

© 2010 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Priority Enabled Transport Layer Protocol for Wireless Sensor Network

¹Atif Sharif, ²Vidyasagar Potdar

*Digital Ecosystems and Business Intelligence Institute,
Curtin University of Technology
Perth, WA, Australia*

¹atif.sharif@postgrad.curtin.edu.au

²v.potdar@curtin.edu.au

A.J.D.Rathnayaka

*Department of Electronic and Telecommunication
Engineering,
University of Moratuwa
Sri Lanka*

ajdr03@ent.mrt.ac.lk

Abstract—Achieving Quality of Service (QoS) objective in Wireless Sensor Network (WSN) handling the multimedia information has significantly gained the importance lately besides energy efficient hardware designing. Transport layer of the WSN communication protocol stack plays a significant role in meeting the QoS objective of WSN. This paper presents a light weight transport protocol for WSN that can handle packets from a numbers of sources having different sensed information and having different priority levels. The protocol assigned middle motes are intelligent enough to achieve prioritization in transmission based on the priority level and packet's Time-To-Live (TTL) information. Extensive simulation is carried out for the three different modes of the envisaged protocol having no prioritized enabled storage, complete prioritized enabled storage and distributed prioritized enabled storage. The results reveal that the significant improvement is observed in case of distributed prioritized enabled storage, approximately 3% data loss occurred, in comparison to 7% data loss for without prioritized enabled storage mode.

Keywords-WSN, transport layer protocol, IEEE 802.11, reliability, congestion control

I. INTRODUCTION

WSN comprised of energy hungry devices also called "motes" that are wirelessly interconnected for communicating the sensed piece of information from the region of interest to the central control station or sink. For the longevity of the WSN, transport layer protocol plays a vital role in assuring the reliability and congestion control aspects of the design. In general the WSN transport layer protocol may include two main functions: congestion control and data reliability or loss recovery. For congestion control in WSN it is firstly important to detect whether or not congestion happens and if it is happen then when and where it happens. Congestion can be detected by a number of ways among them includes:

- Monitoring intermediate mote's buffer occupancy,
- Wireless link load capacity,
- Packet processing time exceeds the packet inter-arrival time limits.

The congestion control and reliability features of the WSN transport layer protocol can be addressed in two ways:

- 1) *End-to-End (E-2-E)*,
- 2) *Hop-by-Hop (H-B-H)*.

The E-2-E approach is quite robust and simple but it will increase the communication overhead with more on-going control packets in network. Whereas the H-b-H approach would require to change the mote's behavior in the entire path from the source to sink but it quickly dampens the local hot

spots (caused by congestion) with fewer number of on-going control packets thus saving considerable mote's energy budget. So when practically designing the WSN's transport layer protocol we should carefully consider the trade-off between E-2-E and H-b-H approaches for WSN.

In WSN there are a number of reasons for data packet loss including:

- Congestion in the network,
- hidden mote problem,
- Poor Signal to Noise Ratio (SNR) due to bad channel (wireless by nature) quality,
- Link breakage due to mote failure.

In WSN the role of data reliability is to allow a data loss recovery mechanism in order to retrieve the correct information. For detecting the data packet loss, in either E-2-E or H-b-H framework, the WSN transport protocol uses the packet sequencing concept that involves the Acknowledgement (ACK)/Not Acknowledgement (NACK) control packets similar to that used in packet-switched networks. Also the WSN, targeted for heterogeneous application i.e. multimedia and scalar sensor support, the retransmission of multimedia traffic flow significantly depletes WSN's allocated energy budget. Significant efficiency in network's energy budget can be achieved by prioritizing the source information [1]. So the source packet with critical E-2-E latency limits (like audio or video data) would need to be processed first by the intermediate motes in comparison to less critical scalar information.

The rest of the paper is organized as following. After introducing the problem definition in section 2 followed by section 3 where we describe the simulation setup used for observing the behavior. In section 4 we will show the simulation results we have taken. The discussion followed by the conclusions will be presented in the section 5.

II. RELATED WORK

Transport protocols are designed to support specific optimization aspects and/or application scenarios. TCP (Transmission Control Protocol) [2] and UDP (User Datagram Protocol) [3] are popular transport control protocols extensively used in Internet, but both of them are not the correct selection for Wireless Sensor Networks (WSN) due to many limitations. Proven transport Protocols for WSNs are identified and classified based on congestion control and/or reliability guarantee in upstream (from sensor motes to sink) or downstream (from sink to sensor motes).

Table: 1 contains several transport control protocols classifications for WSNs.

TABLE I. WSN TRANSPORT LAYER PROTOCOLS

Features Protocol	Direction	Congestion Control Support	Reliability Management Support
CODA	Upstream	Yes	No
ESRT	Upstream	Passive	Yes
RMST	Upstream	No	Yes
PSFQ	Downstream	No	Yes
GARUDA	Downstream	No	Yes
STCP	Upstream	Yes	Yes
TCPW	Upstream	Yes	No

CODA [4] (Congestion Detection and Avoidance) provides sink/receiver based congestion control involving congestion detection, open-loop H-b-H back pressure and closed loop multi source regulation. However it does not offer reliability control feature and offers only unidirectional control in forward (fwd) direction from source to sink.

ESRT [5] (Event-to-Sink Reliable Transport) protocol simultaneously supports reliability and congestion control features in forward direction. However, reliability of single packet is not guaranteed by ESRT instead supports only application level reliability. Also it uses high power mode for data communication over a single hop that may influence the on-going data transmission.

RMST [6] (Reliable Multi-Segment Transport), protocol uses selective ACK (SACK) and designed to run in conjunction with directed diffusion. Although RMST offers high reliability with guaranteed delivery and fragmentation/reassembly however it lacks energy conservation mechanism and does not support the congestion control feature of transport protocol design.

PSFQ [7] (Pump Slowly Fetch Quickly) protocol offers the reliability support in the reverse (rev) direction from sink mote to source motes. It facilitates data from sink to source at relatively slow speed (Pump Slowly). It uses intermediate storage motes for rapid H-b-H recovery of the lost information by using NACK. However, PSFQ unable to detect the loss of single packet when information is communicated as bulk transmission of packets and also require more caching space at intermediate motes.

GARUDA [8] protocol does not support congestion but ensures reliability in rev direction from sink mote to sensor motes. It uses WFP (Wait-for First -Packet) pulse transmission and selects core motes located at a hop count of integral multiple of 3 for storing the packet information that could possibly be used for packet loss recovery using NACK based control attribute and two-tier two-stage loss recovery mechanism.

STCP [9] (Sensor Transmission Control Protocol) offers Sink based controlled variable reliability and congestion control features simultaneously. For continuous and even driven applications STCP uses NACK and ACK based control attribute for assuring E-2-E retransmissions. Reliability is being dictated by the source mote where as the intermediate motes

are used for detecting the congestion and notifying it to sink by setting the congestion notification field of the transmitted packet.

TCPWW [10] (TCP Westwood) [2] is a sender based extension of the existing TCP protocol. It refines the window control and back-off process of the existing TCP design thereby achieving greater design efficiency under sporadic or random data losses. Under congested network conditions TCPWW uses the estimated value of the available channel bandwidth for properly setting the congestion window and slow start threshold limits. It also ensures rapid data retrieval, by avoiding excessively conservative reductions of congestion window and the slow start threshold limits.

III. PROTOCOL OVERVIEW

The proposed transport protocol scheme comprised of following functional modules:

- 1) Congestion Control
 - a) Congestion Detection
 - b) Congestion Notification
 - c) Source rate adjustment
- 2) Reliability or Loss recovery module

The purpose of the congestion control module is to detect the congestion in the network, the congestion index (C_i) is used to measure the severity of the congestion in the network and is given by Eq (1) [15]

$$C_i = \frac{K}{M} * \Gamma + N * ((Q_i * T_{1-p}) + T_{RTS/CTS} + T_{1-h-pr}) \quad (1)$$

where,

K = total available free space in intermediate mote's storage,

M =total available space in intermediate mote's storage,

N = number of hops between source and sink,

Q_i = Current Queue index,

T_{1-p} = Time to execute one packet,

$T_{RTS/CTS}$ =time delay for RTS/CTS information exchange,

T_{1-h-pr} =time required for 1-hop propagation,

Γ = time to process one packet.

This C_i value will be used along with the E-2-E packet latency $T_{delay} (E - 2 - E)$ by the envisaged transport protocol and is helpful in deciding the future source rate plan in order to mitigate the congestion in the network. The newly estimated $T_{delay} (E - 2 - E)$ and rate value involving C_i is given by [13]

$$\hat{T}_{delay} (c_i) = \overline{T_{delay} / C_i} \quad (2)$$

Hence the new estimated rate value comes to be:

$$\hat{R} (c_i) = \frac{\hat{T}_{delay} (c_i)}{N} \quad (3)$$

Based on the computed C_i , the sink broadcast the new rate plan to all sources and upon listening to this broadcasted rate value the sources become aware with the congestion state of the network and immediately switch their sending rate to newly broadcasted value.

The software description of the proposed protocol is shown in the Figure 1. The prioritized packet communication including the retransmission (here we are assuming that link is already established at the lower layers, below transport level, of the communication protocol stack) is done in the following way:

1. Source sense the information, locally process it and sends the data packet to intermediate storage mote.
2. The intermediate mote receives the incoming packet and with ACK = NACK = 0 it assumes the new data. If it is first packet then it clears the storage memory initially else stores the packet to immediate location. The storage along with the duration of storage is done based on some priority.
3. The intermediate mote after storing the information will forward the data packet to the next hop or sink. It will first search the priority index and packet with higher priority will be served first. In case if there are multiple packets with same priority then the decision will be made on the individual's packet TTL value.
4. Sink, with ACK=NACK=0, assumes the incoming new packet and mark its sequence number and return an ACK (ACK=1, NACK=0) packet to the immediate intermediate storage mote, which on reception of it delete the corresponding packet entry from its memory.
5. Sink if misses some packets then upon receipt of the newly arrived packet it will generate a NACK packet with entries marked with the missing sequence numbers and send it to the immediate intermediate mote for possible retrieval of the missing sensed information.
6. Upon receiving the NACK packet the intermediate storage mote search the missing packets in its storage and resend the information to the sink with ACK=NACK=0.
7. If the immediate intermediate storage mote fails to find the missing sequence number it resend the NACK packet back to the next intermediate mote on the way towards source until it will ACK.

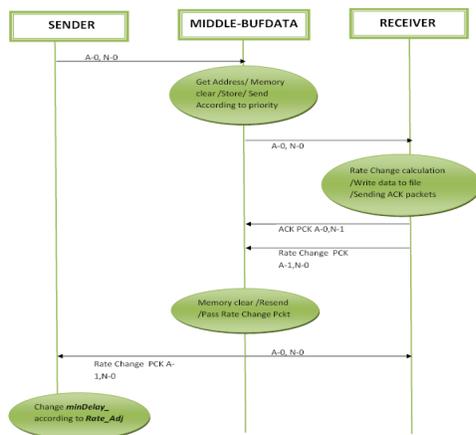


Figure 1. Software description of the proposed transport layer protocol

8. The sink based on Equations 1 -3 start making measurement of the C_i and new rate plan and will then update the sources with this information.

9. Source upon receiving the new rate value will update its rate plan and start sending the scheduled data packets with this rate value in order to suppress the congestion in the network.

In this section we have reviewed the envisaged transport layer protocol for WSN. In the next section we will discuss the simulation setup used for the extensive testing of the envisaged protocol and the possible outcomes of the testing.

IV. SIMULATION SETUP AND RESULTS

This section describes the simulation setup used for an extensive evaluation of the proposed transport layer protocol for WSN. Figure 2 and Table III shows the simulation setup and network parameters used for an extensive testing of the proposed protocol, where Sources A, B and J are multimedia sensor motes while Source C, D, E and H are generic scalar sensor motes. The priority of the multimedia data packet is higher than of the scalar data packet the source priority graph is shown in Table II and obeys the following priority rule:

$$\Pr [A] = \Pr [B] = \Pr [J] > \Pr [C] > \Pr [D] > \Pr [E] > \Pr [H]$$

where,

- Pr(A) = Priority of data packet from Source A,
- Pr(B) = Priority of data packet from Source B,
- Pr(C) = Priority of data packet from Source C,
- Pr(C) = Priority of data packet from Source D,
- Pr(C) = Priority of data packet from Source E,
- Pr(H) = Priority of data packet from Source H,
- Pr(J) = Priority of data packet from Source J.

TABLE II. SOURCE PRIORITY

Mote	A	B	C	D	E	H	J
Priority	5	5	4	3	2	1	5

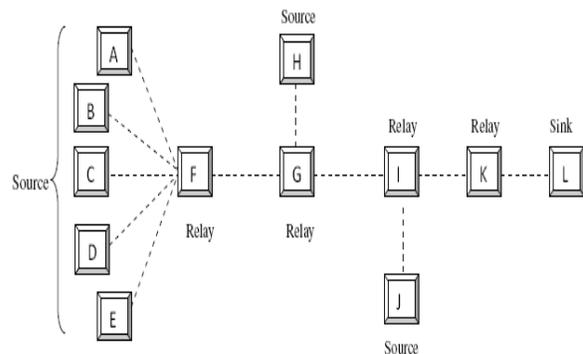


Figure 2. Network topology used for evaluation

The performance of the Prioritization Capability enabled Transport Layer Protocol is evaluated based on following criteria;

- [1] Performance of the Protocol without Prioritization enabled storages,
- [2] Performance of the Protocol with Prioritization enabled storages at Mote F, G and I,
- [3] Performance of the Protocol with Prioritization enabled storages at Mote I.

TABLE III. NETWORK PARAMETERS

Parameter	Values
Frequency	914e+6
Range	100 meters
Transport Protocol	Proposed protocol without priority enabled storage Proposed protocol with complete priority enabled storage Proposed protocol with distributed priority enabled storage
MAC	IEEE802.11 [11]
RX and CS Threshold	9.32665e-10W 8.393985e-10 W
Routing agent	Ad hoc On-Demand Distance Vector (AODV) [14]
Ifqlen (Queue length at MAC level)	200 packets
Energy Model	Yes: NS-2 based Energy computation [12]
CP Threshold	10
Mote Initial power	1000 W
Mote Idle power	712e-6 W
Mote Rx power	35.28e-3 W
Mote Tx power	31.32e-3W
Mote Sleep power	0.001W

A. Performance of the Protocol without Prioritization enabled storages

Table IV shows the average packet drop comparison for the selected network. As we can see from the Table IV that high priority sources A, B and J (having priority of 5) has a packet drop of 0, 11 and 0. Whereas the packets drop for the Sources C, D, E and H are 3, 6, 5 and 0 respectively. So on average around 7% of packet drop occurred for this network topology having no prioritized enabled storage at the intermediate motes and among all significant packet drop occurs for the case of Source B whose total packet drop count is 11.

TABLE IV. AVERAGE PACKET DROP COMPARISON

Mote	No of Packets Send	Received Packets	Packets Dropped
A	36	36	0
B	51	40	11
C	51	48	3
D	51	45	6
E	51	46	5
H	51	51	0
J	51	51	0
Total	342	317	25

The average throughput comparison for the same network is shown in Figure 3. From this it would be obvious that on average throughput rises to 0.435Mbps and then drops to 0.42Mbps when all sources are active for sending with almost equal contribution from every source.

Figure 4. shows the average mote power consumption for the selected network topology. From the graph it is obvious that on average every source consumes 0.2 Watt (W) of power whereas the intermediate and sink motes consumes 0.45 W and 0.25 W of power respectively. So for sources the successful transmission of every packet consumes 0.00556 W, 0.005 W, 0.004167 W, 0.0044 W, 0.004346 W, 0.00392 W and 0.00392 W respectively. Whereas the power (source) consumed in retransmission of the missing packets from Sources B, C, D

and E is 0.0431 W, 0.01176 W, 0.02358 W and 0.0196 W respectively.

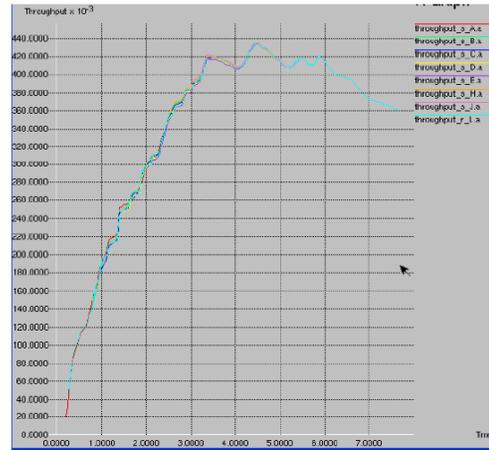


Figure 3. Average Throughput comparison

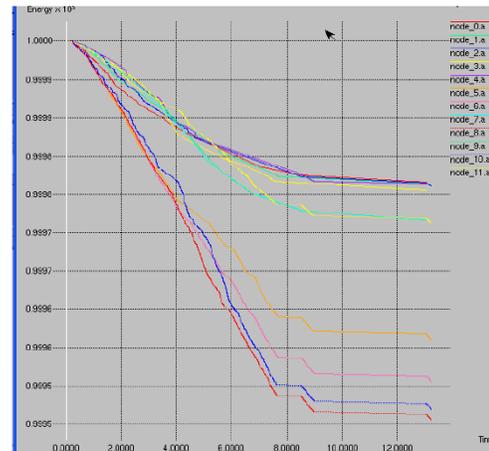


Figure 4. Average Mote power consumption comparison

B. Performance of the Protocol with Prioritization enabled storages at Mote F, G and I

Average packet drop comparison for the network with prioritized enabled storage at motes F, G and I is shown in Table V. From the data given in the Table V it is obvious that high priority sources A, B and J (having priority of 5) has a packet drop of 3, 2 and 0. Whereas the packets drop for the low prior Sources C, D, E and H are 1, 5, 8 and 0 respectively. So on average around 6% of total packet drop (send =327 , received 308) occurred for this network topology having complete prioritized enabled storage at the intermediate motes and among all significant packet drop occurs for the case of Source E whose total packet drop count is 8.

The average throughput comparison for the same network is shown in Figure 5. From this it would be obvious that on average throughput rises to 0.42Mbps and then drops to 0.39Mbps when all sources are active for sending with almost

equal contribution from every source except from Source A and B.

TABLE V. AVERAGE PACKET DROP COMPARISON

Mote	No of Packets Send	Received Packets	Packets Dropped
A	42	39	3
B	30	28	2
C	51	50	1
D	51	46	5
E	51	43	8
H	51	51	0
J	51	51	0
Total	327	308	19

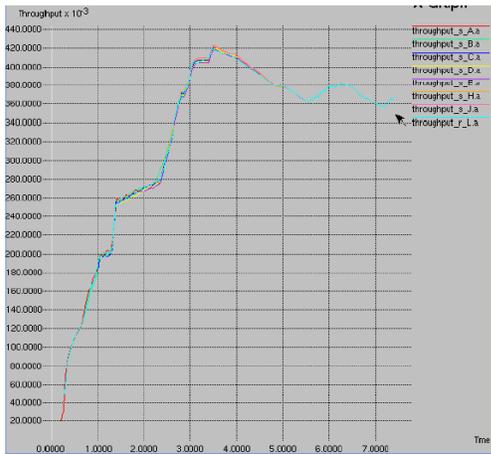


Figure 5. Average Throughput comparison

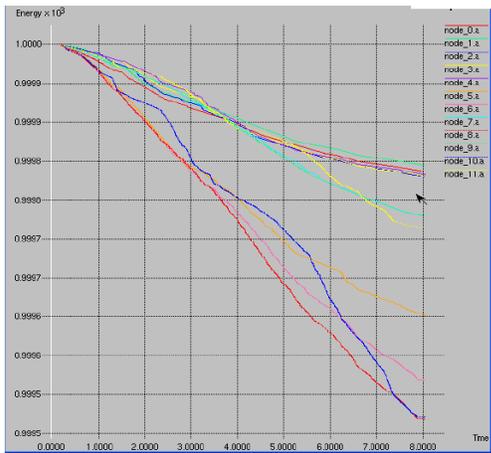


Figure 6. Average Mote power consumption comparison

The Average mote power consumption for the selected network topology with prioritized enabled storage is shown in the Figure 6. From the Figure 6 it is obvious that on average every source consumes 0.2 W of power while the intermediate and sink motes consumes 0.5 W and 0.28 W of power respectively which is slightly higher than the case having no prioritized enabled intermediate storage discussed above. So for sources the successful transmission of every packet

consumes 0.00518 W, 0.00714 W, 0.004 W, 0.00434 W, 0.004651 W, 0.00392 W and 0.00392 W respectively. Whereas the power (source) consumed in retransmission of the missing packets from Sources A, B, C, D and E is 0.014286 W, 0.0133 W, 0.00392 W, 0.0196 W and 0.0313 W respectively.

C. Performance of the Protocol with Prioritization enabled storages (Buffdata Agent) at Mote I

Now we discuss the case of distributed prioritized enabled storage where mote I serve the purpose of intermediate storage. This case is similar to the second case discussed above but with only one mote is used for loss recovery and prioritized forwarding while all other middle motes act as simple relay motes. The average packet drop comparison for this network topology is given in the Table VI. From the Table VI it is obvious that every mote except mote A contribute the same amount of load information and high prior Sources A, B and J has 0% data loss which is quite a significant improvement as compared to the above two cases. Also the low prior Sources C, D, E and H has data loss of 4, 1, 3 and 4 packets respectively. Overall this configuration exhibits a significant improvement with total data loss of only 3% as compared to 7% and 6% for without and complete prioritized enabled storage cases discussed above.

TABLE VI. AVERAGE PACKET DROP COMPARISON

Mote	No of Packets Send	Received Packets	Packets Dropped
A	45	36	0
B	51	40	0
C	51	48	4
D	51	45	1
E	51	46	3
H	51	51	4
J	51	51	0
Total	351	339	12

The throughput comparison for the network is shown in the Figure 7. For this network topology the peak throughput of 0.465Mbps and minimum of 0.445Mbps is achieved with all sources contributed equally (51 data packets except for 45 data packets for Source A). Thus having single storage mote that stores information from all its child relay motes as well as source motes exhibits high throughput in comparison to have no or multiple storage motes. The high data loss for “without” case is quite obvious where as for multiple motes the reason of high loss is due to the fact that since each intermediate mote serve the purpose of storing so during peak data transmission (when all sources contributes) it takes long for every mote to store and rearrange the packets in prioritized order and then prioritized retrieval of the packets for forwarding. As every intermediate mote in complete prioritized enabled storage case perform the similar thing and also the TTL value of every source packet is fixed so there is a chance for some packets to be dropped on their way towards sink because of this long delay.

The Average mote power consumption for the selected network topology with prioritized enabled storage at intermediate mote I is shown in the Figure 8. From the Figure 8 it is obvious that on average every source consumes 0.2 W of power while the intermediate and sink motes consumes 0.45 W

and 0.3 W of power respectively. So for sources the successful transmission of every packet consumes 0.0044 W, 0.00392 W, 0.00425 W, 0.004 W, 0.004167 W, 0.00425 W and 0.00392 W respectively. Whereas the power (source) consumed in retransmission of the missing packets from Sources C, D, E and H is 0.01568 W, 0.00392 W, 0.01176 W and 0.01568 W respectively.

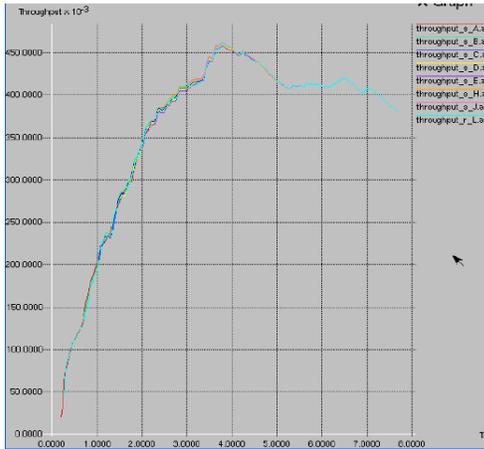


Figure 7. Average Throughput comparison

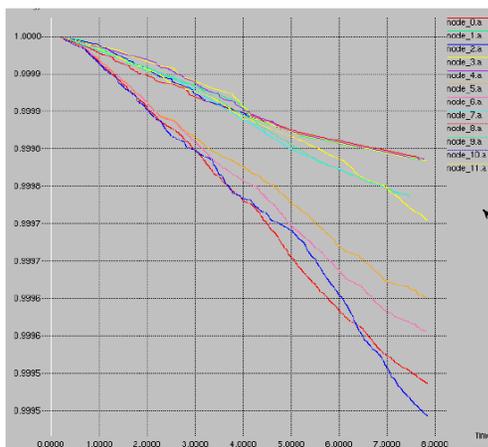


Figure 8. Average Mote power consumption comparison

V. CONCLUSIONS

The existing WSN transport layer protocols are designed for either congestion control or ensuring the data reliability parameter of the design, and none of them simultaneously handles the congestion control and data reliability except for STCP. Source prioritization is also another important aspect which is seldom addressed in conjunction with congestion control and data reliability. In the following paper we have discussed a transport layer protocol scheme for WSN which guarantees congestion control, data reliability and source data priority simultaneously. Stochastic framework for computing the E-2-E delay of source data packet involving the congestion index parameter will be used in finding the updated rate plan for sources and prioritized forwarding of the data packets.

Distributed prioritized enabled storage (which stores data packet for some defined interval of time and enables prioritized forwarding of the data packets based on the source priority and its TTL value) is used to achieve the E-2-E reliability of the data packets thus ensuring the application specific QoS. Distributed prioritized enabled storage and data forwarding showed better results in-terms of average throughput, average data packet drop in comparison to “without” and complete prioritized enabled storage and forwarding. In the next step we will incorporate the crosslayer feature to the existing design and this sets our future research direction.

REFERENCES

- [1] I.F. Akyildiz, T. Melodia, and K.R. Chowdhury, “A Survey on wireless multimedia sensor networks,”*Computer Networks (Elsevier) J.*, vol. 51, pp. 921-960, 2007.
- [2] TCP, <http://www.ibiblio.org/pub/docs/rfc/rfc793.txt>, Aug 2008.
- [3] UDP, <http://tools.ietf.org/html/rfc768>, Aug 2008.
- [4] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, “CODA: Congestion detection and avoidance in sensor networks,” in *Proceedings of ACM Sensys’03*, Nov. 5-7, 2003, Los Angeles, California, USA.
- [5] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz, “ESRT: Event-to-sink reliable transport in wireless sensor networks,” *ACMMobihoc’03*, Jun. 1-3, 2003, Annapolis, Maryland, USA.
- [6] F. Stann and J. Heidemann, “RMST: Reliable data transport in sensor networks,” *IEEE SNPA’03*, May 11, 2003, Anchorage, Alaska, USA.
- [7] 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.
- [8] R. Anantharangachar, C. Reddy and D. Ranganathan, “GARUDA: A Case Study of Service Oriented Application Integration Framework”, HP India.
- [9] Y. G. Iyer, S. Gandham, and S. Venkatesan, “STCP: A generic transport layer protocol for wireless sensor networks,” *IEEE ICCCN 2005*, Oct. 17-19, San Diego, California, USA.
- [10] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang “TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links”, *ACM SIGMOBILE ’01*, Rome, Italy, 2001
- [11] IEEE P802.11, Main General Info Page, Aug 2008.
- [12] The Network Simulator — NS-2, <http://www.isi.edu/nsnam/ns/index.html>, July 2008.
- [13] F.L. Lewis, L. Xie and D. Popa, “Optimal and Robust Estimation with an introduction to Stochastic Control Theory”, Second Edd, CRC Press, 6000 Broken Sound Parkway, NW (2008). ch 1-4, pp. 3-258.
- [14] C.E. Perkins, E.B. Royer and S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing”, RFC 3561, July 2003.
- [15] A.Sharif, V. Potdar, A.J.D Rathnayaka and T. Dillon, “Reliable Transport Protocol for WSN with Congestion Avoidance using Stochastic Control Framework”, Submitted to *ETRI Journal*, Dec 2009.