An Improved Hybla Algorithm in Satellite Networks

Li Yang [1, 2, a], Kui Li [1, 2, b], Debin Wei [1, 2, c]

[1]College of Information Engineering, Dalian University, Dalian 116622, China;

[2]Communication and Networks key Laboratory, Dalian 116622, China.

[a]yangli945@126.com, [b]2583757602@qq.com, [c]weidebin@dlu.edu.cn

Abstract

For the Hybla algorithm, the congestion window growth in the long-delay satellite network is aggressive and it cannot match the degree of satellite network congestion, resulting in packet loss rate increasing. Hybla cannot distinguish the reason of packet loss in Satellite Network with high link error rate. Aiming at the above problems of Hybla algorithm, an improved Hybla algorithm based on bandwidth estimation and queuing delay is proposed—Hybla IM(improved method).The algorithm dynamically detects bandwidth estimate and compares it with the expected bandwidth, then dynamically adjust the window growth factor according to the comparison value. The above approach enables Hybla to achieve rapid growth of the congestion window in long-delay satellite networks and also make window growth to match network congestion.When packet loss occurs in the network, packet loss reason is distinguished through predicting network congestion by calculating packet congestion in satellite networks. reducing the number of calls that congestion control is caused by error packet loss.

Keywords

Bandwidth estimation; Queue delay; Bit error rate; Congestion control; Satellite network.

1. Introduction

With the increase in the demand for satellite communications and the rapid increase in the number of users accessing, people have put forward higher requirements for satellite networks to provide reliable and effective information transmission. Among them, the transport layer congestion control protocol is one of the important guarantees that satellite networks can meet the above requirements. At present, TCP, the most widely used transmission control protocol over the terrestrial Internet, is a window-based end-to-end congestion control mechanism. There are mainly TCP BIC [1], TCP Vegas [2], TCP Peach [3], TCPW [4], TCP Cubic [5].

Among many TCP congestion control algorithms, TCP Hybla introduced the concept of standardized round-trip delay. By modifying the growth mode of the window, the purpose of rapid increase of the congestion window in long-delay satellite networks was achieved, and the utilization of satellite network resources was improved. . So choosing Hybla algorithm as the research object of the congestion control algorithm has very important practical significance.

However, Hybla algorithm also has some problems. Through simulation analysis the literature [6] shows that although the Hybla algorithm can achieve high network throughput, its excessive window scale will cause serious buffer overflow.In addition window growth of Hybla cannot match satellite network congestion status because larger load causes longer delay that Easy to cause Hybla window value to become larger, easily leading to congestion[7]. And Hybla cannot distinguish the reason for packet loss in high bit error satellite network environment, resulting in frequent calls to the congestion mechanism and causing network resources cannot be fully utilized.

On the basis of analyzing the characteristics of satellite networks, this paper deeply studies the TCP Hybla congestion control algorithm. Aiming at the problem of Hybla algorithm in satellite network environment, an improved Hybla algorithm based on bandwidth estimation and queuing delay is proposed—Hybla IM.The algorithm alleviates the problem that growth of the Hybla congestion window cannot match the network congestion through estimating network bandwidth. At the same time, this algorithm distinguishes packet loss reasons by predicting the network congestion during packet loss through calculating the queue delay of the data packet in the satellite network, reducing the number of calls to the Hybla congestion control mechanism.

2. Hybla Algorithm Introduction

Before introducing the Hybla algorithm, first introduce the standard TCP congestion window formula $W(t)$ as follows:

$$W(t)=\begin{cases}2^{t/RTT}, & 0\le t<t_\gamma, \quad SS\\ \dfrac{t-t_\gamma}{RTT}+\gamma, & t\ge t_\gamma, \quad CA\end{cases} \tag{1}$$

SS indicates the slow start phase ,CA indicates the congestion avoidance phase, t_γ indicates the time when the congestion window reaches γ. The data transmission rate is $B(t)=W(t)/RTT$, then the standardized TCP data transmission rate is expressed as:

$$B(t)=\begin{cases}\dfrac{2^{t/RTT}}{RTT}, & 0\le t<t_\gamma, \quad SS\\ \dfrac{1}{RTT}\left(\dfrac{t-t_\gamma}{RTT}+\gamma\right), & t\ge t_\gamma, \quad CA\end{cases} \tag{2}$$

The conclusion that can be drawn from formula (2) is that the data transmission rate is smaller if the RTT is larger. In order to make the networks with long propagation delay obtain the same instantaneous transmission rate as faster as wired networks,this goal can be achieved in two steps: first,by making $W(t)$ independent of RTT through a time scale modification, then by compensating the effect of the division by RTT. Both the steps can be better described after introducing a normalized round trip time, ρ ,defined as:

$$\rho = RTT \big/ RTT_0 \tag{3}$$

Where RTT_0 is the round trip time of the reference connection to which we aim to equalize ourperformance. The former step is performed by multiplying by ρ the time (or the time elapsedfrom the reaching of the ssthresh), achieving a $W(t)$ independent of RTT. The latter, bymultiplying by ρ the congestion window resulting at the first step (including the original ssthresh, γ), to have $B(t)$ independent of RTT. In conclusion we have:

$$W^H(t)=\begin{cases}\rho 2^{\rho t/RTT}, & 0\le t<t_{\gamma,0}, SS\\ \rho\left[\rho\dfrac{t-t_{\gamma,0}}{RTT}+\gamma\right], & t\ge t_{\gamma,0}, \quad CA\end{cases} \tag{4}$$

Where the superscript H identifies the Hybla proposal. As a consequence of the modification introduced in the second step, the switching time $t_{\gamma,0}$,defined as the time at which the congestion window reaches the value $\rho\gamma$,is the same for every RTT, being $t_{\gamma,0}=RTT_0\log_2\gamma$, Taking the standard TCP connection as the reference connection, $RTT_0=25\text{ms}$ .From (4), the segment transmission rate $B^H(t)=W^H(t)/\text{RTT}$ of a TCP Hybla assumes thefollowing expression:

$$B^H(t) = \begin{cases} \dfrac{2^{t/RTT_0}}{RTT_0}, & 0 \le t < t_{\gamma,0},\ SS \\ \dfrac{1}{RTT_0}\left[\dfrac{t - t_{\gamma,0}}{RTT_0} + \gamma\right], & t \ge t_{\gamma,0},\quad CA \end{cases} \tag{5}$$

From Equation (5), it can be concluded that the instantaneous transmission rate of Hybla is no longer dependent on the RTT, and is equal to the transmission rate of the wired reference standard TCP connection.

As far as the feasibility of the proposed approach is concerned, let us observe that the Hybla congestion window dynamic can be obtained by modifying the congestion window update rules as follows:

$$W_{i+1} = \begin{cases} W_i + 2^{\rho} - 1,\ SS \\ W_i + \dfrac{\rho^2}{W_i}, \quad CA \end{cases} \tag{6}$$

Wi is the congestion window value when the ith ACK arrives.

3. TCP Hybla IM Algorithm Satellite Network Model

Although the existing terrestrial network congestion control mainly adopts the TCP Reno algorithm, the Reno algorithm congestion window increases slowly in the long-delay satellite network: the congestion window increases by 1 when an ACK is received during the slow start phase, the congestion window increases by when an ACK is received during the congestion avoidance. This growth pattern leads to slow window updates in long-delay satellite networks. For the Hybla algorithm, the congestion window increases by when an ACK is received during the slow start phase, the congestion window increases by when an ACK is received during the congestion avoidance. Because the long-delay satellite network makes the round-trip delay ,it makes and and the bigger the RTT is, the bigger and are. Hybla uses this new congestion window growth approach to achieve rapid growth in long-delay satellite networks, so the Hybla algorithm is more suitable for satellite networks than the Reno algorithm.Hybla algorithm has the problem that window growth cannot match network congestion so it cause network performance cannot be taken full advantage of. In the case of network congestion, aggressive window growth will increase network congestion and cause packet loss.Because of the high error rate of the satellite network links, the possibility of packet loss due to errors increases.The Hybla algorithm cannot distinguish the reason of packet loss, which leads to its need to go through multiple packet loss recovery phases, which seriously affects the increase of its congestion window and reduces the utilization of satellite resources. Aiming at the above problems of Hybla algorithm, an improved Hybla algorithm based on bandwidth estimation and queuing delay is proposed—Hybla IM.The algorithm dynamically detects bandwidth estimate and compares it with the expected bandwidth, then dynamically adjust the window growth factor according to the comparison value. The above approach enables Hybla to achieve rapid growth of the congestion window in long-delay satellite networks and also make window growth to match network congestion.When packet loss occurs in the network, packet loss reason is distinguished through predicting network congestion by calculating packet congestion in satellite networks. reducing the number of calls that congestion control is caused by error packet loss.

3.1 Adjustment of Window Growth Factor

Because of the long communication distance, the communication between the satellite node and the ground has a long propagation delay. Table 1 shows typical round-trip delays for LEO, MEO, and GEO satellite links.

Table 1 Round-trip delays of LEO, MEO and GEO satellites

Satellite network type	Round trip delay (ms)
LEO	50
MEO	250
GEO	550

The typical round-trip delay of the terrestrial network is 25ms. From Table 1, the round-trip delay of the connection varies from 25ms to 550ms in a mixed network including a terrestrial network and a satellite network. So the ground reference connection delay is set to 25ms.

From the formula (3) we can see that the value of ρ varies from 1 to 22. So the window increment is in the range:

$$increment = \begin{cases} increment \in \left(1, 2^{22} - 1\right), SS \\ increment \in \left(\frac{1}{w_i}, \frac{484}{w_i}\right), CA \end{cases} \tag{7}$$

This may lead to two situations that reduce network performance. First, for multiple-linked networks, such aggressive window growth can lead to congestion or increase the degree of network congestion when the impending or emerging congestion causes delays to increase.The formula (6) deduces that the more congestion the network has, the larger the RTT is, and the larger the growth of the Hybla congestion window $2^{\rho} - 1$ and ρ^2 / W_i is, which easily increases the degree of network congestion and increases the possibility of packet loss. Second, in a long-delay network, a large continuous transmission window may cause buffer overflow and cause unnecessary packet loss. Aiming at the above problems of Hybla algorithm in satellite networks, this paper proposes a dynamically adjusting congestion window growth factor based on bandwidth estimation.

Assume that the sender receives an ACK at time t_k, which indicates that the receiver has received continuous piece of data d_k. t_{k-1} is the arrival time of the previous ACK, and the arrival interval of the two ACKs is $\Delta t_k = t_k - t_{k-1}$. The connection bandwidth value b_k at time t_k is:

$$b_k = \frac{d_k}{t_k - t_{k-1}} = d_k \Big/ \Delta t_k \tag{8}$$

The calculation of network expectation bandwidth is defined as follows:

$$bw\_expect = d_k / min(\Delta t_k) \tag{9}$$

Where bw_expect is the expected bandwidth at time t_k and $min(\Delta t_k)$ is the minimum interval between two ACKs currently measured

The larger the used bandwidth is, the smaller the space for congestion window growth can be, and vice versa. ρ can be updated as follow:

$$\rho_{new} = \rho * [1 - \frac{b_k}{bw\_expect}] \tag{10}$$

ρ_{new} denotes the updated ρ.

After updating ρ, each time the Hybla IM receives an ACK, the congestion window grows dynamically based on the available bandwidth in the satellite network so it eliminates the problem of aggressive growth of the Hybla algorithm congestion window.

3.2 Implementation of Packet Loss Differentiation Based on Queue Delay

Although the Hybla algorithm can quickly recover the reduced congestion window due to link errors, alleviating the problem of degraded transmission performance caused by high bit errors, Hybla cannot distinguish the reason for packet loss, which leads to its need to go through multiple packet loss recovery stages, which seriously affects the performance of the algorithm and reduces the utilization of satellite resources.

3.2.1 Calculation of Propagation Delay

Reference [8] calculates the propagation delay as follows:

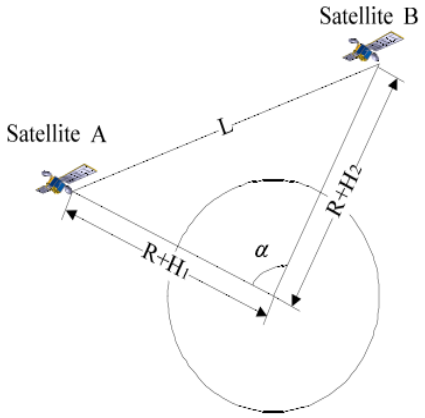

Figure 1. Inter-satellite distance

The formula for calculating the instantaneous heart angle is:

$$\alpha = \arccos[\sin\varphi_1 \sin\varphi_2 + \cos\varphi_1 \cos\varphi_2 \cos(\lambda_1 - \lambda_2)] \tag{11}$$

Where: (λ_1, φ_1) and (λ_2, φ_2) are the satellite latitude and longitude of satellite A and satellite B, respectively.

According to the cosine theorem of the triangle, it can be known that the distance l between two adjacent satellite nodes can be expressed as:

$$l = \sqrt{(R+H_1)^2 + (R+H_2)^2 - 2(R+H_1)(R+H_2)\cos\alpha} \tag{12}$$

Where: R is the radius of the Earth, H_1 is the orbital altitude of the satellite A, and H_2 is the orbital altitude of the satellite B.

The total length of the link is the sum of the distances between the satellites, ie $D = \sum_{i=1}^{n} l_i$, where l_i is the length of the link between adjacent satellites on the link.

Calculate the propagation delay at the current moment according to the length of the inter-satellite communication link. The calculation formula is as follows:

$$T_{PD} = 2 \times D / c \tag{13}$$

Where: T_{PD} is the propagation delay, c is the speed of light, and D is the length of the communication link at the moment.

3.2.2 Calculation of Queue Delay

Round-trip delay RTT includes transmission delay, processing delay, queuing delay, and propagation delay. The transmission delay, processing delay, and queuing delay are called non-propagation delays. Therefore, the round-trip delay RTT can be expressed as:

$$RTT = T_{PD} + T_{NPD} \tag{14}$$

T_{PD} Is propagation delay and T_{NPD} is non-propagation delay. So the non-propagation delay can be expressed as:

$$T_{NPD} = RTT - T_{PD} \tag{15}$$

T_{NPD} Consists of queue delay, processing delay and transmission delay. $\min(T_{NPD})$ Denotes the minimum T_{NPD}, which is generally the time when the algorithm starts sending data packets. At this time, the network has no congestion, so $\min(T_{NPD})$ only include the processing delay and transmission delay. Since the network node processes and sends an MSS (Maximum Segment Size) approximately has the same time, the estimated value of the queuing delay can be expressed as:

$$T_{QD} = T_{NPD} - \min(T_{NPD}) \tag{16}$$

Due to the high-speed movement of nodes in satellite networks especially in LEO satellite network, the distance between satellites and the connection between satellites keep changing result in end-to-end round-trip RTTs are in constant changes. Therefore, RTT cannot accurately reflect the network congestion in satellite networks. The queuing delay T_{QD} indicates the queuing time of the data packet in the satellite buffer and can accurately reflect the congested state of the satellite network.

3.2.3 Distinguishing Packet Loss Algorithm

Step1: The sender calculates the queuing delay T_{QD} for each ACK of the data packet and weights it:

$$T'_{QD}(n) = (1-\alpha)T'_{QD}(n-1) + \alpha T_{QD}(n) \tag{17}$$

$T'_{QD}(n)$ Indicates the weighted value of T_{QD} for the nth packet received, $T'_{QD}(n-1)$ indicates the weighted value of T_{QD} for the n-1th packet received. $T_{QD}(n)$ Indicates the nth data packet queue delay.

Step1: Recording $T_{QD}(n)$, $T_{QD}(n-1)$, $T_{QD}(n-2)$.

$T_{QD}(n)$, $T_{QD}(n-1)$ $T_{QD}(n-2)$ represent the queuing delay calculated by the nth, (n-1) the, and (n-2) the ACKs

It shows the loss of the (n+1)the packet if three repeated ACKs for the nth packet are received, the nth packet need to be quickly retransmitted then go to Step3, Step4:

Step3: Calculating the average number of queuing delays for nth, (n-1) the, and (n-2) the packet.

$$avg = \frac{T_{QD}(n) + T_{QD}(n-1) + T_{QD}(n-2)}{3}$$

Step4: Compare avg and $T'_{QD}(n)$. If $avg > T'_{QD}(n)$, the reason for packet loss is caused by congestion. In this case, the sender needs to reduce the transmission rate and enter the fast recovery phase. Otherwise, packet loss is caused by link error. In this case, the sender keep the sending rate and do not enter the fast recovery phase. (18)

$T'_{QD}(n)$ Represents a roughly queuing delay of the data packet in the network, which can be understood as the queuing delay of the data packet when the network is in a state of congestion and non-congestion. Predict the queue delay of the (n+1) the packet by calculating the first three packet queue delays that are closest to (n+1) the packets because the first three packets closest to (n+1) the packet is in a similar

network condition to that of (n+1) the packet, predicting the degree of congestion of the network where the (n+1) the packet is located.

3.3 Hybla IM Algorithm Flow Chart

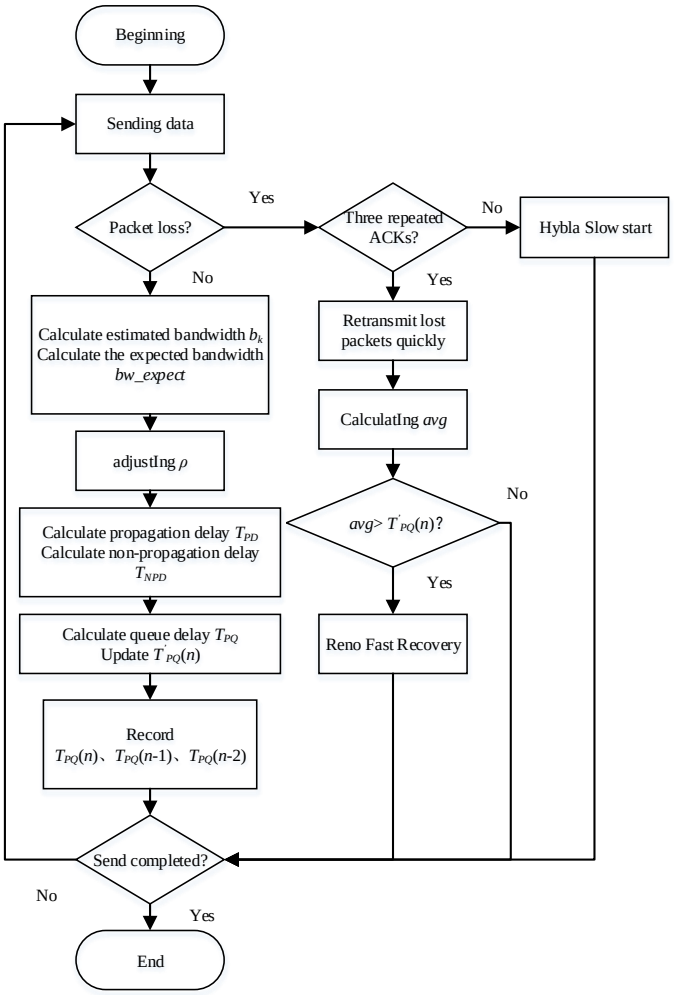

Figure 2. Hila IM algorithm flow chart

4. Algorithm Performance Simulation

In order to verify the performance of the new algorithm, the following two simulation software is used in simulation experiments.（1） The satellite simulation software STK is used to simulate the satellite network topology, the version is 8.1.1;（2） Network simulation software NS2 [9, 10] is used to simulate algorithm performance, the version is version 2.34.

The network topology used in this simulation experiment is shown in Figure 3.

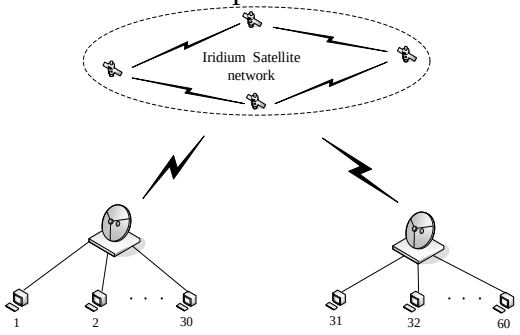

Figure 3. Simulation topology diagram

Select polar orbit satellite constellation Iridium with global coverage as satellite communication network, the specific parameter settings are shown in Table 2. Ground station takes the default configuration of NS2, selecting Berkeley (37.9, -122.3) and Boston (42.3, -71.1) as two ground stations. Each ground station is connected to 30 terminal nodes, the source node is connected to Berkeley and the destination node is connected to Boston ,there is a total of 30 links through the Iridium satellite network. All sources send data to the destination. The link bandwidth between the terminal and the ground station is 10 Mb/s.

Table 2. Iridium satellite constellation parameters

Parameter	Parameter value
Orbital height	780 km
Number of tracks	6
Number of satellites per orbital plane	11
Track inclination	86.4°
Track plane spacing	31.6°
Slit interval	22°
Minimum allowable elevation angle	8.2°
Inter-satellite links per satellite	4
Inter-satellite link bandwidth	25 Mbps
Upstream and downstream bandwidth	1.5 Mbps
Inter-satellite link latitude threshold	60°

The two ground stations are connected together through the LEO satellite constellation, forming a bottleneck link.Using the above-mentioned Iridium satellite network topology and simulation parameter configuration, simulations were performed on the Hybla algorithm and the Hybla IM algorithm. The simulation results are as follows:

(1) Comparison between Hybla and Hybla IM of throughput and packet loss rate at BER of 0.0001

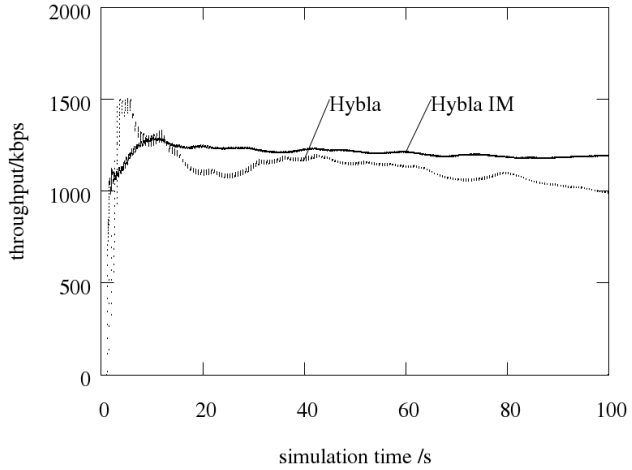

Figure 4. Throughput of different algorithms at BER of 0.0001

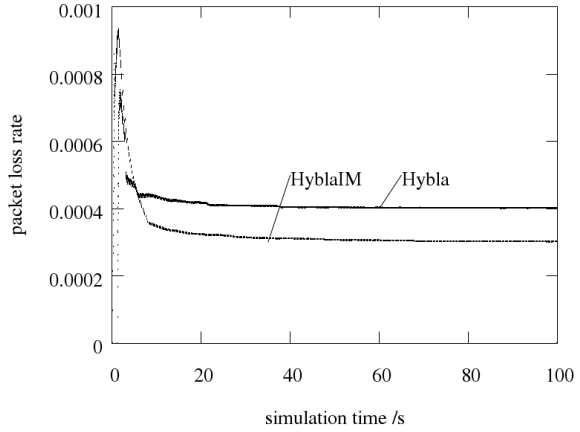

Figure 5. Different algorithm of packet loss rate at BER of 0.0001

In the case of low link error rate of the satellite network, packet loss is mainly caused by congestion. It can be seen from Figure 4 that the throughput of the Hybla algorithm fluctuates greatly. It can be seen from Figure 5 that the packet loss rate of Hybla algorithm is higher than that of Hybla IM. Although Hybla can quickly occupy network bandwidth and improve throughput at the beginning of simulation, the radical window growth of the Hybla algorithm quickly leads to congestion in the network, resulting in packet loss and rapid network throughput degradation. Compared with Hybla, the growth of Hybla IM congestion window can match the bandwidth of the network, so that the throughput can increase steadily and maintain a steady state and the packet loss rate is lower.

(2) Comparison between Hybla and Hybla IM of throughput at BER of 0.001 and 0.01

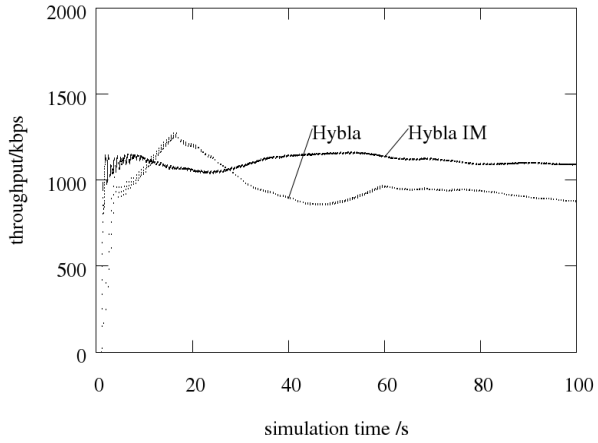

Figure 6. Throughput of different algorithms at BER of 0.001

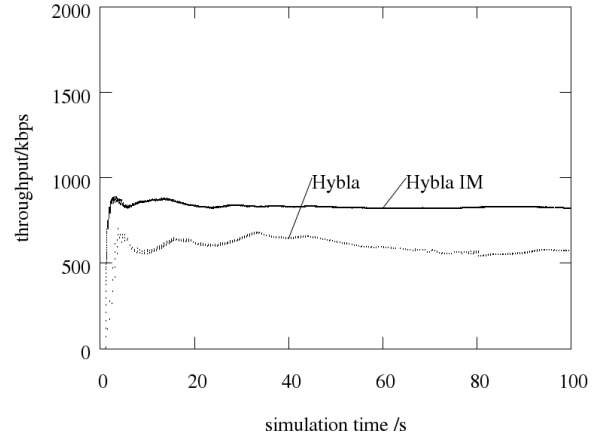

Figure 7. Throughput of different algorithms at BER of 0.01

The Hybla algorithm's throughput at a link error rate of 0.001 is 13% lower than that at the link error rate of 0.0001 and is 32% lower at the link error rate of 0.01 than that at the link error rate of 0.001. The Hybla IM algorithm's throughput at a link error rate of 0.001 is 7.3% lower than that at the link error rate of 0.0001 and is 24% lower at the link error rate of 0.01 than that at the link error rate of 0.001. Hybla IM throughput is 6% higher than Hybla, 13%, and 17%, respectively, when the link error rates are 0.0001, 0.001, and 0.01 respectively, proving that Hybla IM can discriminate between packet loss, reduce the call of congestion control mechanism caused by error packet loss, improve resource utilization, and improve throughput.

5. Conclusion

Compared with the Hybla algorithm, Hybla IM dynamically adjusts congestion window growth through real-time bandwidth estimation and expected bandwidth, effectively mitigating the problem of Hybla's aggressive growth in long-delay networks. At the same time, the reason of packet loss is determined by calculating the packet queue delay on the satellite.Simulation experiments show that compared to the Hybla algorithm,Hybla IM improves the throughput, significantly reduces the packet loss rate, effectively distinguishes the causes of packet loss, and save valuable satellite resources. The next step will be to improve the Hybla algorithm in response to the high dynamics of the satellite network topology so that the Hybla algorithm can be better applied to the satellite network.

Acknowledgements

This subject is supported by National Natural ScienceFoundation of China (NSFC Grant No. 61722105).

References

[1] Jehan M, Radhamani G, Kalakumari T. Experimental Evaluation of TCP BIC and Vegas in MANETs[J]. International Journal of Computer Applications, 2011, 16(1):34-38.
[2] Brakmo L S, Member S, Peterson L L. TCP Vegas: End to End Congestion Avoidance on a Global Internet[C]// IEEE Journal on selected Areas in communications. 1995:1465 - 1480.
[3] Akyildiz I F, Morabito G, Palazzo S. TCP-Peach:a new congestion control scheme for satellite IP networks[J]. IEEE/ACM Transactions on Networking, 2001, 9(3):307-321.
[4] Mascolo S, Casetti C, Gerla M, et al. TCP westwood: Bandwidth estimation for enhanced transport over wireless links[C]// 2001:287-297.
[5] Ha S, Rhee I, Xu L. CUBIC: a new TCP-friendly high-speed TCP variant[J]. Acm Sigops Operating Systems Review, 2008, 42(5):64-74.
[6] Sosic M, Stojanovic V. Resolving poor TCP performance on high-speed long distance links — Overview and comparison of BIC, CUBIC and Hybla[C]// IEEE, International Symposium on Intelligent Systems and Informatics. IEEE, 2013:325-330.
[7] Mao T, Xu Z. An Effective Mechanism for Improving Performance of TCP over LEO/MEO Satellite IP Networks[J]. Journal of Networks, 2013, 8(12):2942-2947
[8] Pan Cheng-sheng,Xuan jing-peng,Wei De-bin,Yang Li. A Congestion Control Algorithm for High Dynamic Satellite Networks[J].Journal of Aeronautics,2014,35(08):953-960.
[9] Issariyakul T, Hossain E. Introduction to Network Simulator NS2[M]. Springer US, 2009.
[10] Ren Y M, Tang H N, Jun L I, et al. Performance Comparison of TCP Variants for High-speed Network by NS2 Simulation[J]. Computer Engineering, 2009, 35(2):6-9.