Study on Congestion Avoidance in MANET

W.R. Salem Jeyaseelan  
Assistant Professor  
J.J. College of Engg & Tech., Trichy, India

Dr. S. Hariharan  
Associate Professor  
Pavender Bharathidasan College of Engg & Tech., Trichy, India

ABSTRACT
Congestion is a main challenge in modern network environment. Unawareness of the network topology the network devices are demand to provide sufficient and structured connection establishment to the network. Nowadays Mobile Adhoc Network (MANET) plays a vital role in the modern networks. Because MANET can constructs its network and its paths based upon the current circumstances. Due to this instant network organization, the congestion is very essential and tough task in MANET. The congestion control mechanism has two basic classifications one is the congestion avoidance and next is slow start. Slow-start is used in conjunction to avoid transmitting huge amount of data in a single path in a network. It is used to control the congestion inside the network and works by increasing the TCP congestion window each time the acknowledgment is received. This is not the fair because any traffic occurs in the network the congestion window size is reduced by half. The window size is calculated by estimating the congestion between the nodes. TCP provides this information to the sender and the sender maintains the congestion window. All segments are received and the acknowledgments reach the sender on time, the window grows exponentially until a timeout occurs or the receiver reaches its limit. This paper focuses on congestion avoidance in the terms of widow size and data rate. The simulation results were obtained from Network Simulator2 (NS-2) version 2.3.9.

General Terms
Congestion control, Mechanism, Simulation.

Keywords
MANET, Slow Start, Congestion Window, TCP.

1. INTRODUCTION
1.1 Congestion in mobile ad-hoc networks
Manet is an infrastructure less wireless network. The network that allows the node mobility. Any number of nodes can participate in the data transmission. The objective is the nodes are free to roam in and around the network and the same time the sender node is forced to update its paths frequently. Route changes due to node mobility causes unsteady packet delivery delays and packet losses[14]. These delays and losses must not be misinterpreted as congestion losses. The use of a wireless multi-hop channel allows only one data transmission at a time within the interference range of one node. It depends on the network type; packet losses which are not caused by network congestion can be much more frequent in wireless networks. This can lead to wrong reactions of TCP congestion control. Moreover, observing packet losses is much harder, because transmission times and thus also round trip times vary much more. Furthermore, due to the comparatively low bandwidth of mobile ad-hoc networks, one single sender is able to be it accidentally or intentionally cause a collapse of the network due to congestion. The extreme effect of a single traffic flow on the network condition can cause severe unfairness between flows. Thus wireless multi hop networks are much more prone to overload-related problems than traditional wire line networks like the Internet. Therefore an appropriate congestion control is absolutely vital for network stability and acceptable performance.

1.2 Basic of congestion control Mechanism
The essence of the congestion avoidance mechanism of Transmission Control Protocol (TCP) is to dynamically control the window size according to the congestion level of the network. TCP is widely used by many Internet services including HTTP (Hypertext Transfer Protocol) WWW (World Wide Web) and FTP (File Transfer Protocol). Even if the network infrastructure may change in the future, it is very likely that TCP and its applications would be continuously used. The TCP protocol is executed at the terminal nodes of a network and it indicates the TCP protocol [1] status by packet traveling time as well as success or failure of the packet delivery. The accuracy of the bandwidth estimation is dependent on stability of network traffic and length of the path. The receiver measures the network bandwidth based on the packet interarrival interval and uses it to compute a congestion window size deemed appropriate for the sender. Due to unawareness of network conditions, regular TCP is not able to fully control the limited resources and distinguish packet loss from congestion loss and random loss. Routers are required to provide some information allowing the sender to estimate more accurately the remaining capacity over the bottleneck node with respect to the path from the sender to the receiver. A receiver can only reduce the data transfer rate by misbehaving. TCP at the terminal nodes will be able to adjust its data rate closer to the network capacity and to improve the performance of both TCP and network accordingly. TCP induces packet losses to estimate the available bandwidth in the network. TCP continues to increase its window size by one during each round trip time. When it experiences a packet loss, it reduces its window size to one half of the current window size. This paper focused to prevent the network congestion by adjusting the window size.
This paper organized that section 2 presents related works. Section 3 presents simulation study. Section 4, presents discussion about the comparison of slow-start and window based cc. Section 5 presents the conclusion and future works

2. RELATED WORKS

This section contains the verity of works carried out by the different research people in their own constraints. Yao-Nan Lien and Ho-Cheng Hsiao [1] said that TCP can resolve congestion efficiently and has higher average throughput than TCP New Reno. TCP outperforms TCP New Reno and TCP Selective ACK due to its ability to estimate the available bandwidth more precisely and the ability to deal with loss. The concept of multi-level data rate adjustment and the details to control the size of CWND remain to be investigated. They left out the Congestion window size reduction. Stefan Savage et al. [2] described that, the receiver can manipulate the TCP congestion control function managed by the sender, and the sender can prevent these manipulations. They also concentrate only on the receiver side. Christian Lochert et al. discussed about MANET congestion control survey [3]. In their survey they focused window-based additive increase, multiplicative decrease mechanism. TCP uses a timeout that depends on the measured round-trip time of the connection. If this retransmission timeout (RTO) elapses without an acknowledgment TCP concludes severe congestion. Then the window size is reduced to one and the unacknowledged segment is sent again. The timeout until the next retransmission attempt if still no acknowledgment arrives is doubled. Thus this timeout grows exponentially. The result of the survey paper leads to motivate in the direction of congestion window size. Kai Shi et al.[4], they proposed a sender and receiver combined congestion control mechanism. The receiver estimates a congestion window deemed to be appropriate from the measured bandwidth and RTT, and then advertises the window size (feeds this information back) to the sender. The sender then adjusts its congestion window according to the advertised window size of the receiver. Through this receiver-assisted method, the sender can increase the congestion window quickly to the available bandwidth, thus improving the network utilization. The performance of the TCP BIC and TCP Vegas congestion control algorithms analyzed [5] in ideal condition without any cross traffic and any other additional flows. In that small MANET scenario, the algorithm BIC provided good throughput after 75 seconds but algorithm Vegas provided stable and excellent result almost all over on the whole run time. So they conclude the algorithm Vegas be the good algorithm for small and short duration communication. They also did not focus on window size reduction.

3. SIMULATION STUDY

The research is carried out using network simulator-2 version 2.34(NS2- 2.34) (www.isi.edu/nsnam/ns). It is one of the most widely used commercial simulators based on Linux platform. The simulation focused on the performance of slow start and window based congestion control with increased in scalability and mobility. Therefore, two simulation scenarios consisting of 40 nodes initially and doubling amount nodes i.e. to 80 is considered. The nodes were randomly placed within certain gap from each other in 800 x 800 m and 1500 x 1500 m campus environment for 30 and 60 nodes respectively. The general parameters of Wireless LAN parameters are listed in Table 1[18].

4. COMPARISON AND RESULTS

4.1 Simulation 1: Change of Congestion Window Size

In this section, we evaluate and compare the performance of congestion control mechanism, slow start and window based congestion control in different kind of node structure by using the network simulator NS-2[17]. We also observe and show the behavior and the performance while TCP congestion window size. The general simulation parameters are listed in Table 2 and the parameter used by Congestion control mechanism of TCP is shown in Table 3.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Nodes</td>
<td>40-80</td>
</tr>
<tr>
<td>Link Bandwidth</td>
<td>2 Mbps</td>
</tr>
<tr>
<td>Transmission Range</td>
<td>250 m</td>
</tr>
<tr>
<td>Packet Size</td>
<td>1460 bytes</td>
</tr>
</tbody>
</table>

4.2 Simulation 2: Comparison of Slow Start and Window Based CC

In this section, we evaluate the performance of TCP in ideal condition and compare it with the window based CC using the network simulator NS-2. We focus on the comparison of slow start and window based CC. The performance of TCP New Reno and TCP Selective ACK is studied in this section. The result is presented in Table 4.

Table 1. Wireless LAN Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wireless LAN MAC Address</td>
<td>Auto Assigned</td>
</tr>
<tr>
<td>BSS Identifier</td>
<td>Auto Assigned</td>
</tr>
<tr>
<td>Physical Characteristics</td>
<td>Direct Sequence</td>
</tr>
<tr>
<td>Data Rate (bps)</td>
<td>11 Mbps</td>
</tr>
<tr>
<td>Channel Settings</td>
<td>Auto Assigned</td>
</tr>
<tr>
<td>Transmit Power</td>
<td>0.030</td>
</tr>
<tr>
<td>RTS Threshold</td>
<td>None</td>
</tr>
<tr>
<td>Packet-Receive Threshold</td>
<td>-95</td>
</tr>
<tr>
<td>Short Retry Limit</td>
<td>7</td>
</tr>
<tr>
<td>Long Retry Limit</td>
<td>4</td>
</tr>
<tr>
<td>AP Beacon Interval (seconds)</td>
<td>0.02</td>
</tr>
<tr>
<td>Max Receive Lifet ime (seconds)</td>
<td>0.5</td>
</tr>
<tr>
<td>Buffer Size (bits)</td>
<td>10240000</td>
</tr>
<tr>
<td>Large Packet Processing Fragment HCF</td>
<td>Promoted</td>
</tr>
</tbody>
</table>

4.3 Simulation 3: Comparison of Different MANET Congestion Control Algorithms

In this section, we evaluate the performance of different MANET congestion control algorithms, such as Vegas and BIC, in terms of throughput and delay. The result is presented in Table 5.

Table 2. Simulation Parameters
4.2 Simulation 2: Throughput and Retransmission Comparison
We compare the throughput and retransmission of Window based cc and slow start with other TCP variants. The network topology and the parameters used in Simulation 2 is the same as those used in Simulation 1. Fig. 2 shows the simulation results from this simulation.

4.3 Simulation 3
From Fig. 3, we observe that Window based congestion control TCP has a higher throughput than the Slow start including TCP when the hop count is less than 16. However slow start no longer performs well with longer networks (> 16 hops) because slow start keeps its congestion window size too small. Window based congestion control TCP is able to avoid the periodic packet loss due to precise controls of congestion window size according to router feedbacks.

4.4 Simulation 4
Fig. 2. Shows that slow start causes much less retransmission than Window based congestion control TCP. With increasing number of hop count, the numbers of retransmissions are all increasing for window-based congestion control mechanism. By using the results from our simulation slow start has some little bit low performance over Window based congestion control.

5. CONCLUSION AND FUTURE WORK
From the results obtained from the simulations, we conclude that Window based congestion control can resolve congestion efficiently and has higher average throughput than slow start. Due to its ability to estimate the available bandwidth more precisely and the ability to deal with random loss. While coexisting with TCP remains a stable performance output and fair utilization of the available bandwidth compared with other major TCP variants. The concept of multi-level data rate adjustment and the details of control the size of CWND remain stable in Window based congestion control. Furthermore, the theoretical formula for slow start and support of mobility are also essential for future work.
6. ACKNOWLEDGMENTS
The authors would like to express their thanks to the, Chairman, Vice Chairperson, Principal and Faculty of Information Technology Department, Pavender Bharathidasan and JJ College of Engineering and Technology for the environment provided.

7. REFERENCES