Issues of Transport Control Protocols for Wireless Sensor Networks

Chonggang Wang[1], Kazem Sohraby[1], Bo Li[2], and Weiwen Tang[3]

[1]University of Arkansas, Fayetteville, AR, USA, 72701

[2]The Hong Kong University of Science and Technology, Hong Kong, P.R.China

[3]Sichuan Communication Research Planning & Designing Co., Ltd., Chengdu, China

Abstract—**This paper gives out a survey on transport control protocol for wireless sensor networks (WSNs). First, it lists the disadvantages of traditional transport control protocols (TCP and UDP) for the environment of WSNs. Second, several design issues of transport control protocols for WSNs are presented. Third, some existing transport control protocols for WSNs are classified and compared. Finally, several problems needing further studying are outlined.**

I. INTRODUCTION

Wireless sensor networks have been experiencing more and more attentions in academia and industry in recent years, especially under the possibility of much more cheap sensors with certain computation and communication capability. WSNs can be used for many applications such as habitat monitoring, in-door monitoring, target tracking, and security surveillance, etc. However there is a path before commercially deploying sensors, because WSNs have some problems to be overcome, for example, energy-conservation, congestion control, reliability data dissemination, security, and management of a WSN itself. These problems often involve in one or several layers top-down from application layer to physical layer, and can be studies separately in each corresponding layer, or collaboratively cross each layer. For example, congestion control may involve in only transport layer, but energy-conservation may be related to physical layer, data link layer, network layer, and high layers. Some researchers recently turn their attentions to transport control protocols, which are important for reliable data dissemination and energy-conservation for WSNs.

Generally speaking, transport control protocols, especially for connection-oriented transport protocols, may include two main functions: *congestion control* and *loss recovery*. As for congestion control, it is firstly required how to detect whether or not congestion happens, and when and where it happens. Congestion can be detected through monitoring node buffer occupancy and link (or wireless channel) load. In traditional Internet, the methods to weaken congestion are packet dropping at congestion point such as AQM (active queue management), rate decreasing in source node such as AIMD (Additive Increase Multiplicative Decrease) in TCP (Transport Control Protocol) [2], and routing techniques. For WSNs, it should be carefully considered how to detect congestion and how to overcome it, because sensors are often with limited resources. These protocols must consider its simplicity and scalability to save energy and possibility to prolong the life-time of whole networks. For example, in order to weaken congestion, we can use end-to-end mechanism like TCP or hop-by-hop backpressure like that in ATM (Asynchronous Transfer Mode) networks or Frame Relay networks. The end-to-end approaches are very simple and robust, but it will bring with more on-going packets in networks. However, hop-by-hop approaches can quickly weaken congestion and bring with less on-going packets in networks, while it needs to change the behavior of each node on the way from source to destination. Since less on-going packets can result in saved energy, there is a trade-off between end-to-end and hop-by-hop mechanism, which should be carefully considered when designing practical congestion control algorithms for WSNs.

Packet loss is usual under wireless sensor networks due to bad quality of wireless channel, sensor failure, and/or congestion. WSNs must guarantee certain reliability in packet-level or application-level through loss recovery in order to abstract correct information. Some critical applications need reliable transmission of each packet and thus packet-level reliability is needed. Other applications need only a proportionally reliable transmission of total packets and thus application reliability is needed. Anyway, we first need to detect packet loss in order to correctly recover missing packets. The traditional methods used in packet-switched networks can be used to detect packet loss for wireless sensor networks. For example, each packet can piggyback sequence number, and a receiver can detect loss though arbitrating the continuity of received sequence number during a time interval (if not considering of packet disorder resulted from multi-path). After detecting packet loss, ACK and/or NACK (and their variant) can be used to recover missing packets based on an end-to-end or hop-by-hop approach. Like that in congestion control, there is still a trade-off between end-to-end and hop-by-hop approach, which should be thought over. When designing transport control protocols for wireless sensor networks, we must consider energy-conservation at the same time.

Intuitively, if there are few on-going packets and few re-transmissions, energy can be saved. Effective congestion control can result in few on-going packets and effective loss recovery approach can result in few re-transmissions. So congestion control and reliability guarantee can additionally save energy in a wireless sensor network. In a summary, the problem of transport control protocols for sensor networks is how to effectively control congestion and how to guarantee reliability while conserving energy as more as possible simultaneously.

This paper aims to give out a survey of transport control protocols for WSNs. The remainder of the paper is organized as follows. Section II lists the disadvantages of traditional transport protocols if used in WSNs. Section III presents several issues to design transport protocol for WSNs. Section IV classifies and compares the existing transport protocols. Section V gives out several problems to be further studied in the future. Finally section VI concludes the whole paper.

II. DISADVANTAGES OF TCP AND UDP

TCP [2] and UDP (User Datagram Protocol) [3] are two well-known transport control protocols widely deployed in Internet. But both of them are not the good choice for WSNs. First let us see the characteristics of TCP protocols as follows:

- TCP is a connection-oriented protocol. Before data transmission, there is a three-way handshake interactive process. If and only if after the TCP connection has been established, TCP sender can begin to transmit data. In WSNs, the sensed data for event-based applications is just several bytes or so (a value of an interest). The three-way handshake process will be a big overhead for the small volume data. Also since wireless link is error-prone under WSNs, the time to setup TCP connection might be much longer than that under Internet. Then the data will be probably outdated after TCP connection has been established.
- In TCP, it is assumed that all segment losses are resulted from congestion and will trigger window-based flow control and congestion control. This style will incur that TCP will unwisely reduce transmission rate under WSNs when there is no congestion, but packet losses from bit-error. The behavior will lead to low throughput especially under multiple wireless hops. Therefore it is hard for sensor nodes, especially the ones far away from sink, to obtain enough throughout to support such WSNs applications that require continual data transmissions.
- TCP uses end-to-end approach to control congestion. This approach generally has longer response time when congestion occurs, and in-turn will result in lots of segment dropping. The segment dropping means useless energy consumption and not energy-efficient. Also the long response time will make it hard to fully fill wireless channel after congestion.
- TCP uses end-to-end ACK and retransmission to guarantee reliability. This approach will cause much lower throughput and longer transmission time if RTT (Round-Trip Time) is larger as that in large-scale WSNs, since the sender will stop to wait for the ACK after each data transmission.
- Under WSNs, sensor nodes may have different hops and different RTT from sink. TCP in such environment may cause unfairness. The sensor nodes near to sink may get more opportunities to transmit data and may deplete their energy first, and the whole wireless sensor network will be disjointed with a high probability.

Although UDP is a connectionless transport control protocol, it is still not suitable for WSNs considering the following reasons:

- There is no any flow control and congestion control mechanism in UDP. If UDP is used for WSNs, it will cause lots of datagram dropping when congestion happens. In this point at least, UDP is not energy-efficient for WSNs.
- UDP contains no ACK mechanism, no any reliability mechanism. The datagram loss can be only recovered by lower MAC algorithms or upper layers including application layer.

Beside the disadvantages listed above, there is no any interaction between TCP (or UDP) and lower layer protocols such as routing and MAC (Medial Access Control) algorithm. But under wireless sensor networks, the lower layers can provide rich and helpful information to transport control layer and make it possible to optimize system performance. In a summary, neither TCP nor UDP are well suitable for wireless sensor networks.

III. THE DESIGN ISSUES OF TRANSPORT CONTROL PROTOCOLS FOR WSNS

Generally speaking, the transport control protocol for WSNs should consider the following factors. First, it should provide congestion control mechanism and guarantee reliability, especially the latter. The most data streams are flowed from sensor nodes to sink in WSNs, so congestion might occur around sink. Also there are some high-bandwidth data streams produced by multi-media sensors. Therefore it is necessary to design effective congestion detection, congestion avoidance, and congestion control mechanisms for WSNs. Although MAC protocol can recover packets loss from bit-error, it has no way to handle packets loss from buffer overflow. Then the transport protocol for WSNs should have mechanism for packets loss recovery such as ACK and Selective ACK [4] used in TCP protocol so as to guarantee reliability. At the same time, the reliability under WSNs may have different meaning from traditional networks where it generally guarantees the correct transmission of every packet. For some application, WSNs only needs to correctly receive packets from a certain area not every sensor nodes in this area, or some ratio of successful transmission from a sensor node. These new reliability can be utilized to design more efficient transport control protocols. It would be better to use hop-by-hop mechanism for congestion control and loss recovery since it can reduce packet dropping and conserve energy. The hop-by-hop mechanism can lower the buffer requirement in intermediate nodes simultaneously. It is helpful for sensor nodes with limited memory.

Second, transport control protocols for wireless sensor networks should simplify initial connecting process or use connectionless protocol so as to speedup start and guarantee throughput and lower transmission delay. Most of applications in WSNs are reactive which passively monitor and wait for event occurring before reporting to sink. These applications may have only several packets for each reporting, and the simple and short initial setup process is more effective and efficient.

Third, the transport control protocols for WSNs should avoid as few packets dropping as possible since packet dropping means energy wastage. In order to avoid packet dropping, the transport protocol can use active congestion control at the cost of a bit lower link utility. The active congestion control (ACC) can trigger congestion avoidance before congestion occurs. An example of ACC is to make sender (or intermediate nodes) reduce sending (or forwarding) rate when the buffer size of their downstream neighbors overruns a threshold.

Fourth, the transport control protocols should guarantee fairness for different sensor nodes in order that each sensor nodes can achieve fair throughput. Otherwise the biased sensor nodes can not report the events in their area and system may misunderstand there is no any event in the area. Fifth, it would be better if the transport control protocol can enable cross-layer optimization. For example, if routing algorithm can tell route failure to transport protocol, the transport protocol will know the packet loss is not from congestion but from route failure and the sender will frozen its status and keep its current sending rate to guarantee high throughput and low delay.

IV. THE EXISTING TRANSPORT CONTROL PROTOCOLS FOR WSNS

There are several transport control protocols [5-11] (see Table 1) for wireless sensor networks. They aim at congestion control and/or reliability guarantee in upstream (from sensor nodes to sink) or downstream (from sink to sensor nodes), and can be classified into four types: *upstream congestion control*, *downstream congestion control*, *upstream reliability guarantee*, and *downstream reliability guarantee*.

CODA [5]

CODA (COngestion Detection and Avoidance) belongs to *upstream congestion control*. It contains three components: *congestion detection*, *open-loop hop-by-hop backpressure*, and *closed-loop end-to-end multi-source regulation*. CODA attempts to detect congestion by monitoring current buffer occupancy and wireless channel load. If buffer occupancy or wireless channel load exceeds a threshold-based value, it means that congestion happens. Then node detecting congestion will notify its upstream neighbor nodes to decrease rate, with the manner of open-loop hop-by-hop backpressure. The upstream neighbor nodes will trigger to decrease output rate like AIMD and to replay backpressure continuously, after they receive backpressure signal. Finally CODA can regulate multi-source rate through closed-loop end-to-end approach, which works as follows: 1) When a sensor rate overruns theoretical throughput, it will set “regulation” bit in even packet. 2) If the event packet received by sink has “regulation” bit, sink should send ACK control message to sensors and to inform them to decrease their rate. 3) If congestion is cleared, sink will actively send ACK control message to sensors and to inform them to increase their rate. CODA uses AIMD-like mode in TCP protocols to regulate sensors rate. CODA brings with such disadvantages: 1) Unidirectional control from sensors to sink; 2) Consider no reliability but congestion control; 3) Result in decreased reliability (although conserving energy) especially under such scenarios with sparse source and high data rate; 4) The delay or response time of closed-loop multi-source regulation will be increased under heavy congestion since the ACK issued from sink would loss with high probability at this time.

ESRT [6]

ESRT (Event-to-Sink Reliable Transport) aims at providing reliability from sensors to sink while congestion control simultaneously. It belongs to *upstream reliability guarantee*. Firstly it needs to periodically compute the factual reliability r according to successfully received packets in a time interval. Secondly and the most importantly, ESRT deduces the required sensor report frequency f from r: $f=G(r)$. Thirdly and finally, ESRT informs f to all sensors through an assumed channel with high power and sensors can report even and transmit packets with frequency f. ESRT is an End-to-End approach to guarantee a desired reliability through regulating sensor report frequency. It provides reliability for applications not for each single packet. The addition benefit resulted from ESRT is energy-conservation since it can control sensor report frequency. ESRT brings with such disadvantages: 1) ESRT regulates report frequency of all sensors using the same value. But it may be more reasonable if using different value since each sensor may have different contributions to congestion. 2) ESRT assumes and uses a channel (one-hop) with high power that will influence the on-going data transmission. 3) ESRT mainly considers reliability and energy-conservation.

RMST [7]

RMST (Reliable Multi-Segment Transport) also belongs to *upstream reliability guarantee.* It is designed to run above *Directed Diffusion* (to use its discovered path from sensors to sink) in order to provide guaranteed reliability from sensors to sink (delivery and fragmentation/reassembly) for applications. RMST is a selective NACK-based protocol. RMST basically operates as follows. Firstly, RMST uses timer-driver mechanism to detect data loss and send NACK on the way from detecting node to sources (Cache or non-Cache mode). Secondly, NACK receivers are responsible for looking for the missing packet, or forward NACK on the path toward sink if it fails to find the missing packet or in non-cache mode. Several advantages of RMST are including: 1) No congestion control. 2) No effective energy conservation mechanism. 3) No application-level reliability.

PSFQ [8]

PSFQ (Pump Slowly Fetch Quickly) aims to distribute data from sink to sensors by pacing data at a relatively slow-speed, but allowing nodes that experience data loss to fetch (recover) any missing segments from immediate neighbors very aggressively (local recovery, "fetch quickly"). It belongs to *downstream reliability guarantee*. The motivation of PSFQ is to achieve loose delay bounds while minimizing the loss recovery cost by localized recovery of data among immediate neighbors. It contains three components: Pump operation, Fetch operation, and Report operation. Firstly, sink slowly broadcasts a packet (with such fields-file ID, file length, sequence number, TTL, and report bit) to its neighbors every *T* until all the data fragments has been sent out. Secondly, a sensor can go into fetch mode once a sequence number gap in a file fragment is detected and issue NACK in reverse path to recover missing fragment. The NACK don't need to be relayed unless the number of times the same NACK is heard exceeds a predefined threshold while the missing segments requested by the NACK message are no longer retained in a sensor's cache. Thirdly, sink can make sensors to feedback data delivery status information to it through a simple and scalable hop-by-hop report mechanism. PSFQ has several disadvantages: 1) PSFQ can't detect the loss of single packet since it used only NACK not ACK. 2) PSFQ uses statically and slowly pump that result in large delay. 3) Hop-by-hop recovery with cache will need more buffer.

GARUDA [9]

GARUDA belongs to *downstream reliability guarantee*. It has three primary components. Firstly, GARUDA uses WFP (Wait-for First -Packet) pulse transmission to guarantee success of single/first packet delivery, in order to choose and construct *Core* sensors. Secondly, GARUDA performs *Core* election using such methods-only sensors with *HopCount* of the form *3*i* where *i* is a positive integer, are allowed to elect themselves as *Core* sensors. Thirdly, GARUDA begins two phase loss recovery-Loss recovery for *Core* sensors and Loss recovery for non-*Core* sensors using out-of-sequence NACK. Several disadvantages are including: 1) Support reliability only on the downstream direction from sink to sensors. 2) It provides no congestion control.

ATP [10]

ATP is a new transport protocol for ad-hoc networks. It is a receiver-based and network-assisted end-to-end feedback control algorithm. It uses selective ACKs (SACKs) for packets loss recovery. In ATP, intermediate network nodes compute the sum of exponentially averaged packet queuing delay and transmission delay, called *D*. The idea is that the required end-to-end rate should be the reverse of *D*. The *D* is computed over all the packets traversing the node and used to update the value piggybacked in each outgoing packet if the new value of *D* is bigger than the old value. After this hop-by-hop computation and piggyback, the receiver can get the largest value of *D* that each packet experience on the way. Then the receiver can calculate the required end-to-end rate, the reverse of *D*, for the sender and feedback it to the sender. Then the sender can intelligently adjust its sending rate according to received *D* from the receiver. In order to guarantee reliability, ATP uses selective ACKs (SACKs) as an end-to-end mechanism for loss detection. But the SACK block in ATP is 20, much larger that that in TCP (only 3). ATP decouples congestion control and reliability and achieves better fairness and higher throughput than TCP. But it doesn't consider energy issues and its end-to-end approach might be not the optimal for sensor networks.

SenTCP [11]

SenTCP is an open-loop hop-by-hop congestion control protocol with two special features: 1) It jointly uses average local packet service time and average local packet inter-arrival time in order to estimate current local congestion degree in each intermediate sensor node. The use of packet arrival time and service time not only precisely calculates congestion degree, but effectively helps to differentiate the reason of packet loss occurrence in wireless environments, since arrival time (or service time) may become small (or large) if congestion occurs. 2) It uses hop-by-hop congestion control. In SenTCP, each intermediate sensor node will issues feedback signal backward and hop-by-hop. The feedback signal, which carries local congestion degree and the buffer occupancy ratio, is used for the neighboring sensor nodes to adjust their sending rate in the transport layer. The use of hop-by-hop feedback control can remove congestion quickly and reduce packet dropping, which in turn conserves energy. SenTCP realizes higher throughput and good energy-efficiency since it obviously reduces packet dropping; however, SenTCP copes with only congestion and guarantees no reliability.

V. THE FUTURE PROBLEMS

The major functions of transport control protocols for wireless sensors networks are congestion control, reliability guarantee, and energy conservation that can be passively realized by congestion control and reliability guarantee. What the existing protocols studied is only either congestion or reliability guarantee in uni-direction (upstream or downstream), and none of them settles congestion control and reliability simultaneously in both directions. Moreover, some protocols such as [5] only focusing on congestion control have decreased reliability. However some applications in wireless sensor networks require both functions in both directions, for example, re-tasking and critical time-sensitive monitoring and surveillance.

The second problem of the existing transport control protocols for wireless sensor networks is that they control congestion either through end-to-end or through hop-by-hop

(Although there are end-to-end and hop-by-hop mechanism for congestion control in CODA [5], CODA only simply uses them at the same time, and has no any adaptive method to integrate the two mechanisms for optimization). But an adaptive congestion control that integrates end-to-end and hop-by-hop may be more helpful for wireless sensor networks with diverse applications on it, and useful for energy-conservation and simplification of sensor operation.

The third problem is that the protocols guaranteeing reliability provide either packet-level reliability or application-level reliability, not both of them. If a sensor network supports two applications (one of them needs only packet-level reliability, and another needs only application-level reliability), the existing transport control protocols will face difficulty and will be not the optimal choices. Therefore, an adaptive recovery mechanism is required to support packet-level and application-level reliability, and to be helpful for energy-conservation.

The fourth problem is that the existing transport control protocols have hardly implemented any cross-layer optimization. However lower-layers such as network layer and MAC layer can provide useful information up to transport layer. A new effective and cross-layer optimized transport control protocol can be available through such cross-layer optimization.

VI. Conclusions

In this paper, we present a survey on transport control protocol for wireless sensor networks. First we give out the limitation of TCP and UDP protocols and explain why they are not suitable for wireless sensor networks. Second we briefly review several existing transport control protocols for wireless sensor networks, and list out several problems of the existing protocols. When designing transport control protocols for wireless sensor networks, we should consider such major problems as: 1) how to design an effective and efficient congestion control mechanism? The effective mechanism should avoid as few packets dropping as possible, while guaranteeing high throughput simultaneously. 2) Whether and how to guarantee reliability in transport layer? If it is required to provide packets loss recovery in transport layer, which mechanism is effective and energy-efficient? It should have low requirement on storage since sensor nodes often have small size of memory. 3) how to guarantee fairness among the sensor nodes that usually have different distance from sink. 4) how to utilize cross-layer optimization to improve performance of transport control protocol?

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," Computer Networks, vol. 38, 2002, pp. 393-422.

[2] J. B. Postel, "Transmission control protocol," *RFC 793*, September 1981.

[3] J. B. Postel, "User datagram protocol," *RFC 768*, August 1980.

[4] M. Mathis, "TCP selective acknowledge options," *RFC 2018*, October 1996.

[5] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: Congestion detection and avoidance in sensor networks," *in Proceedings of ACM Sensys'03*, November 5-7, 2003, Los Angeles, USA.

[6] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyidiz, "ESRT: Event-to-sink reliable transport in wireless sensor networks," *in Proceedings of ACM Mobihoc'03*, June 1-3, 2003, Annapolis, USA.

[7] F. Stann and J. Heidemann, "RMST: Reliable data transport in senor networks," *in Proceedings of IEEE SNPA'03*, May 11, 2003, Anchorage, USA.

[8] C.-Y. Wan, A. T. Campbell, "PSFQ: A reliable transport protocol for wireless sensor networks," *in Proceedings of ACM WSNA'02*, September 28, 2002, Atlanta, USA.

[9] S.-J. Park, R. Vedantham, R. Sivakumar, and I. F. Akyildiz, "A scalable approach for reliable downstream data delivery in wireless sensor networks," *in Proceedings of ACM MobiHoc'04*, May 24-26, 2004, Roppongi, Japan.

[10] K. Sundaresan, V. Anantharaman, H.-Y. Hseeh, and R. Sivakumar, "ATP: A reliable transport protocol for ad-hoc networks," *in Proceedings of ACM Mobihoc'03*, June 1-3, 2003, Annapolis, Maryland, USA.

[11] C. Wang, K. Sohraby, and B. Li, "SenTCP: A hop-by-hop congestion control protocol for wireless sensor networks," *in Proceedings of IEEE INFOCOM 2005* (Poster Paper), Miami, Florida, USA, Mar. 2005.

TABLE1: Several transport Control protocols for WSNs

Attributes		**CODA**	**ESRT**	**RMST**	**PSFQ**	**GARUDA**	**SenTCP**
Direction		Upstream	Upstream	Upstream	Downstream	Downstream	Upstream
Congestion	Support	Yes	Passive	No	No	No	Yes
	Congestion detection	Buffer size& Channel condition	Buffer size	-	-	-	Buffer size& Pkts arrival rare
	Open-loop or Closed-loop	Both	Close	-	-	-	
Reliability	Support	No	Yes	Yes	Yes	Yes	No
	Packet or Application Reliability	-	Application	Packet	Packet	Packet	-
	Loss detection	-	No	Yes	Yes	Yes	-
	End-to-End or Hop-by-Hop	-	E2E	HbH	HbH	HbH	-
	Cache	-	No	Yes or No	Yes	Yes	-
	In-sequence or Out-of-sequence NACK	-	N/A	In-seq	Out-of-seq	Out-of-seq	-
	ACK or NACK	-	ACK	NACK	NACK	NACK	-
Energy-conservation		Good	Fair	No result	No result	Yes	Good

(E2E: End-to-end; HbH: Hop-by-hop; Upstream: from sensor to sink; Downstream: from sink to sensor)