WTLS based Protocol for MANET

P.S. Sujith Kumar¹, Dr. A. Ramesh Babu²
Hindustan Institute of Technology & Science, Chennai¹
Prof & Head S&H, Hindustan Institute of Technology & Science, Chennai

Abstract

The throughput of TCP suffer when it used in mobile ad hoc networks. This is a direct consequence of TCP wrongly attributing packet losses due to link failures (a consequence of mobility) to congestion. While this problem causes an overall degradation of throughput, it especially affects connections with a large number of hops, where link failures are more likely. Thus, short connections enjoy an unfair advantage over long connections. Moreover, if the MAC protocol defined in the IEEE 802.11 standard is used, the problems make worse due to the capture effect induced by this protocol, leading to a larger degree of unfairness and a further degradation of throughput. In this paper we develop a scheme which we call SecSplit TCP. This scheme separates the functionalities of TCP congestion control and reliable packet delivery. For any TCP connection, certain nodes along the route take up the role of being proxies for that connection. The proxies buffer packets upon receipt and administer rate control. The buffering enables dropped packets to be recovered from the most recent proxy. Introducing proxies we emulate shorter TCP connections and can thereby achieve As shown by our simulations, the use of proxies decreases the problems described; i.e., a) it improves the total throughput by as much as 30% in typical scenarios - grid view (straight movement of nodes) b) it reduces unfairness significantly. When comparing the performance of the two protocols, we infer that SecSplit-TCP outperforms Split-TCP by 50% in terms of delay, 4% in terms of delivery ratio, 60% in terms of drop and 48% in terms of throughput.

We conclude that incorporating TCP proxies is beneficial in terms of improving the security TCP performance in mobile adhoc networks.

Keywords: Manets, SecSplit-TCP, TCP Proxies, Standard TCP, end-to-end.

INTRODUCTION

A manet is a collection of wireless nodes connection together to form a network. Every node has its own routing functionality when forwarding packet from one node to another. MANET nodes have stringent resource constrains and they are typically mobile, forming a highly dynamic network topology, absent of any clear network boundaries. However due to its dynamic nature and lack of infrastructure this kind of network is often susceptible to security attacks by malicious nodes [1][2][3].

One distinguishing characteristic of MANETs from the security design perspective is the lack of a clear line of defense. Unlike wired networks that have dedicated routers, each mobile node in an ad hoc network may function as a router and forward packets for other peer nodes. The wireless channel is accessible to both legitimate network users and malicious attackers. There is no well defined place where traffic monitoring or access control mechanisms can be deployed. As a result, the boundary that separates the inside network from the outside world becomes blurred [4] [6].

Figure (1): Mobile Ad hoc Network (MANET) sample

II. AN OVERVIEW OF SPLIT-TCP

In this section, we provide an overview of how TCP proxies work, and provide qualitative arguments that
show the motivation behind their use. Proxies split a TCP connection into multiple local segments. They buffer packets and deliver them to the next proxy or to the destination. Each proxy receives packets from either the source (A proxy P1 receives packets from S in Figure 2) or from the previous proxy, sends LACKs for each packet to the sender (source or proxy) of that packet (as an example in Figure 2, the second proxy P2, upon receiving a packet, sends a LACK for that packet to P1), buffers the packet, and when possible, forwards the packet towards the destination, at a rate proportional to the rate of arrival of LACKs from the next local segment. The source keeps transmitting according to the rate of arrival of LACKs from the next proxy, but purges a packet from its buffer only upon receipt of an end-to-end ACK for that packet (note that this might be indicated in a cumulative ACK for a plurality of packets) from the destination. This essentially splits the transport layer functionalities into that of congestion control and end-to-end reliability. Correspondingly, we propose to split the transmission window at the source into two windows, the congestion window and the end-to-end window. The congestion window would always be a sub-window of the end-to-end window. While the congestion window changes in accordance with the rate of arrival of LACKs from the next proxy, the end-to-end window will change in accordance with the rate of arrival of the end-to-end ACKs from the destination. The dynamics of both these windows vary as per the rules that govern traditional TCP subject to the condition that the congestion window stays within the end-to-end window. At each proxy, there would be a congestion window which would govern the rate of sending between proxies. We suggest that these end-to-end ACK’s be infrequent (one end-to-end ACK for every 100 or so packets that are received by the destination), since the likelihood of a proxy failure might be expected small. We elaborate on the advantages of TCP proxies with regards to alleviating the two effects that cause TCP to perform poorly: (a) mobility, and (b) the link capture effect of the 802.11 MAC protocol.

### Dealing with Mobility:
SecSplit-TCP can handle mobility better than the plain TCP. Mobility in MANETs manifests itself as link failures. As the length (in hops) of a particular session increases, the possibility of link failures on that path also increases. One link failure can cause an entire TCP session to choke, when in fact packets can be transferred on other links that are still up. Split TCP helps take advantage of these links that are up. When a link on a local segment fails, it is possible for TCP with proxies to sustain data transfer on other local segments. Thus, the hit on TCP throughput due to mobility is of much lower impact.

We point out that the higher probability of link failures on longer paths (as mentioned) causes an unfair disadvantage to long TCP sessions when compared with shorter TCP sessions. By splitting the long TCP session into shorter local segments, we essentially create a scenario in which all TCP sessions are of short length. Thus, we can expect that our scheme improves the fairness among TCP sessions in the network.

### Dealing with the link capture effect:
If the IEEE 802.11MAC protocol is used in conjunction with TCP, it causes the channel capture effect. If we have two simultaneous TCP sessions that are initiated in the geographical vicinity of each other, and are both heavily loaded, this effect provides an advantage to one of the sessions at the expense of the other.
unfair advantage to the session that originated earlier or to the session that is of fewer hops.

III Protocol performance evaluation

3. Simulation Results

3.1 Simulation Parameters

We use NS2 to simulate our proposed SecSplit-TCP protocol. We use the IEEE 802.11 for wireless networks as the MAC layer protocol. It has the functionality to notify the network layer about link breakage. In our simulation, the packet size is varied as 250, 500, 750, 1000 and 1250. The area size is 1300 meter x 1300 meter square region for 50 seconds simulation time. The simulated traffic is Constant Bit Rate (CBR).

Our simulation settings and parameters are summarized in table 1

Table 1: Simulation parameters for Grid Architecture

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Nodes</td>
<td>64</td>
</tr>
<tr>
<td>Area</td>
<td>1300 X 1300</td>
</tr>
<tr>
<td>MAC</td>
<td>802.11</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>50 sec</td>
</tr>
<tr>
<td>Traffic Source</td>
<td>CBR</td>
</tr>
<tr>
<td>Packet Size</td>
<td>250, 500, 750, 1000 and 1250</td>
</tr>
<tr>
<td>Propagation</td>
<td>TwoRayGround</td>
</tr>
<tr>
<td>Antenna</td>
<td>OmniAntenna</td>
</tr>
<tr>
<td>Rate</td>
<td>50Kb</td>
</tr>
</tbody>
</table>

Table 2: Simulation parameters for Non-Linear Architecture

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Nodes</td>
<td>20, 40, 60, 80 and 100</td>
</tr>
<tr>
<td>Area</td>
<td>1300 X 1300</td>
</tr>
<tr>
<td>MAC</td>
<td>802.11</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>50 sec</td>
</tr>
<tr>
<td>Traffic Source</td>
<td>CBR</td>
</tr>
<tr>
<td>Packet Size</td>
<td>250, 500, 750, 1000 and 1250</td>
</tr>
<tr>
<td>Propagation</td>
<td>TwoRayGround</td>
</tr>
<tr>
<td>Antenna</td>
<td>OmniAntenna</td>
</tr>
<tr>
<td>Rate</td>
<td>50Kb</td>
</tr>
</tbody>
</table>

3.2 Performance Metrics

We evaluate performance of the new protocol mainly according to the following parameters. We compare our SecSplit-TCP [6] protocol with Split-TCP protocol.

Average Packet Delivery Ratio: It is the ratio of the number of packets received successfully and the total number of packets transmitted.

Throughput: The throughput is the amount of data that can be sent from the sources to the destination.

Packet Drop: It is the number of packets dropped during the data transmission

Delay: It is the time taken by the packets to reach the destination.

3.3 Results & Analysis

The simulation results are presented in the next section.

Case-1 (Grid)

A. Based on Packet Size

In our experiment we vary the packet size as 250, 500, 750, 1000 and 1250 bytes.

Fig 3: Packet Size Vs Delay

Fig 4: Packet Size Vs Delivery Ratio
Figures 3 to 6 show the results of delay, delivery ratio, packet drop and throughput by varying the packet size from 250 to 1250 for the TCP traffic in SecSplit-TCP and Split-TCP protocols. When comparing the performance of the two protocols, we infer that SecSplit-TCP outperforms Split-TCP by 50% in terms of delay, 4% in terms of delivery ratio, 60% in terms of drop and 48% in terms of throughput.

Case-2 (Non-Linear)
A. Based on Packet Size
In our first experiment we vary the Packet size as 250, 500, 750, 1000 and 1250.
B. Based on Nodes

In our second experiment we vary the number of nodes as 20, 40, 60, 80 and 100.

Figures 10 to 11 show the results of delivery ratio and throughput by varying the number of nodes from 20 to 100 for the TCP traffic in SecSplit-TCP and Split-TCP protocols. When comparing the performance of the two protocols, we infer that SecSplit-TCP outperforms Split-TCP by 3% in terms of delivery ratio and 64% in terms of throughput.

IV. CONCLUSIONS

In this paper, we propose a new promising approach to improve the performance of TCP in terms of fairness and throughput in MANETs. We propose to achieve this by introducing proxy agents that Secsplit-TCP into localized segments. Our new version of TCP is called SecSplit TCP. The proxy agents facilitate the separation of the congestion control and the end-to-end reliability semantics of TCP. The introduction of proxy agents especially benefits longer connections. To summarize, TCP proxies succeed in terms of achieving a higher throughput and providing better fairness to longer TCP connections with respect to shorter ones. We show by means of simulations that SecSplit TCP can improve both the fairness among TCP connections (by a factor of 60%) and the throughput (by about 5% to 40%) of individual TCP connections.

REFERENCES