INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2007; **20**:1155–1177
Published online 22 January 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/dac.869

Delay aware reliable transport in wireless sensor networks

Vehbi C. Gungor[1,*,†] and Özgür B. Akan[2]

[1] *School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, U.S.A.*
[2] *Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara 06531, Turkey*

SUMMARY

Wireless sensor networks (WSN) are event-based systems that rely on the collective effort of several sensor nodes. Reliable event detection at the sink is based on collective information provided by the sensor nodes and not on any individual sensor data. 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. Moreover, the reliability objective of WSN must be achieved within a certain real-time delay bound posed by the application. Therefore, the WSN paradigm necessitates a collective delay-constrained event-to-sink reliability notion rather than the traditional end-to-end reliability approaches. To the best of our knowledge, there is no transport protocol solution which addresses both reliability and real-time delay bound requirements of WSN simultaneously.

In this paper, the delay aware reliable transport (DART) protocol is presented for WSN. The objective of the DART protocol is to timely and reliably transport event features from the sensor field to the sink with minimum energy consumption. In this regard, the DART protocol simultaneously addresses congestion control and timely event transport reliability objectives in WSN. In addition to its efficient congestion detection and control algorithms, it incorporates the time critical event first (TCEF) scheduling mechanism to meet the application-specific delay bounds at the sink node. Importantly, the algorithms of the DART protocol mainly run on resource rich sink node, with minimal functionality required at resource constrained sensor nodes. Furthermore, the DART protocol can accommodate multiple concurrent event occurrences in a wireless sensor field. Performance evaluation *via* simulation experiments show that the DART protocol achieves high performance in terms of real-time communication requirements, reliable event detection and energy consumption in WSN.

Received 2 October 2006; Accepted 7 November 2006

KEY WORDS: wireless sensor networks; real-time and reliable transport; congestion detection and control

*Correspondence to: Vehbi C. Gungor, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, U.S.A.
† E-mail: gungor@ece.gatech.edu

Contract/grant sponsor: National Science Foundation; contract/grant number: ECS-0428329
Contract/grant sponsor: Turkish Scientific and Technical Research Council (TUBITAK) Career Award; contract/grant number: KARIYER-104E043

1. INTRODUCTION

Wireless sensor networks (WSN) are, in general, comprised of large number of low-cost, low-power, multifunctional sensor nodes which collaboratively work towards achieving the application-specific objectives through short distance and multi-hop communications. In fact, the existing and potential applications of WSN span a very wide range including homeland security, environmental monitoring, biomedical research, human imaging and tracking and military applications [1]. The practical realization of these currently designed and envisioned applications, however, directly depends on the efficient real-time and reliable communication capabilities of the deployed sensor network.

Recently, there has been considerable amount of research efforts which have yielded many promising communication protocols to address the challenges posed by the WSN paradigm [1]. The common feature of these research results is that they mainly address the energy-efficient and reliable data communication requirements of WSN. However, in addition to the energy-efficiency and communication reliability, there exist many proposed *WSN applications which have strict delay bounds and hence mandate timely transport of the event features from the field*.

Many of the potential WSN applications such as real-time target tracking, homeland security, process control, controlling the vehicle traffic in highways necessitate the reliable event transport to be achieved within a certain application-specific delay bound. For instance, the accuracy and effectiveness of military WSN applications such as border surveillance and intrusion detection are directly related to the timeliness of the reliable event detection at the sink, e.g. military decision centre. Clearly, late detection of a certain event at the sink leads to the failure of the ultimate objectives of the deployed WSN for such applications. Therefore, the communication protocols, which only consider the energy-efficiency and transport reliability, are deemed to be incapable of addressing the needs of applications, which have certain delay bounds.

Consequently, to assure reliable and timely event detection in WSN, reliable event transport to the sink node within a certain delay bound must be effectively handled by an efficient transport protocol mechanism. Several transport protocols have been developed for sensor networks in recent years [1]. These protocols are mainly designed for congestion control and reliable data delivery from the sink to the sensor nodes [2, 3] and from the sensor nodes to the sink [4–6]. However, none of these protocols address the application-specific real-time delay bounds of the reliable event transport in WSN. Clearly, there is an urgent need for a new real-time and reliable data transport solution with efficient congestion detection and control mechanisms for WSN.

To address this need, in this paper, the delay aware reliable transport (DART) protocol is introduced for WSN. The DART protocol is a novel transport solution that seeks to achieve reliable and timely event detection with minimum energy consumption and no congestion. It enables the applications to perform right actions timely by exploiting both the correlation and the collaborative nature of sensor networks. To achieve this objective, based on event transport reliability and event-to-sink delay bound, *a delay-constrained event-to-sink reliability* notion is defined.

We emphasize that the DART protocol has been designed for use in typical WSN applications involving event detection and signal estimation/tracking within a certain delay bound, and not for guaranteed end-to-end data delivery services. Our work is motivated by the fact that the sink is only interested in timely and reliable detection of event features from the collective information provided by numerous sensor nodes and not in their individual reports, as

DOI: 10.1002/dac

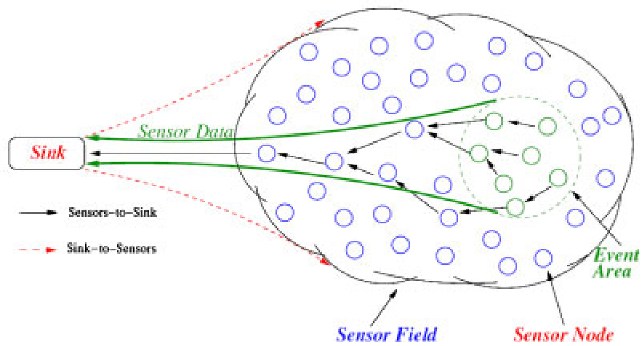

Figure 1. Typical wireless sensor network architecture. The sink is only interested in collective information of sensor nodes within certain delay bound and not in their individual data.

illustrated in Figure 1. The notion of delay-constrained event-to-sink reliability distinguishes the DART protocol 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 and this is the first research effort focusing on real-time and reliable event data transport with minimum energy consumption in WSN.

In this paper, we have also extended our work in [7] by enhancing the protocol operations in order to accommodate the scenarios where multiple concurrent events occur in the wireless sensor field. Such an enhancement is important since the data flows generated by the multiple events occurring simultaneously may not be always isolated in the wireless sensor network. Thus, uncoordinated protocol actions may fail to achieve required delay-constrained event-to-sink 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 delay-constrained event-to-sink reliability with minimum energy expenditure for all of the multiple concurrent events in the sensor field. The description of DART protocol mechanisms for handling multiple concurrent events, and the DART performance study corresponding to these scenarios can be found in Sections 6 and 7, respectively.

The remainder of the paper is organized as follows. In Section 2, we present a review of related work in transport protocols in WSN and point out their inadequacies. In Section 3, we describe the design principles and functionalities of the DART protocol in detail. In Section 4, we explain combined congestion detection and control mechanism of the DART protocol. The detailed protocol operation of the DART is described in Section 5. In Section 6, we explain how the DART protocol operation accommodates the scenarios where multiple concurrent events occur in the wireless sensor field. Performance evaluation and simulation results are presented in Section 7. Finally, the paper is concluded in Section 8.

2. RELATED WORK

Despite the considerable amount of research on several aspects of sensor networking, the problems of real-time and reliable transport and congestion detection and control are yet to be

DOI: 10.1002/dac

efficiently studied and addressed. The urgent need for an efficient congestion detection and control in WSN is pointed out in [8, 9]. In these works, it was shown that exceeding network capacity can be detrimental to the observed throughput. However, these research efforts do not propose any specific solution for the problem they identify.

In another recent work [3], the pump slowly fetch quickly (PSFQ) mechanism is proposed for reliable retasking/reprogramming of the sensors based on strict reliability measures in WSN. However, PSFQ does not address packet losses due to congestion; and also its use for the forward direction can lead to a waste of valuable resources. In [5], the reliable multi-segment transport (RMST) protocol is proposed to utilize in-network caching and provide guaranteed delivery of the data packets generated by the event flows. However, RMST is also based on end-to-end guaranteed reliability *via* in-network caching, which may bring significant overhead. The congestion detection and avoidance (CODA) protocol for sensor networks is presented in [6]. However, the CODA does not address the reliable event transport to the sink in WSN [6]. On the contrary, it has been observed in the experiment results [6] that the congestion control performed at the sensor nodes without considering the reliability objective impairs the event-to-sink transport reliability. A new framework called GARUDA for providing sink-to-sensors reliability in WSN is introduced in [2]. The GARUDA incorporates an efficient pulsing-based solution, which informs the sensor nodes about an impending reliable short-message delivery by transmitting a specific series of pulses at a certain amplitude and period. However, the GARUDA does not address sensors-to-sink reliability. In contrast to the transport layer protocols for conventional end-to-end reliability, event-to-sink reliable transport (ESRT) protocol is developed in [4], which is based on the event-to-sink reliability notion and provides reliable event detection without any intermediate caching requirements and with minimum energy expenditure in WSN. Despite ESRT includes an efficient congestion control component that serves the dual purpose of achieving reliability and conserving energy, it does not take application specific real-time delay bounds into account while avoiding network congestion.

A novel transmission control scheme for use at the MAC layer in WSN is proposed in [10] 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 [11] that provide some form of hop reliability is inadequate for reliable event detection in WSN.

In addition, transport solutions in wireless *ad hoc* networks [12–15] 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 [16]) and processing capabilities.

In summary, although there exist the above discussed transport layer solutions for wireless *ad hoc* and sensor networks, none of them addresses the problem of both reliability and real-time delay bound requirements simultaneously. Clearly, there is a need for real-time and reliable transport layer solution in WSNs that emphasizes on collective reliability, resource efficiency and simplicity.

DOI: 10.1002/dac

3. DART PROTOCOL DESIGN

In the following sections, we first discuss the main design components of the DART protocol in detail and then we present a case study to gain more insight regarding the challenges of WSN.

3.1. Reliable event transport mechanism

In WSNs, sensors-to-sink transport does not require 100% reliability due to the correlation among the sensor readings [4, 17]. Hence, conventional end-to-end reliability definitions and solutions would only lead to over-utilization of scarce sensor resources. On the other hand, the absence of reliable transport mechanism altogether can seriously impair event detection. Thus, the sensors-to-sink transport paradigm requires a collective event-to-sink reliability notion rather than the traditional end-to-end reliability notion. The DART protocol also considers the new notion of *event-to-sink delay bound* (described in Section 3.2) to meet the application deadlines for proper operation of the deployed network. Based on both event-to-sink reliability and event-to-sink delay bound notions, we introduce the following definitions:

- The *observed delay-constrained event reliability* (DR_i) is the number of received data packets within a certain delay bound at the sink in a decision interval i.
- The *desired delay-constrained event reliability* (DR^*) is the minimum number of data packets required for reliable event detection within a certain application-specific delay bound. This lower bound for the reliability level is determined by the application and based on the physical characteristics of the event signal being tracked.
- The *delay-constrained reliability indicator* (δ_i) is the ratio of the observed and desired delay-constrained event reliabilities, i.e. $\delta_i = \mathrm{DR}_i/\mathrm{DR}^*$.

Based on the packets generated by the sensor nodes in the event area, the event features are estimated and DR_i is observed at each decision interval i to determine the necessary action. If the observed delay-constrained event reliability is higher than the reliability bound, i.e. $\mathrm{DR}_i > \mathrm{DR}^*$, then the event is deemed to be reliably detected within a certain delay bound. Otherwise, appropriate action needs to be taken to assure the desired reliability level in the event-to-sink communication. For example, to increase the amount of information transported from the sensors to the sink, reporting frequency of the sensors can be increased properly while avoiding congestion in the network. Therefore, sensors-to-sink transport reliability problem in WSN is to *configure the reporting rate*, f, *of source nodes so as to achieve the required event detection reliability*, DR^*, *at the sink within the application-specific delay bound.*

3.2. Real-time event transport mechanism

To assure reliable and timely event detection, it is imperative that the event features are reliably transported to the sink node within a certain delay bound. We call this *event-to-sink delay bound*, Δ_{e2s}, which is specific to application requirements and must be met so that the application-specific objectives of the sensor network operation are achieved. The event-to-sink delay bound has two main components as outlined below:

1. *Event transport delay* (Γ^{tran}): It is mainly defined as the time between when the event occurs and when it is reliably transported to the sink node. Therefore, it involves the following delay components:
   - (a) *Buffering delay* ($t_{b,i}$): It is the time spent by a data packet in the routing queue of an intermediate forwarding sensor node i. It depends on the current network load and transmission rate of each sensor node.
   - (b) *Channel access delay* ($t_{c,i}$): It is the time spent by the sensor node i to capture the channel for transmission of the data packet generated by the detection of the event. It depends on the channel access scheme in use, node density and the current network load.
   - (c) *Transmission delay* ($t_{t,i}$): It is the time spent by the sensor node i to transmit the data packet over the wireless channel. It can be calculated using transmission rate and the length of the data packet.
   - (d) *Propagation delay* ($t_{p,i}$): It is the propagation latency of the data packet to reach the next hop over the wireless channel. It mainly depends on the distance and channel conditions between the sender and receiver.
2. *Event processing delay* (Γ^{proc}): This is the processing delay experienced at the sink, when the desired features of event are estimated using the data packets received from the sensor field. This may include a certain decision interval [4] during which the sink waits to receive adequate samples from the sensors.

Let Δ_{e2s} be the event-to-sink delay bound for the data packet generated by the detection of event. Then, for a reliable and timely event detection, it is necessary that

$$\Delta_{e2s} \geqslant \Gamma^{tran} + \Gamma^{proc} \tag{1}$$

is satisfied. Here, Γ^{tran} is clearly a function of $t_{b,i}$, $t_{c,i}$, $t_{t,i}$, $t_{p,i}$, and $\hat{N}$, where $\hat{N}$ is the average hop count from the source nodes to the sink node. Note that Γ^{tran} is directly affected by the current network load and the congestion level in the network. In addition, the network load depends on the event reporting frequency, f, which is used by the sensor nodes to send their readings of the event. Hence, the main delay component that depends on the congestion control and thus, can be controlled to a certain extent is the event transport delay, i.e. Γ^{tran}. More specifically, the buffering delay, i.e. $t_{b,i}$, directly depends on the transport rate of the event and the queue management and service discipline employed at each sensor node i. In addition, for the events occurring at further distances to the sink node, the average number of hops that event data packets traverse, $\hat{N}$, increases. Thus, it is more difficult to provide event-to-sink delay bound for further event packets compared to closer ones. Considering that the per-hop propagation delay, $t_{p,i}$, does not vary,[‡] then the buffering delay, $t_{b,i}$, must be controlled, i.e. decreased, in order to compensate the increase in the event transport delay so that the event-to-sink delay bound is met.

‡While the channel access delay can also be controlled to a certain extent *via* priority-based QoS-aware MAC protocols [1], we do not assume the presence of such MAC protocol.

To accomplish this objective, the DART protocol introduces *time critical event first* (TCEF) scheduling policy. In fact, TCEF policy applies the general principles of earliest deadline first service discipline on each sensor node, which is shown to be the optimal scheduling policy when real-time deadlines of the system are considered [18, 19]. However, we also integrate some novel mechanisms so as to fit it to unique challenges of sensor networks. For example, to update the remaining time to deadline without a globally synchronized clock in the network, we measure the elapsed time at each sensor and piggyback the elapsed time to the event packet so that the following sensor can determine the remaining time to deadline without a globally synchronized clock. Then, by using these elapsed time measurements and service index assignments of TCEF policy, the event packets are given high priority at the sensor nodes, as their remaining time to deadline decreases. This way, time critical sensor data obtain high priority along the path from the event area to the sink node and is served first, which is crucial to meet the application deadlines. The details of elapsed time measurement and service index assignment mechanisms of TCEF scheduling are omitted due to the lack of space.

Note that although TCEF policy makes it possible to meet deadlines in the normal operating conditions of the network, in case of severe network congestion, it may become insufficient to provide delay-constrained event reliability. Hence, in addition to TCEF scheduling, the DART protocol considers the event-to-sink delay bounds and congestion conditions in its reporting rate update policies to assure timely and reliable event transport (see Section 5). In the following, we present a case study to gain more insight regarding the communication challenges of sensor network.

3.3. Case study

To investigate the relationship between the event-to-sink delay and the event reporting rate, we develop an evaluation environment using *ns*-2 [20]. The parameters used in our case study are listed in Table I. Event centres (X_{ev}, Y_{ev}) were randomly chosen and all sensor nodes within the event radius behave as sources for that event. In this case study, the sink node receiving the data is placed in the middle of the lower side of the deployment area. To communicate source data to the sink node, we employed a simple CSMA/CA-based MAC protocol and dynamic source routing (DSR) [21]. For each simulation, we run 10 experiments and take the average of the measured values.

First, we investigate the impact of event reporting frequency on average event-to-sink delay and *on-time event delivery ratio*. Here, on-time event delivery ratio represents the fraction

Table I. NS-2 simulation parameters.

Area of sensor field	$200 \times 200\,m^2$
Number of sensor nodes	200
Radio range of a sensor node	20 m
Packet length	30 bytes
Buffer length	65 packets
Transmit power	0.660 W
Receive power	0.395 W
Idle power	0.035 W
Decision interval (τ)	1 s

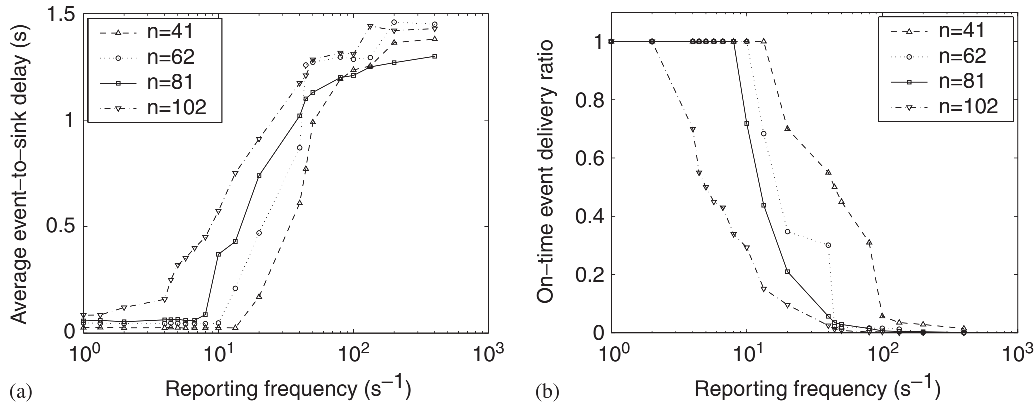

Figure 2. The effect of varying reporting frequency of source nodes on: (a) average event-to-sink delay; and (b) on-time event delivery ratio. The number of source nodes is denoted by n.

of data packets received within event-to-sink delay bound (which we refer to reliable packets) over all data packets received in a decision interval. The results of our study are shown in Figure 2 for different number of source nodes, i.e. $n = 41, 62, 81, 102$. Note that each of these curves was obtained by varying the event reporting frequency, f, for a randomly chosen event centre (X_{ev}, Y_{ev}) and corresponding number of sources, n. These values are tabulated in Table II.

As shown in Figure 2(a) and (b), it is observed that as the event reporting frequency, f, increases, average event-to-sink delay remains constant and on-time event delivery is ensured, until a certain $f = f_{max}$ at which network congestion is experienced. After this point, the average event-to-sink delay starts to increase and on-time event delivery cannot be provided. This is obvious because the increased network load due to higher reporting frequency leads to increase in the buffer occupancy and network channel contention. Moreover, as the number of sources increases, on-time event delivery ratio cannot be provided even at lower reporting frequencies.

To further elaborate the relationship between observed delay-constrained event reliability, DR_i, and the event reporting frequency, f, we have observed the number of packets received at the sink node in a decision interval, τ. As shown in Figure 3, until a certain $f = f_{max}$, observed delay-constrained event reliability and no delay-constrained event reliability[§] coincides, beyond which delay-constrained event reliability significantly deviates from no delay-constrained event reliability. Furthermore, the observed delay-constrained event reliability, DR_i, shows a linear increase (note the log scale) with source reporting rate, f, until a certain $f = f_{max}$, beyond which the observed delay-constrained event reliability drops. This is because the network is unable to handle the increased injection of data packets and packets are dropped because of network congestion and contention. Note that such an initial increase and a subsequent decrease in observed delay-constrained event reliability is observed regardless of the number of source nodes, n. It is also important to note that f_{max} decreases with increasing n, i.e. network

[§]No delay-constrained event reliability represents the number of event packets received at the sink node irrespective of their packet delay.

Table II. Randomly selected event centres used in the simulations.

Number of source nodes	Event centre (X_{ev}, Y_{ev})	Event radius
41	(75.2, 72.3)	30 m
62	(52.1, 149.3)	30 m
81	(59.2, 68.1)	40 m
102	(90.6, 119.1)	40 m

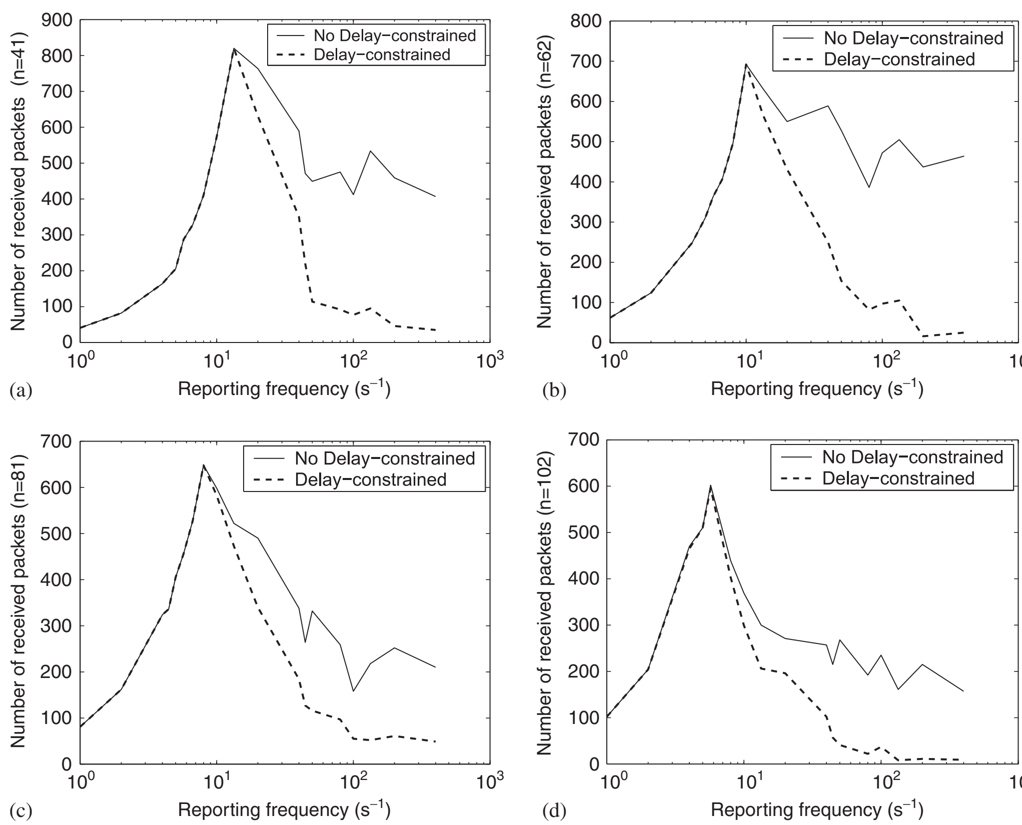

Figure 3. The number of received packets at the sink in a decision interval, when the number of sources: (a) $n = 41$; (b) $n = 62$; (c) $n = 81$; and (d) $n = 102$.

congestion occurs at lower reporting frequencies with greater number of source nodes. In addition, for $f > f_{max}$, the DR_i *versus* f behaviour is rather wavy as seen in Figure 3. An intuitive explanation for such a behaviour is as follows. The number of received packets within a certain delay bound, which is our delay-constrained event reliability, DR, 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.

DOI: 10.1002/dac

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 behaviour always stays well below the maximum delay-constrained event reliability at $f=f_{\max}$.

Furthermore, the evaluation scenarios explored here represent densely deployment cases where congestion is more likely to occur. As it is observed from Figure 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 DR *versus* f behaviour 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 DR *versus* f behaviour for different event radiuses and number of source nodes. Consequently, the results obtained here apply to the cases with lower densities as well.

In summary, with increasing reporting frequency, f, a general trend of an initial increase and a subsequent decrease (due to network congestion) in delay-constrained event reliability is observed in our preliminary studies, as shown in Figure 3. Furthermore, when the application specific event-to-sink delay bound is considered, the observed delay-constrained event reliability decreases significantly with the network congestion. These observations confirm the urgent need for a delay-constrained event-to-sink reliable transport solution with an efficient congestion detection and control mechanism in WSN. In the following section, combined congestion detection mechanism of the DART protocol is described in detail.

4. CONGESTION DETECTION AND CONTROL

In WSNs, because of the memory limitations of the sensor nodes and limited capacity of shared wireless medium, congestion might be experienced in the network. Congestion leads to both waste of communication and energy resources of the sensor nodes and also hampers the event detection reliability because of packet losses [4, 9]. Hence, it is mandatory to address the congestion in the sensor field to achieve reliable event detection and minimize energy consumption.

However, the conventional sender-based congestion detection methods for end-to-end congestion control purposes cannot be applied here. The reason lies in the notion of delay-constrained event reliability rather than end-to-end reliability. Only the sink node, and not any of the sensor nodes, can determine the delay-constrained reliability indicator $\delta_i = \mathrm{DR}_i/\mathrm{DR}^*$, and act accordingly. In addition, for efficient congestion detection in WSNs, the sensor nodes should be aware of the network channel condition around them, since the communication medium is shared and might be congested with the network traffic among other sensor nodes in the neighbourhood [8]. Therefore, because of shared communication medium nature of WSNs, the sensor nodes can experience congestion even if their buffer occupancy is small.

In this regard, the DART protocol uses a *combined congestion detection* mechanism based on both average node delay calculation and local buffer level monitoring of the sensor nodes to accurately detect congestion in the network. Note that average node delay at the sensor node gives an idea about the contention around the sensor node, i.e. how busy the surrounding vicinity of the sensor node. To compute the average node delay at the sensor i, the sensor takes exponential weighted moving average of the elapsed time measurements (see Section 3.2).

In combined congestion detection mechanism of the DART protocol, any sensor node whose buffer overflows due to excessive incoming packets or average node delay is above a certain

DOI: 10.1002/dac

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

Figure 4. A typical data packet with congestion notification (CN) bit.

delay threshold value is said to be congested and it informs the congestion situation to the sink node.[¶] More specifically, the sink node is notified by the upcoming congestion condition in the network by utilizing the *congestion notification* (CN) bit in the header of the event packet as shown in Figure 4. Therefore, if the sink node receives event packets whose CN bit is marked, it infers that congestion is experienced in the last decision interval. In conjunction with the delay-constrained reliability indicator, δ_i, the sink can determine the current network state and adjust the reporting frequency of the sensors.

Importantly, the algorithms of the DART protocol mainly run on resource-rich sink node, with minimal functionality required at resource-constrained sensor nodes. To this end, the DART protocol uses sink broadcast to communicate the updated reporting frequency to the sensor nodes in order to avoid any feedback latency problem as well as to save scarce sensor energy resources, which would be wasted if the new reporting frequency were communicated in a multi-hop manner.

5. REPORTING FREQUENCY UPDATE POLICIES

In the previous sections, based on delay-constrained event reliability and event-to-sink delay bound notions, we had defined a new delay-constrained reliability indicator $\delta_i = \mathrm{DR}_i/\mathrm{DR}^*$, i.e. the ratio of observed and desired delay-constrained event reliabilities. To determine proper event reporting frequency update policies, we also define T_i, which is the amount of time needed to provide delay-constrained event reliability for a decision interval i. In conjunction with the CN information (CN bit) and the values of f_i, δ_i and T_i, the sink node calculates the updated reporting frequency, f_{i+1}, to be broadcast to source nodes in each decision interval. This updating process is repeated until the optimal operating point is found, i.e. adequate reliability and no congestion state is obtained. Note that in WSNs the DR_i *versus* f characteristics 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. In the following sections, we describe the details of the reporting frequency update policies.

5.1. Early reliability and no congestion (ER,NC) state

In this state, the required reliability level specific to application is reached before the event-to-sink delay bound, i.e. $T_i \leqslant \Delta_{\mathrm{e2s}}$. Also, no congestion is observed at the sink, i.e. $\mathrm{CN} = 0$. However, the observed delay-constrained event reliability, DR_i, is larger than desired delay-constrained event reliability, DR^*. This is because source nodes transmit event data more

[¶] To avoid reacting to transient network behaviour and to increase the accuracy of congestion detection, the DART protocol detects congestion, if the node delay measurements exceed a delay threshold more than a certain number of successive times.

DOI: 10.1002/dac

frequently than required. The most important consequence of this state is excessive energy consumption of the sensors. Therefore, the reporting frequency should be decreased cautiously to conserve energy. This reduction should be performed cautiously so that the delay-constrained event-to-sink reliability is always maintained. Thus, the sink decreases the reporting frequency in a controlled manner. Intuitively, we try to find a balance between saving energy and maintaining reliability. Hence, the updated reporting frequency can be expressed as follows:

$$f_{i+1} = f_i \frac{T_i}{\Delta_{\text{e2s}}} \tag{2}$$

5.2. *Early reliability and congestion (ER,C) state*

In this state, the required reliability level specific to application is reached before the event-to-sink delay bound, i.e. $T_i < \Delta_{\text{e2s}}$. Also, congestion is observed at the sink, i.e. $\text{CN} = 1$. However, the observed delay-constrained event reliability, DR_i, is larger than desired delay-constrained event reliability, DR^*. In this situation, the DART protocol decreases reporting frequency in order to avoid congestion and save the limited energy of sensors. This reduction should be in a controlled manner so that the delay-constrained event-to-sink reliability is always maintained. However, the reporting frequency can be decreased more aggressively than the case of early reliability and no congestion. This is because in this case, we are further from optimal operating point. Here, we try to avoid congestion as soon as possible. Hence, the updated reporting frequency can be expressed as follows:

$$f_{i+1} = \min\left(f_i \frac{T_i}{\Delta_{\text{e2s}}}, f_i^{(T_i/\Delta_{\text{e2s}})}\right) \tag{3}$$

5.3. *Low reliability and no congestion (LR,NC) state*

In this state, the required reliability level specific to application is not reached before the event-to-sink delay bound, i.e. $T_i > \Delta_{\text{e2s}}$. Also, no congestion is observed at the sink, i.e. $\text{CN} = 0$, and the observed delay-constrained event reliability, DR_i, is lower than desired delay-constrained event reliability, DR^*. This can be caused by (i) packet loss due to wireless link errors, (ii) failure of intermediate relaying nodes, (iii) inadequate data packets transmitted by source nodes. Packet loss due to wireless link errors might be observed in WSN due to energy inefficiency of powerful error correction 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 [4] and spatially separated channels for communication from event-to-sink in WSN applications. Furthermore, when intermediate nodes fail, packets that need to be routed through these nodes are dropped. This can cause a reduction in reliability even if enough number of data packets is transmitted by source nodes. However, fault-tolerant routing/re-routing in WSN is provided by several existing routing algorithms [1]. DART protocol can work with any of these routing schemes. Therefore, to achieve required event reliability, we need to increase the data reporting frequencies of source nodes. Here, we exploit the fact that the DR *versus.* f relationship in the absence of congestion, i.e. for $f < f_{\max}$, is linear (see Section 3.3). In this regard, we use the multiplicative increase

DOI: 10.1002/dac

strategy to calculate updated reporting frequency, which is expressed as follows:

$$f_{i+1} = f_i \frac{\mathrm{DR}^*}{\mathrm{DR}_i} \tag{4}$$

5.4. Low reliability and congestion (LR,C) state

In this state, the required reliability level specific to application is not reached before the event-to-sink delay bound, i.e. $T_i > \Delta_{e2s}$. Also, congestion is observed at the sink, i.e. CN = 1, and the observed delay-constrained event reliability, DR_i, is lower than desired delay-constrained event reliability, DR^*. This situation is the worst possible case, since desired delay-constrained event reliability is not reached, network congestion is observed and thus, restricted energy of sensors is wasted. Hence, the DART protocol aggressively reduces reporting frequency to reach optimal reporting frequency as soon as possible. Therefore, to assure sufficient decrease in the reporting frequency, the reporting frequency is exponentially decreased and the updated frequency can be expressed as follows:

$$f_{i+1} = f_i^{(\mathrm{DR}_i/\mathrm{DR}^*/k)} \tag{5}$$

where k denotes the number of successive decision intervals for which the network has remained in the same situation including the current decision interval, i.e. $k \geqslant 1$. Here, the purpose is to decrease reporting frequency with greater aggression, if a network state transition is not detected.

```
k = 1;
DART()
  If (CONGESTION)
    If (δ < 1)
    /*Low Reliability and Congestion (LR,C)*/
      f_{i+1} = f_i^{DR_i/(DR* * k)};
      k = k + 1;
    else if (δ > 1)
    /*Early Reliability and Congestion (ER,C)*/
      k = 1;
      f_{i+1} = min(f_i T_i/Δ_{e2s}, f_i^{(T_i/Δ_{e2s})});
    end;
  else if (NO_CONGESTION)
    k = 1;
    If (δ < 1 − β)
    /*Low Reliability and No Congestion (LR,NC)*/
      f_{i+1} = f_i DR*/DR_i;
    else if (δ > 1 + β)
    /*Early Reliability and No Congestion (ER,NC)*/
      f_{i+1} = f_i T_i/Δ_{e2s};
    end;
    else if (1 − β ≤ δ ≤ 1 + β)
    /*Adequate Reliability and No Congestion (AR,NC)*/
      f_{i+1} = f_i;
    end;
end;
```

Figure 5. Algorithm of the DART protocol operation.

DOI: 10.1002/dac

5.5. Adequate reliability and no congestion (AR,NC) state

In this state, the network is within β tolerance of the optimal operating point, i.e. $f < f_{\max}$ and $1-\beta \leqslant \delta_i \leqslant 1+\beta$. Hence, the reporting frequency of source nodes is left constant for the next decision interval

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

Here, our aim is to operate as close to $\delta_i = 1$ as possible, while utilizing minimum network resources and meeting event-to-sink delay bounds. For practical purposes, we define a tolerance level, β, for optimal operating point. If at the end of decision interval i, the delay-constrained reliability indicator δ_i is within $[1-\beta, 1+\beta]$ and if no congestion is detected in the network, then the network is in (adequate reliability, no congestion) state. In this state, the event is deemed to be reliably and timely detected at the sink node and the reporting frequency remains unchanged. Note that greater proximity to the optimal operating point can be achieved with small β. However, smaller the β, greater the convergence time needed to reach corresponding (adequate reliability, no congestion) state. The entire DART protocol operation is presented in the pseudo-algorithm given in Figure 5.

6. MULTIPLE EVENT OCCURRENCES

The DART protocol operation defined in Section 5 directly applies to the scenarios where a single event occurs in the wireless sensor field. In this section, we extend DART protocol in order to accommodate the cases where multiple events concurrently occur in the same wireless sensor field. The DART protocol operation in case of multiple events is inspired by our previous work in [4]. In Section 6.1, we explain how DART mechanisms can accurately detect multiple event occurrences and extract the required information for the protocol operation. Then, we present the DART protocol operation in multiple event scenarios in Section 6.2.

6.1. 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 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 Figure 4. 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 in the field. Hence, the sink achieves the desired event-to-sink reliability with minimum energy expenditure using the DART protocol operation as explained in Section 5.

If the sink receives data packets carrying different event IDs in their Event ID fields as in Figure 4, it infers that multiple concurrent events occurred 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 DART 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. Thus, in this case, DART protocol can address delay-aware reliability requirements of these multiple events individually with the default DART operation explained in Section 5.

If there exist common sensor nodes performing routing for the multiple events occurred simultaneously, 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 DART protocol operation in order to accommodate these cases are explained in Section 6.2.

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 Figure 4. Here, we assume that Event ID field shown in Figure 4 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 multi-dimensional 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 into its event-list and leaves the packet intact and forwards it.

6.2. DART operation in multiple event scenarios

As described in Section 6.1, 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 field, 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 **(AR, NC)** state with the default DART protocol operation as explained in Section 5.

For the multiple event occurrence scenarios, the DART protocol operation varies based on whether the flows generated by these multiple events are isolated or not as explained in

Section 6.1. Hence, the detailed protocol operation for these two distinct cases are explained in the following sections.

6.2.1. 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.

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 DART 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 5. 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 $(\mathbf{AR}, \mathbf{NC})$ state 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 DART operation explained in Section 5.

6.2.2. 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. 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 DART is to achieve ESRT. 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 $((\mathbf{ER}, \mathbf{NC}))$ no congestion is experienced and the observed reliability is higher than required, it is completely opposite in state $((\mathbf{LR}, \mathbf{C}))$ where there is a congestion in the network and the event-to-sink reliability is not achieved. Hence, the event flows whose current network state are $((\mathbf{LR}, \mathbf{C}))$ 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 $((\mathbf{LR}, \mathbf{NC}))$ and $((\mathbf{ER}, \mathbf{NC}))$, the event flows which are currently in state $((\mathbf{LR}, \mathbf{NC}))$ do not receive their desired reliability levels and has higher priority than the ones in state $((\mathbf{ER}, \mathbf{NC}))$. With this respect, we group the network states $\{((\mathbf{LR}, \mathbf{C})), ((\mathbf{LR}, \mathbf{NC})), ((\mathbf{ER}, \mathbf{C})), ((\mathbf{ER}, \mathbf{NC}))\}$ into *high priority states*, i.e. $((\mathbf{LR}, \mathbf{C}))$, $((\mathbf{LR}, \mathbf{NC}))$, and *low priority states*, i.e. $((\mathbf{ER}, \mathbf{C}))$, $((\mathbf{ER}, \mathbf{NC}))$, 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 6.1. Let $\mathbf{S}_i^k$ and f_i^k be the current network state and the reporting frequency for the event k for $k \in N_\mathrm{e}$.

1. The sink determines the network state $\mathbf{S}_i^k$ for each of the flows generated by the event $k \in N_\mathrm{e}$ at the end of decision interval i as described in Section 5.

DOI: 10.1002/dac

2. If there are events whose network states are high priority, i.e. $\exists j \in N_e$ such that $\mathbf{S}_i^j = ((\mathbf{LR}, \mathbf{C}))$ or $\mathbf{S}_i^j = ((\mathbf{LR}, \mathbf{NC}))$:
   (a) The sink immediately performs the default DART operation described in Section 5 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 $\mathbf{S}_i^j = ((\mathbf{LR}, \mathbf{C}))$ or $\mathbf{S}_i^j = ((\mathbf{LR}, \mathbf{NC}))$. 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 within their delay bounds.
   (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 $\mathbf{S}_i^j = ((\mathbf{ER}, \mathbf{C}))$ or $\mathbf{S}_i^j = ((\mathbf{ER}, \mathbf{NC}))$.
   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 DART protocol operation which is to achieve delay-aware reliability.
3. If there are no events whose network states are high priority, i.e. $\mathbf{S}_i^j = ((\mathbf{ER}, \mathbf{C}))$ or $\mathbf{S}_i^j = ((\mathbf{ER}, \mathbf{NC})) \ \forall j \in N_e$, then the sink follows the default DART operation described in Section 5 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 $(\mathbf{AR}, \mathbf{NC})$ state as described in Section 5. As a result, the DART protocol operation described in Section 5 can accommodate the scenarios where multiple events occur simultaneously in the wireless sensor field.

7. PERFORMANCE EVALUATION

To evaluate the performance of the DART protocol for WSN, we once again developed an evaluation environment using ns-2 [20]. For all our simulations presented here, the number of sources, event-to-sink delay bound and tolerance level were selected as $n = 81$, $\Delta_{e2s} = 1$ s and $\beta = 5\%$, respectively. The event radius was fixed at 40 m. We run 10 experiments for each simulation configuration. Each data point on the graphs is averaged over 10 simulation runs. The main performance metrics that we employ to measure the performance of the DART protocol are the convergence time to (adequate reliability, no congestion) state from any other initial network states and average energy consumption per packet (E_i) for each decision interval.

The DART protocol convergence results are shown in Figure 6 for different initial network states. As shown in Figure 6, it is observed that the DART protocol converges to (adequate reliability, no congestion) state starting from any of the other initial network states discussed in Section 5. The performance of our reporting frequency update policies for event-to-sink transport can also be seen from the trace values listed in Figure 6. In this context, DART is self-configuring and can perform efficiently under random, dynamic topology frequently encountered in WSN applications. In addition to convergence time, the average energy

DOI: 10.1002/dac

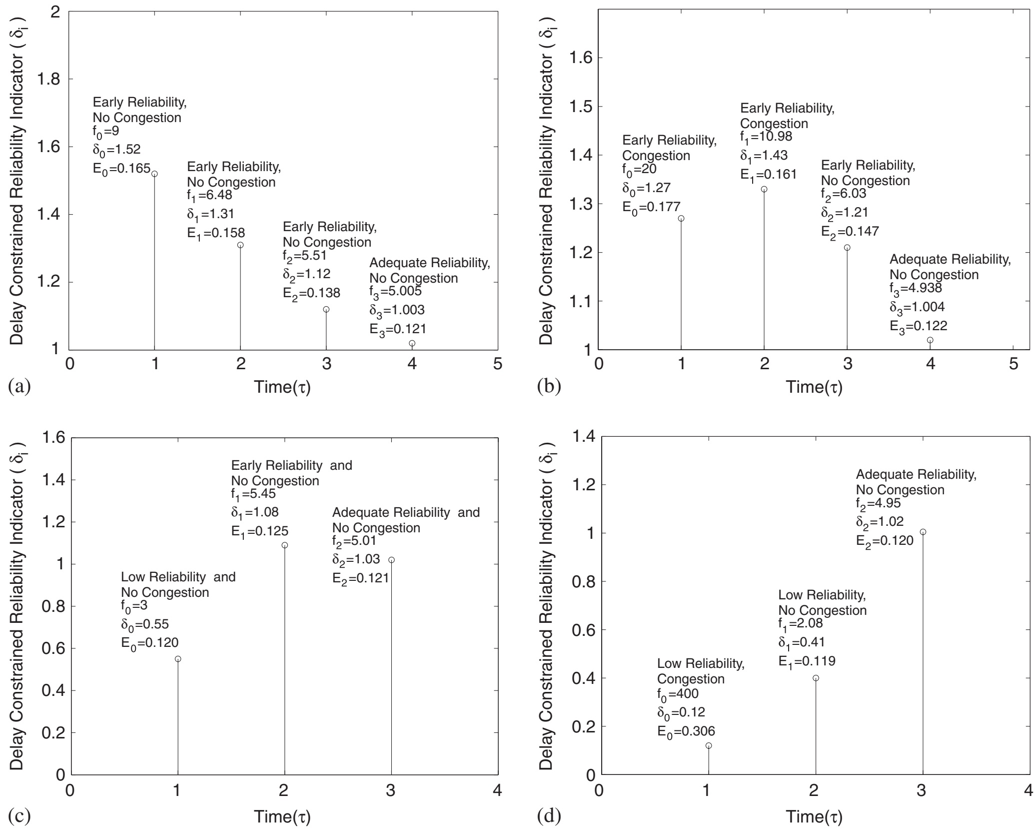

Figure 6. The DART protocol trace, when: (a) early reliability and no congestion; (b) early reliability and congestion; (c) low reliability and no congestion; and (d) low reliability and congestion, is observed.

consumed per packet during event-to-sink transport, i.e. E_i, is also observed. As shown in the Figure 6, E_i decreases as the (no congestion, adequate reliability) state is approached. This shows that energy consumption of the sensor nodes is also decreased while providing reliability constraints and delay bounds. Due to limited energy resources of the sensors, this result is also important for the proper operation of WSN.

To further investigate the DART protocol's convergence results, we have compared DART with ESRT [4] protocol in terms of convergence time to (adequate reliability, no congestion) state. The reason why we compare DART with ESRT is that both of them is based on event-to-sink reliability notion unlike the other transport layer protocols addressing conventional end-to-end reliability in WSNs. As shown in Figure 7, the convergence time of DART is much smaller than that of ESRT for different initial network states. This is because ESRT does not consider application specific delay bounds in its protocol operations.

To elaborate the relationship between the event-to-sink delay notion and the DART protocol operation, in Figure 8, we have also observed the delay distributions of the event packets received at the sink, when there is a transition from (low reliability, congestion)

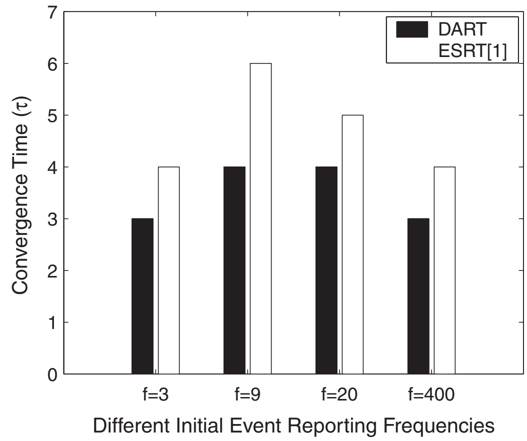

Figure 7. The comparison of DART and ESRT [4] in terms of convergence times.

state to (adequate reliability, no congestion) state. As seen in Figure 8, when the (adequate reliability, no congestion) state is approached, the delay of the event data packets also decreases. This is because the DART protocol takes event-to-sink delay bounds into account, while adjusting reporting rate of sensor nodes and avoiding network congestion.

Furthermore, we run simulation experiments to assess the DART 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. Figure 9 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 (adequate reliability, no congestion) state.

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 Figure 9, the average number of decision intervals it takes for all of the event flows to converge to the (adequate reliability, no congestion) state 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 DART brings the network state of these flows to (adequate reliability, no congestion) individually as explained in Section 5.

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 Figure 10, the average number of decision intervals it takes for all of the event flows to converge (adequate reliability, no congestion) state slightly increases with the number of concurrent events. This is mainly because the flows generated by these multiple events are not isolated and hence the DART protocol considers the priority of the current network states of these flows as explained in Section 6.2. 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

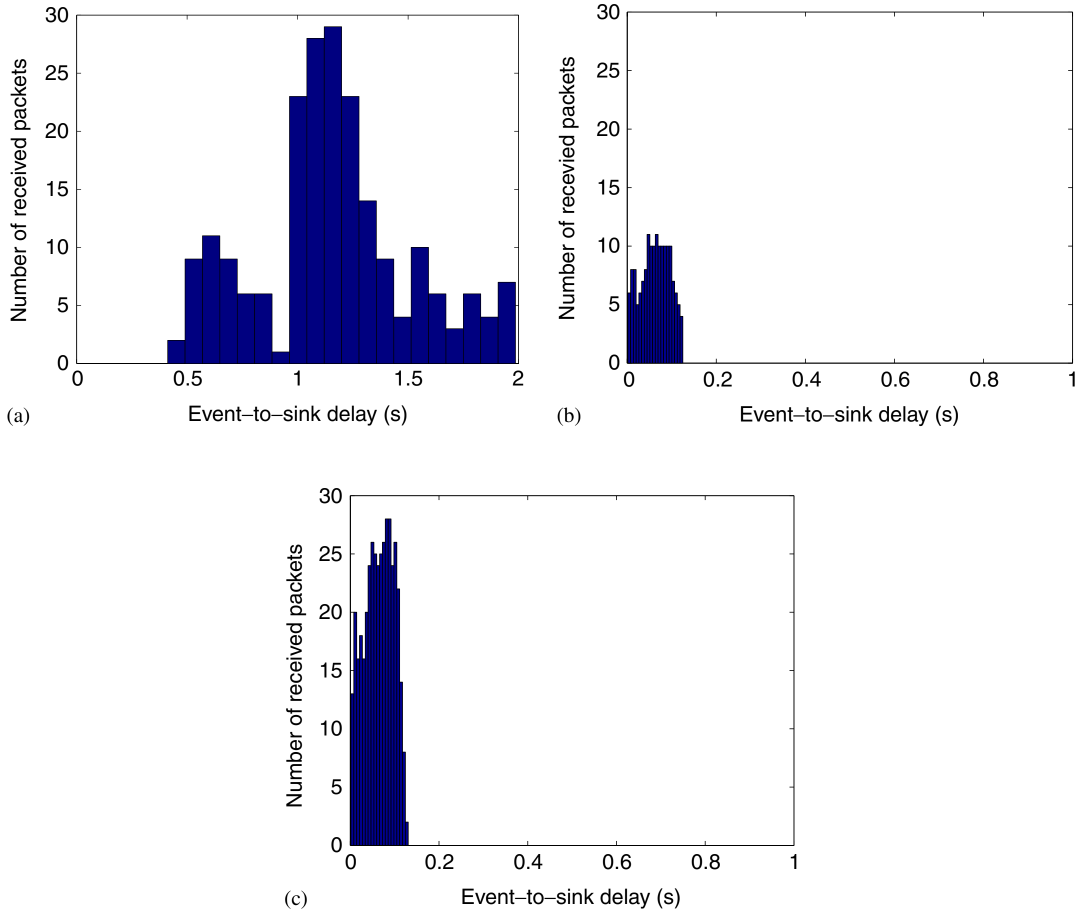

Figure 8. Packet delay distributions in: (a) (low reliability, congestion); (b) (low reliability, no congestion); and (c) (adequate reliability, no congestion) states.

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 Figure 10, the increase in the convergence time is small even in case of five non-isolated concurrent events. Hence, the DART protocol can effectively address the cases where multiple events occur simultaneously.

Note that, in these experiments, we do not assume that the underlying layer protocols, i.e. network, MAC and physical layer protocols, provide any additional support for meeting application-specific real-time delay requirements. Intuitively, we anticipate that the performance of DART protocol further improves, when deployed on top of lower layer communication protocols, which also provide real-time support. The evaluation of such scenario is beyond the scope of this paper and left as a future work.

DOI: 10.1002/dac

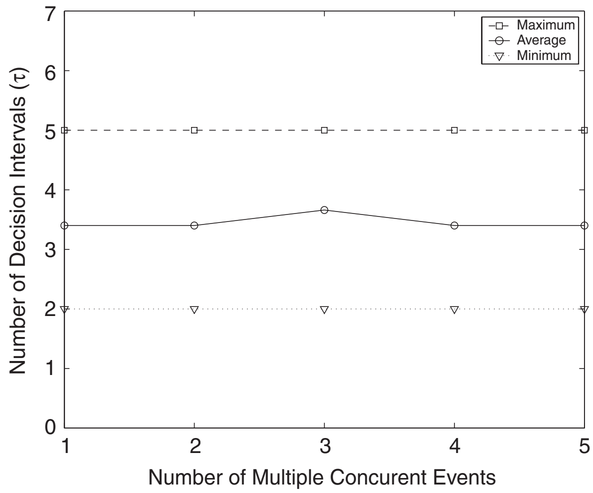

Figure 9. The number of decision intervals for all of the event flows to converge to (adequate reliability, no congestion) state for varying number of multiple concurrent events. In this set of experiments, the multiple concurrent events are isolated and their flows do not pass through any common router sensor node.

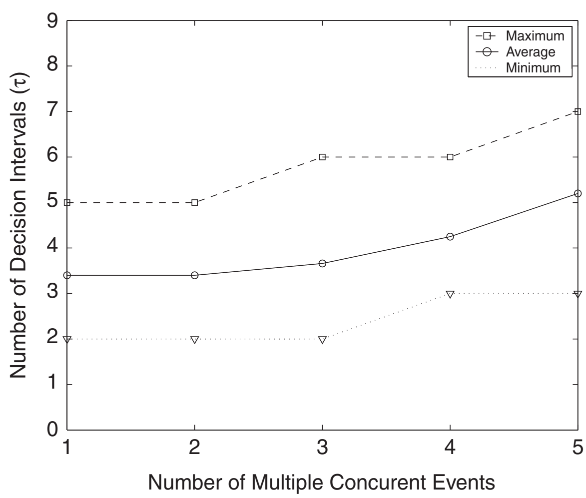

Figure 10. The number of decision intervals for all of the event flows to converge to (adequate reliability, no congestion) state for varying number of multiple concurrent events. In this set of experiments, the multiple concurrent events are not isolated.

8. CONCLUSION

The notion of delay-constrained event-to-sink reliability is necessary for timely and reliable transport of event features in WSN. This is due to the fact that the sink is only interested in timely collective information of a number of source nodes and not in individual sensor reports.

DOI: 10.1002/dac

Based on such a delay-constrained collective reliability notion, the delay aware reliable transport (DART) protocol for WSN is presented in this paper.

The DART protocol is a novel transport solution that seeks to achieve reliable and timely event detection with minimum possible energy consumption and no congestion. It enables the applications to perform right actions timely by exploiting both the correlation and the collaborative nature of sensor networks. To the best of our knowledge, reliable and timely event transport in WSN has not been studied from delay-constrained event-to-sink reliability perspective before.

In addition, the DART protocol has been tailored to meet the unique requirements of WSN. Its combined congestion detection and control mechanism serves the dual purpose of achieving reliability and conserving energy. It also considers event-to-sink delay bounds, while dynamically adjusting reporting frequency of the source nodes and avoiding network congestion. Moreover, it effectively addresses the cases, where multiple events occur in the wireless sensor field. Performance evaluation *via* simulation experiments show that the DART protocol achieves high performance in terms of reliable event detection, communication latency and energy consumption. Future work includes the implementation of the developed protocol on a physical testbed.

ACKNOWLEDGEMENTS

The work of Vehbi C. Gungor is supported by the National Science Foundation under contract ECS-0428329 and the work of Özgür B. Akan is supported by the Turkish Scientific and Technical Research Council (TUBITAK) Career Award under grant KARIYER-104E043.

REFERENCES

1. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor networks: a survey. *Computer Networks Journal* 2002; **38**(4):393–422.
2. Park SJ, Vedantham R, Sivakumar R, Akyildiz IF. A scalable approach for reliable downstream data delivery in wireless sensor networks. *Proceedings of ACM MOBIHOC*, Tokyo, Japan, 2004.
3. Wan CY, Campbell AT, Krishnamurthy L. PSFQ: a reliable transport protocol for wireless sensor networks. *Proceedings of ACM WSNA*, Atlanta, GA, U.S.A., 2002.
4. Akan OB, Akyildiz IF. ESRT: Event-to-sink reliable transport for wireless sensor networks. *IEEE/ACM Transactions on Networking* 2005; **13**(5):1003–1016.
5. Stann F, Heidemann J. RMST: reliable data transport in sensor networks. *Proceedings of IEEE SNPA*, Anchorage, AK, U.S.A., 2003.
6. Wan CY, Eisenman SB, Campbell AT. CODA: congestion detection and avoidance in sensor networks. *Proceedings of ACM SENSYS*, Los Angeles, CA, U.S.A., 2003.
7. Gungor VC, Akan OB. DST: delay sensitive transport in wireless sensor networks. *Proceedings of IEEE ISCN*, Istanbul, Turkey, 2006.
8. Hull B, Jamieson K, Balakrishnan H. Mitigating congestion in sensor networks. *Proceedings of ACM SENSYS*, Baltimore, MD, U.S.A., 2004.
9. Tilak S, Abu-Ghazaleh NB, Heinzelman W. Infrastructure tradeoffs for sensor networks. *Proceedings of ACM WSNA*, Atlanta, GA, U.S.A., 2002.
10. Woo A, Culler DE. A transmission control scheme for media access in sensor networks. *Proceedings of ACM MOBICOM*, Rome, Italy, 2001.
11. Ye W, Heidemann J, Estrin D. An energy-efficient MAC protocol for wireless sensor networks. *Proceedings of INFOCOM*, New York, U.S.A., 2002.
12. Chlamtac I, Conti M, Liu JJN. Mobile ad-hoc networking: imperatives and challenges. *Ad Hoc Networks Journal* 2003; **1**(1):13–64.
13. Holland G, Vaidya NH. Analysis of TCP performance over mobile ad hoc networks. *Proceedings of ACM MOBICOM*, Seattle, WA, U.S.A., 1999.

14. Liu J, Singh S. ATCP: TCP for mobile ad hoc networks. *IEEE Journal on Selected Areas in Communication* 2001; **19**(7):1300–1315.
15. Sundaresan K, Anantharaman V, Hsieh HY, Sivakumar R. ATP: a reliable transport protocol for ad-hoc networks. *Proceedings of ACM MOBIHOC*, Annapolis, MD, U.S.A., 2003.
16. MICA Motes. http://www.xbow.com/Products/Wireless_Sensor_Networks.htm (10 November 2006).
17. Vuran MC, Akan OB, Akyildiz IF. Spatio-temporal correlation: theory and applications wireless sensor networks. *Computer Networks Journal* 2004; **45**(3):245–259.
18. Caccamo M, Zhang LY, Sha L, Buttazzo G. An implicit prioritized access protocol for wireless sensor networks. *Proceedings of IEEE Real-Time Systems Symposium*, Urbana, IL, U.S.A., 2002.
19. Schwartz M. *Mobile Wireless Communications*. Cambridge University Press: New York, 2004.
20. The Network Simulator, ns-2. http://www.isi.edu/nsnam/ns/index.html (10 November 2006).
21. Johnson D, Maltz D, Hu Y, Jetcheva J. The dynamic source routing protocol for mobile ad hoc networks (DSR). *IETF Internet Draft*, 2002.

AUTHORS' BIOGRAPHIES

Vehbi C. Gungor received his BSc and MSc degrees in Electrical and Electronics Engineering from Middle East Technical University, Ankara, Turkey, in 2001 and 2003, respectively. He is currently a Research Assistant in the Broadband and Wireless Networking Laboratory and pursuing his PhD degree at the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA. His current research interests include wireless *ad hoc* and sensor networks, wireless sensor and actor networks, wireless mesh networks and WiMAX.

Özgür B. Akan received the BSc and MSc degrees in Electrical and Electronics Engineering from Bilkent University and Middle East Technical University, Ankara, Turkey, in 1999 and 2001, respectively. He received the PhD 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. Dr Akan is serving as an Editor for ACM-Springer Wireless Networks Journal and Area Editor for AD HOC Networks Journal (Elsevier), and participating in the organization of many international conferences. He is currently an Associate Professor with the Department of Electrical and Electronics Engineering, Middle East Technical University. His current research interests include sensor networks, next-generation wireless communications and bio-inspired communications.

DOI: 10.1002/dac