

A Real-Time and Reliable Transport (RT)² Protocol for Wireless Sensor and Actor Networks

Vehbi C. Gungor, *Student Member, IEEE*, Ozgur B. Akan, *Member, IEEE*, and Ian F. Akyildiz, *Fellow, IEEE*

Abstract—Wireless Sensor and Actor Networks (WSANs) are characterized by the collective effort of heterogenous nodes called sensors and actors. Sensor nodes collect information about the physical world, while actor nodes take action decisions and perform appropriate actions upon the environment. The collaborative operation of sensors and actors brings significant advantages over traditional sensing, including improved accuracy, larger coverage area and timely actions upon the sensed phenomena. To realize these potential gains, there is a need for an efficient transport layer protocol that can address the unique communication challenges introduced by the coexistence of sensors and actors.

In this paper, a Real-Time and Reliable Transport (RT)² protocol is presented for WSANs. The objective of (RT)² is to reliably and collaboratively transport event features from the sensor field to the actor nodes with minimum energy dissipation and to timely react to sensor information with a right action. In this respect, the (RT)² protocol simultaneously addresses congestion control and timely event transport reliability objectives in WSANs. To the best of our knowledge, this is the first research effort focusing on real-time and reliable transport protocol for WSANs. Performance evaluation via simulation experiments show that (RT)² achieves high performance in terms of reliable event detection, communication latency and energy consumption in WSANs.

Index Terms—Wireless sensor and actor networks, Real-time and reliable transport, Congestion detection and control.

I. INTRODUCTION

WIRELESS Sensor and Actor¹ Networks (WSANs) are characterized by the collective effort of densely deployed sensor nodes and sparsely deployed actor nodes. In WSANs, sensor nodes collect information about the physical world, while actors take action decisions and perform appropriate actions upon the environment. The existing and potential applications of WSANs span a very wide range, including real-time target tracking, homeland security, battlefield surveillance, and biological or chemical attack detection [2]. The practical realization of these currently designed and envisioned applications, however, directly depends on real-time and reliable communication capabilities of the deployed sensor/actor network.

Recently, there has been considerable amount of research efforts, which have yielded many promising communication protocols to address the challenges posed by the wireless sensor network (WSN) paradigm [1],[3],[13],[18],[19]. The common

feature of these protocols 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 WSAN applications, which have strict delay bounds and hence mandate timely transport of the event features from the sensor field to the actor nodes [2]. Consequently, the unique features and application requirements of WSANs call for a real-time and reliable data transport solution. The functionalities and design of a real-time and reliable transport solution for WSANs are the main issues addressed in this paper.

The major communication challenges in realization of a real-time and reliable transport protocol for WSANs can be summarized as follows:

- *Heterogeneous reliability requirements:* The transport paradigms of WSANs have different reliability requirements due to the node heterogeneities in the deployment field [2]. For example, while sensor-actor communication may not require 100% reliability due to the correlation among the sensor readings [1],[17], actor-actor communication requires 100% reliability in order to make a decision on the most appropriate way to collaboratively perform the action.
- *Delay bounds:* In WSANs, actor nodes need to immediately react to sensor data based on the application-specific requirements. Hence, real-time communication within certain delay bounds is a crucial concern to guarantee timely execution of the right actions.
- *Wireless channel errors:* The wireless channel errors in WSANs lead to bursts of packet loss [2]. Despite the existence of channel coding schemes, packet-level transport layer reliability mechanisms are required. Furthermore, new congestion detection and control algorithms are necessary to avoid erroneous congestion decisions resulting from channel related packet losses.
- *Node mobility and route failures:* Actor nodes in WSANs might be highly mobile depending on the application requirements. The mobility may lead to route failures and packet losses that must be accurately captured by the developed transport layer solutions to avoid inaccurate congestion control.
- *Energy efficiency:* Although the primary objective of the transport protocols in WSANs is reliable event detection and timely execution of the right actions, an efficient transport layer solution should also accomplish this with minimum energy consumption due to limited energy resources of sensor nodes.

In this paper, to address all above communication challenges,

V.C. Gungor and I.F. Akyildiz are with the Broadband and Wireless Networking Laboratory, Georgia Institute of Technology, Atlanta, GA.

O.B. Akan is with Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey.

This work is supported by the National Science Foundation under contract ECS-0428329.

¹We refer to acting entities as *actors*. They are sometimes referred to as actuators in the related literature.

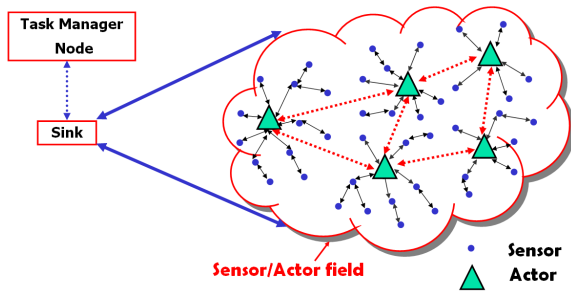


Fig. 1. An illustration of an integrated architecture of the Wireless Sensor and Actor Networks (WSANs).

a real-time and reliable transport (RT)² protocol is presented for WSANs. (RT)² 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 WSANs. Furthermore, (RT)² addresses heterogeneous reliability requirements of both sensor-actor and actor-actor communication. More specifically, for sensor-actor communication, unlike traditional end-to-end reliability notions, (RT)² defines *delay-constrained event reliability* notion based on both *event-to-action delay bounds* and event reliability objectives. On the other hand, for actor-actor communication, it introduces 100% packet-level reliability mechanisms in order to avoid wrong action decisions in the deployment field. This way, the (RT)² protocol simultaneously addresses event transport reliability and timely action performance objectives of WSANs.

Overall, the contribution of this paper is the design of a real-time and reliable transport protocol for typical WSAN applications involving event detection and timely action performance within a certain delay bound. Our work is mainly motivated by the fact that the actor nodes are only interested in timely and reliable detection of event features from the collective information provided by several sensor nodes and not in their individual reports. The notion of *delay-constrained event reliability* distinguishes (RT)² from other existing transport layer models that focus on end-to-end reliability. To the best of our knowledge, reliable event transport has not been studied from this perspective before and this is the first research effort focusing on real-time and reliable event transport and action performance objectives of WSANs.

The remainder of the paper is organized as follows. In Section II, we present the network architecture and describe the design principles and functionalities of the (RT)² protocol in detail. The protocol operation of (RT)² for sensor-actor and actor-actor communication is described in Section III and IV, respectively. Performance evaluation and simulation results are presented in Section V. Finally, the paper is concluded in Section VI.

II. (RT)² PROTOCOL DESIGN PRINCIPLES

Unlike traditional networks, the sensor/actor network paradigm necessitates that the event features are collaboratively estimated within a certain reliability and real-time delay bound. To achieve this objective with maximum resource efficiency, the (RT)² protocol exploits both the correlation and the collaborative nature

of the network. In the following sections, we first describe the characteristics and challenges of both sensor-actor and actor-actor communication and then based on these characteristics, we discuss the main design components of the (RT)² protocol in detail. We also present a case study in order to gain more insight regarding the challenges of sensor/actor network.

A. Network Architecture

A typical network architecture of WSANs is shown in Fig. 1. In this network architecture, the sensors are energy constrained, multi-functional devices with limited processing and low range communication capabilities, while the actors are resource-rich nodes equipped with better processing capabilities, high transmission power and longer battery life time. Furthermore, in WSANs, a large number of sensors, i.e., on the order of hundreds or thousands, are randomly deployed in a target area to perform a collaborative sensing task. Such a dense deployment is usually not necessary for actor nodes, because actor nodes have higher capabilities and can act on large areas.

In WSANs, the collaborative operation of the sensor nodes enables *distributed sensing* of a physical phenomenon. After sensors detect an event occurring in the environment, the event data is distributively processed and transmitted to the actors, which gather, process, and eventually reconstruct the event data. We refer the process of transmission of event features from the sensor nodes to the actor nodes as *sensor-actor communication*. Once an event has been detected in the deployment field, the actors need to communicate with each other to make a decision on the most appropriate way to collaboratively perform the action. We refer to this process as *actor-actor communication*. Therefore, the operation of the WSANs can be considered as a timely event detection, decision and acting loop.

B. Reliable Event Transport

The (RT)² protocol is equipped with different reliability functionalities in order to address heterogeneous requirements of both sensor-actor and actor-actor communication. The main features of these reliability functionalities are described in the following.

1) *Sensor-Actor Transport Reliability*: In WSANs, sensor-actor transport does not require 100% reliability due to the correlation among the sensor readings [1],[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 sensor-actor transport paradigm requires a collective *event transport reliability* notion rather than the traditional end-to-end reliability notions. The (RT)² protocol also considers the new notion of *event-to-action delay bound* (described in Section II-C) to meet the application-specific deadlines. Based on both event transport reliability and event-to-action 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 actor node 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., $\delta_i = DR_i/DR^*$.

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 action². If the observed delay constrained event reliability is higher than the reliability bound, i.e., $DR_i > DR^*$, 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 sensor-actor communication. For example, in order to increase the amount of information transported from the sensors to the actor, reporting frequency of the sensors can be increased properly while avoiding congestion in the network. Therefore, sensor-actor transport reliability problem in WSANs is to *configure the reporting rate, f , of source nodes so as to achieve the required event detection reliability, DR^* , at the actor node within the application-specific delay bound*. The details of the $(RT)^2$ protocol operation for sensor-actor communication is described in Section IV.

2) *Actor-Actor Transport Reliability*: In WSANs, a reliable and timely actor-actor ad hoc communication is also required to collaboratively perform the right action upon the sensed phenomena [2]. The $(RT)^2$ protocol simultaneously incorporates *adaptive rate-based transmission control and point-to-point packet loss recovery* to achieve 100% packet reliability in the required ad hoc communication. To achieve this objective, $(RT)^2$ protocol relies upon new feedback based congestion control mechanisms and probe packets to recover from subsequent losses and selective-acknowledgments (SACK) to detect any holes in the received data stream. These algorithms are shown to be beneficial and effective in recovering from multiple packet losses in one round-trip time (RTT) especially [14]. The details of adaptive rate-based transmission and congestion control algorithms for actor-actor ad hoc communication is explained in Section IV. In the following section, event-to-action delay bound notion of $(RT)^2$ protocol is explained in detail.

C. Real-Time Event Transport

In order to assure accurate and timely action on the sensed phenomena, it is imperative that the event is sensed, transported to the actor node and the required action is performed within a certain delay bound. We call this *event-to-action delay bound*, Δ_{e2a} , which is specific to application requirements and must be met so that the overall objective of the sensor/actor network is achieved.

The event-to-action delay bound Δ_{e2a} , has three 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 actor 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* ($t_{p,i}^i$): It is the propagation latency of the data packet to reach the next hop over the wireless channel. It mainly depends on the node density and communication medium.
- 2) **Event processing delay** (Γ^{proc}): This is the processing delay experienced at the actor node when the desired features of event are estimated using the data packets received from the sensor field. This may include a certain decision interval [1] during which the actor node waits to receive adequate samples from the sensor nodes.
 - 3) **Action delay** (Γ^{act}): The action delay is the time it takes from the instant that event is reliably detected at the actor node to the instant that the actual action is taken. It is composed of the *task assignment delay*, i.e., time to select the *best*³ set of actors for the task and the *action execution delay*, i.e., time to actually perform the action.

More specifically, while event transport delay Γ^{tran} and event processing delay Γ^{proc} occur during sensor-actor communication, action delay Γ^{act} is resulted from actor-actor communication in the deployment field. Let Δ_{e2a} be the event-to-action delay bound for the data packet generated by the detection of event. Then, for a timely action, it is necessary that

$$\Delta_{e2a} \geq \Gamma^{tran} + \Gamma^{proc} + \Gamma^{act} \quad (1)$$

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 actor 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 .

²With in-network data aggregation, one must account for data packets that were aggregated en route to the actor node.

³The best set of actors refers to the actors which are close to the event area, or which has high capability and residual energy, or which has small action completion time upon the sensed phenomenon [2].

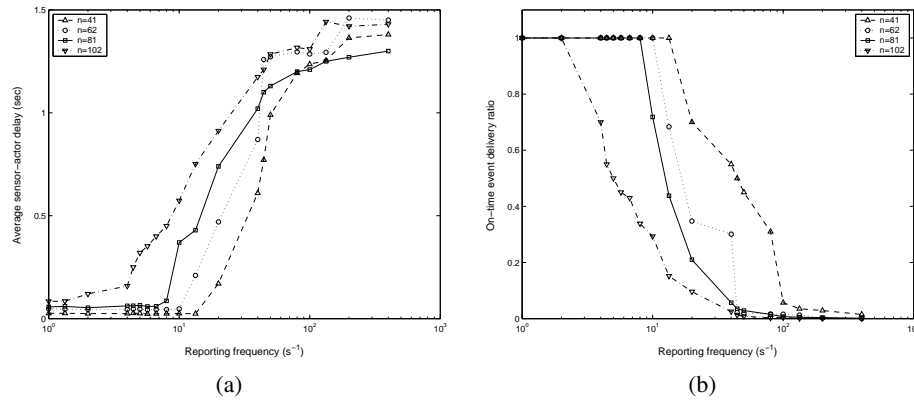


Fig. 2. The effect of varying reporting frequency of source nodes on (a) average sensor-actor delay and (b) on-time event delivery ratio.

In addition, for the events occurring at further distances to the actor node, the average number of hops that event data packets traverse, \bar{N} , increases. Thus, it is more difficult to provide event-to-action delay bound for further event packets compared to closer ones. Considering that the per-hop propagation delay, $t_{p,i}$, does not vary⁴, 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-action delay bound is met. In order to accomplish this objective, we define the following rules for the required service discipline employed at each sensor node:

- The buffering delay at each intermediate forwarding node must be controlled such that the data packets generated by the detection of an event with lower *remaining time to deadline*, $\Delta_{r,i}$, must be served first by the forwarding sensor node. Here, $\Delta_{r,i}$ represents the remaining time to the application-specific delay bound at the sensor node i . Note that with this objective, if there is no congestion in the network, the data packets generated by the further events are served first by the intermediate forwarding sensor nodes.
- In WSNs, the calculation of updated remaining time to deadline at each intermediate sensor node is not trivial due to the *lack of globally synchronized clock*. Therefore, the computation of remaining time to deadline for the dynamic service discipline should be performed even in the lack of globally synchronized clock.

Based on these dynamic service discipline objectives, the (RT)² 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 [5],[12]. However, we also integrate some novel mechanisms so as to fit it to unique challenges of sensor networks.

In order 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 actor 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 presented in the Appendix.

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 (RT)² protocol considers the event-to-action delay bounds and congestion conditions in its reporting rate update policies to assure timely and reliable event transport in WSNs (see Section III). It is also important to note that the measured elapsed time at each sensor node can give an idea of congestion level experienced in the network, since it represents both the buffering delay and the channel contention around the sensor node. This is discussed in Section II-E in detail. In the following, we present a case study to gain more insight regarding the communication challenges of sensor/actor network.

D. Case Study

To investigate the relationship between the event-to-action delay and the event reporting rate, we develop an evaluation environment using *ns-2* [15]. The parameters used in our case study are listed in Table I. In our simulations, 200 sensor nodes were randomly positioned in a 200m x 200m sensor field. Node parameters such as radio range and IFQ (buffer) length were carefully chosen to mirror typical sensor mote values [16]. Event centers (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 actor node receiving the data is placed in the middle of the lower side of the deployment area. To communicate source data to the actor node, we employed a simple CSMA/CA based MAC protocol and Dynamic Source Routing (DSR) [8]. 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 sensor-actor communication delay and *on-time event*

⁴While the channel access delay can also be controlled to a certain extent via priority-based QoS-aware MAC protocols [3], we do not assume the presence of such MAC protocol.

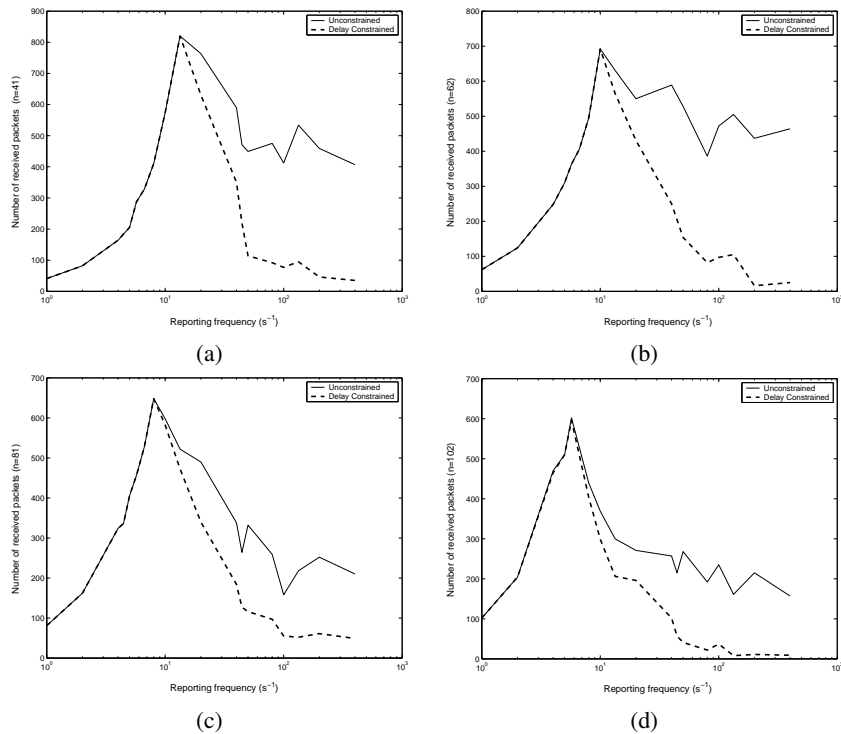


Fig. 3. The number of received packets at the actor node in a decision interval, when the number of sources, (a) $n = 41$, (b) $n = 62$, (c) $n = 81$, (d) $n = 102$.

TABLE I
NS-2 SIMULATION PARAMETERS

Area of sensor field	200x200 m^2
Number of sensor nodes	200
Radio range of a sensor node	20 m
Packet length	30 bytes
IFQ length	65 packets
Transmit Power	0.660 W
Receive Power	0.395 W
Idle Power	0.035 W
Decision interval (τ)	1 sec

delivery ratio. Here, on-time event delivery ratio represents the fraction of data packets received within sensor-actor 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 Fig. 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 center (X_{ev}, Y_{ev}) and corresponding number of sources, n . These values are tabulated in Table II.

TABLE II
RANDOMLY SELECTED EVENT CENTERS USED IN THE SIMULATIONS

Number of source nodes	Event center (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

As shown in Fig. 2(a) and 2(b), it is observed that as the event reporting frequency, f , increases, average sensor-actor transport 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 sensor-actor transport 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 actor node in a decision interval, τ . We make the following observations from Fig. 3:

- i. Until a certain $f = f_{max}$, observed delay-constrained event reliability and unconstrained event reliability⁵ coincides, beyond which delay-constrained event reliability significantly deviates from unconstrained event reliability.
- ii. 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 congestion.
- iii. Such an initial increase and a subsequent decrease in observed delay-constrained event reliability is observed

⁵Unconstrained event reliability represents the number of event packets received at the actor irrespective of their packet delay.

regardless of the number of source nodes, n .

- iv. f_{max} decreases with increasing n , i.e., network congestion occurs at lower reporting frequencies with greater number of source nodes.
- v. After $f=f_{max}$, delay-constrained event reliability starts to drop significantly due to network congestion. Therefore, an accurate congestion detection mechanism is required in order to both provide delay-constrained reliability and an effective congestion control in the network.

In summary, with increasing reporting frequency, 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 Fig. 2. Furthermore, when the application-specific delay bounds are considered, the observed delay-constrained event reliability decreases significantly with the network congestion, regardless of the number of source nodes. These observations confirm the urgent need for a delay-constrained reliable event transport solution with an efficient congestion detection and control mechanism in WSAWs. In the following section, combined congestion detection mechanism of the (RT)² protocol is described in detail.

E. Congestion Detection and Control Mechanism

In WSAWs, 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 [1]. Hence, it is mandatory to address the congestion in the sensor field to achieve real-time and 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 actor node, and not any of the sensor nodes, can determine the delay-constrained reliability indicator $\delta_i = DR_i/DR^*$, and act accordingly.

In addition, for efficient congestion detection in WSAWs, 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 neighborhood [10]. Therefore, because of shared communication medium nature of WSAWs, the sensor nodes can experience congestion even if their buffer occupancy is small.

In order to investigate the impact of the channel contention on the congestion level of the neighboring nodes, we perform a simulation study using ns-2 [15]. The network configuration is shown in Fig. 4, in which node 0 and 1 (sources) send data to node 4 and 5 (destinations), respectively. During the time period between 4 and 6 sec, the node 0 increases its transmission rate, which creates a hot spot around node 2. In Fig. 5 (a) and (b), the resulting packet delay and buffer occupancy at the nodes 2 and 3 are shown, respectively. As seen in Fig. 5 (a), we observe that at node 2 both buffer occupancy ratio and average packet delay between 4 and 6 sec increase significantly and these metrics reflect the congestion level at node 2 accurately. On the other hand, as shown in Fig. 5 (b), we

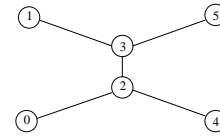


Fig. 4. A simple wireless ad hoc network of 6 nodes. Only the nodes connected by a line are within each other's communication range.

observe that even if the buffer occupancy at node 3 is small during 4 and 6 sec (buffer occupancy ratio is almost 20%), average packet delay increases significantly between 4 and 6 sec. This is because in this time period, although the incoming traffic does not change, the increased channel contention around the node 3 causes packet collisions and retransmissions resulting in increased packet delay. Note that at node 3, it is difficult to detect the level of congestion solely based on the buffer occupancy. Therefore, for an efficient congestion detection in WSAWs, a combined approach is required.

In this regard, the (RT)² 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 node i , the sensor node takes exponential weighted moving average of the elapsed time, $t_{e,i}$, which is calculated in (8) in Section VI. Recall that with the proposed mechanism in Section II-C, the calculation of the average node delay can be performed without globally synchronized clock in the network.

In combined congestion detection mechanism of (RT)² protocol, any sensor node whose buffer overflows due to excessive incoming packets or average node delay is above a certain delay threshold value is said to be congested and it informs the congestion situation to the actor node.⁶ More specifically, the actor 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 transmitted from sensors to the actor node. Therefore, if the actor 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 actor node can determine the current network condition and dynamically adjust the reporting frequency of the sensor nodes.

In order to achieve timely execution of the right action upon the environment, actor-actor ad hoc communication must also be efficiently handled. In this respect, congestion control is also imperative for reliable and timely actor-actor ad hoc communication. Hence, combined congestion mechanism of (RT)² protocol is also utilized for actor-actor ad hoc communication. The details of adaptive rate-based transmission and congestion control algorithms for actor-actor ad hoc communication are explained in Section IV. In the following section, we discuss the details of the (RT)² protocol operation for sensor-actor communication.

⁶To avoid reacting to transient network behavior and to increase the accuracy of congestion detection, (RT)² protocol detects congestion, if the node delay measurements exceed a delay threshold more than a certain number of successive times.

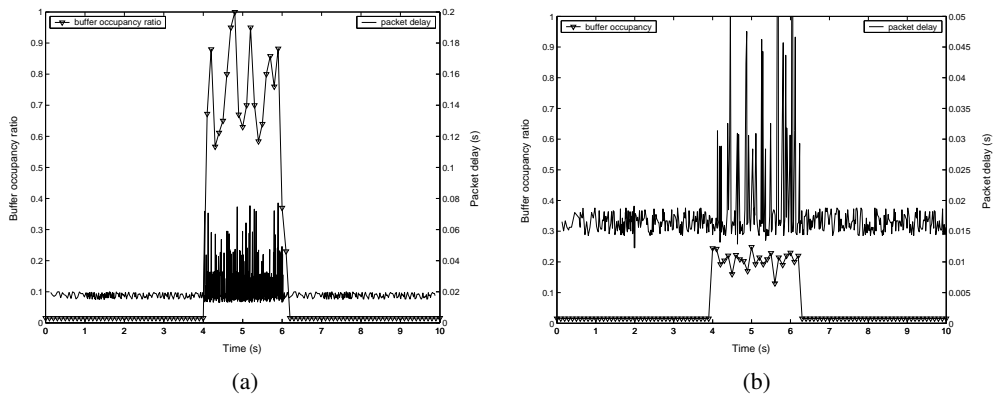


Fig. 5. Buffer occupancy ratio and packet delay observations at (a) node 2 and (b) node 3.

III. (RT)² PROTOCOL OPERATION FOR SENSOR-ACTOR COMMUNICATION

In this section, we describe the (RT)² protocol operation during sensor-actor communication. Recall that in the previous sections, based on the delay-constrained event reliability and the event-to-action delay bound notions, we had defined a new delay-constrained reliability indicator $\delta_i = DR_i/DR^*$, i.e., the ratio of observed and desired delay-constrained event reliabilities. In order to determine proper event reporting frequency update policies, we also define T_i and T_{sa} , which are the amount of time needed to provide delay-constrained event reliability for a decision interval i and the application-specific sensor-actor communication delay bound, respectively. In conjunction with the congestion notification information (CN bit) and the values of f_i , δ_i , T_i and T_{sa} , the actor 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 condition is obtained. In the following sections, we describe the details of the reporting frequency update policies and possible network conditions experienced by the sensor nodes.

A. Early Reliability and No Congestion Condition

In this condition, the required reliability level specific to application is reached before the sensor-actor communication delay bound, i.e., $T_i \leq T_{sa}$, and no congestion is observed in the network, i.e., $CN = 0$. 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 frequently than required. The most important consequence of this condition is excessive energy consumption of the sensors. Therefore, the reporting frequency should be decreased cautiously in order to conserve energy. This reduction should be performed cautiously so that the delay-constrained event reliability is always maintained. Therefore, the actor node 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:

$$f_{i+1} = f_i \frac{T_i}{T_{sa}} \quad (2)$$

B. Early Reliability and Congestion Condition

In this condition, the required reliability level specific to application is reached before the sensor-actor communication delay bound, i.e., $T_i < T_{sa}$, and congestion is observed in the network, i.e., $CN = 1$. However, the observed delay-constrained event reliability, DR_i , is larger than the desired delay-constrained event reliability, DR^* . In this situation, the (RT)² 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 reliability is always maintained. However, the reporting frequency can be decreased more aggressively than the case where there is no congestion and the observed delay-constrained event reliability, DR_i , is larger than the desired delay-constrained event reliability, DR^* . This is because in this case, we are farther from optimal operating point. Here, we try to avoid congestion as soon as possible. Hence, the updated reporting frequency can be expressed as follows:

$$f_{i+1} = \min\left(f_i \frac{T_i}{T_{sa}}, f_i^{(T_i/T_{sa})}\right) \quad (3)$$

C. Low Reliability and No Congestion Condition

In this condition, the required reliability level specific to application is not reached before sensor-actor communication delay bound, i.e., $T_i > T_{sa}$, and no congestion is observed in the network, i.e., $CN = 0$. However, the observed delay-constrained event reliability, DR_i , is lower than the 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 WSNs 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 significantly in successive decision intervals. This is reasonable with static sensor nodes, slowly time-varying and spatially separated communication channels [1]. 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 [3]. (RT)² protocol can work with any of these routing schemes. Therefore, in order to improve delay-constrained event reliability, we need to increase the data reporting frequencies of source nodes. To achieve this objective, the reporting frequency is aggressively increased in the absence of congestion. Hence, the possible updated reporting frequency in this case can be multiplicative, which is expressed as follows:

$$f_{i+1} = f_i \frac{DR^*}{DR_i} \quad (4)$$

D. Low Reliability and Congestion Condition

In this condition, the required reliability level specific to application is not reached before sensor-actor communication delay bound, i.e., $T_i > T_{sa}$, and congestion is observed in the network, i.e., $CN = 1$. However, the observed delay-constrained event reliability, DR_i , is lower than the 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, limited energy of sensors is wasted. Hence, the (RT)² protocol aggressively reduces reporting frequency in order to reach optimal reporting frequency as soon as possible. Therefore, in order to assure sufficient decrease in the reporting frequency, it is exponentially decreased and the new frequency is expressed by:

$$f_{i+1} = f_i \frac{DR_i}{(DR^*)^k} \quad (5)$$

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., $k \geq 1$. Here, the purpose is to decrease reporting frequency with greater aggression, if a network condition transition is not detected.

E. Adequate Reliability and No Congestion Condition

In this condition, the network is within β tolerance of the optimal operating point, i.e., $f < f_{max}$ and $1 - \beta \leq \delta_i \leq 1 + \beta$, and no congestion is observed in the network. Hence, the reporting frequency of source nodes is left constant for the next decision interval:

$$f_{i+1} = f_i \quad (6)$$

Here, our aim is to operate as close to $\delta_i = 1$ as possible, while utilizing minimum network resources and meeting event 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) condition. In this condition, the event is deemed to be reliably and timely detected and the reporting frequency remains unchanged. Thus, a greater proximity to the optimal operating point can be achieved with small β . However, the smaller the β , the greater the convergence time needed to reach corresponding (Adequate reliability, No

```

k = 1;
(RT)2()
  If (CONGESTION)
    If ( $\delta < 1$ )
      /*Low Reliability and Congestion*/
       $f_{i+1} = f_i \frac{DR_i}{(DR^*)^k}$ ;
      k = k + 1;
    else if ( $\delta > 1$ )
      /*Early Reliability and Congestion*/
      k = 1;
       $f_{i+1} = \min(f_i \frac{T_i}{T_{sa}}, f_i^{(T_i/\Delta_{e2a})})$ ;
    end;
  else if (NO_CONGESTION)
    k = 1;
    If ( $\delta < 1 - \beta$ )
      /*Low Reliability and No Congestion*/
       $f_{i+1} = f_i \frac{DR_i}{DR^*}$ ;
    else if ( $\delta > 1 + \beta$ )
      /*Early Reliability and No Congestion*/
       $f_{i+1} = f_i \frac{T_i}{T_{sa}}$ ;
    end;
    else if ( $1 - \beta \leq \delta \leq 1 + \beta$ )
      /*Adequate Reliability and No Congestion*/
       $f_{i+1} = f_i$ ;
    end;
  end;

```

Fig. 6. Algorithm of the (RT)² protocol during sensor-actor communication.

congestion) condition. Therefore, a good choice of β is the one that balances the tolerance and convergence requirements.

In summary, the operation of (RT)² protocol for sensor-actor communication is closely tied to the current network condition and thus, the reporting frequency of the sensors is adjusted accordingly. The entire (RT)² protocol operation is presented in the pseudo-algorithm given in Fig. 6.

IV. (RT)² PROTOCOL FOR ACTOR-ACTOR COMMUNICATION

In WSANs, as discussed before, after receiving event information, actors need to communicate with each other in order to collaboratively make decisions on the most appropriate way to perform the action. Thus, to timely initiate the right actions upon the sensed phenomena, the (RT)² protocol also addresses efficient actor-actor communication. In this section, we first describe the main design principles of the (RT)² protocol for actor-actor communication. Then based on these design principles, we describe the details of the (RT)² protocol operation during actor-actor communication.

A. (RT)² Protocol Overview for Actor-Actor Communication

In this section, we make an overview of the key design elements of the (RT)² protocol for actor-actor communication as follows:

- 1) **Cross-layer interactions:** In the current literature, in order to improve the communication performance in wireless ad hoc communication, some protocols providing an efficient coordination between communication layers are developed [20]. The purpose of such coordination is to react the network dynamics both accurately and timely. Furthermore, some feedback schemes are proposed so that the

sender can distinguish between a route failure and network congestion [4],[7]. In this respect, the $(RT)^2$ protocol benefits from both cross-layer interactions and intermediate node feedback information so as to i) capture route failures accurately and timely, ii) get congestion notification and transmission rate feedback for both initial start up phase and steady state phase.

- 2) **Distinguishing cause of packet loss:** $(RT)^2$ protocol distinguishes congestion and non-congestion related losses by the feedback information from both receiver and the intermediate nodes. In this context, the $(RT)^2$ protocol uses a *combined congestion detection* mechanism based on both the average node delay calculation and the local buffer level monitoring of the actor nodes in order to accurately detect congestion in the network (see Section II-E). When the actor node is notified about the congestion condition, it decreases the transmission rate accordingly to relieve the congestion as soon as possible. In addition, packet transport reliability is provided by the receiver feedback packets and selective ACKs (SACKs).
- 3) **Adaptive rate-based transmission:** Unlike traditional transport protocols, where computations happen solely at the sender, the $(RT)^2$ protocol exploits the receiver to perform rate control computations. Since the available bandwidth in the network is difficult to calculate in multi hop wireless ad hoc communication, our transmission rate computation mechanism relies on both congestion notification (CN) and maximum packet delay information of intermediate nodes. Here, the packet delay represents the sum of queuing, channel access time and transmission time at the bottleneck node along the path. Note that we also compute exponential average of packet delays at the intermediate nodes and the receiver in order to fine tune the fluctuations of the observed delay values (see Section II-E). Furthermore, the average minimum transmission rate (R_{min}) is determined according to the remaining time to event-to-action delay bound (see eq. 7). Consequently, the transmission data rate is dynamically adjusted according to the current conditions of the data path and event-to-action delay bounds.
- 4) **Admission control:** To meet the application-specific delay bounds, each sender probes the available transmission rate before connection establishment. If the available rate is not higher than the average minimum transmission rate (R_{min}), then the transmission rate is adjusted according to the current conditions of the network. For example, in case the average minimum transmission rate cannot be met, the sender can send the most critical information to the receiver other than all existing information. Hence, the $(RT)^2$ protocol tries to shape data traffic based on both delay bounds and the current conditions of the network. Note that no admission control is executed at the intermediate nodes nor are there any resources reserved on behalf of the sender during the life time of the connection.
- 5) **Flow control:** The $(RT)^2$ protocol also considers the packet delay observed at the receiver. Based on this delay observation, the application processing rate (R_p) is calculated at the receiver. If R_p is smaller than the rate feedback (R_f)

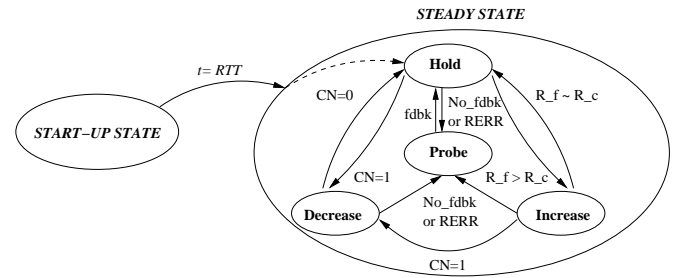


Fig. 7. $(RT)^2$ state transition diagram for actor-actor communication.

provided by intermediate nodes, the receiver sends R_p to the sender as a rate feedback. Thus, $(RT)^2$ also provides flow control at the receiver while dynamically adjusting transmission rate.

B. $(RT)^2$ Protocol Operation for Actor-Actor Communication

In this section, we describe the protocol operation of $(RT)^2$ during actor-actor communication. The protocol operation is composed of two main states: i) start-up state, ii) steady state. In Fig. 7, the $(RT)^2$ protocol state diagram for actor-actor communication is shown. In the following, the operations at each state is described in detail.

- 1) **Start-Up State:** When establishing new connection between sender and receiver, the sender transports a probe packet towards the receiver in order to capture the available transmission rate quickly. Each intermediate node between the sender and receiver intercepts the probe packet and updates the bottleneck delay field of the probe packet, if the current value of delay information is lower than that of the intermediate node. Initially, the delay value of probe packet is assigned to zero. Therefore, after one round-trip-time, the sender gets estimated rate feedback from the receiver (the inverse of maximum delay), which results in quick convergence to available transmission rate. Furthermore, this probing mechanism of start up phase is also applied after route changes.
- 2) **Steady State:** This state consists of four substates: i) Increase, ii) Decrease, iii) Hold and iv) Probe. In the following, we describe the $(RT)^2$ protocol operations in each substate:
 - a) **Increase:** In this state, the sender increases its transmission rate according to the feedback coming from the receiver. Once an increase decision for sender transmission rate is taken, only m fraction of the difference between transmission rate feedback (R_f) and sender current transmission rate (R_c) is performed. The appropriate fraction value (m) for the transmission rate increase is obtained as follows: If the hop count along the data path is greater than or equal to 4 for that connection, m is set to 4. Otherwise, if the hop count is less than 4, then m is set to the actual hop count value along the path. The inherent spatial reuse property of underlying CSMA/CA based MAC protocol requires this normalization in transmission rate. The details can be found in [9],[11]. Note also

that in order to prevent fluctuations, transmission rate is only increased when a certain threshold (Δ_{rate}) is exceeded.

- b) **Decrease:** In this state, the sender reduces its transmission rate according to the feedback coming from the receiver. Note that the transmission rate is decreased until the average minimum transmission rate (R_{min}) is reached. R_{min} represents the minimum average rate that the application event-to-action delay bound can tolerate. R_{min} can be calculated as follows:

$$R_{min} = \frac{B}{\Delta_{re2a}} \quad (7)$$

where B represents the amount of packets that should be transmitted to the actor and Δ_{re2a} corresponds to remaining event-to-action deadline, which is the residual time of event-to-action delay bound Δ_{e2a} (see Section II-C), after the sensor-actor communication is performed.

- c) **Hold:** In this state, the required transmission transmission rate is reached. The sender does not change the transmission rate unless route failure or congestion occurs in the network.
- d) **Probe:** In this state, the sender sends a probe packet to the receiver so as to monitor the available transmission rate in the network like in start up phase. This phase might occur due to route errors (RERR), which is common in ad hoc communication networks. When the route error is observed, i.e., the route error information is received from intermediate nodes, first the sender freezes its transmission and periodically starts to send the probe packet to get transmission rate feedback from the receiver.

Overall, the (RT)² protocol dynamically shapes data traffic based on both delay bounds and the current conditions of the network to achieve the global objective of the network. Note that, in the protocol operation, the sender adjusts its transmission rate in response to the rate feedbacks from the receiver, which are sent with the period of T_{fdbk} . To prevent the sender from over-flooding the network in case all the feedback packets from the receiver are lost, the (RT)² protocol also performs a multiplicative decrease of transmission rate for each feedback periods, in which the sender does not receive feedback from the receiver up to a maximum of two feedback periods. After the second feedback period, if the sender still does not receive any feedback packet, it enters into probe state so as to monitor the available transmission rate in the network. In this respect, the period of feedback packets should be larger than one round-trip-time (RTT) and small enough to capture the network dynamics. Moreover, the period of the probe packets (T_p) should be small enough to capture the available transmission rate as soon as possible. For this purpose, the period of feedback packets (T_{fdbk}) and probe packets (T_p) are selected as $2 * RTT$. Note also that if the receiver rate feedback changes more than a certain threshold (Δ_{fdbk}), then the receiver immediately sends the rate feedback information to the sender without waiting for a feedback timer

```
(RT)2()
Sender:
/*Transmission Rate Update Procedure*/
packetfdbk received with Rf
If (CONGESTION)
/*Decrease transmission rate*/
Rc =  $\frac{R_c}{2}$ 
If (NO CONGESTION)
If (HopCount > 4)
m=4
else
m=HopCount
If ( $\frac{R_f - R_c}{R_c} > \Delta_{rate}$ )
Rc = Rc +  $\frac{R_c - R_f}{m}$ 
else If (Rc > Rf)
Rc = min(Rf, Rmin)
else
Hold Rc
Intermediate Node:
/*Calculate Di for individual packet*/
Avg(Di) =  $\alpha * Avg(D_i) + (1 - \alpha) * Current(D_i)$ 
If (Avg(Di) > Dth)
stamp D = Avg(Di)
Receiver:
Avg(D) =  $\alpha * Avg(D) + (1 - \alpha) * Current(D)$ 
When Tfdbk expires or  $\Delta_{fdbk}$  exceeded
/*Compute rate feedback*/
Rf =  $\frac{1}{Avg(D)}$ 
/*Flow control at the receiver*/
If (Rf > Rp)
Rf = Rp
stamp Rf on packetfdbk
send packetfdbk to sender
```

Fig. 8. Algorithm of the (RT)² protocol operation for actor-actor communication.

timeout event. Thus, the sender can adjust the transmission rate accordingly even for long RTT values.

In addition, in order to provide reliable actor-actor communication, (RT)² protocol relies upon probe packets to recover from subsequent losses and SACK packets to detect any holes in the received data stream. In order to prevent congestion in the reverse path, SACK packets are delayed in the receiver, i.e., one SACK packet for every d data packets received. Hence, this delayed SACK strategy of (RT)² protocol enables the receiver to control the amount of the reverse path traffic accordingly. The pseudo-algorithm of the (RT)² protocol for actor-actor communication is given in Fig. 8.

Note that actor-actor communication in WSNs is similar to the communication paradigm of ad hoc networks due to the small number of resource-rich actor nodes being loosely deployed. In the related literature, there are several transport protocols dealing with ad hoc networks [6]. In general, these research efforts propose either window-based [7] or rate-based solutions [14]. Although, these research efforts improve TCP performance to a certain extent, they do not address the unique requirements of WSNs completely. In Table 3, we summarize the main differences between (RT)² protocol and the previously developed ad hoc transport protocols [7],[14]. To evaluate the performance of (RT)² during actor-actor communication, we have also compared (RT)² with these ad hoc transport solutions. The details of performance evaluations are shown in the following section.

TABLE III
SUMMARY OF DIFFERENCES

Issue	TCP	TCP-ELFN[7]	ATP[14]	(RT) ²
Transmission type	Window-based	Window-based	Rate-based	Rate-based
Explicit congestion detection	Not addressed	Not addressed	Not addressed	Combined congestion detection
Rate feedback	Not addressed	Not addressed	Periodic with constant period	Periodic with dynamic period
Timing constraints	Not addressed	Not addressed	Not addressed	Addressed with R_{min}
Route failures	Packet Loss	ELFN feedback	Intermediate node feedback	Intermediate node feedback

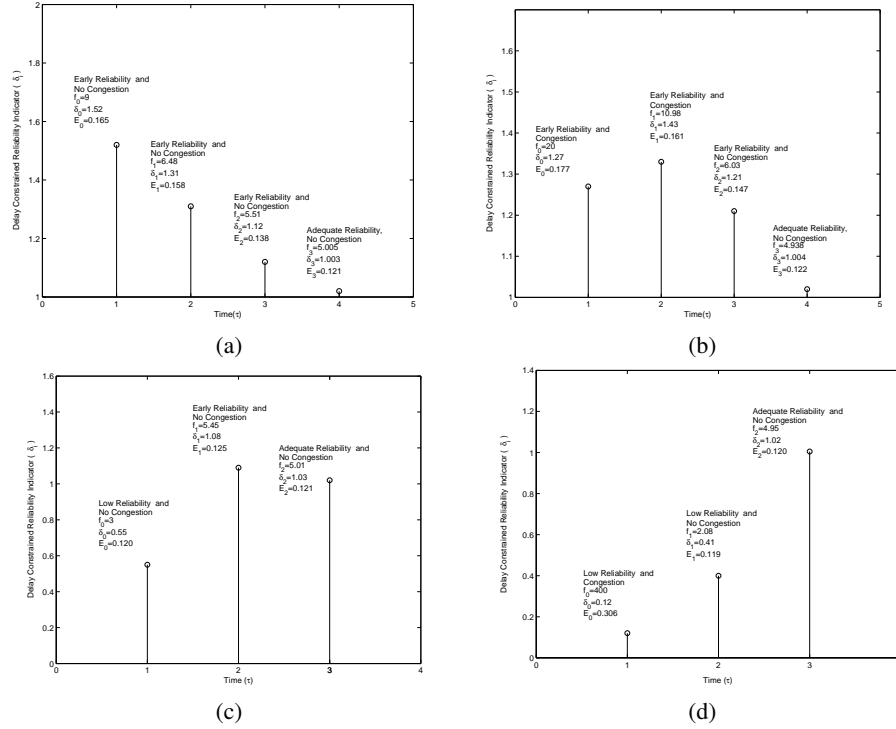


Fig. 9. The (RT)² protocol trace, when (a) early reliability and no congestion, (b) early reliability and congestion, (c) low reliability and no congestion, (d) low reliability and congestion, is observed.

V. (RT)² PERFORMANCE EVALUATION

In this section, we present the performance evaluation of the (RT)² protocol. In Section V-A, we report the performance results for the sensor-actor communication, while in Section V-B, we discuss the performance results for the actor-actor communication.

A. Sensor-Actor Communication

In order to evaluate the performance of the (RT)² protocol during sensor-actor communication, we developed an evaluation environment using ns-2 [15]. For sensor-actor communication scenario, the number of sources, sensor-actor delay bound and tolerance level was selected as $n = 81$, $1sec$ and $\epsilon = 5\%$, respectively. The event radius was fixed at $40m$. We run 10 experiments for each simulation configuration. Each data point on the graphs is averaged over 10 simulation runs. We use the same sensor node and simulation configurations provided in Table I in Section II-D. For this communication scenario, the main performance metrics that we employ to measure the performance of the (RT)² protocol are the convergence time to

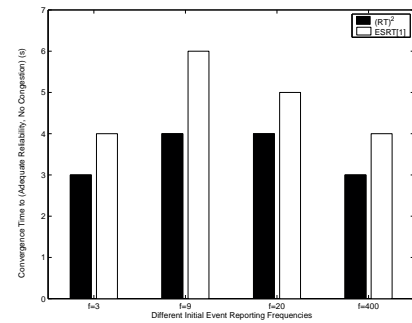


Fig. 10. The comparison of (RT)² and ESRT[1] for sensor-actor communication in terms of convergence times to (Adequate reliability, No congestion) condition.

(adequate reliability, no congestion) condition from any other initial network conditions and average energy consumption per packet (E_i) for each decision interval i .

The (RT)² protocol convergence results are shown in Fig. 9 for different initial network conditions. In Fig. 9, it is observed that (RT)² protocol converges to (Adequate reliability, No congestion) condition starting from any of the other initial network conditions

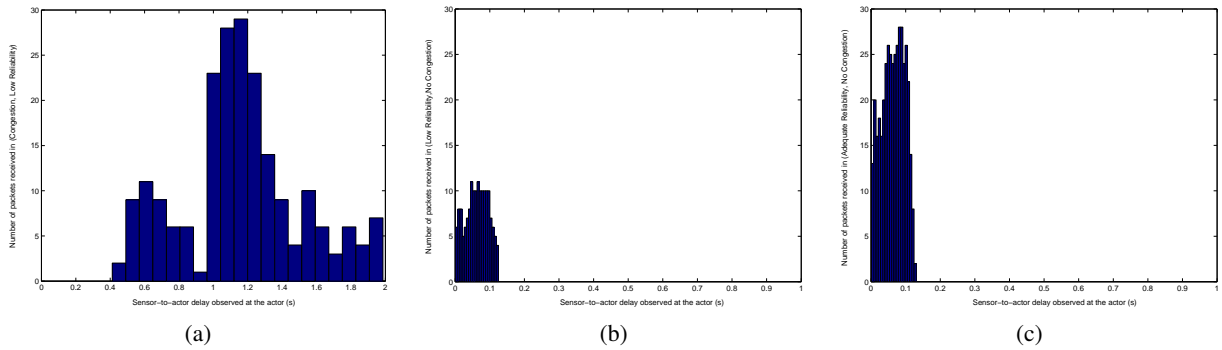


Fig. 11. The packet delay distribution in (a) (Low reliability, Congestion), (b) (Low reliability, No congestion), (c) (Adequate reliability, No congestion), conditions of $(RT)^2$ protocol.

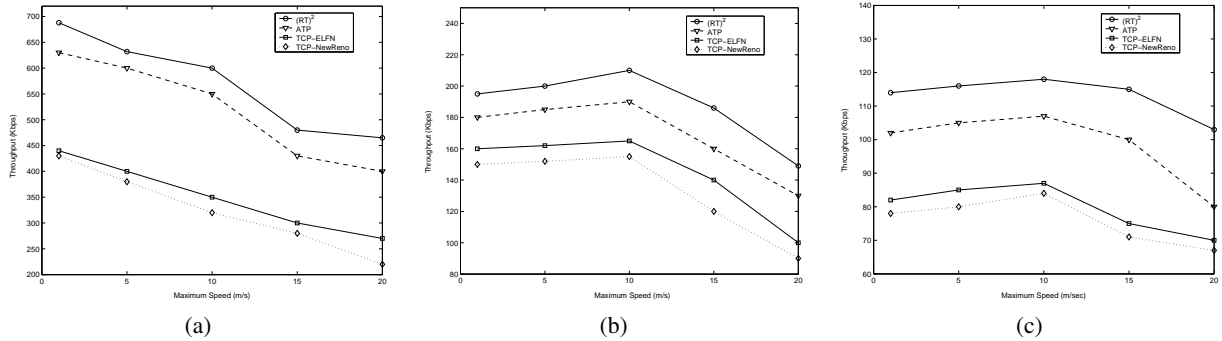


Fig. 12. Aggregate throughput for (a) 1 flow connection, (b) 5 flow connection, (c) 10 flow connection, when the maximum speed of the actors are varying.

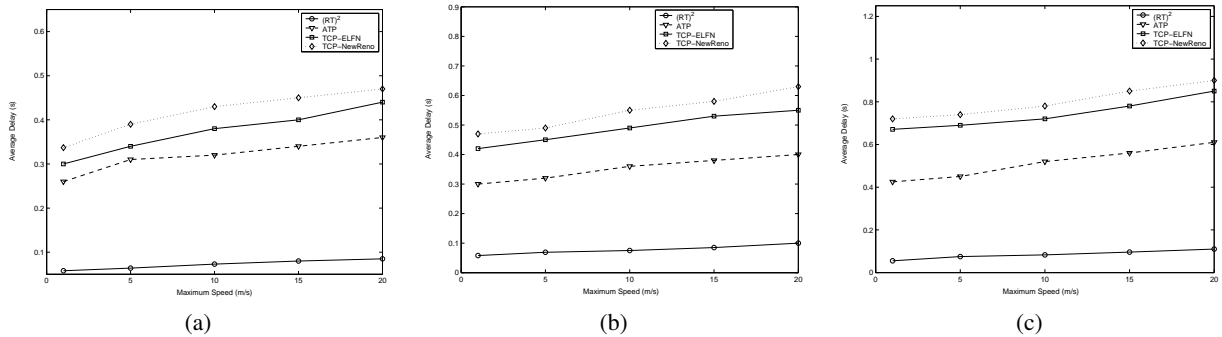


Fig. 13. Average packet delay for (a) 1 flow connection, (b) 5 flow connection, (c) 10 flow connection, when the maximum speed of the actors are varying.

discussed in Section III. In this sense, $(RT)^2$ is self-configuring and can perform efficiently under random, dynamic topology frequently encountered in WSN applications. In addition to convergence time, the average energy consumed per packet during sensor-actor communication, i.e., (E_i) , is also observed. As shown in Fig. 9, E_i decreases as the (no congestion, adequate reliability) state is approached which shows that energy consumption of the sensor nodes is also decreased while providing reliability constraints and delay bounds. Because of limited energy resources of the sensors, this result is also important for the proper operation of WSN. The performance of our reporting frequency update policies for sensor-actor communication can also be seen from the trace values and states listed within the Fig. 9.

To further investigate $(RT)^2$ protocol convergence results, we have compared $(RT)^2$ protocol and ESRT [1] protocol in terms

of convergence time to (Adequate reliability, No congestion) condition. The reason why we compare $(RT)^2$ protocol with ESRT is that both of them is based on event transport reliability notion unlike the other transport layer protocols addressing conventional end-to-end reliability in WSNs. As shown in Fig. 10, the convergence time of $(RT)^2$ protocol is much smaller than that of ESRT for different initial network conditions. This is because ESRT does not consider application-specific delay bounds while avoiding network congestion and adjusting reporting rate of sensor nodes.

To elaborate the relationship between the sensor-actor delay and $(RT)^2$ protocol operation, in Fig. 11, we also observed sensor-actor delay distributions of the event packets received at the actor, when there is a transition from (Low reliability, Congestion) condition to (Adequate reliability, No congestion) condition. Recall that the objective of $(RT)^2$ is to achieve

reliable and timely event data delivery with minimum energy consumption and with no network congestion.

As can be seen in Fig. 11(a), in (Low reliability, Congestion) condition, the packet delay values are as high as 2 sec and the number of received packets is not sufficient to provide delay-constrained event reliability. This is because in this case, the network capacity is exceeded and thus, the data packets at the forwarding sensor nodes are either started to be dropped or experienced large delays due to network congestion. Hence, when (Low reliability, Congestion) condition is detected, $(RT)^2$ protocol aggressively reduces reporting frequency to reach optimal network condition as soon as possible. After reporting frequency is updated accordingly, congestion is released and (Low reliability, No congestion) condition is reached as shown in Fig. 11(b). However, in (Low reliability, No congestion) condition, the delay-constrained reliability objective of $(RT)^2$ protocol is not met, i.e., the number of packets received providing sensor-actor delay bound is not sufficient. Therefore, in order to improve the delay-constrained event reliability, data reporting frequencies of sensor nodes are increased. With this reporting frequency update, (Adequate reliability, No congestion) condition is reached, as seen in Fig. 11(c). In this condition, no congestion is observed and delay-constrained reliability objective is provided. Note that in (Adequate reliability, No congestion) condition, packet delay values are below 0.1 sec for almost 100% of the event packets and the number of packets received is sufficient to provide delay-constrained event reliability. Consequently, $(RT)^2$ protocol converges to optimal network condition by avoiding network congestion and adjusting reporting frequency of the source nodes in a controlled manner.

B. Actor-Actor Communication

In this section, we present the performance results of the $(RT)^2$ protocol during actor-actor communication. For the simulations, we set up an evaluation environment using ns-2 [15]. The simulations for this scenario are performed for a 200m x 200m deployment field consisting of 10 actor nodes, distributed randomly over the field. In addition, to take into account the mobility of the actors during actor-actor communication, we have used the random way-point model. In this mobility model, we consider maximum speeds of 1m/s, 5m/s, 10m/s, 15m/s and 20m/s for mobile actor nodes. The packets generated are of size 1000 bytes. Other simulation parameters are the same as those listed in Table I in Section II-D.

For actor-actor communication scenario, the performance of the $(RT)^2$ protocol is evaluated and compared against TCP-NewReno, TCP-ELFN [7] and ATP [14]. The main performance metrics that we employ to measure the performance of the $(RT)^2$ protocol are aggregate throughput, average packet delay and average transfer time. Here, the aggregate throughput reflects the number of packets successfully received at the destination. By average packet delay, we refer to average latency of data packets during actor-actor communication. The average transfer time represents the time required to transport 10 MB event information in actor-actor communication. All the simulations last for 1000 sec. We run 10 experiments for each simulation configuration and each data point on the graphs is averaged over 10 simulation runs.

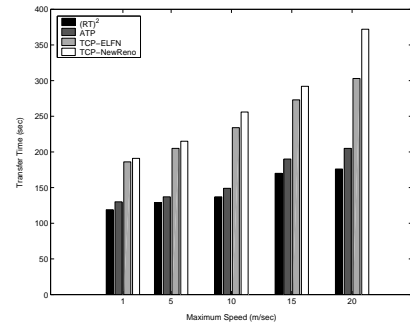


Fig. 14. Average transfer time of 10 MB event information, when the maximum speed of the actors are varying.

1) *Aggregate Throughput*: In Fig. 12, we present the aggregate throughput results of the $(RT)^2$ protocol and other ad hoc transport protocols, i.e., TCP-NewReno, TCP-ELFN [7] and ATP [14]. In this simulation experiment, different number of flow connections are used and source-destination pairs are randomly chosen from 10 actor nodes. In terms of aggregate throughput, the $(RT)^2$ protocol outperforms other transport protocols under comparison, since $(RT)^2$ dynamically shapes data traffic according to the channel conditions and intermediate node feedbacks. In addition, the proper reaction of $(RT)^2$ to congestion related and non-congestion related losses, such as route failures, prevents it from any performance degradation during actor-actor communication. For example, for 5 flow connection and 10m/s speed, we obtain that the aggregate throughput achieved by $(RT)^2$ during actor-actor communication is around 40%, 30% and 15% higher than that of TCP-NewReno, TCP-ELFN and ATP, respectively. Note also that in our simulations, we observe that the aggregate throughput values for rate-based transport protocols, i.e., $(RT)^2$ and ATP, are higher than those of window-based transport protocols, i.e., TCP-ELFN and TCP-NewReno. This is because rate-based transport protocols capture the available bandwidth more quickly compared to window-based transport protocols.

2) *Average Delay*: In Fig. 13, we also show the average packet delay results of the $(RT)^2$ protocol and the other transport protocols. As shown in Fig. 13, for all simulation configurations, the average packet delay values of $(RT)^2$ are much lower than those of other protocols, since $(RT)^2$ captures the available bandwidth in the network quickly and does not allow a burst of packet transmissions with explicit congestion notification and rate feedback based mechanisms, which significantly improves the network performance. For example, for 10 flow connection and 15m/s speed, we obtain that the average packet delays achieved by $(RT)^2$ are approximately eight, seven and five times lower than that of TCP-NewReno, TCP-ELFN and ATP, respectively. This is so crucial because of timely event detection and action performance objectives of the WSANs.

3) *Average Transfer Time*: In Fig. 14, we observe average transfer time of 10 MB event information during actor-actor communication, when $(RT)^2$ and other transport protocols are used separately. For this simulation scenario, one source-destination pair is randomly selected. As shown in Fig. 14, $(RT)^2$ achieves the necessary information transfer in a shorter time amount

compared to other transport protocols. For example, for $15m/s$ speed, we obtain that the average transfer time achieved by $(RT)^2$ during actor-actor communication is around 90%, 80% and 15% lower than that of TCP-NewReno, TCP-ELFN and ATP, respectively. This result is also important to achieve timely action decisions during the operation of the network.

VI. CONCLUSION

To address the communication challenges introduced by the coexistence of sensors and actors in WSANs, a Real-Time and Reliable Transport $(RT)^2$ protocol for WSANs is presented in this paper. The $(RT)^2$ protocol dynamically adjusts its protocol configurations to adapt to heterogenous characteristics of WSANs. It also enables the applications to perform right actions timely by exploiting both the correlation and the collaborative nature of WSANs.

The objective of $(RT)^2$ is to reliably and collaboratively transport event features from the sensor field to the actor nodes and to timely react to sensor information with an accurate action. In this respect, the $(RT)^2$ protocol simultaneously addresses congestion control and timely event transport reliability objectives in WSANs. To the best of our knowledge, this is the first research effort focusing on real-time and reliable event transport and action performance objectives of WSANs. Performance evaluation via simulation experiments show that $(RT)^2$ achieves high performance in terms of reliable event detection, communication latency and energy consumption in WSANs.

APPENDIX

In Time Critical Event First (TCEF) scheduling policy, the main challenge is the computation of remaining time to deadline in the lack of globally synchronized clock. For this purpose, when an event packet is received at a sensor node i , its underlying Medium Access Control (MAC) layer stamps the arrival time, $t_{a,i}$, to the event packet. This packet is processed by the routing layer and transported to the chosen forwarding node k via MAC layer. Note that the MAC layer of sensor node i spends some time to capture the channel and might re-transmit the event packet several times until receiving an ACK from the sensor node k . In order for sensor node i to piggyback the updated remaining time to deadline, the sensor node i measures the elapsed time just before it actually transmits the data packet to the physical link as follows:

$$t_{e,i} = t_{d,i} - t_{a,i} + t_{t,i} + t_{p,i} \quad (8)$$

where $t_{e,i}$, $t_{d,i}$, $t_{a,i}$, $t_{t,i}$ and $t_{p,i}$ represent the elapsed time to deliver the event packets to the next hop successfully, the departure time, the arrival time, transmission delay and the propagation delay for the event packet at sensor node i , respectively⁷. Then, the sensor node i updates the remaining time to deadline for the chosen forwarding node k as follows:

$$\Delta_{r,k} = \Delta_{r,i} - t_{e,i} \quad (9)$$

⁷Since the propagation delay in WSANs is negligibly small because of high sensor node density, it is ignored in (8). In addition, the transmission delay in (8) can be calculated using the transmission rate and the length of the data packet.

where $\Delta_{r,k}$ and $\Delta_{r,i}$ represent the remaining time to deadline at the sensor node k and i , respectively. Thus, once the sensor node k successfully receives the data packet, the data packet contains the updated remaining time to deadline. Then, based on packet arrival time, $t_{a,k}$, and the updated remaining time to deadline, $\Delta_{r,k}$, the intermediate sensor node k calculates the service order index of the data packet, s_k , as follows:

$$s_k = t_{a,k} + \Delta_{r,k} \quad (10)$$

Note that the lower the service order index, the higher the priority of the event data packet and the sooner the packet is served. Thus, using TCEF service index assignment, the event packets are given high priority at the sensor nodes, as their remaining time to deadline decreases. This updating process of TCEF scheduling continues until the event packet is received by the actor node. Therefore, the computation of remaining time to deadline in TCEF scheduling is performed in the lack of globally synchronized clock.

REFERENCES

- [1] O.B. Akan and I.F. Akyildiz, "ESRT: Event-to-Sink Reliable Transport for Wireless Sensor Networks," in *IEEE/ACM Transactions on Networking*, October 2005, also in *Proc. ACM MOBIHOC 2003*, pp. 177-188, Annapolis, Maryland, USA, June 2003.
- [2] I.F. Akyildiz and I.H. Kasimoglu, "Wireless Sensor and Actor Networks: Research Challenges," *Ad Hoc Networks Journal (Elsevier)*, October 2004.
- [3] I.F. Akyildiz, M.C. Vuran, O.B. Akan and W. Su, "Wireless Sensor Networks: A Survey REVISITED," to appear in *Computer Networks Journal (Elsevier)*, 2005.
- [4] K. Chandran et al., "A Feedbackbased Scheme for Improving TCP Performance in Ad Hoc wireless networks," *IEEE Personal Communications Magazine*, vol. 8, no. 1, pp. 34-39, February 2001.
- [5] M. Caccamo et al., "An Implicit Prioritized Access Protocol for Wireless Sensor Networks," *Proc. IEEE Real-Time Systems Symposium*, Texas, December 2002.
- [6] I. Chlamtac et al., "Mobile ad-hoc networking: imperatives and challenges", *Ad Hoc Networks Journal (Elsevier)*, vol. 1, no. 1, pp. 13-64, 2003 .
- [7] G. Holland and N.H. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," in *Proc. of ACM MOBICOM 1999*, pp. 219-230, Seattle, WA, August 1999.
- [8] D. Johnson et al., "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.
- [9] Z. Fu et al., "The Impact of Multihop Wireless Channel on TCP Throughput and Loss," in *Proc. IEEE INFOCOM 2003*, San Francisco, 2003.
- [10] B. Hull, K. Jamieson and H. Balakrishnan, "Techniques for Mitigating Congestion in Sensor Networks" in *Proc. ACM SENSYS 2004*, November 2004.
- [11] J. Li et al., "Capacity of Ad Hoc Wireless Network," in *Proc. ACM MOBICOM 2001*, Rome, Italy, July 2001.
- [12] M. Schwartz, "Mobile Wireless Communications," Cambridge University Press, New York, 2004.
- [13] F. Stann and J. Heidemann, "RMST: Reliable Data Transport in Sensor Networks," in *Proc. IEEE SNPA 2003*, pp. 102-112, Anchorage, Alaska, May 2003.
- [14] K. Sundaresan, V. Anantharaman, H.Y. Hsieh and R. Sivakumar, "ATP: A Reliable Transport Protocol for Ad-hoc Networks," in *Proc. ACM MOBIHOC 2003*, pp. 64-75, Annapolis, Maryland, USA, June 2003.
- [15] The Network Simulator, ns-2, <http://www.isi.edu/nsnam/ns/index.html>.
- [16] MICA Notes, http://www.xbow.com/Products/Wireless_Sensor_Networks.htm
- [17] M.C. Vuran, O.B. Akan and I.F. Akyildiz, "Spatio-Temporal Correlation: Theory and Applications Wireless Sensor Networks," *Computer Networks Journal (Elsevier)*, vol. 45, no 3, pp. 245-259, June 2004.
- [18] C.Y. Wan, S.B. Eisenman and A.T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks," in *Proc. ACM SENSYS 2003*, November 2003.
- [19] 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.
- [20] X. Yu, "Improving TCP Performance over Mobile Ad Hoc Networks by Exploiting Cross-Layer Information Awareness," in *Proc. ACM MOBICOM 2004*, Philadelphia, September 2004.