International Journal of Distributed Sensor Networks
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t714578688

Transport Protocols for Wireless Sensor Networks: State-of-the-Art and Future Directions

To link to this article: DOI: 10.1080/15501320601069861
URL: http://dx.doi.org/10.1080/15501320601069861

International Journal of Distributed Sensor Networks, 3: 119–133, 2007

ISSN: 1550-1329 print/1550-1477 online
DOI: 10.1080/15501320601069861

Transport Protocols for Wireless Sensor Networks: State-of-the-Art and Future Directions

JUSTIN JONES and MOHAMMED ATIQUZZAMAN

School of Computer Science, University of Oklahoma, Norman

Characteristics of wireless sensor networks, specifically dense deployment, limited processing power, and limited power supply, provide unique design challenges at the transport layer. Message transmission between sensor nodes over a wireless medium is especially expensive. Care must be taken to design an efficient transport layer protocol that combines reliable message delivery and congestion control with minimal overhead and retransmission. Sensor networks are created using low cost, low power nodes. Wireless sensors are assumed to have a finite lifetime; care must be taken to design and implement transport layer algorithms that allow maximum network lifetime. In this paper we present current and future challenges in the design of transport layers for sensor networks. Current transport layer protocols are compared based on how they implement reliable message delivery, congestion control, and energy efficiency.

Keywords Wireless; Networking; Wireless Sensor Network; WSN; Transport Layer; Layer 4; End-to-End Reliability

1. Introduction

Wireless sensor networks (WSNs) provide a powerful means to collect information on a wide variety of natural phenomena. WSNs typically consist of a cluster of densely deployed nodes communicating with a sink node which, in turn, communicates with the outside world. WSNs are constrained by low power, dense deployment, and limited processing power and memory. WSNs are composed of small, cheap, self-contained, and disposable sensor nodes. The unique constraints imposed by WSNs present unique challenges in the design of such networks.

The need for a transport layer to handle congestion and packet loss recovery in WSNs has been debated; the idea of a cheap, easily deployable network runs contrary to the costly, lengthy process of implementing a unique and specialized transport layer for a WSN. WSNs have advanced to the level of specialization where congestion control and reliability can be incorporated at each individual node.

Reliable data transmission in WNSs is difficult due to the following characteristics of WSNs:

- limited processing capabilities and transmission range of sensor nodes;
- close proximity to ground causes signal attenuation or channel fading which leads to asymmetric links;
- close proximity to ground and variable terrain also leads to shadowing which can effectively isolate nodes from the network;
- conservation of energy requires unused nodes and wake only when needed;
- dense deployment of sensor nodes creates significant channel contention and congestion.

Address correspondence to Mohammed Atiquzzaman, School of Computer Science, University of Oklahoma, Norman, OK73019-6151. E-mail: atiq@ou.edu

The above characteristics can cause loss of data in WSNs. Fortunately, WSNs also provide unique features that can be leveraged to help mitigate losses and design energy-efficient transport layer protocols by network designers. For example,

1. When the nature of the data allows, it can be aggregated at intermediate nodes.
2. Network density, multiple paths to any given destination, and data aggregation in combination with a good choice of network layer can lessen some of the losses due to channel fading and shadowing.
3. Some amount of loss can be made acceptable by employing data aggregation at the sensor nodes.
4. Data aggregation may result in smaller packet size and consequently lower packet loss.
5. Granularity of sensing an event can be controlled.
6. Some events may require a very rough granularity.

Traditional transport layer protocols, such as TCP, are not suitable for severely resource constrained WSNs having characteristics which are different from traditional wired networks. The *objective* of this paper is to illustrate the need for a standard transport layer in WSNs, outline future challenges involved in designing a transport layer protocol that fits the unique constraints imposed by WSNs, and present current implementations of transport layers for WSNs.

The *difference* between this paper and previous papers on transport layers in WSNs is that, instead of proposing a new transport layer protocol, we discuss the issues and challenges in the design of transport layer protocols. The *contribution* of this paper is to illustrate the unique requirements of a transport layer protocols for sensor networks, and compare a number of transport layer protocols that have been proposed in the literature.

The rest of the paper is organized as follows. Various types of reliability to be handled by the transport layer in a sensor network are discussed in Sec. 2. A number of transport layer protocols that have been proposed in the literature for WSNs are discussed in Sec. 3, followed by a comparison of the protocols in Sec. 4. Concluding remarks are given in Sec. 5.

2. Reliability in Wireless Sensor Networks

Traffic from many applications in WSNs is considered loss tolerant. Loss tolerance in WSNs is due to the dense deployment of sensor nodes and data aggregation properties, giving rise to directional reliability. The design of WSN transport layer protocols should exploit directional reliability to lower the number of transmissions, especially for sensors that are close together and are expected to generate highly correlated data [20], and decrease the computational overhead by lowering the amount of data to be aggregated.

Some transport layer protocols only offer unidirectional reliable message delivery, where the idea of directional reliability is especially important. In the rest of this section, we discuss the following three types of reliability in a WSN:

- Point-to-point – Communication between sink and a remote host,
- Point-to-multipoint – Communication between sink and sensor nodes,
- Multipoint-to-point – Communication between sink and multiple wireless sensors.

2.1 Point-to-point Reliability

The transport connection between the sink and a remote host uses a traditional TCP/IP transport layer. Sinks may either be robust nodes on a network with continual power and

much more computational power than sensor nodes, or they may be a more robust version of a sensor node. In the latter case, a lightweight TCP/IP protocol, as described in section 4.1, may be beneficial to these types of sink/proxy nodes.

2.2 Point-to-multipoint Reliability

Messages originating at the sink may be queries and control messages, such as those related to congestion control and reprogramming the sensor nodes. These messages generally need to be delivered to sensor nodes with a higher degree of reliability than those originating at source sensor nodes. Loss of these messages could be detrimental to the life of the sensor network.

2.3 Multipoint-to-point Reliability

Sensor nodes may process information received from other sensor nodes about an observed phenomenon. This process is called data aggregation and allows nodes to reduce the amount of information that must be forwarded. Data aggregation can reduce the impact of data loss by providing an averaged or smoothed value. Consequently, we may not be able to sense the phenomenon with fine granularity, but the impact of loss is reduced by sensing phenomena at a coarse level.

Even though sensor networks are fault tolerant [11] we still have to guarantee the quality of the received data, i.e. the gathered data should be representative of the region queried, or event sensed. Collecting data tainted by packet loss can be more dangerous than not collecting any data at all. For example, if the sink queries the WSN and receives no response, we can assume we have experienced loss after some interval, but if we receive misleading or skewed data we have no way to verify that the data should be discarded at the sink. Figure 1 illustrates this idea. In Fig. 1 (a), the message never reaches the sink, we do not have the data, but we do not have corrupt data. After some interval, the sink may realize that no data has been received and resend the request.

Figure 1 (b) illustrates, a worse case scenario for loss with data aggregation. The gray areas indicate nodes that are unreachable. The aggregated response of many sensor nodes could be dropped, and data is forwarded from a sensor further from the event source. If the

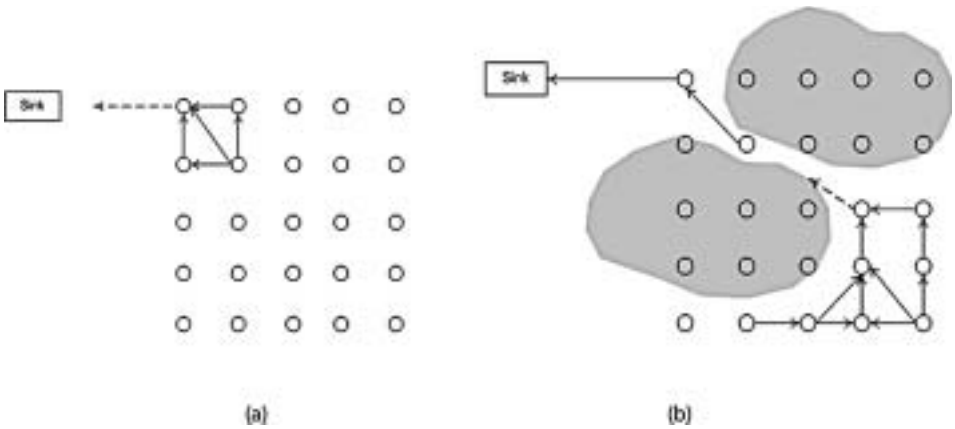

FIGURE 1 Sensor network loss combined with data aggregation could cause data to be skewed in certain situations.

node is sufficiently removed from the event center, the data may not accurately reflect the event. In these cases it would be desirable to have a measure of the "goodness" of the data sent to the sink [9].

In this case, the "goodness" of the data becomes a new measure of the reliability of the data. The accuracy or granularity that is acceptable for the event varies between applications. ESRT is a proposed transport layer protocol for WSNs that allows control over the level of granularity with which the event is detected [9].

3. Transport Protocols for Sensor Networks

In addition to energy-efficient transport layer protocols in resource constrained WSNs, the protocol should also support

- reliable message delivery,
- congestion control, and
- energy efficiency.

The need for a transport layer protocol in WSNs has been debated. Some have suggested that (a) loss detection and recovery can be handled below the transport layer and mitigated using data aggregation, and (b) congestion is not an issue because sensor nodes spend most of the time sleeping resulting in sparse traffic in the network.

In contrast to the above arguments against the need for a transport layer protocol, Yarvis et al. [16] and Dunkels et al. [8] have shown that the generally dense deployment of sensor nodes give rise to congestion in a WSN. Data from sensor nodes to sink (multipoint-to-point) may suffer from channel contention; in the absence of congestion control, the ability of the sensor nodes to deliver data to the sink decreases.

Wan et al. [7] and Stan et al. [8] demonstrated scenarios where data must be delivered reliably in WSNs. In such cases, it is not sufficient to rely only on loss detection and reliability techniques at layers below the transport layer, since layers beneath the transport layer do not provide guaranteed end-to-end reliability.

The need for reliable message delivery and congestion control suggest that WSNs should have a transport layer, just as 802.3 and 802.11 networks need a transport layer. However, WSNs add a new constraint—energy efficiency. To prolong the lifetime of a WSN, an ideal transport layer needs to support reliable message delivery and provide congestion control in the most energy efficient manner possible. In the rest of this section, we discuss a number of transport layer protocols, including those which have been suggested for WSN.

3.1 TCP/IP

TCP/IP has been used successfully in wired 802.3 and wireless 802.11 networks and has been discussed as a possible transport layer for WSN [14]. Certain attributes, such as IP addressing for individual nodes, unnecessary header overhead for data segments, no support for data centric routing, a heavyweight protocol stack, and an end-to-end reliability scheme that attributes segment losses network congestion, of TCP/IP; however, they make it unsuitable for use in WSNs without modification. Even if TCP/IP is not entirely suitable for WSNs, it is informative to compare TCP/IP to transport protocols designed specifically for WSNs. Such a comparison helps to illustrate that WSNs operate in a different paradigm, and thus need specially designed transport layers to meet their unique needs.

TCP/IP may not be suitable for standard sensor nodes in a WSN, but may still be used at the sink to communicate with other remote endpoints. Sensor nodes with high robustness, such

as Crossbow [18], may use TCP/IP as a virtual sink or proxy between the WSN and the remote host to reduce the number of retransmissions of a data segment by less powerful sensor nodes.

3.1.1 Loss Detection/Recovery. TCP/IP, by default, uses an ACK-based end-to-end reliability mechanism; however, an end-to-end reliability mechanism is not appropriate for sensor networks, given their high loss rates due to signal attenuation and path loss arising from low power radios and channel contention from dense sensor deployment. The probability of receiving an errored packet increases exponentially with the increase in the number of hops on a WSN. To reduce this problem, Dunkels et al. [14] suggest Distributed TCP Caching (DTC) which allows intermediate nodes to cache data segments; on detection of loss, the lost packets can be distributed to nodes using local retransmissions.

DTC requires intermediate nodes to cache intermediate segments. In a worst case scenario, when none of the surrounding nodes have the required segment cached, DTC degrades to end-to-end recovery (see Fig. 2). To help mitigate this problem, a sensor node caches the highest segment number it has seen. Although this improves the chances of a local neighbor having the required segment, it does not eliminate the possibility of DTC degrading to end-to-end recovery.

3.1.2 Congestion Control. No modification of the congestion control mechanism has been suggested by Dunkels et al. [14]. However, DTC should localize the reduction in transmission rates when segments can be recovered form neighboring sensor nodes.

Although the overhead needed to run TCP/IP seems prohibitive for a WSN, it may still be desirable to use TCP/IP for certain types of sensor nodes, specifically those which are less resource-restrained.

3.2 Pump Slowly, Fetch Quickly (PSFQ)

Pump Slowly Fetch Quickly (PSFQ) [7] is a transport layer protocol, designed specifically to meet the unique resource challenges presented by WSNs with a focus on point-to-multipoint

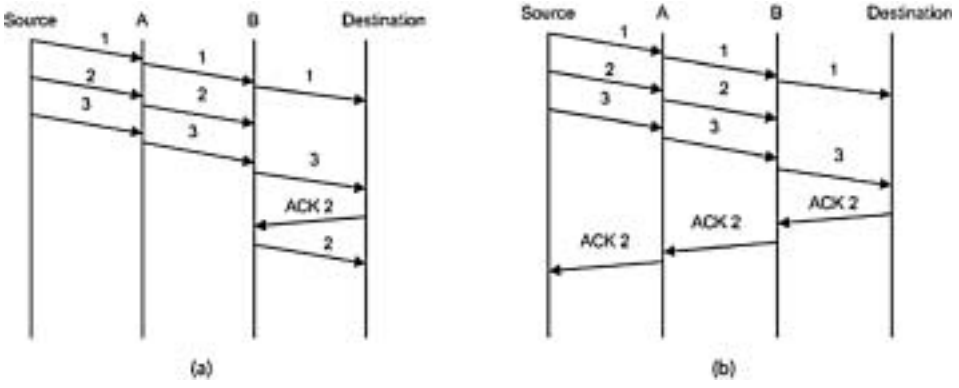

FIGURE 2 DTC caching performs aggressive hop-by-hop recovery when loss is detected; however, if the lost packet has been removed from cache, the NACK must be forwarded on potentially to the destination.

(a) A best case scenario. Neighbor B has cached packet 2 and simply forwards it back to the intended destination.

(b) A worst case scenario. The Source sends a three part message. Packet 2 is dropped by node B, but by the time ACK 2 reaches B it has already forwarded packet 3. Thus the acknowledgement to resend packet 2 has to be sent all the way back to the source.

TABLE 1 Problems and proposed solutions to using TCP/IP on WSN [14]

Problem	Description	Solution
IP addressing architecture	Sensor networks are dense networks with as many as 10 nodes per cubic meter [2]. This combined with the limited memory available to sensor nodes makes traditional IP addressing impractical.	Use spatial IP addressing.
Header overhead	Communication is one of the most costly activities in a WSN [11]. The transmission of large headers of TCP/IP requires lot of energy.	Use header compression.
No support for data centric routing	Routing in IP networks is based on the host and network address. Routing in sensor networks needs to be data centric.	Use an application overlay network.
Sensor Nodes are severely resource limited.	The TCP/IP stack is considered to be too heavyweight for sensors with limited capabilities. Sensor nodes with limited memory may not be able to support a TCP/IP implementation.	Dunkels et al. [15] have shown that a TCP/IP stack can be implemented for 8-bit processors with only a few hundred bytes of memory.
TCP performance and energy inefficiency	End-to-end acknowledgement and retransmission scheme in TCP translate to unnecessary expense in networks with multiple hops and limited energy.	Implement an energy-efficient distributed mechanism for acknowledgements and retransmissions.

reliability. Data is pumped slowly from a root node into the network. Sensor nodes that experience loss can recover data segments by fetching them quickly from their immediate neighbors on a hop-by-hop basis. To reduce signal overhead, nodes signal the loss of segments using negative acknowledgement, rather than acknowledging each received packet.

PSFQ is based on the assumption that a WSN will generate light traffic most of the time; thus, it is designed to avoid loss due to instability of the wireless medium, rather than loss due to network congestion. As such, it does not offer any active congestion control scheme.

PSFQ is designed for tasks that require reliable delivery of all message segments. Its focus is on the transport of binary images, such as new sensor control programs used for sensor retasking in the field. Since PSFQ expects low network traffic and does not provide any active congestion control scheme it may not be efficient for reliable transport of multipoint-to-point sensor events.

3.2.1 Loss Detection/Recovery. Reliability in PSFQ is achieved with a negative acknowledgement (NACK)-based quick fetch mechanism. Loss is detected using gap detection. Each injected message has a sequence number in the message header. If a receiving node determines a gap in sequence number, it begins aggressively broadcasting NACK

messages to try to recover the lost message before the injection interval T_{min} is exceeded, and the next packet is sent.

In case a downstream node needs to quickly recover a lost packet, a NACK-based scheme requires upstream nodes to buffer messages that have been sent downstream, to conserve energy, NACK requests are bundled, as illustrated in Fig. 3. A sending node near the receiving node caches message segments it forwards; this recovery scheme is called "local recovery" PSFQ's assumption that all intermediate nodes store all the segments they forward may not be feasible on a real WSN due to a limited cache size on sensor nodes. At the very least the amount of segments stored would have to be heavily optimized for the small amount of storage space available on sensor nodes.

A negative acknowledgement gap detection scheme leaves holes at the beginning and end of messages potentially undetected. Detecting dropped segments at the beginning of messages can only be done if one message segment is received downstream. If a message consists of only a single segment, and that segment is somehow dropped on the way downstream, it will not be detected. Likewise, a node cannot detect the loss of the last data

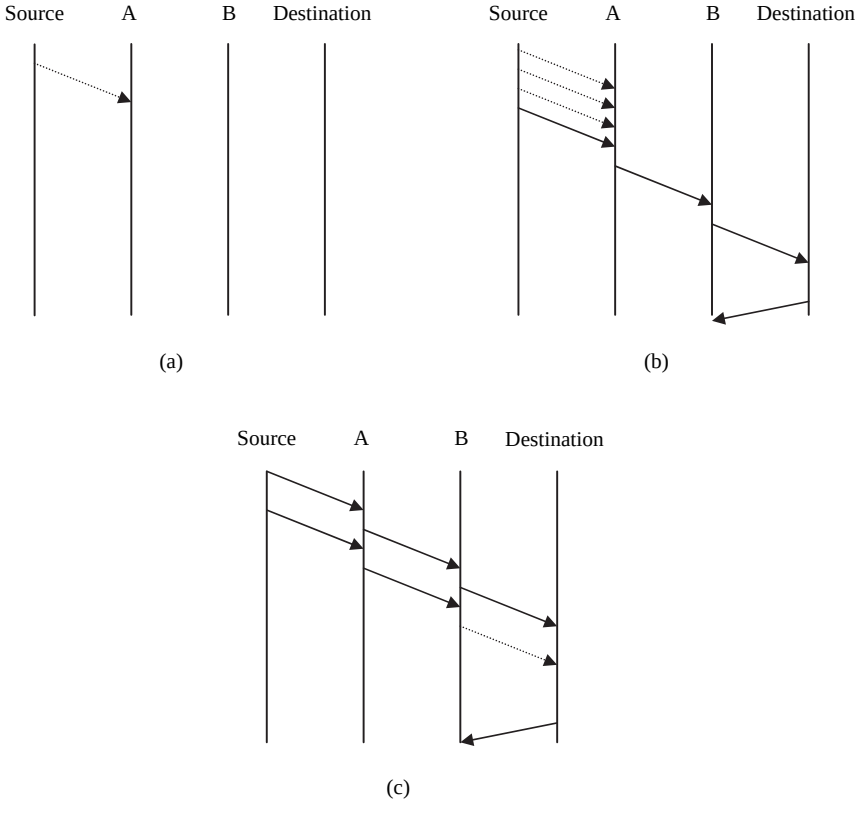

FIGURE 3 Loss detection/recovery in PSFQ. (a) A message consisting of a single data segment is sent from the Source and never received at node A. Since no data is ever received at node A, nothing can be recovered. (b) All data segments up to the last data segement are lost. The Destination receives the last data segment and is able to NACK for retransmission of all the lost data segments at once. (c) The last data segment is lost. The Destination creates a proactive fetch after some interval to retrieve the lost data segment.

segment in a transmission, since it will not be able to tell if the data segment has been lost or has not reached it yet.

To address the shortcomings of gap detection, PSFQ uses a "proactive fetch" [7] scheme that allows it to set a timer that starts from the receipt of the last message until the next message is received. This continues while the total size of the received data segments is less than the file size specified in the header field of the inject message. If no message is received from any upstream neighbor before the timer times out, then a downstream sensor node will manually generate and broadcast a NACK event to actively try to recover the segements that were presumably lost. To save energy, proactive fetches, like the normal fetch mechanism, aggregate missing message segments into one NACK message.

PSFQ will buffer messages received if a gap is detected until the lost data segments have been recovered. As a side effect this means that data is delivered in order.

3.2.2 Congestion Control. PSFQ assumes light traffic in most cases in a WSN; not much is done to detect and control congestion. Instead, PSFQ attempts to avoid introducing congestion into the network through the use of a time-to-live (TTL) field in the segment header. Also, if a message with a sequence number lower than the last forwarded message is received, the message is silently discarded. Silently discarding messages helps to decrease the likelihood of flooding between the sensor nodes.

3.3 Reliable Multi-Segment Transport (RMST)

RMST, first proposed in [8], is a reliable transport layer for WSNs. RMST is meant to operate on top of the gradient mechanism used in directed diffusion [5]. RMST adds two important features to directed diffusion [8],

1. fragmentation and reassembly of segments, and
2. reliable message delivery.

One of the most intriguing features of RMST is that it is an extension of directed diffusion that can be applied to a sensor node and configured without having to recompile. Essentially RMST is a plugin transport layer mechanism for an already widely accepted and studied WSN network layer.

RMST can be configured to allow hop-by-hop recovery (using local broadcast NACK) or end-to-end recovery (end-to-end NACK) at run time, and can be combined with a MAC-level Automatic Repeat Query (ARQ). The configuration between hop-by-hop (cached) recovery and end-to-end (noncached) recovery can be configured at the sensor nodes at runtime.

The main contribution of the paper by Stann et al. [8] was to compare the combination of transport layer reliability and lower layer recovery mechanisms. Reliable delivery was compared to using end-to-end recovery at the transport layer, hop-by-hop recovery at the transport layer, and hop-by-hop recovery at the MAC layer using an Automatic Repeat Request (ARQ).

RMST considers reliable transport in the point-to-multipoint direction and multipoint-to-point direction with special emphasis given to sensor re-programming or transfer of binary objects, when the loss of a single segment would irreparably damage the entire message.

3.3.1 Loss Detection/Recovery Mechanisms. RMST employs a Negative Acknowledgement (NACK) gap detection to detect and recover lost messages similar to the scheme used by PSFQ. However, RMST makes no guarantee of in-order message delivery, rendering loss detection is particularly difficult since it is difficult for sensor nodes to determine whether gaps are caused by out-of-order delivery or lost messages. To help assuage

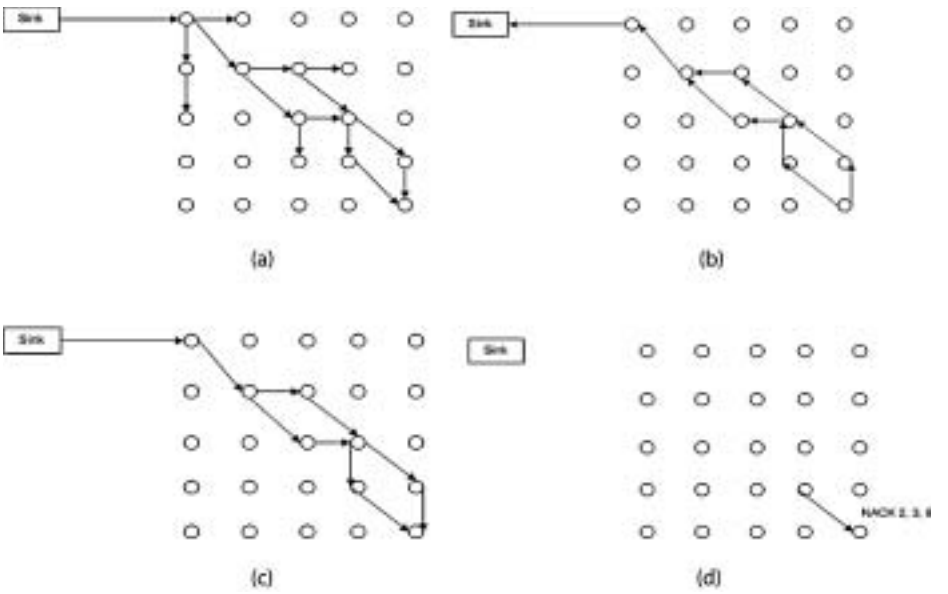

FIGURE 4 An example of the RMST protocol.
(a) Interest is disseminated through the network, using directed diffusion.
(b) A reinforced gradient path is established between the source and the sink.
(c) RMST snoops on the reinforced path at each hop and uses it to establish a backchannel for NACK that need to be sent in the source direction.
(d) RMST allows nodes to NACK multiple fragments of a message at once. Here the node is sending NACK 2, 3, 8 asking for retransmission of packet fragments 2, 3, and 8.

this problem RMST creates a "hole map" for detected gaps and assigns a "watchdog" timer to generate an automatic NACK for any segment that has not been received in the timer interval [8].

Multiple fragment numbers can be combined into a single NACK, as in PSFQ, to cut down on the network traffic generated during message recovery, as shown in Fig. 4. Since RMST uses the same gap acknowledgement scheme as PSFQ, it inherits the same shortcomings when detecting loss of truncated messages. As seen with PSFQ's recovery scheme, at least one data segment must be received downstream for RMST to detect message loss.

3.3.2 Congestion Control Mechanisms. RMST does not specify any congestion control or detection mechanism. It is concerned solely with reliable data transfer between the sensor nodes and the sink. Any congestion control mechanisms are a byproduct of the use of directed diffusion which offers minimal congestion control. For example, sensor nodes having gradients that show interest in the same information, but have different reporting intervals, may "downconvert" to the lower of the two reporting intervals [5].

3.4 ESRT (Event to Sink Reliable Transport)

ESRT introduces the idea of reliable event detection from the sensor nodes to the sink. ESRT leverages the loss tolerant characteristic of WSNs, the goal being to pass a course description of the event rather than providing fine details. Since ESRT will only reliably pass a course description of the event, it is unacceptable for applications that require delivery

TABLE 2 Essential features of ESRT [9]

Feature	Description
Self-configuration	Events must be detected reliably even in adverse network conditions. WSNs may also be randomly deployed. ESRT addresses this by controlling and adjusting the optimal operating interval.
Energy awareness	Sensor nodes have a finite lifetime. ESRT places most of the responsibility for ensuring reliability on the sink, since it is usually more robust. To extend the lifetime of the sensor nodes the sink may decrease the reporting frequency of sensor nodes.
Congestion control	ESRT will decrease the reporting rate of sensor nodes to alleviate congestion on WSNs while still using the event detection threshold to ensure that events are reliably detected.
Collective identification	Since sinks are more often interested in events than individual nodes, ESRT does not require individual node IDs. Instead event IDs are used to correlate data flows with events.
Biased implementation	To conserve energy algorithms used to ensure reliable event detection are mainly run on the sink. Since the sinks nodes are generally more robust nodes in a WSN, this feature conserves energy and preserves the lifetime of the sensor nodes.

of all message segments. Unlike PSFQ and RMST, ESRT would be a good choice for tasks such as sensor retasking or transporting binary objects in general.

ESRT uses a different paradigm to measure reliability in wireless sensor networks. The assumption is not made that only messages in the point-to-multipoint direction, i.e. from the sink to the sensor nodes, is the only type of message that needs to be reliably delivered. Instead a measure of goodness is created using a defined event detection threshold and that threshold is used to define reliability in the multipoint-to-point direction.

The five essential features of ESRT are summarized in Table 2.

3.4.1 Loss Detection/Recovery Mechanisms. ESRT's loss detection and recovery mechanism is tied inextricably to its congestion control mechanism. It does not prevent all losses, nor does it guarantee delivery of all message segments from all source nodes. Instead ESRT tries to find the correct frequency, f, to send messages.

Sankarasubramaniam et al. [9] introduce definitions for observed event reliability, r_i, and desired event reliability, R. Observed event reliability, r_i, is defined as the number of data segments received over some interval i at the sink, and desired event reliability, R, is defined as the number of packets required for reliable event detection, i.e. R is the threshold for reliable event detection. Data segments are given event IDs, and thus r_i can be computed in real time by incrementing a counter at the sink for all correlated segments.

Sankarasubramaniam et al. [9] controlreliable event detection and network congestion by relating r_i and R to f. The problem of reliable event detection then becomes adjusting f to maintain r_i in an optimal interval around R. To help illustrate this Sankarasubramaniam et al. [9] define five operating intervals, which are discussed in section 3.4.2 and summarized in Table 4.

Vuran et al. [20] go on to further explore the idea of maximizing energy efficiency on WSNs by minimizing the transmission of highly correlated data flows. Eliminating the

TABLE 3 ESRT defined operation intervals [9]

Operation Interval	Abbreviation	Characteristics
No congestion, Low reliability	(NC, LR)	$f < f_{max}$ and $\eta < 1 - \epsilon$
No congestion, High reliability	(NC, HR)	$f \leq f_{max}$ and $\eta > 1 + \epsilon$
Congestion, High reliability	(C, HR)	$f > f_{max}$ and $\eta > 1$
Congestion, Low reliability	(C, LR)	$f > f_{max}$ and $\eta \leq 1$
Optimal Operating Region	OOR	$f < f_{max}$ and $1 - \epsilon \leq \eta \leq 1 + \epsilon$

need to send data from all sensor nodes allows for some redundancy for the sensor nodes in WSNs and can prolong the lifetime of the network.

3.4.2 Congestion Control Mechanisms. ESRT recognizes the need for avoiding and controlling congestion in WSNs. To this end, ESRT defines the following five intervals illustrated in Table 3.

ESRT provides a new twist on providing reliability in WSNs. It introduces the idea that reliable data on a sensor network can mean not only delivering an entire binary object reliably, but for tasks where some loss is acceptable we should still provide a measure of reliability that provides the gathering entity with a measure of the "goodness" of the data.

4. Transport Layer Protocol Comparison

As mentioned in Sec. 3, a transport protocol for WSNs should provide energy-efficient reliable message delivery and congestion control. In this section, we evaluate the transport layer protocols described in Sec. 3 in terms of their effectiveness in reliable message delivery, congestion control, and energy efficiency.

4.1 Reliable Message Delivery

We will use the following criteria to compare the transport protocols in providing reliable message delivery:

1. End-to-end versus hop-by-hop recovery: Is recovery done along the node path, or is it handled at the end points. This is a very important question when the error rate between source and destination nodes on a multihop network is high. The destination node should be able to signal the source node that loss has occurred. Hop-by-hop recovery is traditionally done at the MAC layer, and therefore transparent to the transport layer; however the goal of hop-by-hop recovery is to provide a reliable message flow that is cost-efficient and scalable for dense networks [7].
2. Intermediate node caching: This criterion is important when nodes with limited resources are deployed in a very dense network. A lot of traffic may be routed through intermediate nodes, and this criterion may adversely affect our ability to recover from losses.
3. Loss signaling mechanism: What mechanism does the destination use to signal the source of unacceptable loss?

Table 4 summarizes the effectiveness of the transport protocols in addressing the above criterion. It is seen that hop-by-hop recovery has been used in most of the WSN transport protocols. For WSNs, end-to-end recovery is not an efficient recovery mechanism, because we have an exponential increase in loss rate for each hop; this is a significant

TABLE 4 Categorization of protocols by recovery

Transport Protocol	End-to-end recovery	Hop-by-hop recovery
TCP/IP (with DTC)		X
PSFQ		X
RMST	X	X
ESRT	X	

problem in dense sensor networks [7]. For example, with 10% physical layer loss, the success rate of delivering a message across only seven hops is approximately 50%.

RMST is configurable with either end-to-end or hop-by-hop recovery. ESRT specifies that only the sink can detect loss, and is also the only protocol with strict end-to-end recovery between events and the sink. ESRT assumes that the sink can communicate directly with any node and adjust its reporting frequency. This suggests that end-to-end recovery does not incur the high cost of transmission from sink-to-event.

4.2 Congestion Control

Table 5 summarizes the congestion control schemes used by each of the four protocols we have considered in this paper. TCP/IP and ESRT are the only two protocols that implement congestion control. PSFQ makes the explicit assumption that congestion is not likely to be a problem in WSN; loss is seen as far more likely from signal loss due to attenuation [7]. Current research has shown that congestion poses a significant problem in WSNs, and should be considered an important part of transport layer protocols [9, 17].

TCP/IP assumes all packet loss is due to congestion, and the addition of DTC does nothing to change this default behavior. Unfortunately, in an environment of interference and channel contention, such as a WSN, it is not possible to consider all losses being due to congestion.

ESRT uses buffer overflows to signal congestion. Buffer overflows are estimated based on the current size of the data buffer and the observed change in buffer size over past intervals, calculated as $\Delta b = b_k - b_{k-1}$, where, b_k and b_{k-1} are the buffer sizes at the end of the k^{th} and $(k-1)^{\text{th}}$ interval, respectively. If B is the buffer size at some node X, then, $b_k + \Delta b > B$ at the end of some interval k, suggests that X will experience congestion during the $(k+1)^{\text{th}}$ interval, resulting in X setting a special bit (called Congestion Notification (CN)) in the header. Upon receipt of a header with CN flag set to 1, the sink can determine the current network state and adjust the reporting frequency accordingly.

4.3 Energy Efficiency

Sensor nodes should sleep most of the time to conserve and replenish energy through scavenging. To illustrate the difference in energy consumption, a standard mote spends as

TABLE 5 Congestion control mechanisms used by different transport layer protocols

Transport Protocol	Congestion control mechanism
TCP/IP (with DTC)	Dropped packets signal congestion.
PSFQ	None is provided.
RMST	None is provided.
ESRT	Congestion is signaled by computing projected buffer use based of current buffer size and observed buffer increment.

TABLE 6 Comparison of energy efficiency of WSN transport protocols

Transport Protocol	Energy efficiency
TCP/IP	Large header size, even with compression, makes TCP/IP intuitively non-energy efficient.
PSFQ	NACK recovery mechanism and required in order delivery.
RMST	Configurable for NACK local recovery. Can recover multiple segment loss with a single NACK, provided the lost segments are still available locally.
ESRT	Reliability algorithm runs on sinks not sensor nodes. Reliability based on rough detection of event.

little as 16 microamps while sleeping compared to 18 milliamps while awake, and as much as 33 milliamps during data transmission [17].

Table 6 intuitively compares the transport layer protocols in terms of energy efficiency. It should be noted that none of the protocols suggested in [7, 8, 9, 14] explicitly compare energy efficiency, thus this may be an important area of future research. Based on intuition and cost of message transmission discussed above, we can say with a degree of certainty that TCP/IP is the most inefficient protocol among all that are being considered in this paper.

PSFQ and RMST only give a rough idea of the header size, but we may assume that the header size need not be large given that they have been specially created for the sensor environment. In fact PSFQ and RMST header sizes should be of comparable size, since each only requires a sequence number for the event and a fragmentation number for messages within the flow of the event. If we assume, for simplicity, that PSFQ and RMST headers are the same size, then RMST will be the more energy efficient protocol, since it can handle out-of-order delivery of segments and has the ability to signal several missing segments with one NACK.

Based on its low granularity, acceptance of loss, and congestion avoidance, ESRT may be the most energy efficient transport layer protocol among those considered in this paper. Because PSFQ and RMST do not specifically implement congestion control, ESRT has an advantage over PSFQ and RMST in networks with congestion. ESRT should fair very well when compared to TCP/IP for energy efficiency, since TCP/IP was not initially intended for a wireless, much less a micro-wireless, environment.

Finally, Table 7 summarizes the transport layer protocols based on the various comparison criteria considered in this paper.

5. Conclusion

A transport layer is needed in wireless sensor networks to control congestion and ensure reliable delivery of messages from the sensor nodes to the sink. The limited energy, memory, and computational resources of sensor nodes require an energy-efficient transport layer. Traditional transport protocols, such as TCP/IP, do not provide an efficient enough alternative without serious modification; however, modifying TCP/IP may prove useful at sink nodes to optimize communication between regions in the sensor fields and hosts on foreign networks.

More research is needed on congestion control in sensor networks. A measure of data "goodness" to supplement a protocol, such as ESRT, may be beneficial in determining whether a data needs to be retransmitted. If the current aggregated data at a node does not

TABLE 7 Transport layer protocols compared by reliability, energy efficiency, and congestion awareness

Protocol Name	Reliability mechanism	Energy efficient	Supports congestion control
TCP/IP	Hop-by-hop	Not energy efficient.	Yes.
PSFQ	Hop-by-hop	NACK recovery mechanism and required in order delivery.	No.
RMST	Hop-by-hop, or end-to-end	Configurable for NACK local recovery. Can recover multiple segment loss with a single NACK, provided the lost segments are still available locally.	No.
ESRT	End-to-end	Reliability algorithm runs on sinks not sensor nodes. Reliability based on rough detection of event.	Yes.

measure up to the goodness level, the node could hold the data until more neighbors report information to be aggregated, thereby reducing the amount of data repeated on the network.

Sliding granularity protocol is another area of future research. A protocol similar to ESRT, that when notified of an event, dynamically shifts granularity so that messages can be watched more closely. This way protocols such as RMST or PSFQ that provide reliability based off a negative acknowledgement system would not have to account for the overhead of sending NACKs unless some event has been sensed.

About the Authors

Justin "Bo" Jones (bojones@ou.edu) received a B.Sc. degree in computer science in 2004 from the University of Oklahoma. He is currently working toward his M.Sc. degree at the University of Oklahoma, and currently works at RiskMetrics Group. His research interests are in the areas of IP mobility, wireless and mobile networks, wireless sensor networks, and transport protocols.

MOHAMMED ATIQUZZAMAN (atiq@ou.edu) received his M.Sc. and Ph.D. degrees in electrical engineering from the University of Manchester, England. Currently he is a professor of Computer Science at the University of Oklahoma. He is Co-Editor-in-Chief of the Computer Communications Journal, and serves on the editorial boards of IEEE Communications Magazine, Wireless and Optical Networks Journal, Real Time Imaging Journal, International Journal of Sensor Networks, and Telecommunication Systems. He was technical co-chair of HPSR 2003 and the SPIE Quality of Service over Next-Generation Data Networks Conference (2001, 2002, and 2003). He serves on the technical program committee of many national and international conferences, including IEEE INFOCOM and IEEE GLOBECOM. He is the co-chair of the Next Generation Networks Symposium at Globecom'06 and Wireless Communications Symposium at ICC'06. His current research interests are in wireless, satellite, and mobile networks, QoS for next-generation Internet, broadband networks, and multimedia over high speed networks.

He is a coauthor of the book TCP/IP over ATM Networks. He has taught many short courses to industry in the area of computer and telecommunication networking. His research has been supported by state and federal agencies like NSF, NASA, the U.S. Air Force, the

Ohio Board of Regents, and DITARD (Australia). He has over 150 refereed publications in the above areas, most of which can be accessed at http://www.cs.ou.edu/~atiq.

References

1. A. Tanenbaum. "Computer Networks," Prentice Hall, 4th Ed. 2002.
2. I. F.Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. "A Survey on Sensor Networks," *IEEE Communications*., **40**, 8, pp 102–114 Aug. 2002.
3. J. M. Rabaey, M. J. Ammer, J. L. da Silva Jr., D. Patel, S. Roundy. "PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking," *IEEE Computer*, **33**, 7, pp. 42–48 July 2000.
4. D. Estrin, et al. "Next Century Challenges: Scalable Coordination in Sensor Networks," *Mobicom 1999*, pp. 263–270 15–20 Aug. 1999 Seattle, WA.
5. C. Intanagonwiwat, R. Govindan, D. Estrin. "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *MobiCOM '00*, pp. 56–67 August 6–11, 2000, Boston, MA.
6. J. Zhao, R. Govindan. "Understanding Packet Delivery Performance in Dense Wireless Sensor Networks," *1st International Conference on Embedded Networked Sensor Systems 2003 (SenSys' 03)*, pp. 1 – 13 November 5–7, 2003.
7. C. Wan, A. T. Campbell, L. Krishnamurthy. "PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks," *1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 1–11 2002.
8. F. Stann, J. Heidemann. "RMST: Reliable Data Transport in Sensor Networks", *IEEE International Workshop on Sensor Net Protocols and Applications (SNPA)*, pp. 1–11, May 11, 2003.
9. Y. Sankarasubramaniam, O. B. Akan, I. F. Akyildiz. "ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks," *MobiHoc 2003: 4th ACM Symposium on Mobile Ad Hoc Networking & Computing*, pp. 177–188 June 2003.
10. W. R. Heinzelman, J. Kulik, and H. Balakrishnan. "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," *5th ACM/IEEE Mobicom*, pp. 174–185 August 1999.
11. D. Culler, D. Estring, M. Srivastava, "Overview of Sensor Networks," *Computer*, **37**, 8, pp. 41–49 August 2004.
12. A. Gurtov, S. Floyd, "Modeling Wireless Links for Transport Protocols," *Computer Communication Review*, **34**, 2, pp. 85–96 April 2004.
13. J. Hill, M. Horton, R Kling, L. Krishnamurthy. "The Platforms Enabling Wireless Sensor Networks," *Communications of the ACM*, **47**, 6, pp. 41–46 June 2004.
14. A. Dunkels, J. Alonso, T. Voigt. "Making TCP/IP Viable for Wireless Sensor Networks," *First European Workshop on Wireless Sensor Networks (EWSN'04), work-in-progress session*, Berlin, Germany, January 2004.
15. A. Dunkels, T. Voigt, J. Alonso, H. Ritter, and J. Schiller. "Connecting Wireless Sensornets with TCP/IP Networks," *Second International Conference on Wired/Wireless Internet Communications (WWIC2004)*, Frankfurt, Germany, pp. 143–152 February 2004.
16. M. Yarvis, W. S. Conner, L. Krishnamurthy, et al. "Real-World Experiences with an Interactive Ad Hoc Sensor Network," *International Conference on Parallel Processing Workshops*, 2002, pp. 143–151 18–21 Aug. 2002, Vancouver, Canada.
17. C. T. Ee, R. Bajcsy. "Congestion Control and Fairness for Many-to-One Routing in Sensor Networks," *SenSys '04*, pp. 148–161 November 3–5, 2004, Baltimore, Maryland.
18. MICA MOTE/CROSSBOW WEBSITE
19. S. Park, R. Sivakumar. "Sink-to-Sensors Reliability in Sensor Networks," ACM Mobile Computing and Communications Review, **7**, 3, pp. 27–28 July 2003.
20. M. Vuran, O. B. Akan, I. F. Akyildiz. "Spatio-Temporal Correlation: Theory and Applications for Wireless Sensor Networks," *Computer Networks*, **45**, 3, pp. 245–259 June 2004.