2010 International Conference on Broadband, Wireless Computing, Communication and Applications

Wireless Sensor Network Transport Protocol - A State of the Art

[1]A. J. Dinusha Rathnayaka, [2]Vidyasagar M. Potdar
Digital Ecosystems and Business Intelligence Institute
Curtin University of Technology
Perth, Western Australia
[1]abekoon.rathnayaka@postgrad.curtin.edu.au,
[2]v.potdar@curtin.edu.au

[3]Atif Sharif, [4]Saeed Sarencheh, [5]Samitha Kuruppu
Digital Ecosystems and Business Intelligence Institute
Curtin University of Technology
Perth, Western Australia
[3]atif.sharif@postgrad.curtin.edu.au,
[4]sarencheh@gmail.com, [5]kjayamal@gmail.com

Abstract— **In this article, we present a survey of Wireless Sensor Networks (WSNs) existing Transport Protocols. We have evaluated the design concepts of different protocols based on congestion control, reliability support and source traffic priority support. Then we draw the concluding remarks, while highlighting up-and-coming research challenges for WSN transport protocols, which should be addressed further in prospective designs.**

Keywords- Wireless Sensor Network; Transport protocol; Reliability; Congestion control; Priority

I. Introduction

WSN [1] is comprised of tiny embedded devices termed as "motes" that has inbuilt features for sensing, processing and communicating over wireless links. The key objective of a transport protocol is to attain the reliable data transport, while avoiding congestion and achieving energy efficiency. However the proven transport protocols like User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) are inappropriate for WSN due to many constraints [2, 3]. Currently, many researchers have made significant advancements in protocol design and the Table I illustrates one of the common classifications based on their capabilities in congestion control and reliability. The generic structure of the transport protocol (Figure 1) is comprised of three main functional modules: (i) congestion control module, (ii) reliability module, and (iii) priority module. Congestion module prevents congestion, thereby reducing the packet drops thus resulting in increased throughput. The reliability module ensures the successful delivery of each segment to the ultimate destination. The priority module differentiates the source traffic based on the importance of the application. The detailed description of each module is available in [30]. In this paper, we evaluate the functionalities of existing transport protocols based on above three modules (Table II).

TABLE I. Classification of existing transport protocols

<table>
<tr><th>Reliability support only</th><th>Congestion control only</th></tr>
<tr><td>• ERTP: Energy-efficient and Reliable Transport Protocol [15]
• GARUDA [16]
• DTSN: Distributed Transport for Sensor Networks [17]
• RBC: Reliable Bursty Convergecast [18]
• DTC: Distributed TCP Caching [19]
• RMST: Reliable Multi-Segment Transport [20]
• PSFQ: Pump Slowly Fetch Quickly [21]</td><td>• PHTCCP : Prioritized Heterogeneous Traffic-oriented Congestion Control Protocol [22]
• PCCP : Priority-based Congestion Control Protocol [23]
• Siphon [24]
• Fusion [25]
• CCF: Congestion Control and Fairness [26]
• Trickle [27]
• CODA: Congestion Detection and Avoidance[28]
• ARC: Adaptive Rate Control [29]</td></tr>
<tr><th colspan="2">Both congestion control and reliability support</th></tr>
<tr><td colspan="2">• TRCCIT: Tunable Reliability with Congestion Control for Information Transport [4]
• CRRT: Congestion aware and Rate controlled Reliable Transport[5]
• CTCP: Collaborative Transport Control Protocol [6]
• RT2: Real-Time and Reliable Transport [7]
• ART: Asymmetric and Reliable Transport [8]
• RCRT: Rate-Controlled Reliable Transport [9]
• Flush [10]
• DST: Delay Sensitive transport [11]
• PORT: Price-Oriented Reliable Transport[12]
• STCP: Sensor Transmission Control Protocol [13]
• ESRT: Event-to-Sink Reliable Transport [14]</td></tr>
</table>

II. Evaluation

A. Congestion Control

TRCCIT, CRRT, CTCP, RT^2, ART, RCRT, Flush, DST, PORT, STCP, ESRT, PHTCCP, PCCP, Siphon, Fusion, CCF, Trickle, CODA and ARC claim to achieve congestion control. Now we discuss the Congestion control attributes [30] of these protocols based on the congestion detection, notification and avoidance.

1) Congestion Detection

Congestion detection refers to the identification of possible events, which create congestion in the network. Different protocols use different parameters and events to detect congestion. STCP, ESRT and Fusion solely detect the congestion when the buffer usage is higher than the predefined threshold, whereas RT^2 and DST monitor the node delay threshold in addition to the buffer occupancy. CTCP uses both transmission error loss rates and the buffer

978-0-7695-4236-2/10 $26.00 © 2010 IEEE
DOI 10.1109/BWCCA.2010.177

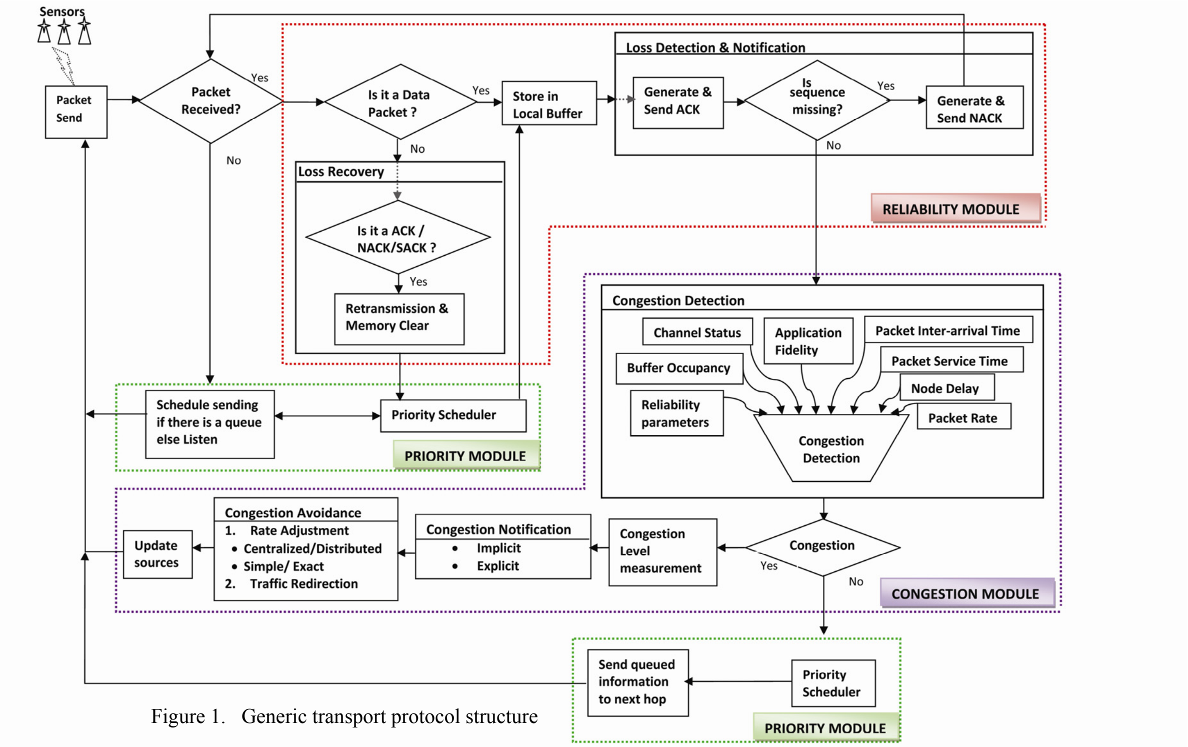

Figure 1. Generic transport protocol structure

usage. ARC, ART, PORT and RCRT detect congestion based on feedback parameters of the reliability module. For example, ART assumes congestion if ACK is not received to selected dominating sensors within timeout and similarly ARC monitors unsuccessful packet deliveries at sink. RCRT detects congestion based on the time to recover the packet loss and PORT uses the number of transmission attempts made before a successful delivery, also termed as 'node price' [12], and the loss rates of the links. CODA, Siphon, and Flush consider channel condition in congestion detection; Flush measures the intra-path interference [10] at each hop. In CODA and Siphon, sensors listen to the channel only when the buffer occupancy is high, trace the channel busy time and calculate the local channel loading. In addition, Siphon also checks whether the event detection rate at the sink is below the perceived application fidelity. In contrast, PCCP identifies the congestion when packet service time (PST) is higher than the packet inter-arrival time (PIAT) at the MAC layer. PST is termed as the time duration it takes to process one packet at a node, whereas PIAT means the time interval between the two sequential arriving packets at each node. PHTCCP relies on the rate ratio calculated using PST at the MAC layer, and CCF uses PST at transport layer. TRCCIT identifies congestion when the packet incoming rate is higher than the packet outgoing rate and CRRT assumes congestion when nodes experience the reduced packet forwarding rate and excessive buffer usage.

As a whole, the most common techniques in congestion detection would be to use the buffer occupancy. But the mere prediction of congestion based on the high buffer usage of specific sensor is not sufficient, because even though the buffer usage is low, the sensors may experience congestion due to the network traffic among other sensor nodes in the neighbourhood as shared communication medium nature of WSNs. Therefore it is necessary to realize the network channel condition around sensors. However other than the channel status and buffer occupancy, the other influential factors like PST, PIAT and node delay also must be considered to estimate an accurate congestion degree. Some protocols detect the congestion based on the reliability issues. Solely depending on such factors like time to recover the loss or successful reception of packets is not very feasible, as a single packet drop may force the rate reduction to avoid the congestion, which can further result in energy consumption.

1) Congestion Notification

The detected congestion should be informed to relevant neighbours either explicitly by means of a control message or implicitly by embedding the information into the normal data packets. All the transport protocols implicitly notify the congestion, whereas CTCP and CODA send explicit notifications. CTCP generates control messages when congestion occurs as well as when congestion is resolved. In CODA, a suppression message is sent to their upstream neighbours via a backpressure method. When we compare two notification techniques; implicit and explicit, the former is more power efficient as it avoids the overhead associated with control messages.

2) Congestion Avoidance

The protocols avoid congestion mainly utilizing three mechanisms; (a) rate adjustment, (b) traffic redirection, and (c) polite gossip policy.

a) Rate adjustment: Rate regulation schemes can be categorized as either centralized or distributed based on the location, which the rate adjustment plans are implemented;

- *Centralized rate adjustment:* In CRRT, RCRT, ESRT, and DST, the rate control functionalities are implemented centrally at the sink.
- *Distributed rate adjustment:* STCP, CODA, Flush, ARC, ART, Fusion, CCF, PCCP, RT^2, TRCCIT and PHTCCP develop rate control functionality at each hop of the network.

As a whole, when we compare the centralized scheme with distributed scheme, the centralized scheme may take unbiased decision about the rate, since the sink has the broader view of the network and controls the aggregate rate of the network. It is also more energy efficient to perform decision making tasks at the base station, as the sensor nodes are energy constrained and limited in computational abilities. On the other hand, distributed scheme may reduce the congestion quicker as the rate is adjusted at each hop.

These rate adjustment techniques are either simple or exact based on the rate regulation algorithms used.

- *Simple rate adjustment like AIMD:* STCP, CODA, Flush, ARC, ART, Fusion, ESRT, CRRT and RCRT use merely single congestion bit to inform whether there is congestion or not. In such cases, the rate adjustments are done using local policies like AIMD (Additive Increase and Multiplicative Decrease) and its variants. Fusion also uses similar method, which stops sending packets to congested nodes, when the congestion is heard. ART also temporarily terminates the traffic of non essential nodes. CRRT uses AIAD (Additive Increase and Additive Decrease) [5] scheme, which additively decreases the rate in congestion, avoiding aggressive multiplicative rate reduction in AIMD.
- *Exact rate adjustment:* Here the protocols estimate and notify the additional congestion information, other than just notifying whether there is congestion or not. For example, CCF, TRCCIT inform the allowable data rate, which should be updated in next data transmission. PCCP and PHTCCP inform the congestion degree and RT^2 and DST inform the delay constrained reliability parameter [7], which the rate is adjusted accordingly.

When we evaluate the two techniques; simple rate adjustment technique is not feasible since it is difficult to precisely adjust the transmission rate using limited information given by the binary congestion notification bit. And also AIMD like local rate control policies also result greater sending rates at the sources closer to the sink when compared to other nodes, causing uneven number of packet reception at the sink, while negatively influencing the fairness and link utilization. Therefore the exact rate control technique is more suitable to implement more accurate rate adjustment plan.

b) Traffic redirection: Here the protocols divert the outgoing traffic to optimal uncongested paths to ease the congestion. Siphon solely depends on traffic redirection and, PORT and STCP consider this in addition to rate adjustment. PORT selects the alternative paths, based on the node prices [12] and the loss rate feedback obtained from neighbours. STCP uses the congestion bit enabled acknowledgement packets to realize the congested path. Siphon distributes virtual sinks [24] across the sensor network, which siphons the events with high traffic loads.

c) Polite gossip policy: Trickle uses polite gossip policy [27] that works in such a way that, each node tries to broadcast a summary of its data to local neighbours periodically, but if nodes hear identical data from neighbours it "politely" suppress its own broadcasting. If a new code of data is received, nodes shorten their broadcast period to broadcast the new code sooner. When motes hear older data than its own, the protocol sends small trickle of packets to update nodes.

B. Reliability Support

TRCCIT, CRRT, CTCP, RT^2, ART, RCRT, Flush, DST, PORT, ESRT, STCP, ERTP, GARUDA, DTSN, RBC, DTC, RMST, and PSFQ protocols claim to achieve reliability. As summarized in Table II, the reliability support mechanisms [30] of transport protocols vary in different criteria; reliability direction, reliability level, loss detection, notification and recovery.

1) Reliability direction

Transport protocols offer reliability either in upstream for the dataflow traffic from sources to sink or in downstream for the control and query traffic from sink to source. All the protocols offer upstream reliability, except PSFQ and GARUDA which only support downstream reliability.

In reliability point of view it will be more perfect if the bidirectional reliability can be achieved. From these protocols, only ART achieves the bidirectional reliability.

2) Reliability level

The level of reliability means the extent of reliability supported by the protocol. It can be packet level reliability, which considers the successful delivery of every packet, event level reliability, which considers only the successful event detection and the destination reliability, where the guaranteed data transfer is enabled only for a cluster of sensors. Most protocols discussed here concern about packet reliability except PORT, ART, ESRT and DST which provide event level reliability. GARUDA offers destination reliability in addition to the packet reliability.

The packet level reliability is much more trustworthy than the event reliability as it guarantees the delivery of each and every bit of information. But on the other hand, packet level reliability involves more energy utilization. Therefore the protocol design should be more flexible to adapt both event and packet reliability depending on the targeted application.

3) Loss recovery

The loss recovery refers to repairing the packet drops by means of retransmission. This is done in two ways; end-to-end loss recovery, and hop-by-hop loss recovery, which the

TABLE II. COMPARISON OF EXISTING TRASNSPORT PROTOCOLS

	Protocols	Congestion Detection	Congestion Notification	Congestion Avoidance	Reliability Direction	Reliability Level	Loss Recovery	Loss Detection & Notification	Diverse Traffic precedence
Both Reliability and Congestion control	TRCCIT(2010)	Packet rate	Imp	Rate adjustment	Up	Packet	Hop-by-hop	iACK , ACK	Yes
	CRRT (2009)	Queue occupancy Packet rate	Imp	Rate adjustment	Up	Packet	End-to-end Hop-by-hop	NACK, MAC	-
	CTCP(2008)	Transmission error loss Queue occupancy	Exp	Rate adjustment	Up	Packet	Hop-by-hop	ACK Double ACK	Yes
	RT^2(2008)	Node delay Queue occupancy	Imp	Rate adjustment	Up	Packet	Hop-by-hop	SACK	Yes
	ART(2007)	ACK received to set of nodes (core)	Imp	Reduce traffic of set of nodes	Both	Event	End-to-end	ACK, NACK	-
	RCRT(2007)	Time to recover loss	Imp	Rate Adjustment	Up	Packet	End-to-end	NACK, cumulativeACK	-
	FLUSH(2007)	Queue occupancy Link interference	Imp	Rate adjustment	Up	Packet	End-to-end	NACK	-
	DST (2006)	Node delay Queue occupancy	Imp	Rate adjustment	Up	Event	End-to-end	-	Yes
	PORT(2005)	Node price Link-loss rates	Imp	Traffic redirection Rate adjustment	Up	Event	-	-	-
	ESRT (2005)	Queue occupancy	Imp	Rate adjustment	Up	Event	-	-	-
	STCP (2005)	Queue occupancy	Imp	Rate adjustment Traffic redirection	Up	Packet	End-to-end	NACK, ACK	Yes
Reliability only	ERTP (2009)	-	-	-	Up	Packet	Hop-by-hop	iACK, ACK	-
	GARUDA (2008)	-	-	-	Down	Packet Destination	Two-tier loss recovery	NACK	-
	DTSN (2007)	-	-	-	Up	Packet	End-to-end	ACK,NACK	-
	RBC (2005)	-	-	-	Up	Packet	Hop-by-hop	iACK	-
	DTC (2004)	-	-	-	Up	Packet	Hop-by-hop	ACK, SACK	-
	RMST(2003)	-	-	-	Up	Packet	Hop-by-hop End-to-end	NACK, MAC	-
	PSFQ (2002)	-	-	-	Down	Packet	Hop-by-hop	NACK	-
Congestion control only	PHTCCP(2008)	Packet service ratio	Imp	Rate adjustment	-	-	-	-	Yes
	PCCP(2006)	Packet inter-arrival time Packet service time	Imp	Rate adjustment	-	-	-	-	Yes
	Siphon (2005)	Queue occupancy Application fidelity	-	Traffic redirection	-	-	-	-	-
	Fusion (2004)	Queue occupancy	Imp	Rate adjustment	-	-	-	-	-
	CCF (2004)	Packet service time	Imp	Rate adjustment	-	-	-	-	-
	Trickle (2004)	-	-	Polite gossip	-	-	-	-	-
	CODA (2003)	Queue occupancy Channel status	Exp	Drop Packets Rate adjustment		-	-	ACK	-
	ARC (2001)	Successful/ unsuccessful delivery of packets	Imp	Rate adjustment	-	-	-	-	-

recovery process is performed between two end points and two adjacent nodes respectively. CRRT, RCRT, Flush, STCP, ART, DST, and DSTN offer end-to-end loss recovery, whereas TRCCIT, CRRT, DTC, CTCP, RT^2, PSFQ, RBC, ERTP, RMST, DTC, and RBC use hop-by-hop method for loss recovery. From the transport protocols discussed, PORT, DST and ESRT dealing with event level reliability, do not focus on loss recovery mechanisms.

In comparison of two methods, end-to-end loss recovery is not very feasible in large networks with multiple hops due to energy consuming retransmissions. This also causes the loss of control messages as it flows through number of hops. Hop-by-hop mechanism consumes lesser amount of energy as only two adjacent nodes involve in loss recovery. Eventually it alleviates the congestion at a quicker rate. But the hop-by-hop technique uses more memory as each intermediate sensor maintains a data cache. So we need to investigate further to see which approach is better.

4) Loss detection and notification

The transport protocols implement different feedback mechanisms to communicate the packet loss to neighbours in order to initiate retransmissions. Positive acknowledgements (ACKs) are generated for all the packets received and negative acknowledgements (NACKs) are sent for the missing sequence numbers in received stream. The receiver sends selective acknowledgements (SACK) to inform the sender about all the sequentially received packets using one control packet. Implicit acknowledgement (iACK) is the interpretation for the transmitter's ability to overhear the forwarding transmissions in physical wireless links.

Now we discuss how existing protocols use different loss notification techniques to achieve the desired loss recovery.

a) ACK/ iACK based system

CTCP achieves two hop by hop reliability levels; fist level is obtained using ACKs and the second higher reliability level is obtained using ACKs and 'Double ACKs'[6]. The term 'Double ACK' means that a node sends ACKs to the preceding node once the data are received from it and also when the ACK is obtained from the following node for transmitted data. RBC uses iACK with windowless block acknowledgement scheme [18]. ERTP uses stop-and-wait hop-by-hop iACKs that the sender retransmits the packet if it does not hear the iACK within the timeout and also uses ACKs to confirm the packet reception at sink. TRCCIT also uses timer driven iACK in similar way and the receiver notifies the reception of the retransmitted packet using ACK.

b) NACK based loss detection

In Flush, the sink sends a single cumulative NACK packet, which can hold up to three sequence numbers corresponding to three missing packets. CRRT achieves end-to-end reliability using NACK and hop-by-hop reliability using MAC layer retransmissions. Loss detection in RMST is timer driven and implements both end-to-end and hop-by-hop selective request NACK. RMST also considers MAC level ARQ (Automatic Repeat Request) for the reliability of control packets. GARUDA utilizes a NACK-based two stage recovery process; which the first stage deals with recovering all lost packets in a set of dominating nodes (core nodes). In second stage, non-core nodes request retransmission from core nodes, only after the completion of all the retransmissions of its core node. In PSFQ, loss repair request is made using NACK messages. If the packets are lost in a bursty event, single fetch would be sent, by aggregating the loss windows with missing sequence number pairs.

c) ACK and NACK based system

STCP differentiates dataflow as continuous and event-driven and uses NACKs for continuous flow and ACKs for event-driven flow to ensure successful data delivery. ART recovers upstream data using ACK, and downstream queries using NACK. DTSN achieves "full reliability level" using selective repeat ARQ, employing both ACK and NACK and "differentiated reliability level" using ARQ together with Forward Error Correction (FEC) [17]. RCRT employs NACK for loss recovery, but also sink sends a cumulative ACK sequence number to clear the retransmit buffer.

d) SACK based system

DTC, an attempt to optimize TCP, identifies successful packet delivery by using SACK and ACK. RT^2 also achieves loss recovery solely using SACK.

When we evaluate different loss notification schemes, we find that NACK based protocols are unable to detect when the last fragments of a stream are lost. Generally here the packet loss is concluded when a packet with a sequence number higher than expected is received and the last packet is not followed by a packet with higher sequence number. NACK based protocol is also unable to detect if the entire message is lost. In such scenarios, the receiver would never aware of that message even to send a notification. When compared to the NACK-based schemes, ACK-based protocols can accomplish more reliability, but if single ACKs are sent for each packet, it consumes more energy and it is minimized in SACK. iACK is more energy efficient since it does not deal with overhead. But in order to utilize an iACK based technique, sensor nodes should be able to overhear the physical channel. Therefore iACK may not be the correct choice when the channel exhibits transmission errors and when the channel is not duplex.

C. Priority Support

TRCCIT, PCCP, PHTCCP, STCP, CTCP, DST and RT^2 claim to achieve flexible differentiation of source traffic by means of precedence levels or application/flow identifiers. These are assigned based on nature of flows, importance of application, and packets' remaining time to deadline [7].

1) Priority Scheduler

PCCP defines local source traffic priorities and transit traffic priorities at each node. Similarly PHTCCP also assigns the precedence levels to diverse sensors. DST and RT^2 determine the packet's remaining time to deadline [7] and the event packets are given higher priority at the nodes, as their remaining time to deadline decreases. In TRCCIT, the sources define tunable reliability [3] levels based on the application requirement. STCP differentiates the source traffic as continuous and event-driven and obtains different reliability levels. Similarly CTCP also assigns two different reliability levels for heterogeneous applications.

III. Discussion and Conclusion

This article evaluated existing transport layer protocols proposed for WSNs, based on their capability to handle congestion control, reliability and source priority. However there are still several research concerns such as cross-layer optimization, weighted fairness, and active queue monitoring in congestion control, which should be addressed further in future transport protocol improvements.

Cross-layering enhances the performance of the transport protocol by merging the functionalities of other layers of the protocol stack. For example this can be used to select better paths for retransmission and to get error reports. Very few protocols such as RT^2, PCCP and PHTCCP constructively use cross-layering. The protocols such as RMST and DTSN also rely on link layer ARQs but do not take prominent cross-layering benefits.

Achieving application-specific QoS and weighted fairness that allocates more bandwidth for imperative sensors are crucial in handling heterogeneous applications. But only few protocols such as PCCP, PHTCCP DST, RT^2, STCP and CTCP address the node prioritization and flexible reliability levels for diverse sensors.

Most congestion control mechanisms in current protocols monitor the channels and dynamically regulate the data transmission rate only when the congestion is detected. But it is vital to monitor the channel intelligently to control the possible anticipated congestion scenarios, before the real congestion occurs. Only few protocols like TRCCIT, RT^2 concern on active queue monitoring and congestion control.

References

[1] D. Culler, D. Estrin, and M. Srivastava,“Overview of Sensor Networks”, *IEEE Computer*, 37(8), pp. 41-49, 2004.

[2] TCP. Retrieved August 2008: http://www.ibiblio.org/pub/docs/rfc/rfc793.txt.

[3] UDP. Retrieved August 2008: http://tools.ietf.org/html/rfc768.

[4] F. K. Shaikh, A. Khelil, A. Ali, and N. Suri, "TRCCIT: Tunable Reliability with Congestion Control for Information Transport in Wireless Sensor Networks," in *the Proceedings of the International Wireless Internet Conference (WICON)*, Singapore, 2010.

[5] M. Alam and C. S. Hong, "CRRT: Congestion-Aware and Rate-Controlled Reliable Transport in Wireless Sensor Networks," *IEICE Trans. Commun,* vol. E92-B, pp. 184-189, January 2009.

[6] E. Giancoli, F. Jabour, and A. Pedroza, "CTCP: Reliable Transport Control Protocol for Sensor Networks," in *the Proceedings of Fourth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, Sydney,Australia, 2008, pp. 493-498.

[7] V. Cagri, Gungor, Ö. B. Akan, and I. F. Akyildiz, "A real-time and reliable transport $(RT)^2$ protocol for wireless sensor and actor networks," in *the IEEE/ACM Transactions on Networking (TON)*, Piscataway, NJ, USA, 2008, pp. 359-370.

[8] N. Tezcan and W. Wang, "ART: an asymmetric and reliable transport mechanism for wireless sensor networks," *International Journal of Sensor Networks,* vol. 2, pp. 188-200, June 2007.

[9] J. Paek and R. Govindan, "RCRT: rate-controlled reliable transport for wireless sensor networks," in *the Proceedings of the 5th international conference on Embedded networked sensor systems*, Sydney, Australia, 2007, pp. 305 - 319.

[10] 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," in *the Proceedings of the 5th international conference on Embedded networked sensor systems* Sydney, Australia, 2007, pp. 351 - 365.

[11] V. C. Gungor and O. B. Akan, "DST: Delay sensitive transport in wireless sensor networks," in *the Proceedings of 7th IEEE International Symposium on Computer Networks*, Istanbul, Turkey, 2006, pp. 116–122.

[12] Y. Zhou and M. R. Lyu, "PORT: a price-oriented reliable transport protocol for wireless sensor network," in *the proceedings of 16th IEEE International Symposium on Software Reliability Engineering* Chicago, 2005, pp. 117-126.

[13] H. Zhang, A. Arora, Y.R.Choi, Y.G.Iyer, S.Gandham, and S.Venkatesan, "STCP: a Generic Transport Layer Protocol for Wireless Sensor Networks," in *the Proceedings of the 14th IEEE International Conference on Computer Communications and Networks (ICCCN)*, USA, 2005, pp. 449–454.

[14] Y.Sankarasubramaniam, O.B.Akan, and I.F.Akyildiz, "ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks," in *the Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM Mobihoc)*, USA, 2003, pp. 177–188.

[15] T. Le, W. Hu, Peter, Corke, and S. Jha, "ERTP: Energy-efficient and Reliable Transport Protocol for data streaming in Wireless Sensor Networks," *Computer Communications,* vol. Volume 32, pp. 1154-1171, May 2009.

[16] S.J.Park, R.Vedantham, R.Sivakumar, and I.F.Akyildiz, "A scalable approach for reliable downstream data delivery in wireless sensor networks," in *the Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM Mobihoc),* Japan, 2004, pp. 78-79.

[17] B.Marchi, A.Grilo, and M.Nunes, "DTSN – Distributed Transport for Sensor Networks," in *the Proceedings of IEEE Symposium on Computers and Communications (ISCC)*, Aveiro, Portugal, 2007.

[18] M.G.Gouda, "Reliable Bursty Convergecast in Wireless Sensor Networks," in *the Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM Mobihoc)*, USA, 2005, pp. 266-276.

[19] A.Dunkels, T.Voigt, H.Ritter, and J.Alonso, "Distributed TCP Caching for Wireless Sensor Networks," in *the Proceedings of the 3rd Annual Mediterranean Ad Hoc Networking Workshop*, Turkey, 2004.

[20] F.Stann, and J.Heideman, "RMST: Reliable Data Transport in Sensor Networks," in t*he Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications (SNPA)*, USA, 2003, pp. 102–113.

[21] C.Y.Wan, A.T.Campbell, and L.Krishnamurthy, "PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks," in *the Proceedings of ACM International Workshop on WSN and Applications (WSNA),* USA, 2002, pp. 1–11.

[22] M. M. Monowar, M. O. Rahman, A.-S. K. Pathan, and C. S. Hong, "Congestion control protocol for wireless sensor networks handling prioritized heterogeneous traffic," in *the Proceedings of the 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, Dublin, Ireland, 2008.

[23] C. Wang, K. Sohraby, V. Lawrence, B. Li, and Y. Hu, "Priority-based Congestion Control in Wireless Sensor Networks," *Trustworthy Computing,* vol. 1, pp. 22-31, 2006

[24] C. Y. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft, "Siphon: overload traffic management using multi-radio virtual sinks in sensor networks," in *3rd international conference on Embedded networked sensor systems*, San Diego, California, USA, 2005, pp. 116 -129

[25] B.Hull, K.Jamieson, and H.Balakrishnan, "Mitigating Congestion in Wireless Sensor Networks," in *the Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (ACM SenSys)*, USA, 2004, pp. 134-147

[26] C. T. Ee and R. Bajcsy, "Congestion control and fairness for many-to-one routing in sensor networks," in *the Proceedings of 2nd international conference on Embedded networked sensor systems*, Baltimore, MD, USA, 2004, pp. 148 - 161

[27] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *the Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation - Volume 1 table of contents*, San Francisco, California, 2004, pp. 29-31.

[28] C.Y.Wan, S.B.Eisenman, and A.T.Campbell, "CODA: Congestion detection and avoidance in sensor networks," in *the Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems (ACM SenSys)*, USA, 2003, pp. 266–279.

[29] A. Woo and D. E. Culler, "A transmission control scheme for media access in sensor networks," in *the Proceedings of the 7th annual international conference on Mobile computing and networking*, Rome, Italy, 2001, pp. 221 - 235.

[30] A. J. D. Rathnayaka, V. M. Potdar, and A. Sharif, "Wireless Sensor Networks: Challenges Ahead," accepted by *5th International Conference on Broadband and Wireless Computing, Communication and Applications*, Fukuoka, Japan, 2010, in press.