CTCP: A CROSS-LAYER INFORMATION BASED TCP FOR MANET

Gaurav Bhatia[1] and Vivek Kumar[2]

[1]Department of Information Technology,
Ibri College of Technology, Wilayat Ibri,Sultanate of Oman.
[2]Department of Computer Science,
Gurukul Kangri Vishwavidyalya, Haridwar-249404,India.

ABSTRACT

Traditional TCP cannot detect link contention losses and route failure losses which occur in MANET and considers every packet loss as congestion. This results in severe degradation of TCP performance. In this research work, we modified the operations of TCP to adapt to network states. The cross-layer notifications are used for adapting the congestion window and achieving better performance. We propose Cross-layer information based Transmission Control Protocol (CTCP) which consists of four network states. Decelerate state to recover from contention losses, Cautionary state to deal with route failures, Congested state to handle network congestion and Normal state to be compatible with traditional TCP. Decelerate state makes TCP slow down if the packet loss is believed to be due to contention rather than congestion. Cautionary state suspends the TCP variables and after route reestablishment resumes with conservative values. Congestion state calls congestion control when network is actually congested and normal state works as standard TCP. Simulation results show that network state based CTCP is more appropriate for MANET than packet loss based traditional TCP.

KEYWORDS

MANET, IEEE MAC 802.11, DSR, TCP, Congestion Control, Cross Layer Information

1. INTRODUCTION

Mobile Ad hoc Network (MANET) is wireless, infrastructure less, mobile nodes network which uses multihop path to transmit the information [1]. The nodes composing a MANET are free to move and to organize themselves arbitrarily. The topology of the network may change rapidly and unpredictably. Unfortunately, the TCP congestion control reveals to be suboptimal in MANET [2]. The traditional TCP is remarkably successful for wired network, where the key assumption that packet loss is due to congestion works well. In MANET, beside congestion, there are various factors for packet loss such as heavy MAC contention, route failure etc. TCP in MANET misinterpret every packet loss as congestion where in practice each one is required to be addressed differently. In this section we illustrate the different factors of TCP performance deterioration in MANET.

DOI : 10.5121/ijasuc.2014.5101

1.1. Excessive MAC Contention

In MANET, the underlying protocol used to access the wireless channel is IEEE 802.11 MAC. The multiple nodes contend together to access the shared channel for transmission. A contention based channel reservation mechanism is used to avoid collisions. One of the factors for packet loss in MANET is excessive MAC contention in the shared wireless channel. The MAC contention level is intensified by the TCP window mechanism. TCP aggressively increases the packet transmission rate by doubling its congestion window on each arrival of ACK. The exponential growth mechanism makes network overloaded by transmitting multiple packets together. The TCP aggressive window causes higher amount of contention among nodes as all of them try to access the channel. It causes more collision in ad hoc wireless network. TCP sends packets beyond the limit that causes excessive MAC layer retransmission and leads to MAC contention losses. This further is wrongly perceived and dealt by TCP as congestion. A TCP sender should limit the size of its congestion window in order to achieve better performance.

1.2. Routing impact on TCP

In MANET, nodes move unrestrictedly in the network changing the topology dynamically. This mobility causes frequent route failures. When a route is broken, the underlying reactive routing protocols send a route failure message to the source and then source routing protocol initiates a route reestablishment process. The route reestablishment process may take a significant amount of time to obtain a new route to the destination. The number of nodes in the route, traffic load in network, stale cache of routing protocol and the nature of the routing protocols are some reasons that increase route reestablishment time. The large delays in route repair have a severe impact on TCP performance. Since the TCP is not aware of the route failure, it continues to transmit / re-transmit packets. If discovering a new route takes longer time then the RTO times out at the sender and this further invokes TCP congestion control. The unnecessary reduction of CWND and doubling the RTO deteriorate the TCP throughput. Thus, if packet loss is because of route failure TCP should not invoke the congestion control.

1.3. Malapropos Congestion Control

Congestion can be defined as the state of a network when there is not sufficient bandwidth to support the network traffic. As in wired networks, MANET also suffers from congestion. The dynamic topology of MANET causes congestion quickly as one node can be the center of transmission for multiple nodes with arbitrary movement of nodes. This congestion frequently happens in MANET and can lasts for a long period of time. Traditional TCP assumes that packet loss indicates network congestion. In a MANET, packet losses are usually not caused by network congestion. The heavy contention from MAC layer and frequent route failure due to mobility lead packet losses, but the TCP invokes congestion control for packet losses. Thus, the current method of TCP to detect the congestion is inappropriate for MANET.

In our proposal, cross-layer interactions convey network state information to TCP that utilizes it to disambiguate congestion from heavy contention and route failure. Cross-layer interaction is a promising architecture to support flexible layer approach. In the cross-layer design [3], lower layers protocols interact with the upper layers irrespective of the OSI model in which the layers are restricted to communicate with the layers above and below to it. The paper is organized as follows: Sections 2 present the literature review. Section 3 presents a detailed description of our proposed work. Section 4 shows the simulation results and conclusion is given in Section 5.

2. LITERATURE REVIEW

Several efforts have been made to optimize the performance of TCP over MANET [4]. We focus on research work based on lower layer feedbacks. These methods can be categorized based on route failure [5-7], and MAC contention [8-10] as problem domain.

Chandran et al. [5] proposed TCP-feedback (TCP-F) to overcome the TCP over reaction towards route failures in MANETs. As soon as the network layer at any node detects the broken route, it explicitly sends a route failure notification packet to the source. Consequently, the TCP sender goes into the snooze state and freezes all its variables, timers and congestion window size. When one of the intermediate nodes sends a route re-establishment notification to the source then it resumes the transmission from the frozen state. In [6], Holland and Vaidya proposed a similar approach which uses an Explicit Link failure Notification to inform the TCP sender about the route failure. Unlike TCP-F using an explicit notice to signal that a new route has been found, the sender, while on stand-by, periodically sends a small packet to probe the network to see if a route has been established. If there is a new route, the sender leaves the stand-by mode, restores its RTO and continues as normal. They also introduced a new metric, expected throughput, which provides a more accurate means of performance comparison by accounting for the differences in throughput when the number of hops varies. They, then, used this metric to show how the use of explicit link failure notification can significantly improve TCP performance. In [7], authors do not impose changes to the standard TCP itself. A thin layer called ATCP is inserted between TCP and IP layers. The ATCP layer relies on the ICMP protocol and ECN scheme to detect network partition and congestion, respectively. The ATCP layer monitors TCP state and takes appropriate action. The ATCP's four possible states are: Normal, Congested, Loss and Disconnected. When three duplicate ACKs are detected, indicating a packet loss, ATCP puts TCP in persist mode and quickly retransmits the lost packet from the TCP buffer. After receiving the next ACK the normal state is resumed. Congestion control is invoked normally when ECN message is received. In case of temporary network partitioning, the ATCP receives an ICMP Destination Unreachable message. Hence, it puts the TCP sender in the persist state, sets TCP's congestion window to one and enters itself in the disconnected state. TCP periodically generates probe packets until it starts receiving their ACKs. This removes TCP from persist mode and moves ATCP back into normal state.

These papers focused on the route failures. The common shortcoming of these schemes is that, they cannot distinguish packet loss caused by MAC contention failure. Some related work for TCP optimization based on differentiating MAC contention losses from congestion is mentioned in [8], [9] and [10]. In [8], authors introduced a systematic solution named Wireless Congestion Control Protocol (WCCP) to address TCP performance degradation problem. They first illustrate that severe medium contention and congestion are intimately coupled, and TCP's congestion control algorithm becomes too coarse in its granularity, causing throughput instability and excessively long delay. Further, they illustrate TCP's severe unfairness problem due to the medium contention and the tradeoff between aggregate throughput and fairness. Then, based on the novel use of channel busyness ratio, a more accurate metric to characterize the network utilization and congestion status, they propose a WCCP to efficiently and fairly support the transport service in multihop ad hoc networks. WCCP uses channel busyness ratio to allocate the shared resource and accordingly adjusts the sender's rate so that the channel capacity can be fully utilized and fairness is improved. In this protocol, each forwarding node along a traffic flow exercises the inter-node and intra-node fair resource allocation and determines the MAC layer feedback accordingly. The end-to-end feedback, which is ultimately determined by the bottleneck node along the flow, is carried back to the source to control its sending rate. In [9], the authors address the aspect: how to properly set TCP's congestion window limit (CWL) to achieve optimal performance. They turned the problem of setting TCP's CWL into identifying the bandwidth-delay product (BDP) of a path which is the maximum number of packets that the path can support without causing congestion. By considering the transmission interference of the IEEE 802.11 MAC layer protocol, they derived an upper bound of BDP, which is approximately 1/5 of the round-trip hop-count (RTHC). Based on this upper bound, they proposed to use an adaptive CWL setting strategy to adjust TCP's maximum window size according to the current path's RHTC. In MANET, RHTC of a path can be obtained through a source routing protocol or by using a TTL-like counter in the IP header to carry the hop count of the path. In [10], authors presented a Contention Aware Transport (CAT) protocol which is hop-by-hop, rate-based control, over IEEE 802.11 MANETs. CAT introduces a novel control metric, Sending Rate Control Factor (SRCF), for the congestion control. The SRCF considers the contention density which represents the degree of contention level experienced at a given node. CAT adopts a cross-layered method to use MAC layer information and introduces SRCF and adjusts the sending rate based on it. In addition, to support reliable delivery, CAT uses a hop-by-hop acknowledgement scheme. The main purpose of CAT is, to find optimal traffic load that the channel bandwidth is fully utilized by sending rate adjustment while minimizing the packet losses from medium contention.

Our scheme, which shares the same target and also makes use of cross-layer feedback, differs from these approaches in that we deploy the different states for TCP to deal with contention, congestion and route failure losses. In our approach, nodes in MANET implement a notification mechanism that generates a message when it detects either of these events, so that TCP discriminate packet losses factors and react accordingly.

3. CROSS-LAYER INFORMATION BASED TCP (CTCP)

We introduce a C*ross-layer information based TCP* (CTCP) that percept the network state by the information provided by the lower layers notification messages. The objective of notification is to provide the CTCP sender with information about network status so that it can avoid responding to the failures as congestion occurs. These notifications elucidate packet loss signals as heavy contention, network congestion or route-failure. These messages bring CTCP at the sender into the appropriate state when it receives network state notification. In this section, we present methods to measure the contention, estimating the congestion and detection of route failure.

3.1. Contention Measure

The IEEE 802.11 MAC use control packets RTS and CTS to perform a handshake before any data transmission. Figure 1, illustrates the RTS /CTS communication before the data packet is transmitted. In regions where there is high contention for channel from other nodes, number of RTS packets sent is high due to the contention of the channel.

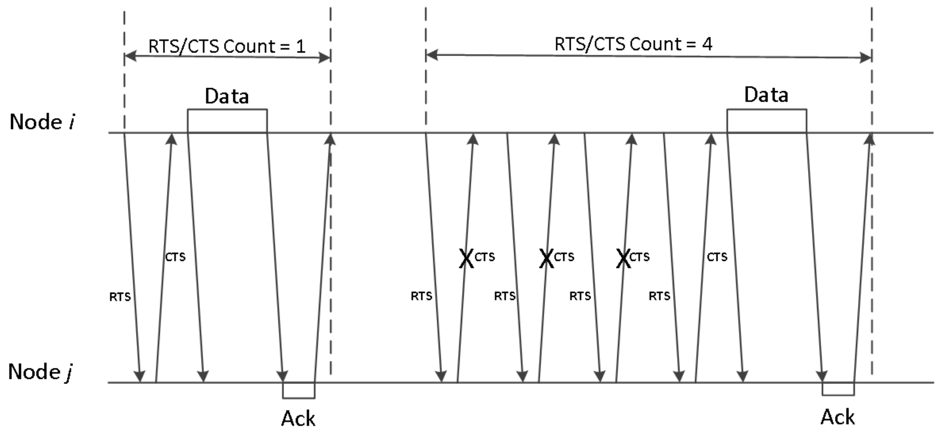


Fig. 1 IEEE802.11 MAC contention mechanism

In RTS / CTS scheme, each node maintains a variable *ssrc* to record the retransmitted count of RTS packet in the current node. When a node fails to transmit RTS packet, it increases *ssrc* by 1 and retransmits the packet. If the value of *ssrc* reaches 7, the sender discards the transmission, reports a link failure and resets *ssrc* to an initial value of 0. If the node successfully transmits the RTS packet then it also resets *ssrc* with an initial value of 0. Thus, when a node starts a new transmission, regardless of the previous value of *ssrc*, the initial value of *ssrc* will be 0. The node does not record the RTS retry count of the last transmission and thus remains ignorant about its channel state.

The contention stage appears to be the blockage of channel access in MAC as network load increases. To measure the contention, first, we introduce a *RTS Retry Count* (RRC) parameter to record the value of ssrc for each data transmission. The RRC indicates the contention of the medium and can be considered as a cross-layer metric for contention status of the node. Then, we propose a *MAC Contention Measure* (MCM), equation 1, parameter based on exponential moving average to evaluate channel contention status. The calculation of the MCM of a node is performed by applying the exponential moving average method to the previous RRC and the current RRC, as follows,

$$M_i = \kappa \times \Re + (1 - \kappa) \times M_j \qquad (1)$$

Where, $\boldsymbol{M_i}$ is the MAC Contention Measure of current attempt and $\boldsymbol{M_j}$ is the MAC Contention Measure of previous attempts, $\Re$ is RTS Retry Count. To calculate MCM, an exponential smoothing constant $\boldsymbol{\kappa}$ between 0 and 1 is selected. This constant is related to the number of time RRC is included, $\boldsymbol{N}$, by the following equation 2:

$$\kappa = \frac{2}{N+1} \qquad (2)$$

The MCM value is used by MAC layer for notifying the high contention level of the medium to CTCP. Once MCM exceeds the *contention threshold* (CTThresh), an *Explicit Heavy Contention Notification* (EHCN) message is sent to set the state of CTCP which indicates a heavy contention in MANET. The notification is sent with the help of ICMP. When EHCN message is sent, the value of κ and *M* resets to zero.

3.2. Congestion Estimation

The number of packets waiting in the interface queue is a metric reflecting the traffic load of the node. Generally, the congestion is defined as the node status where the offered load on interface queue approaches the capacity of the queue. We use the number of packets in the queue to compute *Link Congestion Measure* (LCM), equation 3. It is a simple measure of congestion as queue length results in long packet latency or packet drop.

$$LCM = \frac{Current\ Queue\ Length}{Max\ Queue\ Length} \qquad (3)$$

A high LCM indicates that the node suffers from congestion. When LCM reaches the *congestion threshold* (CNGThresh), an *Explicit Congestion Control Notification* (ECCN) is triggered to inform the sender CTCP about the congestion in the network. The ECCN notification sent to the CTCP by using ICMP messages.

3.3. Route Failure Detection

As a routing protocol, *Dynamic Source Routing* (DSR) [11] protocol has been chosen. When MAC layer detects the link failure at the intermediate node it then informs to the upper routing protocol. The DSR tries to find another route from its cache and if not successful then it transmits the route failure message to the source. In our proposed method, the DSR at intermediate node informs the sender immediately when it receives link failure notification from MAC. The advantage of informing the sender about route failures on time is that unnecessary retransmission and congestion control can be avoided. To implement notification message, *Route Error* (RERR) message of DSR protocol is modified. This message is further used as *Explicit Route Failure Notification* (ERFN) to notify the CTCP about the route failure.

3.4. CTCP Network States

In this section, we present CTCP network states utilizing each of the above components. The intermediate nodes keep track of its status and inform the transport layer for contention, congestion and route failure. Our key idea is based on cross-layer parameter for deciding whether to increase, decrease or suspend TCP congestion window. We modify the standard TCP, Figure 2, to use cross-layer notification to put the sender into either of four network states. Decelerate state to recover from contention losses, Cautionary state to deal with route failures, Congested state to handle network congestion and Normal state to be compatible with TCP. Decelerate state makes CTCP slow down if the loss is believed to be due to contention rather than congestion. Cautionary state suspends the CTCP variables and after route reestablishment resumes with conservative values. Congestion state calls congestion control when network is actually congested and Normal state works as standard TCP. CTCP will prevent unnecessary calling of congestion control when packets are lost due to node contention or routing errors.

(i) Decelerate State

On receiving the EHCN message, CTCP sender gradually reduces offered load. The CTCP decelerate state slowly decreases the MAC contention by subtracting one MSS per RTT. This method controls the amount of the outstanding data in the network while avoiding unnecessary reduction in the CTCP congestion window by implementing subtractive decrease. The discussion in section 1 suggests that a TCP sender should limit the size of its congestion window in order to achieve better performance. The CTThresh value for linear subtraction is set as one-half of CWND. Once the CWND reaches to the CTThresh, it returns to the normal state.

(ii) Cautionary State

When the sender receives the ERFN message, the CTCP enters into cautionary state, suspending all its operations and variables until a new route is found, ensuring that the sender does not invoke congestion control. While in the cautionary state, the CTCP refrains from transmitting any packet, waiting for a route re-establishment. After a route has been repaired, routing protocol transmits *Explicit Route Establishment Notification* (EREN) message. On receiving of EREN message, the CTCP leaves the suspension mode, sets the congestion window to the default *slow start threshold* (SSThresh), resumes RTO, and starts sending packets.

Instead of reducing the CWND to one MSS as TCP does when RTO timeouts, or to reduce it to one-half of the CWND plus three MSS as in Fast Recovery, or to resume it to same value from suspend values as in TCP-F, we propose to set the CWND as initial value of SSThresh. If packet loss is caused due to route failure, CTCP should not reduce the CWND as for congestion control because it reduces throughput. Also, it should not use the old window size because it may cause the contention or congestion in the network.

(iii) Congested State

When the network is truly congested, i.e., when the sender receives a ECCN message, the CTCP invokes Fast Retransmit / Fast Recovery congestion control methods as in TCP.

(iv) Normal State

The CTCP starts from normal state. It consists of standard TCP slow start and congestion avoidance phase. When CTCP receives notification message, it discontinues normal state and starts executing new state indicated by the notification message. After execution of new state or after receiving the EREN message, CTCP goes back to the previous phase in normal state.

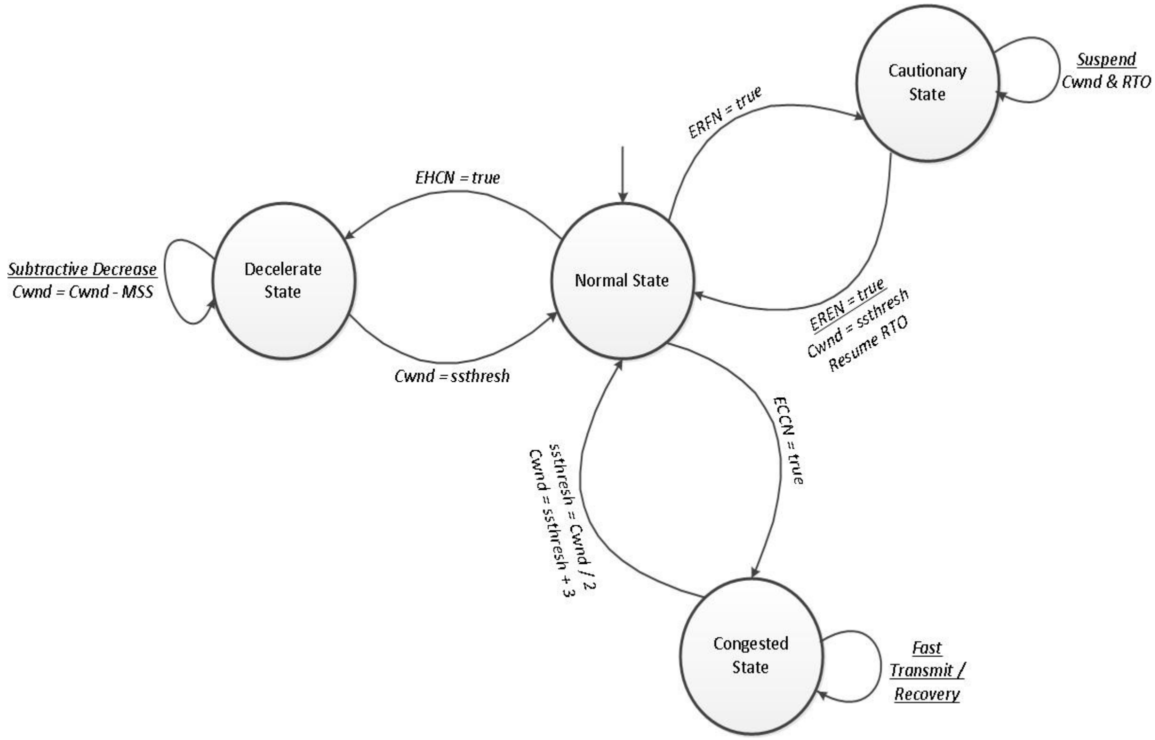


Fig. 2 State transition diagram for CTCP

4. SIMULATION AND RESULT ANALYSIS

We have compared the performance of our proposed CTCP Scheme with TCP Reno, the most widely deployed variant of TCP, via network simulator ns-2 [12]. The random waypoint mobility model is used for topology generation. All the scenarios consist of 50 mobile nodes, distributed randomly, moving in an area of 1000×1000 sq. m. The link layer model is the Distributed Coordinated Function (DCF) of the IEEE 802.11 wireless LAN standard. FTP application is used over TCP and CTCP for all the flows in the network. The default transmission range is 250 meters and channel capacity is 2 Mbits/sec. We have considered packet size of 512 bytes and transfer rate of 4 packets per second. Also, we have used 10s as pause time for all scenarios and each simulation lasts for 500s.

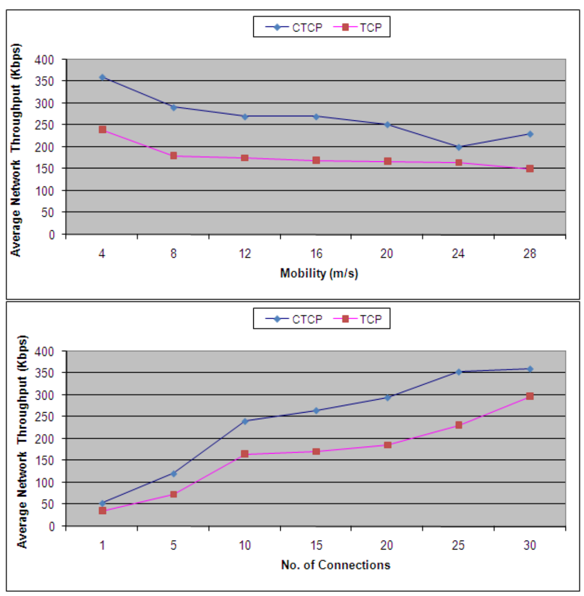
We considered two scenarios for performance analysis. For first scenario, varying mobility, the nodes move with speed range set to 4, 8, 12, 16, 20, 24 and 28 m/s for different simulation runs with 15 source and destination connections. For second scenario, varying load, defined to evaluate the performance based on varying number of connections between nodes are set to 1, 5, 10, 15, 20, 25 and 30 with constant mobility speed of 16 m /s. In this simulation, the metrics employed for the performance evaluation of proposed protocol are (i) the average network throughput, (ii) packet loss ratio, (iii) average congestion control calls and (iv) control overhead.

4.1. Average Network Throughput

The average network throughput is defined as the number of data packets successfully received by destination nodes in network per second. Figure 3, shows the total throughput generated in network using CTCP and TCP. For both the scenarios, the throughput achieved by CTCP is noticeably higher than the TCP.

Figure 3(a) shows output of mobility speed varying from 4 m/s to 28 m/s. TCP in comparison to CTCP experiences a low throughput. Due to DSR route reestablishment time on a route failure, TCP causes a frequent timeout and calls to unnecessary congestion control leading to throughput deterioration. However, when CTCP experiences a route failure it forces CWND to suspend and after route establishment it cautiously selects the size of congestion window maintaining the high throughput.

Similarly, Figure 3(b) shows load varying from 1 to 30 numbers of connections. As the load increases, the CTCP outperforms the TCP. Since CTCP does not use packet loss as an indicator of congestion it linearly decreases their congestion window instead of multiplicative decrease and maintains the high throughput. This is in contrast to TCP which aggressively increases its window size beyond the network capability and degrades the throughput.

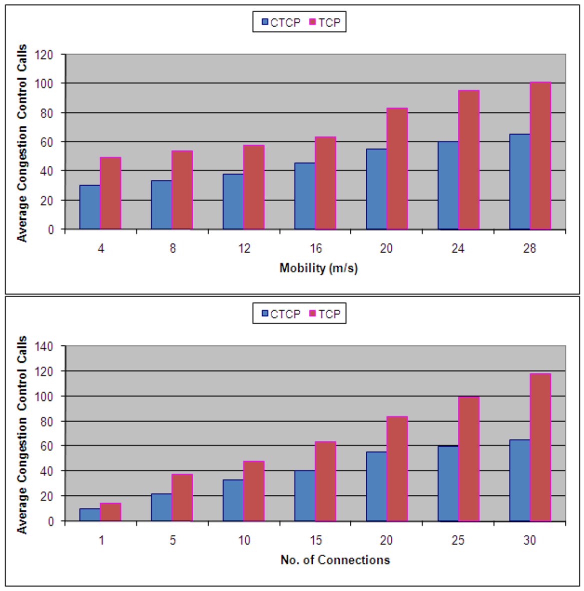


(a) Varying Mobility (b) Varying Load

Fig. 3 Average network throughput

4.2. Average Congestion Control Calls

The average congestion control calls is defined as the total number of calls to congestion control among all the connections per second. For both the scenarios, Figure 4, CTCP yields much lesser congestion control calls count than TCP. This can be attributed to the fact that CTCP is aware of contention and route failure losses and it does not call congestion control unnecessarily. TCP relies on its packet loss based indications and calls congestion control excessively. CTCP disassociates the congestion control from the loss recovery mechanism. Since CTCP uses linear decrease for heavy contention and suspension of congestion window in route failure, its congestion control calls count is very low in comparison to TCP.

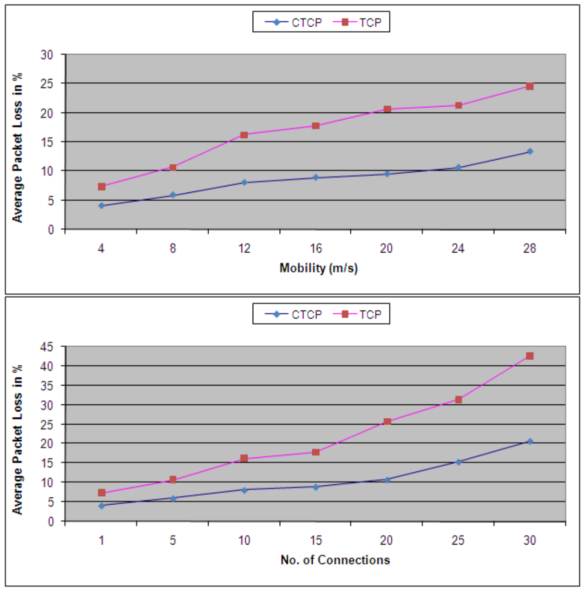


(a) Varying Mobility (b) Varying Load

Fig. 4 Average congestion control count

4.3. Packet Loss Ratio

It is the ratio of the number of data packet lost during transmission to the number of packets transmitted by the source. Figure 5 shows the packet loss ratio of CTCP and TCP for different number of mobility speeds and connections. As mobility of nodes increases, Figure 5(a), the CTCP packet losses decrease in comparison to TCP. This is attributed to the fact that the number of unnecessary packet re-transmissions during the route failure interval reduce. It is observed in Figure 5(b) that for varying loads the packet loss for CTCP is also significantly less than that of TCP. The result can also be explained by the fact that linear decrease of congestion window by CTCP mitigates medium contention losses and reduces the probability that a node drops the packet due to MAC contention.

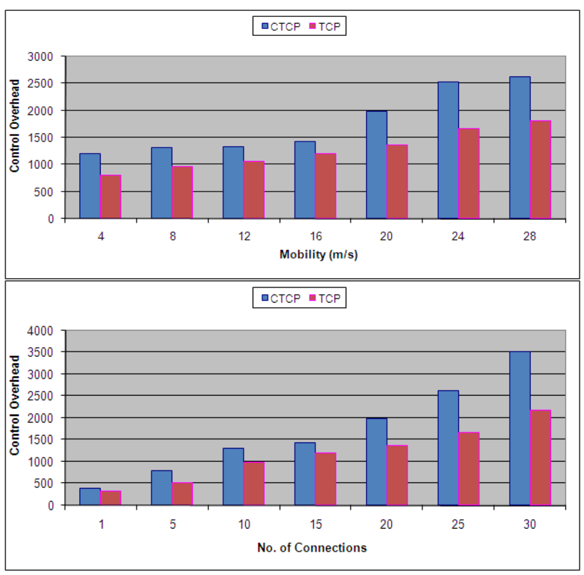


(a) Varying Mobility (b) Varying Load

Fig. 5 Packet loss ratio

4.4. Control Overhead

It is defined as the total number of control packets transmitted per seconds. Figure 6 shows that the control overhead for CTCP is higher than TCP. This is because the network status signals sent by the intermediate nodes to the source are more for CTCP. The CTCP generates EHCN, ECCN, ERFN and EREN additional messages. Eventually, this leads to the high control overhead for CTCP in comparison to TCP.



(a) Varying Mobility (b) Varying Load

Fig. 6 Control overhead

5. Conclusion & Future Work

In this paper, we considered the problem of TCP misinterpretation of contention and route-failure as congestion in MANET. We proposed a cross-layer information based TCP. The cross-layer information based messages give an explicit notification to CTCP to react according to the contention, congestion or route failure. This is done by putting CTCP into four states. Decelerate state to recover from contention losses, Cautionary state to deal with route failures and Congested state to handle network congestion and Normal state to be compatible with TCP. Result section shows considerable improvements are possible when cross-layer parameters are used to differentiate between congestion based packet loss and contention or route failure based packet loss. In future, the impact of channel errors on TCP may also be investigated.

References

[1] Chlamtac, I., Conti, M., Liu, J. J.N. (2003): Mobile ad hoc networking: imperatives and challenges. Ad Hoc Networks. 1, 13–64.

[2] Lochert, C., Scheuermann, B., Mauve, M. (2007): A survey on congestion control for mobile ad hoc networks. Wireless Communications & Mobile Computing.7 (5). 655-676.

[3] Srivastava, V., Motani, M. (2005): Cross-layer design: a survey and the road ahead. IEEE Communication Magazine. 43(12). 1112–119.

[4] Mast, N., Owens, T.J. (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.

[5] Chandran, K., Raghunathan, S., Venkatesan, S., Prakash, R. (1998): A feedback based scheme for improving TCP performance in ad-hoc wireless networks. In: Proceedings of 18th International Conference on Distributed Computing Systems (ICDCS).

[6] Holland, G., Vaidya. N. H. (1999): Analysis of tcp performance over mobile ad hoc networks. In: Proceedings of Annual International Conference on Mobile Computing and Networking. Seattle.

[7] Liu, J., Singh, S. (2001): ATCP: TCP for Mobile Ad hoc Networks. IEEE Journal on Selected Areas in Communications. 19. 1300-1315.

[8] Zhai, H., Chen, X., Fang, Y. (2007): Improving transport layer performance in Multihop ad hoc networks by exploiting MAC layer information. IEEE Trans. Wireless Commun. 6 (5). 1692 -1701.

[9] Chen, K., Xue, Y., Nahrstedt, K.: On setting TCP's congestion window limit in mobile Ad Hoc networks. In: Proceedings of IEEE ICC, Anchorage, Alaska, USA, May 2003.

[10] Lee, K. Y., Joo, S., Ryoo, J.: CAT (2006): Contention Aware Transport Protocol for IEEE 802.11 MANETs. In: Proceedings of IEEE 63rd VTC, vol. 2, pp. 523-527.

[11] Johnson, D., Maltz, D., Hu, Y.C. (2004): The Dynamic Source Routing for mobile ad hoc networks. IETF Internet Draft. http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt.

[12] Information Sciences Institute (ISI).The Network Simulator – ns-2. http://www.isi.edu/nsnam/ns/. Accessed 13 Jan 2013.