Wireless Sensor Network, **2011, 3, 106-113**
doi:10.4236/wsn.2011.33011 Published Online March 2011 (http://www.SciRP.org/journal/wsn)

Energy-Efficient and Reliable Transport Protocols for Wireless Sensor Networks: State-of-Art

Ahmed Ayadi
Institut Telecom/Telecom Bretagne, Cesson-Sévigné, France
E-mail: *ahmed.ayadi@telecom-bretagne.eu*

Received February 14, 2011; *revised February* 28, 2011; *accepted March* 8, 2011

Abstract

New wireless sensor network applications (e.g., military surveillance) require higher reliability than a simple best effort service could provide. Classical reliable transport protocols like Transmission Control Protocol (TCP) are not well suited for wireless sensor networks due to both the characteristics of the network nodes (low computing power, strong energy constraints) and those of the main applications running on those nodes (low data rates). Recent researches present new transport protocols for wireless sensor networks providing various type of reliability and using new mechanisms for loss detection and recovery, and congestion control. This paper presents a survey on reliable transport protocol for WSNs.

Keywords: Wireless Sensor Networks, Transport Protocol, Reliability, Congestion Control, Energy Efficiency

1. Introduction

Wireless sensor networks (WSNs) [1] have important applications such as remote environment monitoring (temperature and humidity) and target tracking (for military purpose).

In monitoring area, a WSN is deployed over a region where some phenomena (e.g., floods, fires) are to be monitored. This has been enabled by the availability of sensors that are smaller, cheaper and intelligent. Sensors are equipped with low-power wireless interfaces, which are using it to communicate with each other to form a network.

Firstly, WSN applications are not interested in reliability because WSNs have been considered as fault- tolerant networks where sensor nodes collect environment information and send it to the base station (sink). Intermediate nodes offer their best effort services to relay generated packets to the base station. However, new WSN applications like military applications (e.g., battlefield surveillance) require more and more reliability. Moreover, in [2], the authors present a need of re-tasking /reprogramming sensor nodes, and thus a need to send a binary file or a script file to sensor nodes.

A transport protocol for WSNs should be reliable (or provides different levels of reliability for each kind of applications) and energy efficient (reduces the amount of exchanged messages to reduce total consumed energy and thereafter increases the network lifetime). The reliability requires two essential mechanisms: congestion control (detection and avoidance), and loss detection and recovery.

The congestion control mechanism is an essential component for a reliable transport protocol because the congestion leads to packet losses. Losses in WSNs are not only due to congestion but also to bad channel errors, collisions and interference. To distinguish between the two types of losses, an explicit congestion notification is proposed. These two mechanisms can be implemented in a distributed form (in sensor nodes) or in a centralized form (in the based station).

In a distributed form, sensor nodes use the sequence numbers of packets to signal packet losses. A gap in the sequence numbers signals a loss. When a loss happens, an intermediate node may request the retransmission of the data from its neighbor nodes. Sensor nodes can detect congestion based on the buffers overflows and then slow down their sending rate.

However, in centralized protocol (e.g., RCRT [3]), the base station is the only node responsible for the packet loss detection and recovery. The base station detects also congestion in the network using the arrival time of the out-of-order packets.

This paper presents a survey of reliable transport protocols for WSNs and identifies some challenge's research. The next section presents the needs of an energy-efficient and reliable transport protocol for wireless sensor networks and presents the recent proposed transport protocols for wireless sensor networks in the literature. Section 3 evaluates and compares different protocols in term of reliability, congestion control and energy efficiency. The last section concludes the paper.

2. Transport Protocol in WSNs

Transport layer ensures the reliability and the quality of data at the sender and the receiver. Transport protocols in WSNs should support multiple applications (industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, health care applications, home automation, and traffic control) and provide variable reliability level, packet loss recovery and congestion control mechanism. We can distinguish between two types of data in WSNs:

1) Data sent by sensor network to the base station (Sink),

2) Data sent by the base station to one or a subset of sensor nodes for different purposes (control, management, re-tasking, reprogramming).

The development of a transport protocol should be generic and independent. It should provide various reliability levels for different applications. WSNs suffer from a high loss rate. Packet loss may be due to bad radio communication, congestion, packet collisions, full memory capacity, and node mobility or fail. Thus, transport protocol should provide two functions: reliable data transport and congestion control. A reliable application requires that all segments sent by a source arrive to the destination. Missing segments that may be lost in the WSN should be recovered by reliable schemes. Congestion happens when the data packets generated by sensor nodes exceed the network capacity. When the networks get congested, intermediate sensor nodes may drop packets. This leads to retransmissions of the dropped packets and thus a waste of energy that is a very important factor in wireless sensor network. In the following, we present a summary of recent transport protocol proposed for WSNs.

2.1. Reliable Multi-Segment Transport

F. Stann *et al.* present in [4] the first transport layer with hop-by-hop recovery scheme using caching mode as additional control traffic for Direct Diffusion [5]. The main goal of RMST is to minimize the cost of end-to-end retransmissions. RMST protocol provides two transmission modes: caching mode (with hop-by-hop recovery) and non-caching mode (with end-to-end recovery). In non-caching mode, only sources and sinks maintain a cache, and only sinks set timers to detects loss.

In caching mode, RMST protocol assumes that each sensor node has a cache memory where recently received segments can be saved. RMST protocol reduces end-to-end retransmissions by introducing hop-by-hop retransmissions from caches of neighbor nodes. In link layer, lost packets are retransmitted using Automatic Repeat reQuest (ARQ) [6].

The RSMT receivers are responsible for detecting losses and for trigger the recovery of the missing segments through the generation of Negative Acknowledgments (NACKs). The RSMT receivers are not only sinks, but also intermediate nodes. To handle losses, an RSMT intermediate node should store data traffics and constructs a map of received segments. When an out-of-order segment is received, an RSMT receiver sends a NACK requesting the retransmissions of the lost messages. Firstly, the one-hop neighbors process NACKs. Then, if one of the neighbors finds the missing segments in cache, it suppresses the NACK message and retransmits the missing segments to the sink. Else, the NACK message is relayed to the next node toward the source.

RSMT protocol provides 100% reliability even for applications that do not require total reliability. Moreover, RMST does not include real time guarantees or congestion control mechanisms. Require that each node has enough memory to store all received segments is a very strong condition difficult to be satisfied with memory-constraint wireless devices.

2.2. Pump Slowly, Fetch Quickly

Pump Slowly, Fetch Quickly [2] mechanism is proposed for re-tasking/re-programming a group of sensors over-the-air. PSFQ is based on slowly injecting packets into the network "pump operation" and performing aggressive hop-by-hop recovery in case of packet losses "fetch operation". Like RSMT, PSFQ provides a hop-by-hop error recovery mechanism in which intermediate nodes take the responsibility of loss detection and recovery. To enable a hop-by-hop loss recovery and in-sequence data delivery, a data cache is created and maintained at intermediate nodes.

The PSFQ "pump operation" consists in a timely controlled data forwarding. In intermediate nodes, when a packet is received in an out-of-order sequence, it is stored. However, instead of forwarding it, the immediate node requests retransmission of the missing segment.

The PSFQ "fetch operation" is a proactive act of requesting a retransmission from neighboring nodes once loss is detected at the receiving node. It corresponds to sending NACK for a retransmission request containing the sequence number of the missing segment. If the upstream neighbors do not posses the missing segment, they forward the NACK further, until it reaches a node having the missing segments.

However, the use of PSFQ for the forward direction can lead to a waste of energy. Besides this, PSFQ does not address packet losses due to congestion. PSFQ give good performance in a chain scenario where a sensor node has only two neighbors. Moreover, in randomly distributed network, PSFQ could flood network in fetch phase by NACK messages.

2.3. Event-to-Sink Reliable Transport protocol

ESRT [7] is a transport protocol that seeks to achieve reliable event direction with minimum energy expenditure and congestion resolution. The main goal is to configure the reporting frequency rate to achieve the desired event detection accuracy with minimum energy expenditure. In fact, a sensor node in ESRT sends messages with an announced reporting frequency to the base station. An ESRT base station regulates the reporting rate of sensors in response to a congestion detected in the network. Congestion control mechanism is implemented in the base station, which informs all sensor nodes using a different technology about the new reporting frequency.

An ESRT node monitors its local buffer level and sets a congestion notification bit in the packets it forwards to base station if the buffer overflows. If the base station receives a packet with the congestion notification bit set, it broadcasts a control signal informing all source nodes to slow down their common reporting frequency. The ESRT base station must broadcast this control signal at high energy so that all sources can hear it or use another technology.

Such a signal has several potential drawbacks, however, particularly in large sensor networks. Any on-going event transmission would be disturbed by such high-powered congestion signal to sources. In addition, ESRT always regulates all sources regardless the congestion region. ESRT does not retransmit lost packets.

2.4. Distributed TCP Caching

In [8], Dunkels *et al*. present Distributed TCP Caching, a new scheme for TCP [9] in multi-hop wireless networks that uses segment caching and local retransmission in cooperation with link layer for TCP/IP-based wireless sensor network. DTC is an extension of Snoop [10] idea towards multi-hop sensor networks.

The authors assume that each intermediate node is able to cache a single TCP data segment. DTC relies mainly on timeouts to detect packet losses. Thus, each node measures the round-trip time (RTT) to the receiver and adapts a retransmission timeout RTT to 1.5 x RTT.

The authors propose to compute the RTT in the TCP connection setup phase and to use RTT = 1.5 RTT as a timeout value. The sensor nodes cache the TCP segment that has the highest segment number seen with a certain probability (p = 50%). An unacknowledged packet in link layer should be locked and retransmitted after the timeout. Locked data segments should not be overwritten by a TCP segment with higher sequence number. A locked segment is removed from the cache only when a TCP ACK that acknowledges the cashed segment is received, or when the segment times out.

DTC uses also TCP SACK option [11] to both packet loss detecting and signaling mechanism between DTC nodes. The TCP SACK option is used by sensor node to inform other nodes about segments locked in their caches.

2.5. TCP Support for Sensor Networks

In [12], Braun *et al*. present TCP Support for Sensor networks (TSS), which is a new layer between TCP and network layer. TSS requires storing state information for each TCP connection that contains sequence numbers, acknowledgments numbers, and RTT.

TSS uses Implicit ACK (IACK) for loss detection: a sensor node is assumed to listen to packet transmission of their neighbor to detect whether the next node have forwarded TCP segment. A node using TSS always caches a packet until it is sure that the successor node towards the destination has received the segment. The retransmissions are mainly triggered by timeouts, which requires careful setting of timeout values. Like DTC [8], the retransmission timeout is set to 1.5 RTT. To avoid congestion, a TSS node should stop forwarding its packets until it knows that all earlier packets have been received and forwarded by its successor node.

The simulation results show that TSS gives more throughput than TCP and less exchanged messages than DTC [8]. However, hearing all neighbor node traffics is not energy-efficient because listening power is important as well as transmission power. In addition, a message transmission fail of one sensor node leads to stop the transmission of all its previous sensor nodes.

2.6. Asymmetric Reliable Transport

Asymmetric Reliable Transport [13] is an asymmetric and reliable transport mechanism that does not address the reliability of event notifications (sensor-to-sink) but also the queries (sink-to-sensors), being thus a bidirectional transport protocol. ARP classifies the sensor as essential (E) nodes and non-essential (N) nodes. The E nodes are selected periodically to cover the entire sensible terrain, based on residual energy.

Using both asymmetric acknowledgment and negative acknowledgment provides the end-to-end reliable communications. To distinguish the last query message of sequence, a Poll (P)/Final (F) bit is used.

2.7. Distributed Transport for Sensor Networks

Distributed Transport for Sensor Networks [14] is an energy-efficient hop-by-hop reliable transport protocol using both ACK and NACK message for delivery confirmation. A DTSN node analyzes the sequence numbers of received packets and detects losses by finding gaps.

Every source node sends an Explicit Acknowledgment Request (EAR) every one Acknowledgment Window (AK) to ask for an ACK or a NACK. The sink node replays by an ACK message if no gap is detected or by a NACK message containing the sequence numbers of missing segments.

DTSN protocol is a hop-by-hop recovery protocol; all intermediate node caches received packet in their cache. Upon reception of an ACK message, intermediate node deletes acknowledged segment. Otherwise (*i.e.* reception of NACK message) an intermediate node checks if its cache contains one of the missing segment.

DTSN node retransmits missing segments and updates the NACK message. DTSN offers two types of service: total reliability service and differentiated reliability service. The difference between the two types of service is the probability of caching a segment in an intermediate node. For example, in full reliability scenario, all segments are cached in intermediate nodes. DTSN algorithm does not threat congestion detection and control.

2.8. Wisden

Wisden [15] provides a reliable data transport from the sensor nodes to the base station. In Wisden, nodes self-organize themselves into a routing tree rooted at the base station. Wisden implements both end-to-end and hop-by-hop NACK based reliability scheme. Nodes keep a small cache of recently transmitted packets. Intermediate node detects packet loss based on gap in the sequence numbers in a received segment. Entries in the “missing packets” list are piggybacked in the outgoing transmissions, and children infer losses by overhearing this transmission. Lost packets are often recovered hop by hop, however, two factors necessity end-to-end recovery: the large list of missing packet that can exceed the memory of the sensor node and the topology changes. Limit of Wisden is the constant value of sending rate, which should be measured and configured based on the bandwidth and the number of nodes.

2.9. Rate-Controlled Reliable Transport protocol

Rate-Controlled Reliable Transport protocol [3] is a multipoint-to-point reliable transport protocol for wireless sensor networks. RCRT uses an explicit end-to-end loss recovery and places all congestion detection, recovery and rate adaptation schemes in the base station (Sink). RCRT sink has three distinct logical components:

a) End-to-end retransmission

The main goal of RCRT is to achieve 100% reliability. The RCRT sink uses NACK-based end-to-end loss recovery to request the retransmissions of missing packets from the source. Each source (sensor node) has a retransmission buffer where is saved the not acknowledged segment. The sink node keeps a list of lost segments then sends a NACK feedback message to the source containing the sequence numbers of missing segments. Upon receiving a NACK, the source node retransmits the requested segments.

b) Congestion detectio*n*

To distinguish between congestion and transmission losses, RCRT congestion detection mechanism is based on the length of the losses. The sink node maintains a list of the out-of-order messages and computes the Time to recover loss. If this value exceeds $2 \times RTT$, congestion is then signaled.

c) Rate adaptation

RCRT uses AIMD to adapt the transmission rate of each source. Whenever the RCRT sink determines the network is congested, it applies the rate decrease and computes the new rate for all flows:

- Increase: $R(t+1) = R(t) + A$,
- Decrease: $R(t+1) = M(t) - R(t)$,

where A is a constant and $M(t)$ is a function of loss rate, $M(t) = \frac{p_i(t)}{2 - p_i(t)}$ and $p_i(t)$ is the loss rate value of the source i at the instant t.

2.10. Interference-Aware Fair Rate Control

Interference-aware Fair Rate Control [16] is a distributed rate allocation scheme that uses queue size to detect congestion, shares congestion state through overhearing messages and converges to fair and efficient rates for each node. IFRC scheme consists of three components:

a) *Measure of level congestion*

An IFRC node uses an exponentially weighted moving average of instantaneous queue length as a measure of congestion $avg_q = \left(1 - w_q\right) \times avg_q + w_q \times inst_q$.

The average packet length is updated whenever a packet is inserted into the queue. IFRC detects incipient congestion by using multiple thresholds $U\left(k\right)$ where $U\left(k\right) = U\left(k-1\right) + I/2^{k-1}$ where k is a small integer and I is a constant increment of queue length.

b) *Congestion Sharing*

In IFRC node i includes the following information in the header of each outgoing transmission packet: its rate r_i, current average queue length, a bit indicating whether any child of i is congested, small state r_i among all its congested children, and l's average queue length. IFRC introduces two simple constraints:

- Rule 1: r_i can never exceed r_j, the rate of i's parent j.
- Rule 2: Whenever a congested neighbor j of i crosses a threshold $U\left(k\right)$, i sets its rate to the lower of r_i and r_l.

The same rule is applied for the most congested child l of any neighbor of i, i sets its rate to the lower of r_i and r_j where l is the most congested child of i's neighbor.

c) *Rate Adaptation*

All nodes start from a fixed rate r_{init}. IFRC implements multiplicate rate increase initially. After slow start phase, an IRFC node increases it rate r_i every $1/r_i$. If node i is congested, then when threshold $U\left(k\right)$ is crosses, the node halves it current rate. The base station, even if it does not source messages, it maintains its "rate" r_h and adapts using the same mechanism described bellow. Because the base station does not send messages, it broadcasts a control message after the reception of $m = 5$ messages to share its rate r_h. IFRC presents a shared congestion control mechanism but not reliable. Moreover, IFRC adds a lot of overhead in the header of transport protocol stack and a significant amount of control messages.

2.11. Energy-efficient and Reliable Transport Protocol

ERTP [17] is an energy-efficient and reliable transport protocol for low data streaming in WSNs. ERTP is a hop-by-hop recovery algorithm using Implicit Acknowledgment. ERTP requires that each node i after sending a packet to the next node to the sink overheads the next forwarding. The forward of a packet by node $i+1$ is considered as an implicit acknowledgment to node i.

The authors present a hop-by-hop reliability control, which adjusts the maximum number of retransmission of a packet in each node based on the link loss rate. They present also an algorithm for computing the extents time in witch node i is expected to "overhead" the forwarding packet of node $i+1$.

In the result section, authors show that the use of ERTP algorithm for computing the RTO time is better that Jacobson algorithm. They show also that using ERTP gives more delivery ratio than using simple explicit acknowledgment. However, hearing all neighbor node traffics is not energy-efficient because listening consumes energy as well as sending.

2.12. PORT

In [18], authors provide in-network dynamic rate-control and congestion-avoidance transport scheme. PORT minimizes energy consumed by avoiding high communication cost. PORT minimizes energy consumption with two schemes.

The first is based on the application-based optimization approach of sink that feedbacks the optimal reporting rates for source nodes. These source report feedbacks allow the sink to adjust the reporting rate of each data source. PORT adds a price form each node. Node price is the total number of transmission attempts made before a successful packet is delivered from the source and the sink. It is a metric used to evaluate the energy cost communication. The sink adjusts the reporting rate of each source based on the source's node price and the information provided about the physical phenomenon.

The second scheme is based on feedbacks from source node to the sink to inform it about congestion and increase the nodes costs. The sink uses the communication cost information to slow down the reporting rate of the appropriate source and increase the reporting rate of other sources that have lower communication cost since reliability must be maintained.

2.13. Flush

Flush [19] is a reliable single-flow bulk transport protocol for large diameter WSNs. However, Flush only supports one data flow. Flush uses an end-to-end reliable transport protocol to robust to node failures. Flush requires that the sink node sends the sequence numbers of packets it did not receive back to the data source.

When a source node receives a NACK packet, it retransmits the missing data. Flush proposes also a rate allocation scheme for adapting dynamically the sending rate of the sensor nodes. This scheme take into count the broadcast nature of the medium and the interference between nodes. The rate allocation algorithm follows two basic rules:

1) Rule 1: A node should only transmit when its successor is free from interference.

2) Rule 2: A node's sending rate cannot exceed the sending rate of its successor.

These two rules reduce contention and thus collision in the wireless network and minimize losses due to the queue overflows for all nodes.

3. Transport Protocol Comparison

In this section, we present a comparison between the presented protocols in Section 3. The listed protocol can be classified in categories using some criteria. We can distinguish between IP-based solution like DTC, TSS and non IP-based solution like RCRT, ERTP, etc. We can differentiate between transport protocols by the manner it recovers losses (end to end recovery or hop by hop recovery), the use or not have caching in intermediate node, kind of message used to loss detection and recovery (ACK, NACK, IACK) and the level of reliability.

3.1. IP and Not IP-Based Solutions

Table 1 presents a classification of all protocol based on its network and transport layer. DTC and TSS are new protocols based on IP networks. These two protocols add caching and hop-by-hop recovery of TCP lost segments. The use of IP-based transport protocol enables to connect between wireless sensor network and an external network (e.g., Internet). Using IP solution in WSN allows not providing a new proxy at the network border.

3.2. Reliability and Loss Recovery

Reliability is very important for some applications like health and military applications. Other applications are loss-tolerant and require a ratio/level of reliability. DTC, TSS, PSFQ, RMST, RCRT provide 100% reliability for wireless sensor network applications. However, these protocols do not use the same mechanism for loss detection and recovery. For example, ESRT, ERTP, DTSN and SCTP provide classes of probability for the applications. However, the need for reliability is not the same for all applications but depends on the importance of the application and even on the importance on some packets than others.

Table 1. IP and not IP-based transport protocol.

	TCP/IP	Non-TCP/IP
Transport Protocol	DTC, TSS	RSMT, PSFQ, ESRT, DTSN, RCRT, ERTP, IFRC

Loss detection and recovery methods differ from a protocol to another. In SCTP and RCRT, the receiver use gaps in the sequence numbers of received segments as a signal of packet losses. The receiver asks the retransmission of missing segments from the source node. This mechanism is called an end-to-end recovery. However, in multi-hop wireless sensor networks, end-to-end recovery is not energy-efficient and thus new transport protocols enable intermediate nodes to cache segments.

Loss detection and recovery mechanisms from intermediate nodes allow reducing the total exchanged messages from the source to the receiver. We distinguish between two kinds of intermediate nodes. The first does not detect losses but reacts when it receives a NACK message. Then, it retransmits the missing segment. The second kind detects losses and requests a retransmission from its neighbors.

Acknowledging a packet can be done explicitly by sending an ACK message, thus if the receiver does not receive an ACK message, it detects a loss. The explicit acknowledging approach requires adding a lot of control messages (ACK) to data messages and increases the contention. Another solution is to relay by a NACK message when a packet loss found.

The third mechanism proposed by TSS and ERTP is to use implicit acknowledgment (IACK). This mechanism requires that each node i after sending a packet the next node to the sink overheads the next forwarding. The forward of a packet by node i is considered as an implicit acknowledgment to node i .

3.3. Congestion Control

Congestion has a significant impact on the performance of a reliable transport protocol. Some transport protocols for wireless sensor network (like PSFQ, DTSN, ERTP) make an assumption that congestion is not likely to be problem in WSNs. Others assume all packet loss due to congestion (like DTC, TSS and ESRT).

Some of the proposed solution are distributed, thus all sensor nodes detect congestion and then share the information using a flag named congestion notification.

Thus, the based station, after receiving a message, which the congestion notification bit is set, slow down the transmission rate.

On the other hand, centralized solution implements congestion detection component on the base station. Because DTC and TSS are extension for TCP of wireless sensor network, they inherit congestion detection and avoidance mechanism from TCP. RCRT congestion detection scheme is based on time to recover loss. They assume that the network is not congested as long as end-to-end losses are recovered "quickly enough". Thus, RCRT permit the sender to transmit at a light rate even if there are occasional end-to-end losses, since those rate can be recovered “quickly”.

All other protocols are based on buffer overflows to signal congestion. SCTP specifies two thresholds: t_{lower} and t_{higher}. When the buffer reaches t_{lower}, the congestion bit is set with certain probability. When the buffer reaches t_{higher}, the node sets the congestion notification bit in every packet it forwards. ESRT node calculates $\Delta b = b_k - b_{k-1}$, where b_k and b_{k-1} are the buffer size at the end of the k^{th} and the $(k-1)^{th}$ internal. If the $b_k + \Delta b \succ B$ (where B is the buffer size) then the node signals a congestion by setting the algorithm. IFRC uses an exponentially weighted moving average of instantaneous queue length as a measure of congestion $avg_q = (1 - w_q) \times avg_q + w_q \times inst_q$. Unlike SCTP, IFRC uses multiple thresholds $U(k)$

3.4. Energy Efficiency

Transport protocols should provide reliability with the minimum number of exchanged messages. This constraint comes from the low capacity of energy of sensor node batteries. ESRT proposes to reduce the sending events frequency to reduce the total consumed energy. Others propose to use hop-by-hop recovery instead of end-to-end recovery to reduce the retransmissions. For example, RMST, PSFQ, Wisden, DTC and TSS reduce the amount of exchanged messages by caching not already acknowledged segments in intermediate nodes and process a recovery once a lost is detected.

DSTN proposes to reduce the consumed energy by using Selective Acknowledgment (ACK and NACK) after an Acknowledgment Windows of messages. DSTN reduces the number of control messages, which make it more energy-efficient than other protocols. TSS and ERTP propose not to use explicit acknowledgment instead of implicit acknowledgment.

These approaches need a cross-layer mechanism between the link layer and transport layer (e.g. DTC and TSS). These mechanisms permit to reduce the transport acknowledgments. However, some of these schemes are difficult to be implemented in memory-constraint wireless devices.

Finally, ART increases the lifetime of the network by choosing node with more energy capacity to relay packets from sources to the sink nodes.

All these works have tried to reduce the amount of control messages in the WSNs and thus increase the lifetime of all the networks. **Table 2** summarizes the details of all the protocols mentioned above.

4. Conclusion

In this paper, we have reviewed recent researches on reliability and congestion control in wireless sensor networks. We presented different methods of packet loss detection and recovery, congestion detection and avoidance and energy-efficiency. We compared protocol in term of reliability, congestion control and energy efficiency.

Table 2. Reliable transport protocols.

Transport Protocol	Direction	Type of flows	Congestion Control	Congestion Detection	End-to-End of Hop-by-Hop	Caching	Acknowledging	Energy Aware
RSMT [4]	Sensor to Sink	Continuous	-	-	Hop-by-Hop End-to-End	Yes	NACK/ACK	No
PSFQ [2]	Sensor to Sink Sink to Sensor	Event-Driven	-	-	Hop-by-Hop	Yes	NACK	No
ESRT [7]	Sensor to Sink	Continuous	Yes	Buffer size	End-to-End	No	-	No
DTC [8]	Sink to Sensor	-	Yes	Yes	Hop-by-Hop	Yes	ACK/SACK	Yes
TSS [12]	Sink to Sensor	-	Yes	Yes	Hop-by-Hop	Yes	IACK/ SACK	Yes
ART [13]	Sensor to Sink Sink to Sensor	Continuous Event-Driven	-	-	End-to-End	No	NACK/ACK	No
PORT [18]	Sink to Sensor	Event-Driven	Yes	Packet loss	-	-	-	Yes
Flush [19]	Sensor to Sink	Continuous	-	No	End-to-End	No	NACK	No
DTSN [14]	Sensor to Sink		-	-	Hop-by-Hop	Yes	ACK/NACK	Yes
Wisden [15]	Sensor to Sink	Continuous	-	-	Hop-by-Hop End-to-End	Yes	ACK/NACK	Yes
IFRC [16]	Sensor to Sink	Continuous	Yes	Buffer size	-	No	-	No
RCRT [3]	Sensor to Sink	Continuous	Yes	Time to recover losses	End-to-End	No	NACK	No
ERTP [17]	Sensor to Sink	Continuous	-	-	Hop-by-Hop	-	IACK	Yes

5. Acknowledgements

The work of Ahmed Ayadi has been funded by the Pôle de Recherche Avancée en Communications (PraCOM).

6. References

[1] J. Yick, B. Mukherjee and D. Ghosal, "Wireless Sensor Network: Survey," *Computer Networks*, Vol. 52, No. 12, 2008, pp. 2292-2330. doi:10.1016/j.comnet.2008.04.002

[2] C. Wan, A. Campbell, and L. Krishnamurthy, "Pump-Slowly, Fetch Quickly (PSFQ): A Reliable Transport Protocol for Sensor Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 23, No. 4, April 2005, pp. 862-872. doi:10.1109/JSAC.2005.843554

[3] J. Paek and R. Govindan, "RCRT: rate-Controlled Reliable Transport for Wireless Sensor Networks," *Proceedings of the* 5*th International Conference on Embedded Networked Sensor Systems*, New York, 2007, pp. 305-319. doi:10.1145/1322263.1322293

[4] F. Stann and J. Heidemann, "RMST: Reliable Data Transport in Sensor Networks," *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003, pp. 102-112. doi:10.1109/SNPA.2003.1203361

[5] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann and F. Silva, "Directed Diffusion for Wireless Sensor Networking," *IEEE/ACM Transactions on Networking*, Vol. 11, No. 1, 2003, pp. 2-16. doi:10.1109/TNET.2002.808417

[6] G. Fairhurst and L. Wood, "Advice to Link Designers on Link Automatic Repeat ReQuest (ARQ)," IETF, RFC 3366, August 2002.

[7] Y. Sankarasubramaniam, O. B. Akan and I. F. Akyildiz, "ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks," *Proceedings of the* 4*th ACM International Symposium on Mobile Ad Hoc Networking & Computing*, New York, 2003, pp. 177-188. doi:10.1145/778415.778437

[8] A. Dunkels, J. Alonso and T. Voigt, "Distributed TCP Caching for Wireless Sensor Networks," *Proceedings of the* 3*rd Annual Mediterranean Ad-Hoc Networks Workshop*, 2004.

[9] J. Postel, "Transmission Control Protocol," IETF RFC 793, September 1981.

[10] H. Balakrishnan, S. Seshan, E. Amir and R. H. Katz, "Improving TCP/IP Performance over Wireless Networks," *Proceedings of the* 1*st Annual International Conference on Mobile Computing and Networking*, New York, 1995, pp. 2-11. doi:10.1145/215530.215544

[11] S. Floyd, J. Mahdavi, M. Mathis and M. Podolsky, "An Extension to the Selective Acknowledgement (SACK) Option for TCP," RFC 2883, 2000.

[12] T. Braun, T. Voigt and A. Dunkels, "TCP Support for Sensor Networks," *Proceedings of Fourth Annual Conference on Wireless on Demand Network Systems and Services*, January 2007, pp. 162-169. doi:10.1109/WONS.2007.340494

[13] N. Tezcan and W. Wang, "ART: An Asymmetric and Reliable Transport Mechanism for Wireless Sensor Networks," *International Journal of Sensors Networks*, Vol. 2, No. 3-4, 2007, pp. 188-200. doi:10.1504/IJSNET.2007.013199

[14] B. Marchi, A. Grilo and M. Nunes, "DTSN: Distributed Transport for Sensor Networks," *Proceedings of* 12*th IEEE Symposium on Computers and Communications*, July 2007, pp. 165-172. doi:10.1109/ISCC.2007.4381601

[15] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan and D. Estrin, "A Wireless Sensor Network for Structural Monitoring," *Proceedings of the* 2*nd International Conference on Embedded Networked Sensor Systems*, New York, 2004, pp. 13-24. doi:10.1145/1031495.1031498

[16] S. Rangwala, R. Gummadi, R. Govindan and K. Psounis, "Interference Aware Fair Rate Control in Wireless Sensor Networks," *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, New York, 2006, pp. 63-74.

[17] T. Le, W. Hu, P. Corke and S. Jha, "ERTP: Energy-Efficient and Reliable Transport Protocol for Data Streaming in Wireless Sensor Networks," *Computer Communications*, Vol. 32, No. 7-10, 2009, pp. 1154-1171. doi:10.1016/j.comcom.2008.12.045

[18] Y. Zhou, M. Lyu, J. Liu and H. Wang, "PORT: A Price-Oriented Reliable Transport Protocol for Wireless Sensor Networks," *Proceedings of* 16*th IEEE International Symposium on Software Reliability Engineering*, November 2005, pp. 10-126. doi:10.1109/ISSRE.2005.32

[19] S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, D. Culler, P. Levis, S. Shenker and I. Stoica, "Flush: A Reliable Bulk Transport Protocol for Multihop Wireless Networks," *Proceedings of the* 5*th International Conference on Embedded Networked Sensor Systems*, New York, 2007, pp. 351-365. doi:10.1145/1322263.1322296