An Effective Mechanism for Improving Performance of TCP over LEO/MEO Satellite IP Networks

Tengyue Mao
College of Computer Science, South-Central University for Nationalities, Wuhan, Hubei, China, 430074
Email: 734813320@qq.com

Zhengquan Xu
State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, Hubei, China, 430079
Email: xuzq@whu.edu.cn

Abstract—TCP protocol is widely used in terrestrial Internet, but the terrestrial TCP protocol satellite cannot be directly applied to satellite network with high-transmission delay and high bit error rate. The congestion control is the most core mechanism of TCP protocol, and has direct impact on the overall performance of transmission. The improved scheme of existing TCP control mechanisms such as TCP-Vegas, TCP-Peach and TCP-westwood have improved congestion control mechanisms and error control mechanism of TCP from different angles in order to make it adapt to satellite link's characteristics like signal attenuation and high bit error rate. However, there exist some gaps between these strategies in the multimedia service transmission and user's needs. Based on LEO/MEO network structure, aiming at the issues in transmission such as the slow growth of the congestion window, the low utilization of bandwidth and constrained throughout, this paper put forward an improved algorithm TCPQ (TCP QoS) of transmission on the basis of multi-indicators guarantee. TCPQ includes three parts: accelerated start algorithm, nonlinear congestion avoidance algorithm, and adaptive threshold error-identification algorithm. Analysis and simulation experiments show that TCPQ algorithm can make full use of link resource and have a better performance in terms of throughout, transmission efficiency, etc.

Index Terms—Congestion Control; Nonlinear; LEO/MEO; TCP

I. INTRODUCTION

TCP protocol which is widely used in terrestrial and has a good performance can be applied to satellite network. Unlike terrestrial links, satellite links have some features like high error, a long delay and asymmetric bandwidths. As a result, the performance of TCP protocol with a good performance in the ground is greatly affected when TCP protocol with a good performance in the ground operates in the satellite network. Therefore, the traditional TCP/IP protocol in the ground is difficult to manifest its advantages in satellite network. It is necessary to combine with characteristics of satellite network and improve or redesign the protocol of transmission layer [1, 2].

TCP protocol operates beyond the network layer, providing reliable end-to-end services. Its basic working mechanisms include congestion control, traffic control and error control, among which congestion control is the core mechanism. Four algorithms consist of slow start, congestion avoidance, fast retransmission and fast recovery. As TCP protocol has been quite widely used in the terrestrial Internet and has shown a good performance, to use TCP protocol for data transmission has become the mainstream of network applications.

From the perspective of the current researches, the main work has focused on improving or optimizing transmission control protocol of the current terrestrial internet. TCP-Vegas [3] proposed a method which uses a loop response time to control congestion. Its basic idea is to control the congestion window by observing the changes of values of loop responses in TCP connection so as to make the congestion window stabilize at an appropriate value. However, the cause of packet loss judged by TCP-vegas is link error instead of congestion, where there exist limitations of a single judgment. TCP-Westwood [4] proposed a more accurate bandwidth estimation algorithm. ACK stream through monitoring determines the available bandwidth. Once the packet loss occurs, restore the window quickly to the appropriate level of bandwidth, so as to eliminate the impact of the high error rate. But when delay increases, the performance of TCPW drops quickly, which limits the application of TCPW in satellite network. The authors [5] proposed a recursive, explicit, fair and window adjustment (REFWA) protocol, which may improve the efficiency of satellite systems and TCP fairness. Literature [7] proposed a new TCP congestion control mechanism (TCP-Cherry) in order to improve TCP performance in satellite IP networks. Compared with other existing programs, TCP-Cherry has obtained more...

II. LEO/MEO NETWORK MODEL

Kimura’s [14] DLSC model is the representative of double networks. Considering network stability, the model requires satellite in every layer to establish redundant connections. However DLSC model overly depends on redundant connections to increase network stability, and it leads to excessively higher complexity of satellite network systems. Based on the limitations of DLSC model, the structure of this section is as follows: use backbone / access network model and the “moderate Connection” ISLs to ensure network stability, while making use of group management model and slot allocation rules to reduce the complexity of satellite network.

A. Backbone/Access Network

In existing multi-satellite networks, "Redundant Connection" model has a defect: high complexity defect. “Backbone / access” model” proposed in this section simplifies multi-satellite network structure through backbone layer and access layer. The structure of LEO/MEO double-layer satellite network shown in Figure 1.

![Figure 1. The structure of LEO/MEO double-layer satellite network](image)

B. LEO/MEO Network Parameters

Coverage, delay and delay jitter of satellite network are the factors that should be taken seriously in structure design of satellite network. These factors directly affect transmission guarantee performance of satellite network. The table of LEO / MEO satellite parameters is shown in Table 1.

<table>
<thead>
<tr>
<th>Layer of satellite</th>
<th>Height of the orbit</th>
<th>Satellite number</th>
<th>Obliquity of the orbit</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEO</td>
<td>1414</td>
<td>8x4</td>
<td>52°</td>
</tr>
<tr>
<td>MEO</td>
<td>10390</td>
<td>4x2</td>
<td>45°</td>
</tr>
</tbody>
</table>

III. IMPROVED ALGORITHM

TCPQ algorithm consists of three parts: accelerated start algorithm, nonlinear congestion avoidance algorithm and error detection algorithm. The overall structure of the algorithm is shown in Figure 2.

![Figure 2. The overall structure of TCPQ algorithm](image)

Aiming at slow start of TCP-Reno and TCP-NewReno and small initial window of sudden start strategy of TCP-Peach protocol, accelerated start algorithm makes start more quickly by optimizing slow start algorithm. Aiming at the issue that the stability of TCPW is not enough when delay is larger, nonlinear congestion avoidance algorithms makes window changes adjust dynamically in real time according to estimation value of network bandwidth through a nonlinear growth so as to obtain better utilization of bandwidth and better stability. Aiming at cause judgment of single packet loss of Reno, TCPW and Vegas, adaptive threshold error detection algorithm sets a reasonable threshold and congestion control window by using the values bandwidth estimates, and distinguish network congestion so as to determine the causes of the loss of packet segment [15].

A. Accelerated Start Algorithm

Generally, the slow start time of TCP is a number of RTT , about hundreds of milliseconds for satellite networks with large delay. In previously designed LEO/MEO double-layer networks, the end-to-end maximum delay of the system is about 285 milliseconds. This delay leads to the fact that slow start needs long time, and it takes a very long time to reach full capacity throughout.

To increase the value of the initial window can reduce the time the slow start need. After having increased the
initial window, within the first $RTT$ more data can be sent and the congestion window will increase even faster. When at the receiving end $ACK$ is sent for each segment, assume that $RTT$ stays unchanged, and no congestions occur in the process of $cwnd$ growth, the time for the slow start is: $RTT \cdot \log \frac{rwnd}{cwnd} \cdot 2$.

By increasing the initial value of $cwnd$, within the first $RTT$ more data packet can be transmitted, and the sender receives more recipient's response so that the congestion window is quickly increased. In RFC2581 the value of the initial window value is allowed to increase to $2 \cdot MSS$, particularly in the satellite network.

Considering delay and bandwidth of LEO/MEO double-layer network, the value of the initial window is set to $2 \cdot MSS \cdot ssthresh = rwnd$. After sender and receiver have been established connection successfully, the sender first set initial $cwnd = 2$, and the sender first enters into accelerated start-up phase and begins sending data. In order to reduce the instantaneous congestion burden of the network caused by data bursts, TCPQ algorithm uses the strategy which sends segment intervally. That is to say, the sender passes a length of time every time $\tau$, send a message segment. In it:

$$\tau = RTT / cwnd = RTT / 2$$  \hspace{1cm} (1)

After having sent the $rwnd$ data, the sender stops sending packets segment temporarily, waiting for NACK response. When receiving a NACK response from the receiver, the sender will increase the value of the congestion window to $rwnd$, and then recalculate sending interval $\tau$ of packet segment according to (1), and ends accelerated start state and enters into the state of congestion avoidance in advance.

From the above analysis, we can see that accelerated start strategy can make the value of congestion window increase quickly to the maximum within the lengths of time of $2 \cdot RTT$, which is $cwnd = rwnd$.

Accelerated start algorithm is responsible for the increase of the transmission rate and the data transmission of TCPQ algorithm after the connection is established. In order to compare with TCP-Reno, TCP-Peach, TCPWest-wood protocol, the simulation environment is as follows: each link is $10 \ Mbits/s$, $RTT$ is $250 \ ms$, the length each TCP data segment is $1000 \ bytes$, the value of the receiving window of the receiver is $rwnd = 64$ and the length of simulation step length is $50 \ ms$. The following makes comparison between accelerated start algorithm with TCP-Reno and TCP-Peach from the perspective of the start-up time and size indicators of the initial window, as shown in Figures 3 and 4.

From Figure 3 and Figure 4 we can conclude that comparing with slow start strategy used by TCP-Reno and TCP-NewReno protocol, $cwnd$ in accelerated start strategy only need two $RTT$ and then reach $rwnd$ value. The performance of sudden start strategy of TCP-Peach protocol is equal to the accelerated start strategy in TCPQ, but the initial value is $1$ which is too small. The value of the initial congestion window of accelerated start strategy is: $cwnd = rwnd / 2$. Therefore, at the initial stage the quantity of data transmission of accelerated start strategy is greater than that of the sudden strategy.

B. Nonlinear Congestion Avoidance Algorithm

Aiming at high bandwidth and long delay, based on TCPW algorithm, this section makes window changes adjust dynamically in real time according to estimation value of network bandwidth through a nonlinear growth so as to obtain better utilization of bandwidth and better stability. Meanwhile, before the critical point of packet loss, using conservative window adjustment mechanism and detecting possibility of congestion occurrence in advance, do not make the congestion of the network occurs and avoid segment loss due to congestion.

1) Nonlinear Avoidance Strategy

TCPQ nonlinear congestion avoidance strategy is described as follows: when each sender receives a response message, calculate and obtain the difference $B$ between the available bandwidth and the actual bandwidth, and calculate $window_i$. When the difference $B$ between the available bandwidth and the actual bandwidth is greater than or equal to $3$, the growth of the window increases rapidly in a nonlinear way, which belongs to rapid growth phase of the window. When the value of the window the available bandwidth corresponds to be similar to the value of the current window, it belongs to slow growth phase of the window [16, 17].

In it, the available bandwidth $BE$ the sender has measured is:

$$BE = cwnd / RTT_{\min}$$  \hspace{1cm} (2)

the actual bandwidth $BA$ is:

$$BA = cwnd / RTT$$  \hspace{1cm} (3)

the difference of their bandwidth is:

$$B = BE - BA$$  \hspace{1cm} (4)
Introduce nonlinear increasing coefficient $\zeta$:

$$\zeta = B / BE$$

(5)

the window value of the available bandwidth corresponds to:

$$\text{window} = \frac{BE \cdot RTT \min}{\text{seg _ size}}$$

(6)

The specific algorithm is as follows:

$$\text{cwnd} = \text{cwnd} + \frac{1}{\text{cwnd}} \cdot (1 + \zeta)$$

(7)

$$\text{window} = \text{cwnd} + \frac{1}{\text{cwnd}}$$

(8)

$$\text{cwnd} = \text{cwnd}$$

(9)

$$\text{window} = \text{cwnd}$$

(10)

Shown in Figure 2, after having received a response message determines whether there is data loss. If the data is lost, then performs error judgment stage, entering different states respectively according to judgment results.

2) The Comparison and Analysis of the Throughput when There is No Error

When error rate is 0, it is unnecessary to take an error into account, and packet loss is entirely caused by congestion. At this time, to determine whether window is set reasonable is the most accurate. If the window is set unreasonable, the throughput generated by congestion fluctuates greatly. Assume that each link is 100 Mbps, RTT is 350 ms, the length of each TCP data segment is 1000 bytes, the value of the receiving window of the receiver is rwnd = 64, the simulation step size is 50 ms. Next we compare TCPW and TCPQ from the perspective of throughput indicators, shown in Figure 5.

![Figure 5](image)

Figure 5. Comparisons between TCPW and TCPQ from the perspective of throughput when there is no error

IV. COMPARISON AND ANALYSIS OF PACKET LOSS RATE WHEN THERE IS AN ERROR

Take BER = 0.001 and compare TCPW with TCPQ, shown in Figure 6:

![Figure 6](image)

Figure 6. Comparison between TCPW and TCPQ from the perspective of the throughput when there is an error

When RTT is quite small, the error increases TCPW throughput, but when RTT increases, it will quickly reduce TCPW throughput. Although TCPW congestion control strategy is more accurate than TCP, TCPW adjusts the window through packet loss. Simulation results show that when RTT increases, TCPW can not timely converge to the steady state, and the throughput performance of the system is not more stable than TCPQ.

A. Adaptive Threshold Error-Identification Algorithm

In terms of error-identification strategy, Reno and TCPW think that all segment losses are caused by congestion. On the contrary, Vegas think that packet loss is caused by link errors instead of congestion. In satellite networks, it is hard to judge the cause of packet loss accurately. The cause of packet loss cannot just be determined as error or as congestion. So the premises of a
single cause put forward by Reno and Vegas have their own limitations [18]. Adaptive error-identification strategy absorbs Vegas’s design idea of how to determine the cause of packet loss and Reno’s selecting idea of using congestion control algorithm.

1) Adaptive Threshold Error-Identification Strategy

Firstly use buffer queue number at the switching code to estimate the available bandwidth of the network. Then identify the saturation of network bandwidth, find out the cause of packet loss and take different strategies. After having received a response message, judge whether there is data loss and boot error-identification strategy. The procedure is as follows:

a) Calculate Buffer Queue Number at the Switching Code

\( \bar{N} \) represents a buffer queue at the switching node. According to the previous definition, the expected bandwidth of the current time is \( cwnd / RTT_{\text{min}} \), \( \alpha \) is the reference value of a buffer queue, and the value is 3. 

\[ \bar{N} = (cwnd / RTT_{\text{min}} - BE / \text{seg}_\text{size}) \times RTT_{\text{min}} \] (11)

b) Identification

If \( \bar{N} > \alpha \), judge that network bandwidth is saturate, congestion leads to packet loss, TCPQ enters into congestion recovery state and reset the window and threshold size;

If \( \bar{N} < \alpha \), network bandwidth is non-saturated, network errors lead to packet loss, and lost data is retransmission. The algorithm is performed as follows:

1) If receive three repeated ACK, modify \( ssthresh \):

\[ ssthresh = (BE \times RTT_{\text{min}}) / \text{seg}_\text{size} \] (12)

2) If \( cwnd > ssthresh \)

\[ cwnd = ssthresh \] (13)

3) If it is overtime:

\[ ssthresh = (BE \times RTT_{\text{min}}) / \text{seg}_\text{size} \]
\[ cwnd = 1 \] (14)

4) If \( ssthresh < 2 \)

\[ ssthresh = 2 \] (15)

c) The Comparison and Analysis of Utilization Rate of the Channel

To calculate the effective utilization rate of the channel, set up a duration time of a connection 600 s, 100 s began calculating. The channel BER is respectively set as \( 10^{-4}, 10^{-7}, 10^{-4}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2} \). The effective utilization rate of the channel statistics obtain is shown in Figure 8.

From Figure 8 we can see that the utilization rates of Vegas, Reno and TCPQ channel are relatively close when BER is quite small, while the utilization rates of Vegas, Reno and TCPQ channel differentiate significantly when Reno and Vegas declines significantly when BER increases to \( 10^{-4} \), but TCPQ is relatively stable. After error rate increases to \( 10^{-3} \), TCPQ falls markedly. In terms of the overall utilization performance of channel, TCPQ is higher than Reno and Vegas.

![Figure 8. Comparisons among Vegas, Reno and TCPQ in terms of utilization rate of the channel](image)

![Figure 9. Comparisons between TCPQ and TCPW in terms of congestion window](image)

We can see that under the error rate \( 10^{-4} \), TCPQ has a slightly larger value of congestion window than that of TCPW. At the initial stage of transmission, TCPQ reasonable bandwidth-threshold value detection mechanisms make the window rapidly increase to a larger value. Before the critical point of packet loss, TCPQ uses a conservative window adjustment mechanism, and makes network not congested through early detecting the possibility of congestion occurrence so as to avoid segment loss due to congestion. Since when the error leads to packet loss, instead of using TCPW way which blindly reduces congestion window, TCPQ sets the threshold value and congestion window in a reasonable range according to the estimation result of bandwidth, avoids judging congestion cause falsely, and erroneously reduces transmission rate.

V. conclusion

Aiming at the limitations of TCP-Vegas, TCP-Peach and TCP-westwood algorithms, combined with congestion problems TCPQ proposed accelerated start algorithm, nonlinear congestion avoidance algorithm and...
adaptive threshold error-identification algorithm. Theoretical analysis and simulation results show that TCPQ can rapidly increase the window value of data transmission in a very short period of time and effectively improve the efficiency of the channel's transmission. Aiming at high-bandwidth delay of satellite network, through a nonlinear mode of growth, the algorithm makes the window changes adjust dynamically in real time according to the estimation value of network bandwidth so as to obtain better bandwidth utilization rate. Before the critical point of packet loss, TCPQ uses a conservative window adjustment mechanism, and makes network not congested through early detecting the possibility of congestion occurrence so as to avoid segment loss due to congestion. In the current proposed transmission control protocol of satellite network, TCPQ algorithm can make full use of link resource and have a better performance in terms of throughout, transmission efficiency, etc.

ACKNOWLEDGMENT

The project was supported by Natural Science Foundation of Hubei province (No. 2013CFB444).

REFERENCES