for all of the event flows to converge state **OOR** slightly increases with the number of multiple 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. Consequently, 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. 17, the increase in the convergence time is very small even in case of 10 non-isolated concurrent events. Hence, the ESRT protocol can effectively address the cases where multiple events occur simultaneously.

Furthermore, as shown in Fig. 18, the average power consumed by the sensor nodes also show the same pattern we observed for the previous case in Fig. 16. However, the decrease in the average consumed power is slightly slower in this case. This is also 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 shown in Fig. 17.

C. Suitable Choice of ϵ

For practical purposes, ESRT uses a tolerance zone of ϵ around the optimal operating point P_1 in Fig. 4. If at the end of decision interval i, the reliability η_i is within [1-ϵ,1+ϵ] and if no congestion is detected in the network, then the network is in state **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 VI-A, smaller the ϵ, greater 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 $\mathbf{S}_0$=**(NC,HR)**. Note however that reliable event detection is maintained all along (Lemma 2 in Section VI-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. This is due to the fact that the sink is only interested in the collective information of a number of source nodes and not in individual sensor reports. This is also the reason why traditional end-to-end reliability notions and transport solutions are inappropriate for 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.

ESRT is a novel transport solution developed to achieve reliable event detection with minimum energy expenditure and congestion resolution functionality. To the best of our knowledge, this is the first study of reliable transport in WSN from the event-to-sink perspective.

ESRT has been tailored to meet the unique requirements of WSN. Its congestion control component serves the dual purpose of achieving reliability and conserving energy. The algorithms of ESRT mainly run on the 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. Thus, ESRT protocol operation is determined by the current network state based on the reliability achieved and the congestion condition. In this regard, five possible network states $\mathbf{S}_i \in \{$**(NC,LR),(NC,HR),(C,HR),(C,LR),OOR**$\}$ were identified and ESRT operation in each of these states was discussed in detail in Section IV-A. The main ideas are summarized in Table 4.

We also extend ESRT protocol operations to accommodate the scenarios where multiple events concurrently occur in the wireless sensor field. The sink exploits the collective identification of the sensor nodes in order to accurately capture if a single or multiple events occur and in case of multiple events whether the flows generated by these events are isolated or not. Hence, according to this information, the sink uses the ESRT protocol to achieve the required event-to-sink reliability levels for each of these concurrent events.

Analytical performance evaluation and simulation results show that ESRT converges to state **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. This self-configuring aspect of ESRT is valuable under random, dynamic topology frequently encountered in WSN applications.

REFERENCES

[1] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", *IEEE/ACM Trans. Networking*, Vol. 5, No. 6, pp. 756-769, December 1997.

[2] S. Floyd, V. Jacobson, C. Liu, S. Macanne, and L. Zhang, "A Reliable Multicast Framework for Lightweight Sessions and Application Level Framing," *IEEE/ACM Trans. Networking*, Vol. 5, No. 6, pp.784-803, Dec. 1997.

[3] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *Proc. ACM MOBICOM 2000,* pp. 56-67, Boston, Massachussetts, August 2002.

[4] D. Johnson, D. Maltz, Y. Hu, and J. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," *http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-07.txt,* IETF MANET working group, Internet draft, February 2002.

[5] MICA Motes and Sensors, `\http://www.xbow.com/Products/Wireless_Sensor_Networks.htm`

[6] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz, "ESRT: Event-to-Sink Reliable Transport for Wireless Sensor Networks," in *Proc. ACM MOBIHOC 2003*, pp. 177-188, Annapolis, Maryland, USA, June 2003.

[7] S. D. Servetto and G. Barrenechea, "Constrained Random Walks on Random Graphs: Routing Algorithms for Large Scale Wireless Sensor Networks," *In Proc. WSNA 2002,* pp. 12-21, Atlanta, GA, September 2002.

[8] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks," in *Proc. ACM MOBICOM 2001,* pp. 272-286, Rome, Italy, July 2001.

[9] K. Sohrabi, B. Manriquez, and G. Pottie, "Near-Ground Wideband Channel Measurements," in *Proc. IEEE VTC 1999,* Vol. 1, pp. 571-574, New York, 1999.

[10] F. Stann and J. Heidemann, "RMST: Reliable Data Transport in Sensor Networks," in *Proc. IEEE SNPA 2003*, pp. 102-112, Anchorage, Alaska, May 2003.

[11] The Network Simulator - ns-2, `http://www.isi.edu/nsnam/ns/index.html`

[12] S. Tilak, N. B Abu-Ghazaleh, and W. Heinzelman, "Infrastructure Tradeoffs for Sensor Networks," in *Proc. ACM WSNA 2002*, pp. 49-58, September 2002, Atlanta, GA, USA.

[13] C. Y. Wan, A. T. Campbell, and L. Krishnamurthy, "PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks," in *Proc. ACM WSNA 2002,* pp. 1-11, September 2002, Atlanta, GA, USA.

[14] A. Woo and D. E. Culler, "A Transmission Control Scheme for Media Access in Sensor Networks," in *Proc. ACM MOBICOM 2001,* pp.221-235, Rome, Italy 2001.

[15] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proc. INFOCOM'02,* pp. 1567-1576, New York, USA, June, 2002.