IMPACT OF CONTENTION WINDOW ON CONGESTION CONTROL ALGORITHMS FOR WIRELESS ADHOC NETWORKS

B.Nithya, C.Mala and Vijay Kumar B

Department of Computer Science and Engineering, NIT, Tiruchirappalli-620015
{nithya,mala}@nitt.edu,vijay.chaos@gmail.com

ABSTRACT

TCP congestion control mechanism is highly dependent on MAC layer Backoff algorithms that predict the optimal Contention Window size to increase the TCP performance in wireless adhoc network. This paper critically examines the impact of Contention Window in TCP congestion control approaches. The modified TCP congestion control method gives the stability of congestion window which provides higher throughput and shorter delay than the traditional TCP. Various Backoff algorithms that are used to adjust Contention Window are simulated using NS2 along with modified TCP and their performance are analyzed to depict the influence of Contention Window in TCP performance considering the metrics such as throughput, delay, packet loss and end-to-end delay.

KEYWORDS

TCP, Congestion Control, MAC layer, Backoff, Wireless Adhoc Network, NS2.

1. INTRODUCTION

Wireless Ad Hoc Networks have become increasingly popular in many applications such as battlefield communication, disaster rescue, entertainment, inimical environment monitoring, telemedicine, outdoor business environments. Due to the nature of wireless communication, there are several constraints, such as topology dynamism, limited link bandwidth and quality, variations in node capabilities and energy constrained operation, which make the adverse impact on TCP performance. TCP – the Internet's transport layer, connection oriented, reliable, byte stream transport protocol- was designed for the traditional networks having wired links and stationary hosts. It assumes a packet loss only due to congestion by noticing missing acknowledgements and as a result, slows down the transmission rate dramatically to mitigate the congestion[1]. However, this assumption is no longer valid in wireless networks as packet losses may also due to channel errors, medium contention, and route failures as well as congestion. So it is highly required to have efficient congestion and contention control mechanism in ad hoc networks to achieve reliable transmission and enhance end-to-end throughput.

The greedy properties of TCP and the coupling of congestion and contention lead to instability in TCP throughput and excessively long delay. Until the packet loss is detected, TCP sender will exponentially/ additively increase the congestion window. Consequently the sending rate of TCP sources is increased, the packets are cumulated along the transmission path. Since the wireless medium is shared by all nodes in a neighbourhood, the congestion is not only a phenomenon

Sundarapandian et al. (Eds): CoNeCo,WiMo, NLP, CRYPSIS, ICAIT, ICDIP, ITCSE, CS & IT 07,
pp. 159–170, 2012. © CS & IT-CSCP 2012 DOI : 10.5121/csit.2012.2414

related to one pair of transmitting – receiving nodes, but also one to all nodes in the neighbourhood[2]. When congestion occurs and queues build up, congestion and contention will become severe, leading to significant reduction in end-to-end throughput for any communication pair in the neighbourhood.

In highly congested condition, the quick changes such as decreasing contention window after successful transmission or doubling contention window after a collision, cause bandwidth wastage and high variation in delay. Hence the intelligent and cooperative congestion and contention algorithm is a must to provide end-to-end QoS at packet level and at flow level.

The rest of the paper is organized as follows. Section 2 gives various schemes presented in the literature which incorporates MAC layer information to improve the TCP performance in wireless environment and presents the detailed method to calculate the Retransmission Time-out and focuses the existing Modified TCP congestion control algorithm. Section 3 shows the various situations where the intelligent and collaborative congestion and contention algorithm are highly recommended and the Finite State Machine definition is also given for Binary Exponential Backoff (BEB), Exponential Increase Exponential Decrease(EIED), Double Increment Double Decrement(DIDD), Linear/Multiplicative Increase and Linear Decrease(LMILD), Binary Negative Exponential Backoff (BNEB) algorithms. Section 4 analyzes the performance of modified TCP congestion control method with various backoff algorithms using NS2.34 under different performance metrics. Finally the section 5 concludes the paper.

2. RELATED WORK

The contention based congestion control method is proposed in [3] which uses local information such as successful transmissions and collisions undergone by each station. By monitoring number of entries into the Backoff (BO) state, the station detects the collision rate. If the station gets into BO state frequently, then it will suffer serious congestion. To indicate the degree of congestion, the metric Backoff State Ratio (BSR) is determined based on the number of node's entry into the BO state. By comparing BSR ratio with threshold values, the packet is either sent out or dropped or marked at the MAC layer to allow the ECN enhanced TCP flows. From the simulation results, it has been shown that [3] this method increases the TCP performance in static wireless environment by providing an early sign of network congestion.

Based on the channel busyness ratio, Wireless Congestion Control Protocol (WCCP) is proposed in [4] to enhance the TCP performance in static multi hop ad hoc networks. Each intermediate station monitors and allocates channel resources to the passing flows and modifies the feedback field of the data packets according to its measured channel busyness ratio. The destination uses this information to calculate its congestion window size which is sent along with ACK to the sender. Finally, the sender adjusts the sending rate accordingly. Since the usable congestion window of the TCP sender is determined by channel utilization status at each bottleneck node along the transmission path, it has been shown that [4] WCCP significantly enhances the TCP performance by solving the starvation problem.

The joint congestion and contention window control scheme is designed in [5] to help the TCP to differentiate congestion losses from transmission errors to take proper remedial actions. The snoop agent is implemented in both the transport layer and the MAC layer, which monitors every packet that passes through the connection in either direction. The snoop agent caches TCP packets sent from the TCP source and keeps track of the last ACK sent from mobile host. It removes the TCP packets from cache if their corresponding ACKs are received. By estimating probability of transmission failure and collision, the snoop agent adjusts the contention window of the MAC layer and informs the TCP source to reset its congestion window size. With the simulation, it has been shown that [5] this mechanism enhances the TCP performance in WLANs by successfully alleviating the effects of radio channel errors on TCP protocol.

A cross layer approach called Link Layer CLAMP-TCP (L^2CLAMP-TCP) is proposed in [6]. It requires two additional software modules L^2CLAMP-Agent inside Base Station / Access Point and L^2CLAMP-Client on the receiver which are included in the protocol stack above the Link Layer. When the TCP packet is received by L^2CLAMP-Agent, it calculates the desired size of the Advertised Window based on the values of congestion measure, receiver rate and RTT. It generates and stores a TCP ACK with the calculated information in the buffer. Then it forwards the TCP packet to the receiver and waits for Link Layer acknowledgement from L^2CLAMP-client. Once it is received, L^2CLAMP-Agent sends the generated ACK with optimal Advertised Window value to the TCP sender. If it is not, the generated ACK is dropped from the buffer. Since the TCP ACK is suppressed from the receiver over wireless channel, it saves time which can be utilized for data packet delivery. It has been shown from the simulation results [6], this approach improves the TCP performance under error free and error prone environment.

2.1. Retransmission Time-Out (RTO)

TCP limits the transmission rate by adjusting the congestion window which gives a range of sequence numbers of TCP packets which can be transmitted at a moment without waiting for ACK. After transmitting a window's worth of data, the TCP sender may stall while waiting for an ACK. If an ACK packet is not received within RTO value, the TCP sender will assume that the packet is lost and will retransmit the lost packet. To calculate RTO, Round Trip Time (RTT) is required, which is the time needed for the packet to reach the destination and be acknowledged. If an RTT measurement is started, no other measurement starts until the value of this RTT is finalized. Due to Wireless Network dynamics, RTT of one packet could be different from that of another. So the Exponential weighted moving average RTT ($\dot{t}$) and RTT variation (σ_t) are determined based on the collected RTT samples and are used to compute the RTO value [7]. At time k, let t(k) be the RTT sample, $\dot{t}(k)$ be the smoothed RTT value, $\sigma_t(k)$ be the RTT variation and Δ refer to $t(k+1) - \dot{t}(k)$, then

$$\dot{t}(k+1) = \alpha\dot{t}(k) + (1-\alpha)\,t(k+1) - - - (1)$$

where $\alpha \in (0,1)$, usually set to 7/8

$$= \frac{1}{3}\left(7\dot{t}(k) + t(k+1)\right)$$

$$= \frac{1}{8}\left(7\dot{t}(k) + \dot{t}(k) + t(k+1) - \dot{t}(k)\right)$$

$$= \frac{1}{8}\left(8\dot{t}(k) + \Delta\right) - - - (2)$$

$$\sigma_t(k+1) = \beta\sigma_t(k) + (1-\beta)\left|t(k+1) - \dot{t}(k+1)\right| \qquad (3)$$

where $\beta \epsilon$ (0,1), usually set to 3/4

$$= \frac{1}{4}\left(3\sigma_t(k) + |\Delta|\right)$$

$$= \frac{1}{4}\left(3\sigma_t(k) - 4\sigma_t(k) + 4\sigma_t(k) + |\Delta|\right)$$

$$= \frac{1}{4}\left(-4\sigma_t(k) + 4\sigma_t(k) + |\Delta|\right) - - - (4)$$

By having equations (2) and (4), an unbound RTO value is found instead of using eqn (1) and (3) to reduce the number of arithmetic operations.

$$\text{Unbounded RTO} = \dot{t}(k+1) + m\,\sigma_t(k+1) - - - (5)$$

The stall time after transmitting the kth window is given as follows [8]

$$stall\ time = \frac{S}{R} + RTT - 2^{k-1}\left(\frac{S}{R}\right) --- (6)$$

where S refers to Maximum Segment Size (MSS) in bits and R denotes the transmission rate of the link. The component (S/R + RTT) denotes the time the TCP sender begins to transmit the kth window until the time when the sender receives an ACK for the first segment in the window. The component 2^{k-1} (S/R) is the transmission time of the Kth window. The RTO value needs to be optimized according to the tradeoffs: a small RTO value leads to unnecessary packet retransmission while a large RTO value results in high latency of packet loss detection.

2.2. Modified TCP Congestion Control

According to the Modified TCP which is proposed by A. Elrahim et al in [9], let w be the current congestion window and w_{th} be a slow start threshold, then, if w < w_{th}, for every received ACK, w is incremented as follows

$$w_{new} = w + MSS\ X\ \left({}^{MSS}/_{w}\right)$$

$$w_{th} = \frac{3w}{4}$$

When a time-out occurs, then the Modified TCP sets the value of the w_{th} to ¾ of the previous w and reduces w back to 4 MSS. The Fast Retransmit phase is started when 2 duplicate ACKs are received. The differences between Modified TCP and TCP are summarised in Table 1.

Table 1 : Differences between TCP and Modified TCP

TCP	**m in eqn (5)**	**w_{th}**	**Loss Detection**	
			Timeout	**Fast Retransmission**
TCP	4	½ * w	Reset w to 1	Entered when 3 ACKs are received
Modified TCP	5	¾ * 3	Reset w to 4	Entered when 2 ACKs are received

2.2.1. Benefits of Modifies TCP

(i)Unnecessary spurious timeouts are reduced due to increase in the value of m in eqn (3).
(ii) Early start of Fast Retransmission phase to reduce delay.
(iii) More transmission by a small increment of w for each ACK that arrives.
(iv)Fast recovery of the w by increasing slow start initialization point to 4 MSS .
(v)Longer slow start period by increasing w_{th}.

3. PROPOSED METHODOLOGY

This section identifies the necessities for cooperative contention and congestion control algorithms in wireless adhoc networks and gives the Finite State Machine definition for various Backoff Algorithms.

3.1. Need for cooperative algorithm

It is required to have intelligent and collaborative mechanism between TCP congestion control and MAC contention window algorithm to reduce the effect of wireless interference on the TCP performance. The most important reasons are listed below.

(i) Due to the nature of wireless communication (high probability of path loss, channel noise, fading and interference), an unacknowledged packet could result from not only collisions but also frame loss. But TCP always presumes that the packet was lost due to collision. Consequently, slow start phase is started which reduces the congestion window and BO parameters are exponentially increased. As a result, TCP throughput is not stabilized. So a proper approach dealing with lost but non-collided packets should try to resend them again as quickly as possible without unnecessarily exponential increase of BO parameters and decrease of congestion window.

(ii) Even though MAC layer provides retransmission mechanism, packet may still be lost without delivered to transport layer due to spurious interference or collision [5]. So TCP triggers the retransmission with the help of recovery routines which degrade TCP performance since the original packets need end-to-end retransmissions, which leads to an undesirable reduction in the TCP congestion window.

(iii) The greediness of TCP flows, hidden terminal and receiver blocking problems cause a severe unfairness problem which leads to flow starvation and packet collision [4].

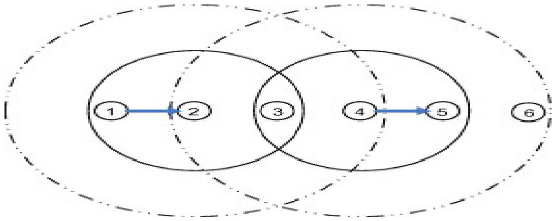

Figure 1. Linear topology

Figure 1 shows the transmission range of node 4 and node2 by solid circle and carrier sensing range by dotted circle. When node 4 is transmitting to node 5, node 2 is a blocked receiver of node 1, since it is in carrier sensing range of node 4. Due to this, node 1 continues to double its contention window and retransmitting packet until dropping it. After completion of the transmission, node 4 has the highest priority to capture the medium by resetting its contention window to minimum. If node 4 wants to start a new transmission again, it has the highest priority for accessing the medium because it has to wait only for the minimum duration. Thereby flow between node 1 and node 2 will be starved. The situation will get worse, if TCP flows have to traverse more hops to reach their destinations.

(iv) Mobility is one of the factor that causes TCP performance degradation in Ad hoc networks even at very light traffic. MAC layer has to take a decision about whether the link failure is caused by mobility or by congestion[10] and corrective mechanism has to be started. Based on this, route discovery by network layer protocol or congestion control by TCP will be initiated. Hence MAC layer highly influences the upper layer operations.

From the aforementioned reasons, the cross layer interaction between the transport layer and the MAC layer is an extremely important mechanism in wireless environment to help the TCP sender to determine the reason for packet loss based on the information provided from the MAC layer protocol.

3.2 Backoff Contention Algorithms

IEEE 802.11 medium access mechanism uses a random access scheme with carrier sense and collision avoidance through random backoff as shown in Figure 2 [11]. If the medium is idle for at least the DIFS duration, a station can access the medium. If the medium is not idle, stations have to wait for the DIFS duration and entering into the contention phase. Then the station has to choose random BO time within the contention window(CW) and delays medium access for this random time. Tremendous BO algorithms were proposed in the literature which differ from other algorithms by means of selecting the random BO time from the chosen contention window. In this paper, Binary Exponential Backoff (BEB) [11], Exponential Increase Exponential Decrease (EIED)[12], Double Increment Double Decrement (DIDD) [13], Linear/ Multiplicative Increase and Linear Decrease(LMILD)[14], Binary Negative Exponential Backoff (BNEB)[15] algorithms are analyzed based on the NS2 simulation.

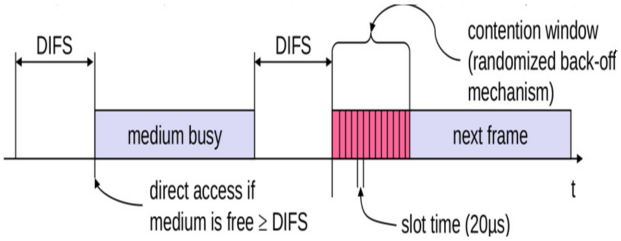

Figure 2. Medium Access Mechanism of IEEE 802.11

The behaviour of these algorithms are depicted as Finite State Machine (FSM) definition in Figure. 3. The arrows in the FSM description indicate the transition of the BO Algorithm from one state to another. The event causing the transition is shown above the horizontal line and the actions taken when the event occurs are shown below the horizontal line. The state transition occurs based on the event, if no action is taken on transition, then the symbol ʌ is used below the horizontal line. The initial state of the FSM is indicated by the dashed arrow.

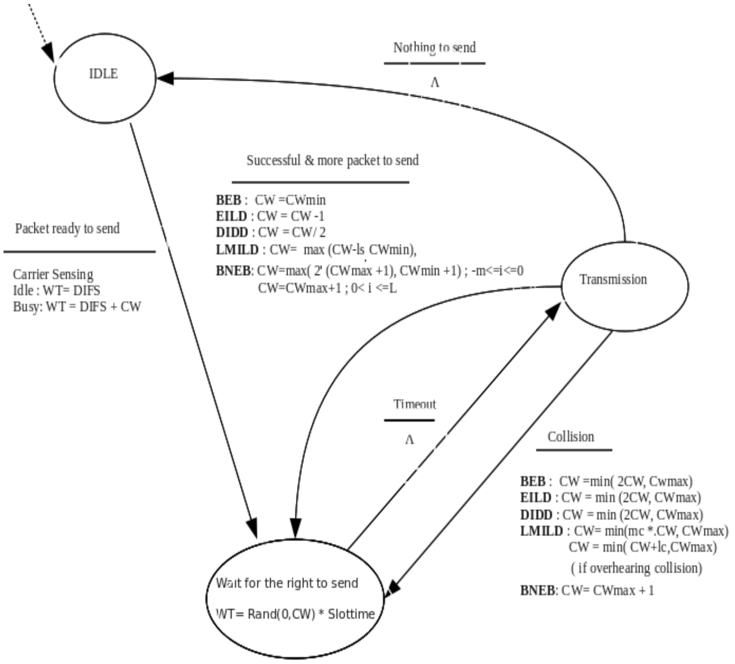
IDLE
Nothing to send
Λ
Successful & more packet to send
BEB : CW =CWmin
EILD : CW = CW -1
DIDD : CW = CW/ 2
LMILD : CW= max (CW-ls CWmin),
BNEB: CW=max(2ᶦ (CWmax +1), CWmin +1) ; -m<=i<=0
CW=CWmax+1 ; 0< i <=L
Packet ready to send
Carrier Sensing
Idle : WT= DIFS
Busy: WT = DIFS + CW
Transmission
Timeout
Λ
Collision
BEB : CW =min(2CW, Cwmax)
EILD : CW = min (2CW, CWmax)
DIDD : CW = min (2CW, CWmax)
LMILD : CW= min(mc *.CW, CWmax)
CW = min(CW+lc,CWmax)
(if overhearing collision)
BNEB: CW= CWmax + 1
Wait for the right to send
WT= Rand(0,CW) * Slottime

Figure 3. FSM Definition for BO Algorithms

Whenever the packet is ready to send, station performs the carrier sensing and goes to the next state in which it waits for the amount of duration given by Waiting Time (WT). Once WT expired, it goes to Transmission state. There are three possible events at this state. If there are no more packets to send, then station goes to Idle state. When collision occurs during the transmission or transmission is successful and more packets are to be sent, CW is adjusted based on the BO algorithms used. Then the station goes to waiting state to wait for the newly calculated WT.

4. Performance Analysis

The simulation to evaluate the performance of Modified TCP congestion control with various BO algorithms is done by using NS2 simulator [16] for the metrics throughput, end-to-end delay, packet loss, congestion window variation. For simulation, 20 mobile nodes and 10 TCP flows to transfer data during the simulation time (200sec) are considered. The source and destination pairs are randomly chosen by cbrgen.tcl script [16] and random mobility pattern is applied by setdest command. To calculate the movement of nodes, pause time of 2 sec and maximum speed of 5 m/s are used in the simulation. Each node moves towards a random destination with a speed uniformly distributed between 0 and maximum speed. When it moves to target position, it waits for pause time and then selects another target and moves towards it. This mobility will continue till the end of the simulation.

4.1 Performance of Modified TCP

The benefits of modified TCP listed in section 2.2.1 are validated by NS2 simulation under the metrics throughput and energy consumption. The results obtained from the simulation are plotted in Figure 4 and Figure 5. From Figure 4, it is inferred that higher throughput is achieved by modified TCP by having larger, stable and optimal congestion window size due to the above mentioned benefits, whereas in Traditional TCP, the throughput is decreased due to unstable congestion window size.

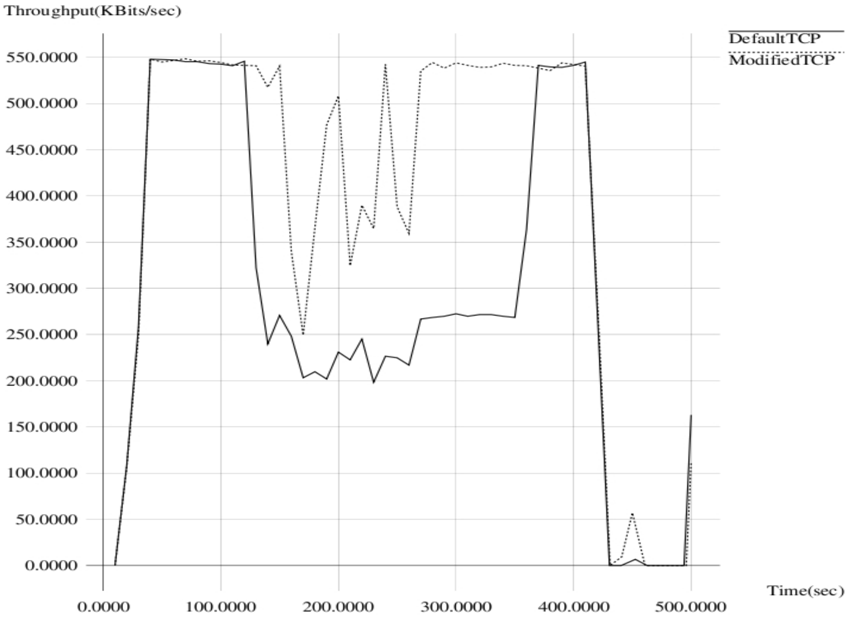

Figure 4. Throughput Comparison

It is observed from Figure 5, modified TCP has consumed less energy by avoiding incorrect triggering of usable congestion window reduction. Consequently it reduces number of retransmission leading to lesser energy consumption. Based on these results, Modified TCP is considered further for analyzing the impact of MAC layer contention algorithms in the TCP congestion control algorithm.

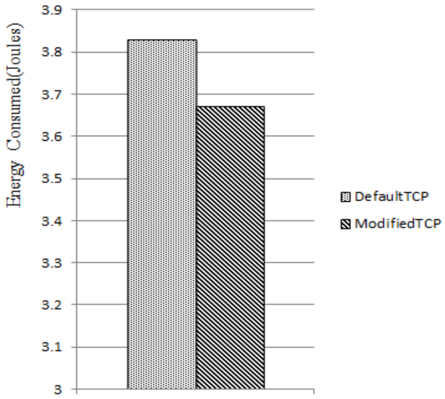

Figure 5. Energy Consumption

4.2 Changes in Congestion Window

By simulating various BO Contention algorithm with modified TCP, the changes in the congestion window are noted and plotted in Figure 6. It is inferred that from the Figure 6, from the simulation time 120sec onwards, the congestion window size is unstable because of increased traffic load. Among the analyzed BO algorithms, DIDD has fairly unstable congestion window sizes. This is because of the characteristic of DIDD contention scheme[13]. In this, every data packet is being retransmitted until its successful transmission. Due to this repeated retransmission, frequent time-out event occurs which requires frequent changes in congestion window. It is also observed that LMILD has similar fluctuation of congestion window sizes at the beginning and ending of the simulation upto 150sec. LMILD assumes that packet loss is only due to congestion. It fails to detect the collided packets due to channel fading[17] in mobile environment.

As a result, usable congestion window is reduced. BNEB gives better performance in ideal channel condition, but not in the presence of transmission errors. Whenever the traffic load and transmission errors are increased, BNEB keeps a maximum CW size. Moreover in EILD, after successful transmission, CW size is linearly decreased. Whereas in some BO algorithms, CW is set to either minimum CW(as in BEB) or half of the current CW (as in DIDD).Due to this, EILD also chooses longer waiting time. This increased contention time triggers the time-out frequently in TCP congestion control mechanism, and hence more fluctuation in congestion window size after 120sec.

4.3 Throughput

Throughput refers to the amount of data received by the receiver per unit seconds. The simulation is carried out to compute the throughput while executing various BO algorithms. The results are plotted in Figure 7 . It is inferred that, EILD has lesser throughput. Because usable congestion window is nearly 1 from 60sec to 120 sec and after that high fluctuation in congestion window size as shown in Figure 7. It implies lesser throughput compared with other algorithms. LMILD has higher throughput after 150sec, because its congestion window is more stable during that time which implies high throughput.

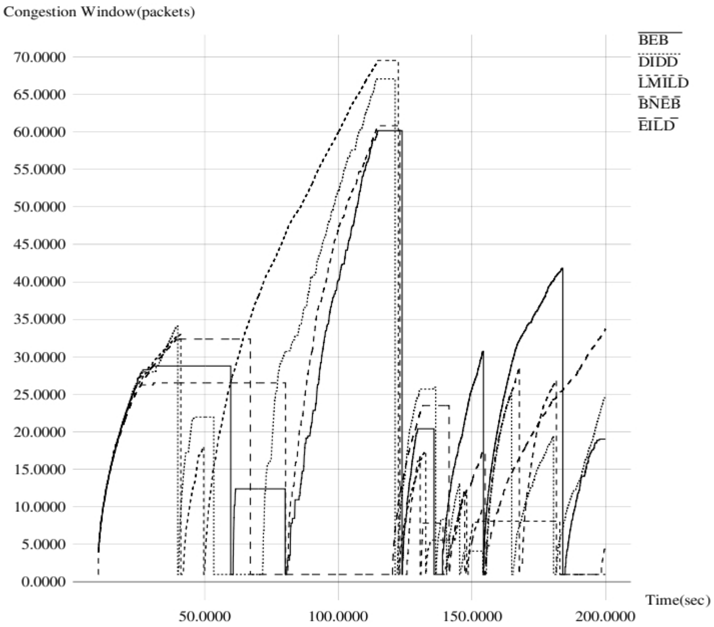

Figure 6. Congestion Window Variations

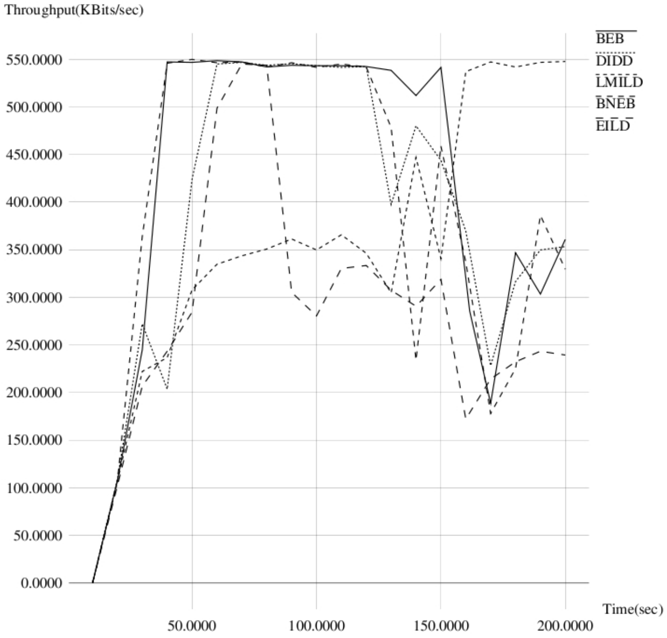

Figure 7. Throughput comparison

4.4 Packet loss

The number of lost packets are calculated at different simulation time which is plotted in Figure 8. It is inferred that from Figure 8, packet loss is less in DIDD from 60sec onwards. This is because, the packets that reach their maximum number of retransmission are not discarded in DIDD scheme [13]. In BNEB,when collision occurs, CW is incremented linearly, so it doesn't give much difference between previous and current CW size, which is not favorable in case of highly congested situation as it increases packet loss.

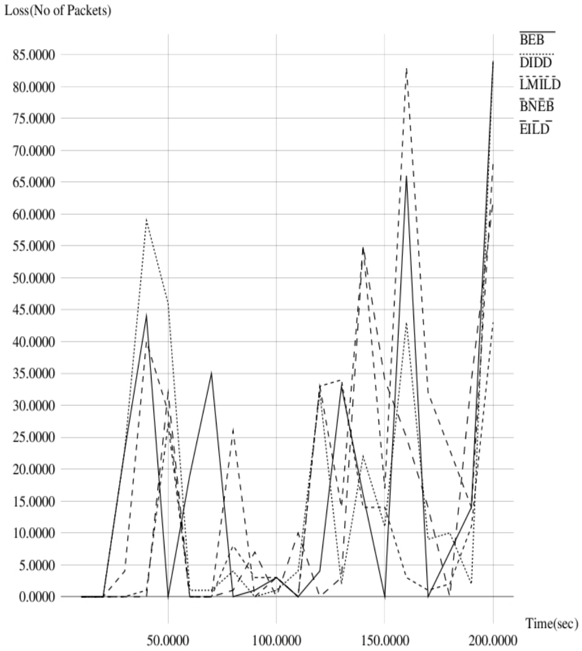

Figure 8. Packet Loss Comparison

4.5 End-to-end delay

End-to-End delay is the sum of delays experienced at each node from source and destination. It includes transmission delay, propagation delay and processing time. The simulation is carried out to calculate the end-to-end delay while executing various BO algorithms. The results are plotted in Figure 9. It is inferred that BNEB increases the end-to-end delay, since it keeps a maximum CW size to avoid collision probability. After successful transmission, EILD linearly decreases the CW size. Due to this, larger BO time is chosen from the CW range thereby increasing delay. Whereas in some BO algorithms, CW is set to either minimum CW(as in BEB) or half of the current CW (as in DIDD) to reduce the CW range.

5. CONCLUSIONS

TCP congestion control implicitly assumes the packet loss only due to congestion. But in wireless environment, packet loss may also due to transmission error, collisions, link breakage and medium contention, which must be correctly attributed by MAC layer. The incorrect reaction to the packet loss may lead to a curbing of congestion window size thereby limiting the performance of TCP in wireless ad hoc networks. From the simulation results, it is evident that the contention window greatly influences the congestion control mechanism which affects the overall performance of the network. Hence the designing of an intelligent and collaborative MAC layer contention and TCP layer Congestion control algorithm is an extremely important issue in wireless adhoc network.

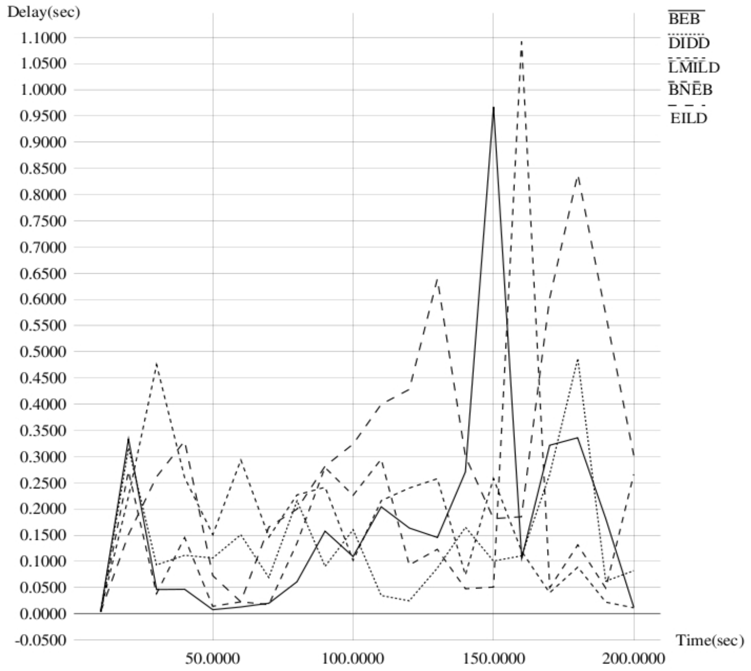

Figure 9. End to End Delay Comparison

REFERENCES

[1] Ahmad Al Hanbali, Eitan Altman and Philippe,(2005) " A survey of TCP over Ad Hoc Networks", *IEEE communications surveys & tutorials,* Third Quarters, Vol**:** 7 pp. 22 - 36.

[2] H. Zhai, J. Wang, X. Chen, and Y. Fang, (2004) "Medium access control in mobile ad hoc networks: Challenges and solutions", *Journal on Wireless Communications & Mobile Computing,* Vol 6, pp. 151 – 170.

[3] Lin Ma, Jun Zhang and Kai Liu, (2010) " Contention-based congestion control in wireless ad hoc networks", *International Conference on Information, Networking and Automation (ICINA),* Vol. 2, pp**.** 28 –32.

[4] H. Zhai, X. Chen, and Y. Fang, (2007) "Improving Transport Layer Performance In Multihop Ad Hoc Networks by Exploiting MAC Layer Information", *IEEE Transactions on Wireless Communications,* vol. 6, no. 5, pp. 1692 – 1701.

[5] Der-Jiunn Deng, Rung-Shiang Cheng, Heng-Jia Chang, Hui-Tang Lin and Ruay-Shiung Chang, (2009) "A cross-layer congestion and contention window control scheme for TCP performance improvement in wireless LANs" , *Telecommunication Systems*, Volume 42, pp. 17-27.

[6] Priya, S.S., Murugan, K., (2009) "Cross Layer Approach to Enhance TCP Performance over Wireless Networks", *In:Proc. IEEE Students' Technology Symposium(TechSym),*pp.171–176.

[7] Teerawat Issariyakul and Ekram Hossain, (2009) "Introduction to Network Simulator NS2", *Springer publication*.

[8] James F.Kurose, Keith W.Ross, (2005) "Computer Networking- A Top Down Approach Featuring the Internet", Third Edition, *Pearson Education.* .

[9] Adel Gaafar A Elrahim, Hussein A Elsayed, Salwa El Ramaly, Magdy M Ibrahim, (2011) "Energy efficient congestion control operation in WSNs ", *International Conference on Computer Engineering Systems ,* pp . 41-46.

[10] Noor Mast and Thomas J Owens, (2011), "A survey of performance enhancement of transmission control protocol (TCP) in wireless ad hoc networks", *EURASIP Journal on Wireless Communications and Networking, DOI :* http://dx.doi.org/10.1186/1687-1499-2011-96

[11] Jochen H. Schiller, (2003) "Mobile Communication", Second Edition, Pearson Education.

[12] Jian Li, Xianwen Zeng, Q. Su, (2008) "Performance Investigation of Backoff Algorithms in Multihop Wireless Networks", *International Conference for Young Computer Scientists*, pp.564-569.

[13] P. Chatzimisios, V. Vitsas, A.C. Boucouvalas, M. Tsoulfa, (2007) “Achieving performance enhancement in IEEE 802.11 WLANs by using the DIDD backoff mechanism”, *International Journal of Communication Systems*. Vol 20, pp. 23-41.

[14] J. Deng, P. K. Varshney, and Z. J. Haas, (2009) “A new backoff algorithm for the IEEE 802.11 distributed coordination function”, *In Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2009)*, Vol 3,pp. 455-459 .

[15] Ki, H.J., Choi, S.-H., Chung, M.Y., Lee, T.-J, (2006) “Performance evaluation of Binary Negative-Exponential Backoff Algorithm in IEEE 802.11 WLAN”, *In Proceedings of Mobile Adhoc ans Sensor Networks (MSN),* vol. 4325, pp. 294–303.

[16] http://isi.edu/nsnam

[17] Bum-Gon Choi, Hyung Joo Ki, Min Young Chung, and Tae-Jin Lee, (2007) “Performance Evaluation of Binary Negative-Exponential Backoff Algorithm in Presence of a Channel Bit Error Rate”, *International Conference on Computational Science*, Part IV: ICCS 2007, pp. 554–557.