Improved Transport Layer Protocol for Congestion Control in Wireless Sensor Networks

Praveen Budhwar
M.E. Student, Department of Computer Science and Engineering
PEC University of Technology, Chandigarh, India

ABSTRACT
In wireless sensor networks, congestion occurs when multiple sensor nodes try to transmit data at the same time. Congestion may lead to packet drops and increased delay which in turn leads to lower throughput and delivery ratio. Thus congestion is an important aspect that should be considered while designing transport layer protocols for sensor networks. In this paper, we present an improved transport layer protocol for congestion control in wireless sensor networks. This protocol uses buffer size to detect congestion. Congestion is notified by overriding ACK packet with congestion information. Congestion is mitigated by adjusting transmission rates of congested nodes' neighbors to optimal values. Simulation results show that this improved protocol provides an efficient congestion control mechanism and thus leads to increased throughput and delivery ratio with reduced delay and less energy loss as compared to CTCP.

Keywords
Congestion control, Reliability, Transport layer, Wireless sensor networks.

1. INTRODUCTION
Wireless sensor network is an emerging class of networks applicable to habitat monitoring, structural monitoring, environment monitoring or military surveillance etc [1]. Wireless sensor networks are comprised of thousands of sensor nodes with one or more sinks. Sensor networks can have one or more source nodes depending upon the requirement of the application. Transmission of packets in these networks poses numerous challenges because of limited processing power, restrained memory and resource constraint nature of sensor nodes [2].

The source nodes of sensor networks generate upstream traffic which can be classified into two classes: event driven and periodic [3]. In event driven flows, sudden flow of data is triggered from the event area towards the sink on the occurrence of an event. This sudden flow of data causes congestion in the network. In periodic flows, data is periodically generated from the source nodes. This traffic is sent to sink nodes periodically. In periodic flows, congestion is most likely to occur when multiple source nodes try to send data at the same time period.

Transport layer protocols can be classified as: protocols that provide only congestion control, protocols that provide only reliability and protocols that provide both congestion control and reliability. Collaborative Transport Control Protocol (CTCP) [4, 5] is a reliable data delivery protocol which provides both end-to-end reliability and congestion control. CTCP is an excellent protocol for scenarios with single source node and less number of sensor nodes. But the average delivery ratio of CTCP is not good for increased number of nodes and multiple sources. CTCP has no proper rate control mechanism. It is up to the nodes to maintain their own transmission rates. Thus an improved protocol is proposed which provides an efficient mechanism for congestion control. The improved protocol uses buffer size to detect congestion. When congestion occurs, implicit congestion notification information is embarked in the ACK packet. It uses the mechanism of rate adjustment to optimal value for congestion control.

The rest of the paper is structured as follows: section II presents the related work. Section III describes the proposed scheme. Section IV presents performance evaluation. At the end, section V draws conclusion.

2. RELATED WORK
A number of transport layer protocols have been proposed which provide both reliability and congestion control as shown in Figure 1. Following are some of the transport layer protocols which provide either congestion control or both reliability and congestion control.

STCP [6] Sensor Transmission Control Protocol is a generic protocol where most of the functionalities are implemented at the sink. STCP offers both congestion control and reliability. It makes use of buffer occupancy for congestion detection. Congestion is notified implicitly and mitigated via rate adjustment and traffic redirection. STCP provides controlled variable end-to-end reliability by using NACK for continuous data flow and ACK for event driven data flow. STCP has several drawbacks such as: i) it makes heavy use of source node caching, due to which it becomes impractical for many wireless senor network applications, ii) the ACK based reliability mechanism for event driven data flows is very costly in terms of delay and memory, iii) STCP requires strict clock synchronization between the sink and the sensor nodes, which can be a performance issue with STCP

CODA [7] Congestion Detection and Avoidance is an energy efficient congestion control protocol. It uses buffer occupancy and channel status for congestion detection. When buffer occupancy crosses the threshold, it samples the channel to detect congestion. It can handle both transient and persistent congestion using open loop hop-by-hop backpressure and closed loop multi source regulation respectively. CODA provides a mechanism of implicit priority and maintains acceptable fidelity. CODA has several drawbacks such as: i) it uses additive increase multiplicative decrease (AIMD) [8] for rate adjustment which leads to packet losses, ii) AIMD is biased towards sources closer to sink, iii) CODA is not scalable as it shows poor performance if the data rate and number of source nodes are increased, iv) sensor nodes may deplete extra energy in sampling the channel periodically.

CTCP [4, 5] Collaborative Transport Control Protocol supports both reliability and congestion control. It provides controlled variable packet reliability.

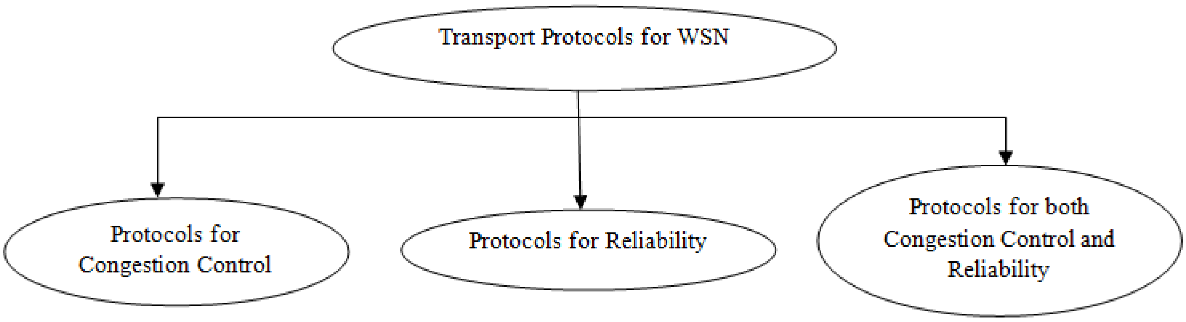

Fig 1: Transport Layer Protocols

It achieves hop-by-hop reliability by sending single and double ACK to each sensor node for reliability level 1 and level 2 respectively. CTCP has proposed a mechanism of congestion control which uses buffer overflows and transmission error losses for congestion detection. STOP and START messages are used to notify neighboring nodes of congestion. If a node receives STOP message from a node, it stops sending packets to that node. When the buffer is empty or below the threshold, node sends START message so that neighboring nodes can start sending packets to this node. CTCP has several drawbacks, viz. i) it has no specific mechanism of rate control, ii) the use of STOP and START messages may not eliminate congestion precisely, iii) CTCP also has control overhead in terms of single and double ACKs to provide reliability.

IFRC [9] Interference-Aware Fair Rate Control is a distributed rate allocation scheme which provides congestion control. It uses buffer size to detect congestion. It makes use of two buffer thresholds. It communicates the congestion information via overhearing. IFRC is most relevant to our work as it achieves efficient transmission rate to avoid congestion. . IFRC uses AIMD scheme to adjust outgoing rate to control congestion at each link. IFRC assigns the data rate which is lowest among the interfering neighbors of the congested node. IFRC has several drawbacks, viz. i) its scheme of collecting rate information from neighbors increases processing overhead and energy consumption, ii) IFRC requires specific parameter tuning for stability, iii) it reduces packet drops by reducing the throughput.

3. PROPOSED SCHEME

In the following three situations, the average delivery ratio and throughput of CTCP is reduced and delay in the network is increased:

- Increasing the number of sources into the network
- Increasing the number of sensor nodes into the network
- Reducing the time interval between the generation of two consecutive packets

The congestion control mechanism used by CTCP makes use of STOP and START signaling messages to manage congestion. Buffer overflows are used to detect congestion. If some node's buffer overflows, it sends a STOP message to its neighboring nodes. Upon reception of this STOP message neighboring nodes stop sending messages to the congested node. When the node's buffer goes below the threshold or when the buffer is empty, it sends a START message to the neighbors and neighbors start sending packets to that node. This mechanism of congestion control does not make use of rate control algorithm. The use of STOP and START messages reduces the average delivery ratio in case of multiple sources and for increased number of sensor nodes. This congestion control mechanism also introduces delay in the network as no node can send packets to the nodes which have sent STOP messages. Also, the use of explicit messages to notify congestion consumes network energy so we need a congestion control mechanism which does not use explicit control messages instead uses rate control to optimal values. To calculate new data rate in case of congestion, the proposed solution makes use of the time interval between the generation of consecutive packets. When the node's buffer size goes below the threshold, the node piggybacks it new state in the ACK packet so that the neighboring nodes can adjust their data transmission rates towards that node. This new method of congestion mitigation is better than the old method used by CTCP because CTCP reduces the transmission rate to zero whenever a node sends STOP message instead the transmission rate should be calculated using the exact buffer occupancy. The new method uses the exact buffer occupancy to calculate the transmission rate. This new transmission rate gives better results in terms of average delivery ratio than the zero transmission rate of the old congestion control method of CTCP.

Figure 2 and 3 show the flow charts for level 1 and level 2 of the improved protocol. Both reliability and congestion control are shown here.

Pseudo code for the improved protocol is presented in the specifying language CCS (The Calculus of Communicating Systems) of Robert Milner [10]. First the notations for the algorithm are given followed by the actual algorithm for both levels.

Notations:

Q_A: Buffer size of node A

Q_B: Buffer size of node B

Q_C: Buffer size of node C

BS: Buffer Size=15

Timer_rate=30

B!dt: sent of dt to B

A?dt: reception of dt sent by A.

Reliability Level 1 code in CCS:

Level 1 = A|||B|||Timer

A =!dt.Wait_ack_A

Wait_ack_A =?ack(A).Check_cong_A

Check_cong_A=if Q_B>=T then Rate_control_A else Rate_set_A

Rate_control_A=Timer_rate_A.A

Timer_rate_A=((BS-Q_B/BS)+0.5)/Timer_rate_A.Rate_control_A

Rate_set_A=Timer_rate.A

B =?dt.!ack.!dt(C).!starttimer.Wait_ack_C

Wait_ack_C=?ack(C).!reset_timer.Check_cong_B+?t_out.!dt(C).!starttimer.Wait_ack_C

Check_cong_B=if Q_C>=T then Rate_control_B else Rate_set_B

Rate_control_B=Timer_rate_B.B

Timer_rate_B=((BS-Q_C/BS)+0.5)/Timer_rate_B.Rate_control_B

Rate_set_B=Timer_rate.B

Timer =?starttimer.(!t_out.Timer+?reset_timer.Timer)

Reliability Level 2 code in CCS:

Level 2 = A|||B|||Timer

A = B!dt.Wait_ack1_A

Wait_ack1_A = B?ack.Check_cong_A.Timer!start.Wait_ack2_A

Check_cong_A=if Q_B>=T then Rate_control_A else Rate_set_A

Rate_control_A=Timer_rate_A.A

Timer_rate_A=((BS-Q_B/BS)+0.5)/Timer_rate_A.Rate_control_A

Rate_set_A=Timer_rate.A

Wait_ack2_A = B?ack.Timer!reset.A+ Timer?tout.B!dt.Timer!start.Wait_ack2_A

B = A?dt.C!dt.A!ack.Check_cong_B.Wait_ack1_B

Check_cong_B=if Q_C>=T then Rate_control_B else Rate_set_B

Rate_control_B=Timer_rate_B.B

Timer_rate_B=((BS-Q_C/BS)+0.5)/Timer_rate_B.Rate_control_B

Rate_set_B=Timer_rate.B

Wait_ack1_B =C?ack.A!ack.T imer!start.W ait_ack2_B

Wait_ack2_B = C?ack.Timer!reset.B+timer?t out.C!dt.timer.start.Wait_ack2_B

Timer =?start(!t out.timer+?reset.timer)

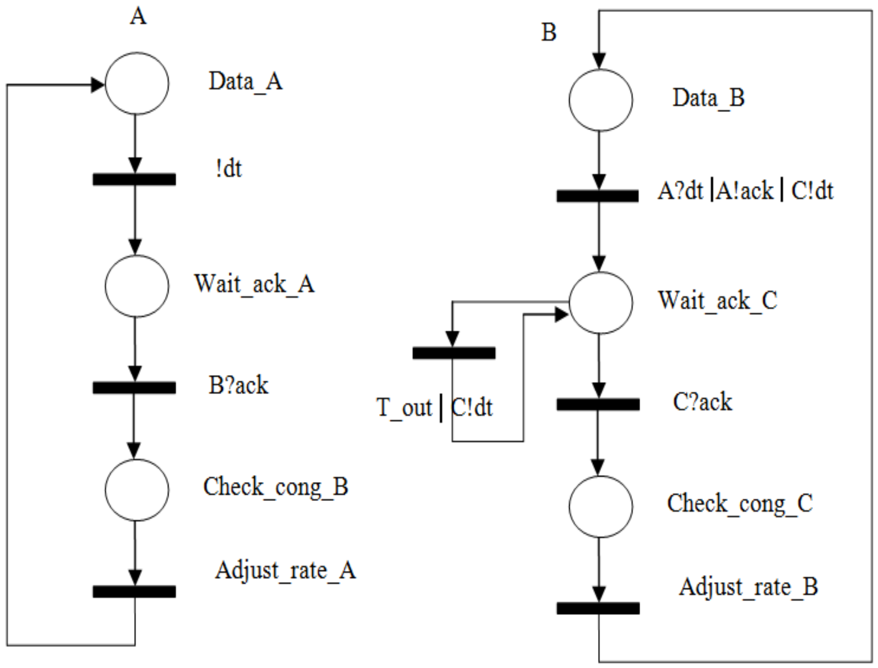

Fig 2: Level 1 flowchart

4. PERFORMANCE EVALUATION

4.1 Simulation Environment

Performances of proposed method are shown by simulation in this section. To understand the behavior of existing and improved protocol simulations are performed using TinyOS simulator TOSSIM [11]. The simulation scenario consists of grid topology where multiple source nodes send traffic packets to the sink node and form many-to-one upstream traffic. Simulation parameters are listed below:

Table 1 Simulation Parameters

Topology used	Grid
No. of nodes	50, 100, 150, 200, 250
Simulation time	1000 sec
Packet size	216
Buffer size	15
Retransmission limit	10
MAC protocol	CSMA MAC
Routing protocol	MintRoute

Performance comparisons of proposed protocol with existing CTCP protocol on average delivery ratio, throughput, average delay and energy loss are provided as follows:

4.2 Average Delivery Ratio

Figure 4 and 5 compares the two schemes on the basis of average delivery ratio for level 1 and level 2 respectively. It can be seen that the proposed protocol performs better than the existing protocol because proposed protocol makes use of rate adjustment to optimal value for congestion control. Figure 6 shows that level 2 performs better than level 1 for both schemes because level 2 is more reliable than level 1.

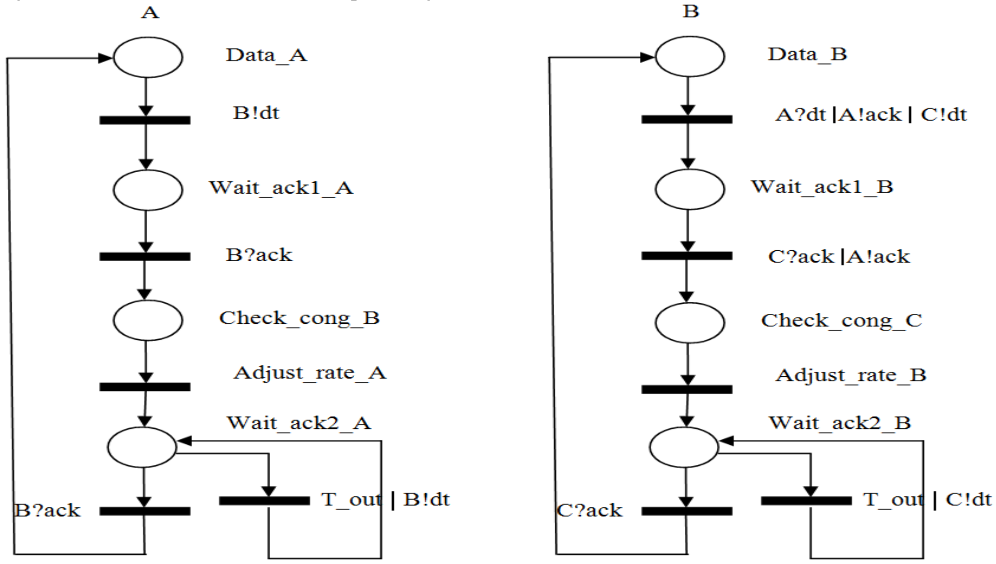

Fig 3: Level 2 Flowchart

4.3 Throughput

Figure 7 and 8 compares the two schemes on the basis of throughput for level 1 and 2 respectively. The proposed protocol performs better than the existing protocol because proposed protocol makes use of multiple sources which generate large number of packets as compared to the single source of existing protocol. Thus the throughput is better for proposed scheme. Figure 9 shows that the throughput of level 1 is greater than the throughput of level 2 because in level 2, a node has to wait twice for two ACKs which reduces the overall number of packets to be transmitted.

4.4 Average Delay

Figure 10 and 11 show that there is no significant difference in the average delay faced by both schemes. In existing scheme, delay occurs because nodes have to wait for the START message to resume transmission. While in proposed scheme, delay is introduced due to the multiple sources which transmit packets simultaneously. Figure 12 shows that Level 2 shows higher delay as compared to level 1 because of the doubled waiting time for two ACKs.

4.5 Energy Loss

Figure 13 and 14 show that proposed scheme has lower energy loss than the existing scheme. This is because the congestion control mechanism in proposed scheme makes use of the rate adjustment policy which handles congestion effectively due to which fewer packets are dropped in proposed scheme and hence less energy loss. Figure 15 shows that Level 2 shows less energy loss as compared to level 1. This is because level 2 is more reliable than level 1. Level 2 ensures successful delivery of packets with the help of double ACKs so it has fewer packet drops and hence less energy loss.

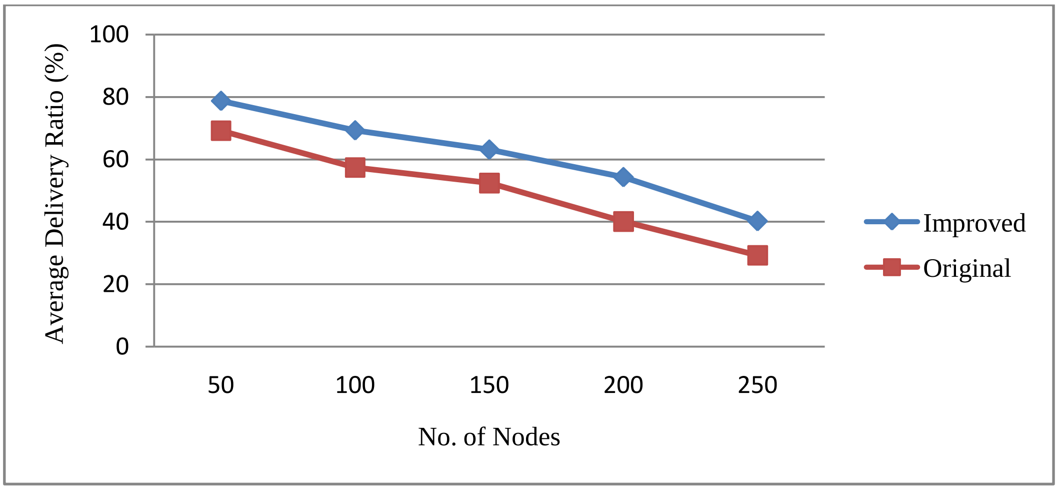

Fig 4: Average delivery ratio for Level 1

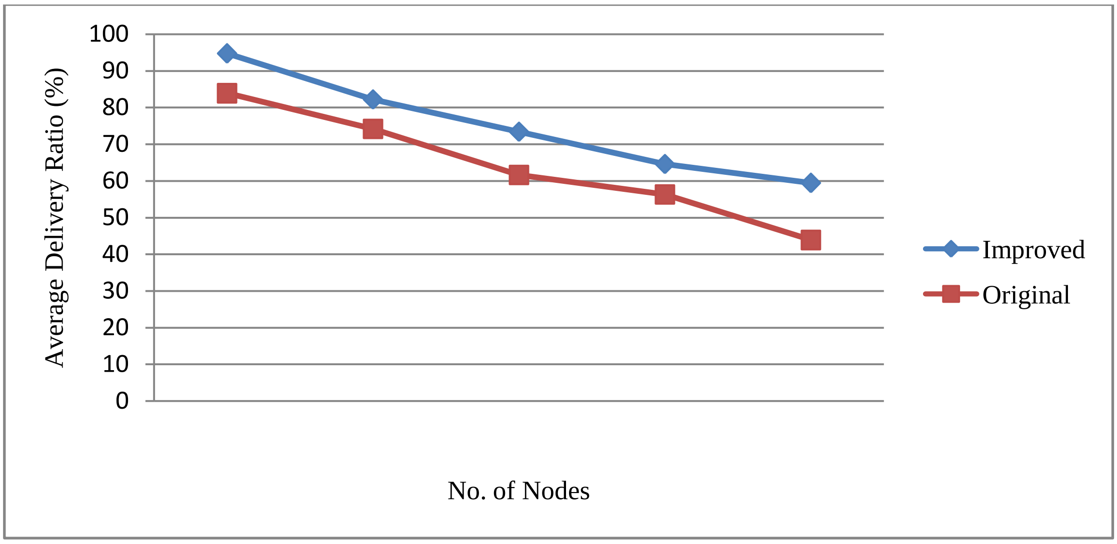

Fig 5: Average delivery ratio for Level 2

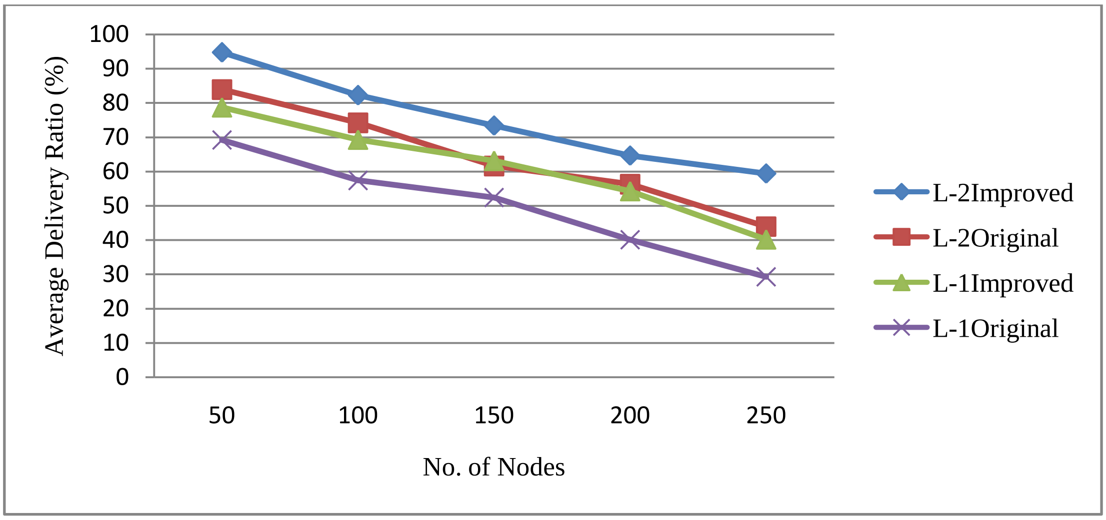

Fig 6: Comparison of average delivery ratio for Level 1 and Level 2

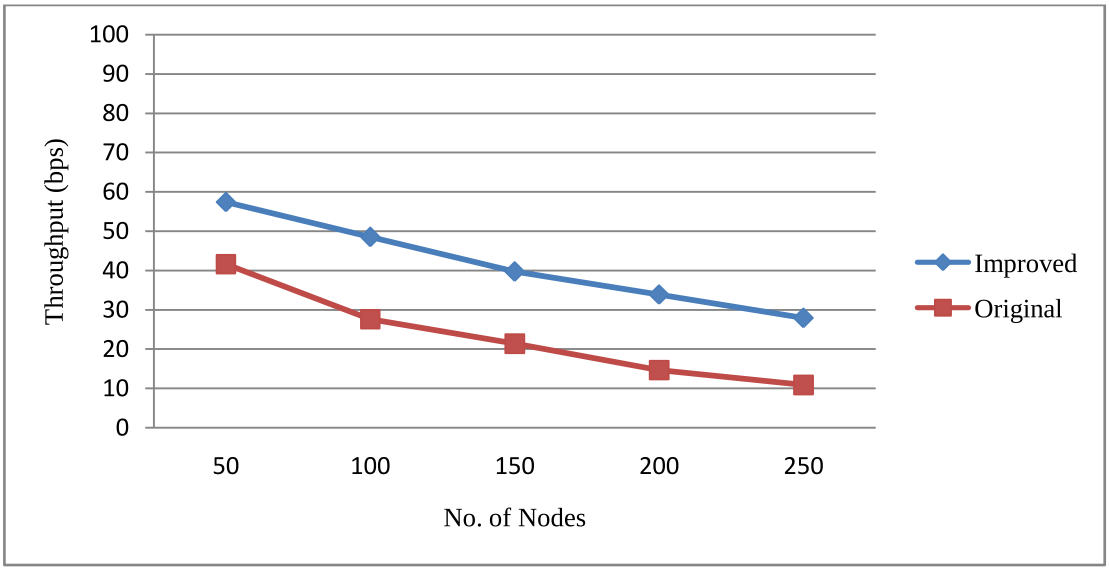

Fig 7: Throughput for Level 1

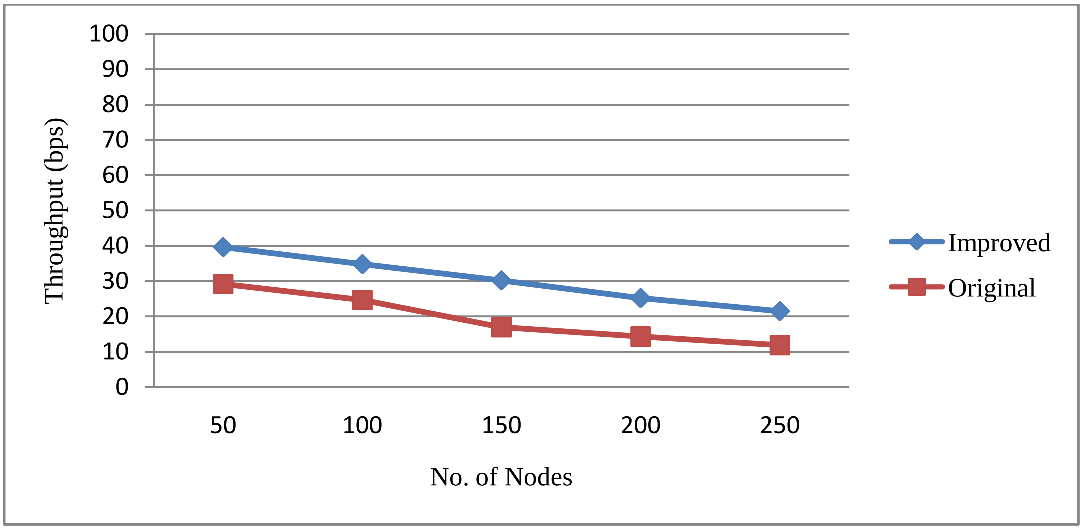

Fig 8: Throughput for Level 2

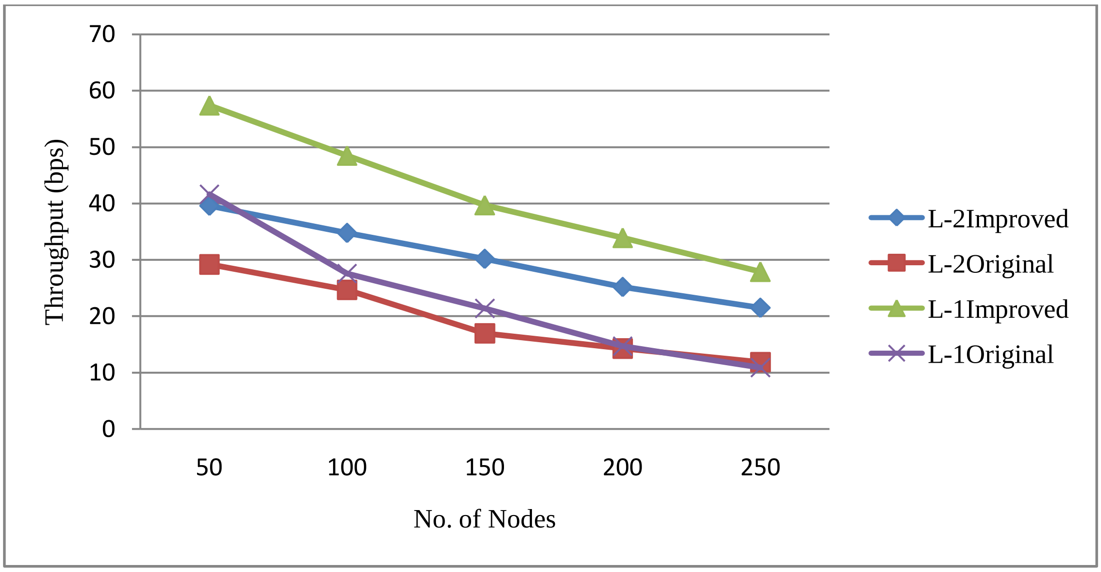

Fig 9: Comparison of throughput for Level 1 and Level 2

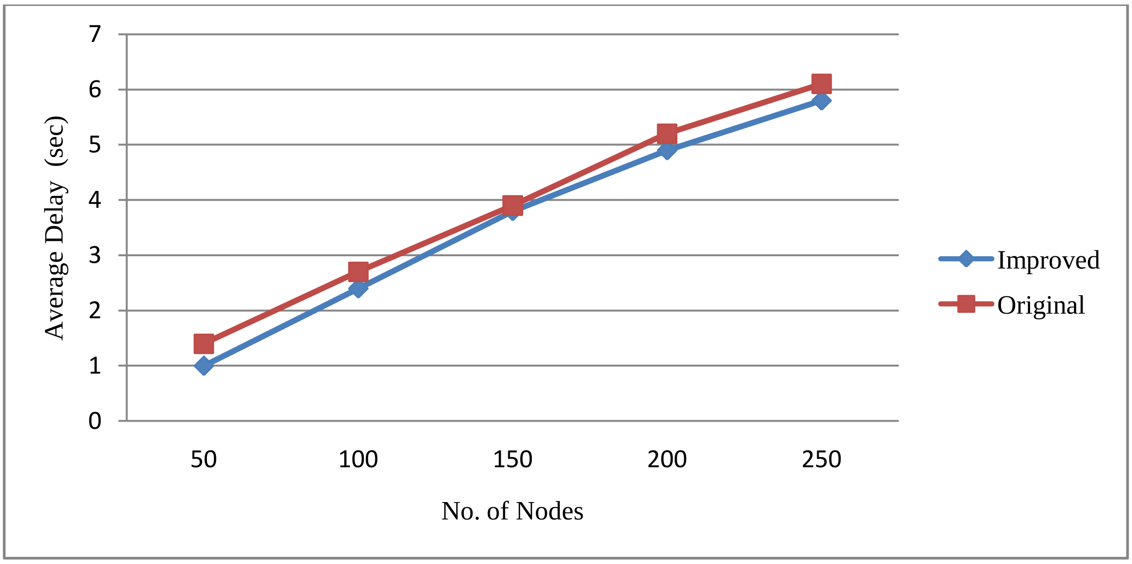

Fig 10: Average delay for Level 1

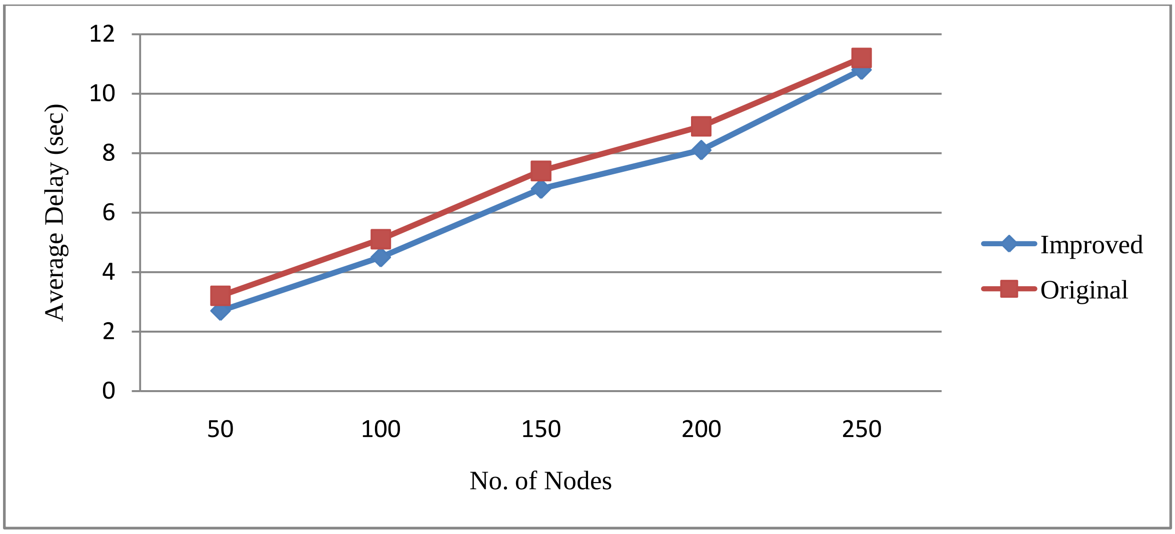

Fig 11: Average delay for Level 2

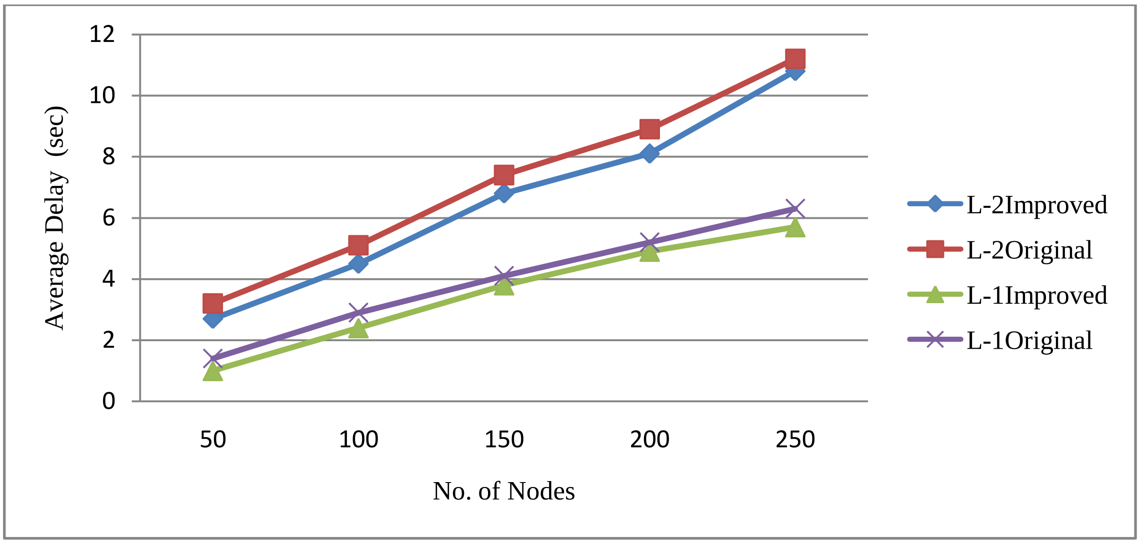

Fig 12: Comparison of average delay for Level 1 and Level 2

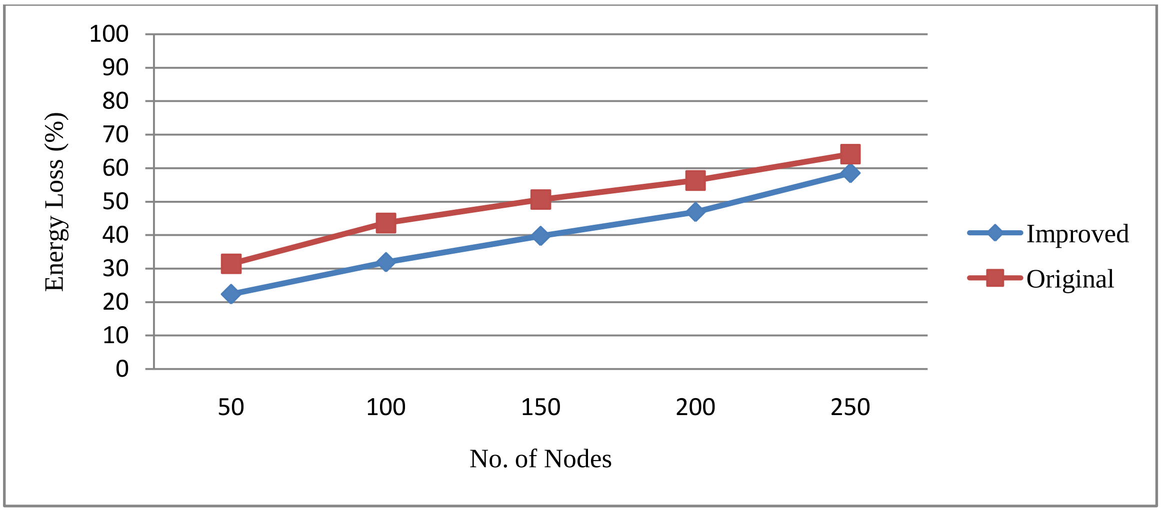

Fig 13: Energy loss for Level 1

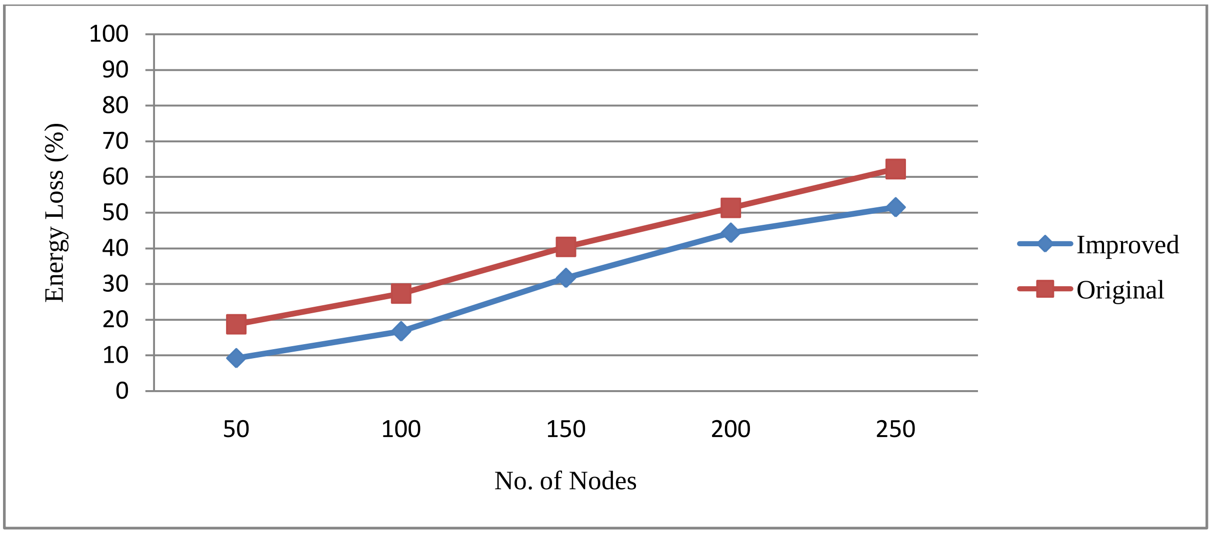

Fig 14: Energy loss for Level 2

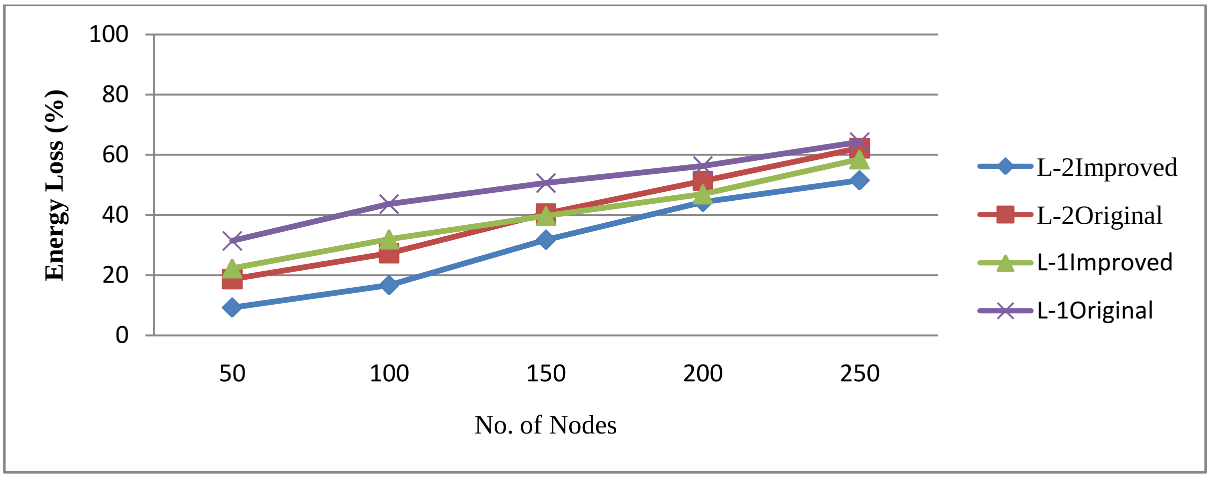

Fig 15: Comparison of energy loss for Level 1 and Level 2

5. CONCLUSION

In this paper, a transport layer protocol is proposed which is an improvement over the existing CTCP protocol. It provides an efficient mechanism for congestion control. This improved protocol makes use of buffer size to detect congestion. Congestion is notified implicitly and rate adjustment to optimal value is used to adjust the rate of upstream traffic towards the congested node. The improved protocol shows a tremendous improvement over the existing CTCP protocol. Simulation results show that the improved protocol is better than the existing protocol in terms of average delivery ratio, throughput and energy loss. This protocol does not ensure fairness for multiple sources. In future, we will improve this protocol to provide fairness among the traffic of different sources.

6. REFERENCES

[1] I. Mahgoub, and M. Ilyas, Smart Dust: Sensor network applications, architecture, and design, CRC Press, New York, 2006.

[2] J. Yick, B. Mukherjee and D. Ghosal, Wireless sensor network survey, Computer Networks, vol. 52 no. 12, pp. 2292-2330, 2008.

[3] B. Hull, K. Jamieson and H. Balakrishanan, Mitigating congestion in wireless sensor networks, In Proceedings of ACM Sensys'04, pp. 134-147, 2004.

[4] Giancoli, F. C. Jabour and A. C. P. Pedroza, Collaborative transport control protocol, In Proceedings of IEEE International Conference on Computer and Electrical Engineering, IEEE Computer Society, pp. 373-377, 2008.

[5] Giancoli, F. C. Jabour and A. C. P. Pedroza, CTCP: reliable transport control protocol for sensor networks, In Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP '08), pp. 493–498, 2008.

[6] Y. G. Iyer , S. Gandham and S. Venkatesan, STCP: a generic transport layer protocol for wireless sensor network, In Proceedings of IEEE International Conference on Computer Communications and Networks, pp. 449-454.

[7] Y. Wan, A. T. Campbell and S. .B. Eisenman, CODA: congestion detection and avoidance in sensor networks, In Proceedings of 1st ACM Conference on Embedded Networked Sensor Systems: (Sensys'03), pp. 266-279, 2003.

[8] Chiu and R. Jain, Analysis of the increase and decrease algorithms for congestion avoidance in computer networks, Computer Networks and ISDN Systems, vol. 17 no. 1, pp. 1-14, 1989.

[9] S. Rangwala, R. Gummadi, R. Govindan and K. Psounis, Interference aware fair rate control in wireless sensor networks, In Proceedings of ACM SIGCOMM, pp. 63-74, 2006.

[10] R. Milner, A Calculus of Communicating Systems, Springer, Verlag, 1980.

[11] P. Levis, N. Lee, M. Welsh and D. Culler, TOSSIM: Accurate and scalable simulation of entire tinyos applications. In Proceedings of the ACM Symposium on Networked Embedded Systems, pp. 1-17, 2003.