TCP Performance Analysis by Loss Differentiation Algorithm over Lossy Wireless Links

Mr Chandan H. Purohit 1 Professor Ridhdhi I Satoniya 2
1,2Department of Electronics & Communication
Darshan Institute of Engineering & Technology, Rajkot.
Gujarat Technological University, Ahmedabad.

Abstract--- TCP continues to be an important transport-layer communication protocol that is typically tuned to perform well in traditional wired networks, where bit error rate is low and congestion is the primary cause of packet loss. Wireless networks are prone to High BER, which is responsible for large number of packet losses. TCP incorrectly interprets these losses as signs of network congestion and throttles its transmission rate unnecessarily, which degrades end-to-end performance in wireless networks. Objective is to distinguish between congestion loss and wireless loss and to fine tune the TCP to perform well in wireless environment using Loss Differentiation Algorithm.

Keywords:- LDA, UMTS, Ns2, RTT, DUPACK, Congestion Window, Slow Start Threshold, BER.

I. INTRODUCTION

From the very beginning, mobile wireless networks and data networks such as Internet developed separately. With the GPRS (General Packet Radio System), we could see a beginning of unification, where the mobile terminals could access a small part of the existing data services, but the main shortfall of these technologies was the small available bandwidth. This is the reason why the claim of the new 3G (Third Generation) networks is to deliver a high data rate service. Those who want to speak about the Internet, have to speak about TCP (Transmission Control Protocol). Indeed, TCP accounts for about 95% of the bytes, 90% of the packets and 80% of the flows on the Internet. Therefore, this kind of data traffic will be a main part of the traffic carried by the emerging 3G technologies.

TCP, the most widely used protocol in the transport layer on the Internet (e.g. HTTP, TELNET, and SMTP), plays an integral role in determining overall network performance. The design of TCP was heavily influenced by what has come to be known as the End-to-End argument. The key component of the End-to-End argument for the purposes is in its method of handling congestion and network overload. Wireless networks are becoming increasingly popular in the world of telecommunication. As a consequence, a significant effort has been devoted to the provisioning of reliable data delivery for a wide variety of applications over different wireless infrastructures. One of the major challenges in modern communication system is to provide wireless access to the Internet. TCP supports the most popular suite of applications on the Internet today and it has been enhanced in recent years to improve robustness and performance over network of varying capacities and quality.

II. MOTIVATION

The Existing versions of TCP, like Reno or NewReno, experience heavy throughput degradation over channels with high error rate, such as wireless channels. The main reason for this poor performance is that TCP congestion control mechanism cannot distinguish between packet losses occurring randomly in wireless channels and those due to network congestion. In other words, the assumption that packet loss is always an indicator of network congestion does not apply especially to heterogeneous networks that include wireless links, in which packet loss may be induced also by noise or any other reason other than congestion. These random losses due bit corruption are misinterpreted: upon loss detection, the TCP sender reduces its transmission rate unnecessarily thus degrading the throughput performance.[6][7]

III. LITERATURE REVIEW

Given here is an overview of some optimizations that have been proposed in the literature and describe how they differ in terms of their retransmission and recovery mechanisms... The proposed optimizations can be categorized into four groups; split-connection, link-layer, explicit notification and end-to-end protocols.

A. Split Connection Protocols: The idea of split-connection approaches is to divide each TCP connection into two separate connections at an intermediate node; one is between the fix host (FH) and the base station (BS) and the other between the BS and mobile host (MH). [1]

The transmission characteristics of the wireless link such as high bit error probability and disruptions due to radio shadow or handoffs influence only the transport connection over the wireless hop. This way any effects due to wireless link can be hidden from nodes within the wire-line network and makes it possible for the wire-line and wireless medium to be employed with different transport protocols. However, since TCP connections are terminated at the base station, data buffers and TCP state information should be maintained at the base station in any split connection scheme. I-TCP, M-TCP and METP are examples of SplitConnection Protocols.

B. Link Layer Protocols: Link-layer protocols are another alternative for improving the poor performance of TCP over wireless links. The concept behind this is to make the wireless link layer look similar to the wired case for TCP by recovering the wireless error locally [5]. There have been several proposals for reliable link-layer protocols. The main techniques employed by these protocols are forward error correction (FER) and automatic repeat request (ARQ). ARQ is frequently used in data communications protocols. When a frame is detected to contain errors after decoding, it is discarded and an ACK is sent back to the sender requesting a retransmission of the frame. This is called a selective retransmission and the most efficient way of retransmission. TCP SNOOP, DDA, Radio Link Protocol are examples of


All rights reserved by www.ijsrd.com
C. Explicit Notification Protocols: A various explicit notification schemes have been proposed to enable the TCP sender to distinguish between different types of packet losses. The receiver explicitly informs the sender about the cause of packet loss, primarily to discriminate between congestion and wireless losses, so that the sender can take appropriate steps, depending on the nature of error reported by the receiver. The receiver, in turn, depends on intermediate routers or BSs to get the information about packet drop. The intuition behind this approach is that there is a known between congestion and wireless congestion and when it disappears, or the state of the link it is controlling. So the router not only knows that a packet is dropped it also knows the reason of that packet drop. By using this information, TCP is made aware of the fact that some of the losses occurred are not due to congestion and thus prevents TCP from invoking the congestion control algorithm each time it detects a loss. However, this scheme may fail when the path between the FH and the MH is changed frequently [5]. Examples of this approach include Explicit Congestion Notification (ECN), Explicit Loss Notification (ELN).

D. End-To-End Protocols: The standard TCP implementations rely on packet loss as an indicator of network congestion and lack the ability to distinguish congestion losses from losses invoked by noisy links. In wireless connections, overlapping radio channels, signal attenuation and additional noises have a huge impact on such losses. As a consequence, the standard TCP reacts with drastic reduction of the congestion window, thus degrading the performance of TCP. End-to-end proposals make the TCP sender handle packet losses caused by both congestion and random wireless errors and requires minimal or no processing at the base station. Another advantage of these schemes is that the end-to-end semantics of TCP is maintained.

IV. CONGESTION CONTROL SCHEMES
The end-to-end approach can have its congestion control mechanism realized in two ways, Reactive and Proactive. By reactive congestion control, the sender rectifies the congestion window when the network situation becomes marginal or has crossed a threshold. By proactive congestion control, feedbacks from the network guide the sender to reallocate network resources in order to prevent congestion.

A. Reactive Congestion Control: The congestion window is adjusted based on the collective feedback of ACKs and DUPLICATE ACKs generated at the receiver. TCP probes for the available bandwidth by continuously increasing the congestion window gradually until the network reaches the congestion state. In this sense, congestion is inevitable. TCP will then fall back to a much slower transmission rate, which may be unnecessary for wireless random loss. TCP Tahoe, TCP Reno and TCP NewReno are the examples of reactive congestion control algorithms. [2]

B. Proactive Congestion Control: In proactive congestion control, the sender attempts to adjust the congestion window proactively to an optimal rate according to the information collected via feedback, which can be translated to an indication of the network condition. By doing so, the sender reacts intelligently to the network condition or the cause of the packet loss, due to either congestion or random errors, and therefore prevents the network from entering an undesired state (e.g., congestion or unnecessary decrease of the congestion window). TCP Vegas, TCP Veno, TCP Spike, etc. are the examples of Proactive Congestion control Algorithms.[2]

V. TCP ENHANCED WITH LOSS DIFFERENTIATION
Loss Differentiation Algorithms (LDA) are used to provide TCP with an estimate of the cause of packet loss, to improve performance over heterogeneous networks including wired and wireless links. As per the literature survey, I found that most of schemes proposed exhibits poor performance in estimating the cause of packet loss. So, I propose the implicit end to end Spike [3] Scheme, which achieves higher accuracy in all the scenarios. The loss differentiation decision can be obtained by processing TCP state variables, such as congestion window (cwnd), slow start threshold (ssthresh), round trip time (RTT), bandwidth estimation, etc.

VI. TCP SPIKE SCHEME
The Round Trip Time (RTT) is a measure of the time a packet takes to travel from the sender to the receiver. Since the sending and receiving times are measured at the sender and receiver separately, the absolute value of delay is difficult to obtain due to the clock skew between the two, thus the name “relative.”

Fig. 1: Generic scheme of a Loss differentiation algorithm

Fig. 2: TCP SPIKE Scheme
The RTT is used to identify the state of the current connection. If the connection is in the spike state, losses are assumed to be due to congestion; otherwise, losses are assumed to be wireless. The spike state derives its name from the fact that plots of RTT vs. time tend to show spikes during periods of congestion. The spike state is determined as follows. On receipt of a packet with sequence number i, if the connection is currently not in the spike state, and the
RTT for packet exceeds the threshold $B_{\text{spikestart}}$, then the algorithm enters the spike state.

![SPIKE Schematic](image)

Otherwise, if the connection is currently in the spike state, and the RTT for packet $i$ is less than a second threshold, $B_{\text{spikestart}}$, the algorithm leaves the spike state. When the receiver detects a loss because of a gap in the sequence number of received packets, it classifies the loss based on the current state.

$$B_{\text{spikestart}} = RTT_{\text{min}} + \frac{1}{2} (RTT_{\text{max}} - RTT_{\text{min}}) \quad [3]$$

$$B_{\text{spikeend}} = RTT_{\text{min}} + \frac{1}{3} (RTT_{\text{max}} - RTT_{\text{min}}) \quad [3]$$

Where, $RTT_{\text{max}}$ and $RTT_{\text{min}}$ is the maximum and minimum relative one-way trip time. To use these formulas, we need to determine values for the parameters $\alpha$ and $\beta$ (suggested values are 0.5 and 0.33). Suppose we consider all buffers along the route from sender to receiver as one big buffer. The $RTT_{\text{min}}$ occurs when buffer is empty and $RTT_{\text{max}}$ occurs when buffer is full [3]. The current value of RTT is compared with two thresholds, to detect if the connection is experiencing congestion or not, according to following rules:

- If the current values of the RTT is greater then $B_{\text{spikestart}}$, ($RTT > B_{\text{spikestart}}$), the scheme assumes that the RTT is too high and a loss due to congestion is imminent. If the current value of RTT is less then $B_{\text{spikeend}}$, ($RTT < B_{\text{spikeend}}$), the connection is experiencing a low delay, and eventual packet losses are ascribed to random impairments on the wireless channel. Finally, if the RTT is comprised between $B_{\text{spikestart}}$ and $B_{\text{spikeend}}$ ($B_{\text{spikestart}} \leq RTT \leq B_{\text{spikeend}}$), the spike scheme assumes that network state is not changed from the previous state.

VII IMPLEMENTATION

Table I: MODIFIED FILES AND FUNCTIONS IN NS-2

<table>
<thead>
<tr>
<th>File</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp.cc</td>
<td>void newack(Packet*)</td>
</tr>
<tr>
<td></td>
<td>void slowdown(int)</td>
</tr>
</tbody>
</table>

VIII RESULT AND ANALYSIS

The key feature of Loss Differentiation Algorithm is to be accurate in ascribing the cause of packet losses, as TCP recovery based on Differentiation reacts gently or aggressively depending on the LDA decision. To summarize, when packet random error rate is low and most of the packet losses are due to congestion, LDA accuracy in ascribing losses is necessary to achieve fairness with concurrent TCP flows. On the other hand, packet error rate is high such as in wireless links, LDA accuracy is necessary in order to achieve throughput gain.

A. Network Scenario:

![Wired Cum Wireless Topology](image)

B. Simulation Parameters:
- S - Source  R - Router  D – Destination
- Simulation End time - 50 sec
- MAC layer Data Rate - 54 Mb
- Link Layer Delay - 2 msec
- TCP type - NewReno and SPIKE

C. Performance Comparison

<table>
<thead>
<tr>
<th>BER</th>
<th>Throughput (NewReno) (Kbps)</th>
<th>Throughput (SPIKE) (Kbps)</th>
<th>Dropped packets (NewReno)</th>
<th>Dropped Packets (SPIKE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>37.876</td>
<td>445.969</td>
<td>96</td>
<td>104</td>
</tr>
<tr>
<td>0.02</td>
<td>286.871</td>
<td>319.783</td>
<td>148</td>
<td>176</td>
</tr>
<tr>
<td>0.04</td>
<td>204.812</td>
<td>220.164</td>
<td>222</td>
<td>234</td>
</tr>
<tr>
<td>0.06</td>
<td>133.716</td>
<td>65.9393</td>
<td>216</td>
<td>66</td>
</tr>
<tr>
<td>0.08</td>
<td>105.241</td>
<td>16.8713</td>
<td>218</td>
<td>48</td>
</tr>
<tr>
<td>0.10</td>
<td>80.6055</td>
<td>1.15693</td>
<td>104</td>
<td>4</td>
</tr>
<tr>
<td>0.12</td>
<td>1.48661</td>
<td>13.6688</td>
<td>6</td>
<td>28</td>
</tr>
<tr>
<td>0.14</td>
<td>1.48661</td>
<td>4.17697</td>
<td>6</td>
<td>20</td>
</tr>
</tbody>
</table>

Table II: NUMERICAL RESULT

![Graph of Throughput comparison – NewReno v/s SPIKE](image)
TCP Performance Analysis by Loss Differentiation Algorithm over Lossy Wireless Links
(IJSRD/Vol. 2/Issue 03/2014/196)

D. Result Analysis: Numerical Results above shows that as compared to NewReno, throughput is higher in case of TCP SPIKE in presence of high BER. The reason for High Throughput is that during Error Loss, TCP does not reduce the value of Congestion Window. The graphs of CWND v/s time for NewReno and SPIKE shows that value of Average Congestion window is higher in case of TCP SPIKE. But, for some case of BER, Throughput of SPIKE reduces though the No. of dropped packets are less. The reason for this reduction in throughput is Self Introduces Congestion by SPIKE. Whenever the Interface Queue (IFQ) gets full, No. of packets start dropping due to congestion. Thus, TCP enters in the phase of Congestion Loss recovery by reducing the value of congestion window and hence, the throughput degrades.

VII. CONCLUSIONS
In wired network congestion is main issue for packet loss. In wireless networks, there are several issues for packet loss like High BER, handoff, limited bandwidth, etc. But, due to lack of intelligence TCP treats packet loss in wireless networks as a loss due to congestion. Thus, unnecessarily it reduces flow rate and network performance degrades. SPIKE is an optimized scheme which discriminates the congestion and wireless losses intelligently and also provides higher throughput but it introduces self-congestion and some loss occurs due to misclassification. Thus it fails to provide higher throughput in some cases.

For cases of BER 0.01, 0.02, 0.04, 0.12 and 0.14 Throughput is higher, but no. of packet drops is also higher which is not a good sign for TCP, because as the no. of packets drop is increased, TCP has to go for loss recovery. Thus, new flavour can be implemented to reduce no. of packet drops with higher throughput.

For cases of BER 0.06, 0.08 and 0.10 losses are misclassified as well as the problem of self-induced congestion occurs due to which throughput degrades. Thus, new Flavour can be implemented so as to reduce the problem of self-induced congestion and misclassification and enhance the throughput.

VIII. ACKNOWLEDGMENTS
With deep sense of gratitude, I express my sincere thanks to my esteemed and worthy supervisor, Prof. Ridhdhi I. Satoniya, Assistant Professor, Department of Electronics & Communication Engineering, Darshan Institute of Engineering & Technology, Rajkot for her valuable guidance in carrying out this work under her effective supervision, encouragement, enlightenment and cooperation. Her vision and resourcefulness have been a constant inspiration to me. The ease with which she transcends the boundaries of individual subjects has helped me widen my perspective and realizing the true meaning of research. I would like to express here also my profound appreciation for all the staff and academicians of Darshan Institute of Engg. & Tech. Rajkot for sharing their knowledge and experiences and being helpful to me directly or indirectly.

IX. REFERENCES