Improved RTO Algorithm for TCP Retransmission Timeout

Xiao Jianliang[1,a], Zhang Kun[1,b]
[1]City Institute of Dalian University of Technology, Dalian, 116600, China
[a]email:gzszdnb@163.com, [b]email:zkly2005@126.com

Keywords: retransmission timeout; round-trip time; congestion control

Abstract. The transmission control protocol (TCP) retransmission timeout (RTO) is calculated based on a round-trip time (RTT) with weighted smoothing. This algorithm is sensitive to the initialization, and convergence is slow. The RTO changes lagging behind the RTT. The algorithm only takes into account the size of RTT, without considering the tendency of it. But the changing trend of RTT more directly reflects the traffic state of network. This paper is based on the traditional RTO algorithm; both the size and the change trend of the RTT are considered. Modify the weight coefficient, the weight coefficients of adaptive changes of RTT, so that RTO can quickly follow the changes of RTT.

Introduction

TCP protocol is a reliable transport protocol connection oriented, its reliability is mainly based on three points: the byte stream oriented sequence index; confirmation of the number; retransmission when timeout. The sender should start a timer after sending out a data packet, in a fixed time interval, did not receive the ACK for the data packet sent by receiver, the TCP sender should resend the unrecognized data packet of the earliest. The timer interval is called RTO. RFC2988 provides the calculation method of RTO. The algorithm proposed by Jacobson in 1988 [1]. The Jacobson's algorithm based on exponential weighted moving average method is non adaptive filtering algorithm, while the whole calculation process, the algorithm uses the same gain value (α, β). It failed to reflect the traffic of the network in a timely manner. In fact, the tendency of RTT can reflect the network traffic better. Brakmo L. S. TCP Vega is proposed based on the variation trend of RTT to predict congestion situation [2]. TCP Vega perception network is congested or not by observing the value change in RTT, so as to control the congestion window size.

In 1990 Jacobson proposed the fast retransmission algorithm [3], according to the 3 duplicate acknowledgement packets (ACK) to determine the data package loss, as far as possible to avoid waiting for the retransmission timer timeout, effectively improve the efficiency of transmission. However, when the last packet transmitted is missing or in short HTTP connection service environment (in the current network traffic, a short connecting business share larger), ACK information is insufficient, or there is no ACK can be used to trigger high packet loss recovery mechanism, can only rely on retransmission timer timeout events to determine the packet loss, retransmission. Therefore, the retransmission timer management is crucial [4]. The Jacobson algorithm is sensitive to the initial value, conservative, response to sudden large amplitude changes of RTT serious lag; this will impact on the performance of transmission control [5].

RFC 2988 indicates that when multiple samples are taken per RTT α and β defined in Jacobson's algorithm may keep an inadequate RTT history. A method for changing these constants is currently an open research question [6].

In this paper, the weight coefficient α, β is improved. The constant (α, β) become as the variable, to adaptive to the changes of RTT. The improved algorithm not only considers the size of RTT, but also considers the tendency of RTT.

Analysis of the Jacobson Algorithms

The Jacobson algorithm is pointed out, to calculate the current RTO, TCP sender need to

maintain two state variables, SRTT (smoothed round-trip time) and RTTVAR (round-trip time variation). Calculated as follows: SRTT, RTTVAR and RTO rules:

Until a round-trip time (RTT) measurement has been made for a packets sent between the sender and receiver, the sender should set RTO to 3 seconds [6]. After completing the first RTT measurement, a TCP sender must be set

$$SRTT_1 = RTT_1 \tag{1}$$

$$RTTVAR_1 = RTT_1 / 2 \tag{2}$$

As the initial value of SRTT, RTTVAR, when second RTT is made, executing the following algorithm for RTO:

$$SRTT_{n+1} = (1 - \alpha) * SRTT_n + \alpha * RTT_{n+1} \tag{3}$$

$$RTTVAR_{n+1} = (1 - \beta) * RTTVAR_n + \beta * \left| SRTT_{n+1} - RTT_{n+1} \right| \tag{4}$$

$$RTO_{n+1} = SRTT_{n+1} + 4 * RTTVAR_{n+1} \tag{5}$$

In the formula, n is the value of 1, 2, 3, 4 ....., Jacobson recommendation: α=1/8, and β=1/4, and was confirmed by the RFC2988.

In addition, according to the RFC2988, once the RTO calculation is over, if it is less than 1 second, then the RTO should be set to 1 seconds [7]. Therefore, in any case, the minimum value of RTO is 1 second.

Based on the above algorithm, this paper analyzes the situations of RTT increases rapidly and reduction rapidly. In theory, RTO is closer to RTT, retransmission the more timely, network bandwidth utilization rate is higher, the user experience better.

(1) The situation of RTT rapid growth

Assume a burst in network, tend to congestion. During a TCP connection, the round-trip time RTT will increase rapidly. Eq. (3), the most current RTT accounted for only 12.5% of the weight, while the historical SRTT has accounted for 87.5% of the weight, this shows that, the rapid change of RTT are not fully taken into account, leading to SRTT response lag. To enable SRTT timely reflect changes in RTT, should be in the Eq. (3) increased the weight of RTT, namely, let α timely reflect changes in RTT. If the RTT increases rapidly, the weight α is also at the same rate, even more radical point of increase rate. The increase in α value, mean increasing the impact of the current RTT to RTO, reducing the impact of the history‘s RTT to RTO. Look at the Eq. (4), because SRTT slow growth, no fast follow changes of RTT, RTT and SRTT results deviation increases more. This bias amplification are 4 times amplification applied to the RTO, makes the RTO fast deviation RTT. RTT change more quickly, the faster the RTO deviation of RTT. When the network restored, RTT will rapidly decrease, because of the lag effect of Jacobson algorithm, the convergence time become much longer. RTO needs more time to follow RTT.

(2) The situation of RTT rapid reduction

Consider another situation, assume that a TCP connection is just established, the network is busy, the initial RTT of the measured relatively large. However, because of its sudden application of Internet, congestion may soon become better; making the follow-up of RTT rapidly becomes smaller. In this case, by Eq. (3), the latest RTT accounted for only 12.5% of the weight, while the historical SRTT has accounted for 87.5% of the weight. Therefore, the rapid change of RTT has not been fully considered, leading to SRTT response lag. From the Figure 1 can be seen, RTO lags behind RTT extremely. RTO is too large, resending overtime too late, the network bandwidth is underutilized, and transmission efficiency is greatly reduced. In order to fully reflect the changes in the current RTT, should increase its weight, which should increase the value of α and reduce the influence of history SRTT. Re-investigation Eq.(4), RTT is rapidly dwindling, it will lead to increased deviation of RTT and SRTT, after operation by the Eq. (4), and 4 times magnification to RTO, it can make RTO increase rapidly. The deviation from RTT become much more, as shown in Figure 2. Therefore, need to adjust the weights of β small value, in order to reduce the influence factors of the deviation, so that RTTVAR can quickly reduce.

Jacobson algorithm's weight α, β value is constant, no matter what, RTT rapid increase or decrease quickly, according to Eq. (5), the calculation value of RTO rapidly divergence of RTT, leading to further extend the convergence time of RTO, is not conducive to the full use of cyber source.

The Improved Algorithm

Through the above analysis, we can see that, if the network becomes congested, RTT will increase rapidly, while RTO will quickly deviate from RTT. After network restoration, the time of convergence would become longer, so easy to cause the waste of cyber source. If the network from congestion becomes more open, RTT will rapidly decrease, the RTO reaction is seriously lagging behind, the sender greatly delayed packet retransmission, will all be wasted cyber source, reduce the transmission efficiency.

This due to Jacobson's algorithm of retransmission timeout RTO, based only on the size of the RTT, without considering the tendency of RTT. In addition, the weight coefficient α, β fixed, lack of flexibility, adaptive ability is insufficient.

Therefore, the thesis provides a train of thought, according to the dynamic change trend of RTT to adjust weight α, β. Whether the change trend of RTT is a rapid increase or decrease quickly, increase the weight of RTT, which is to increase α, making the RTT's changes are reflected in RTO rapidly. In the Jacobson algorithm, the change of RTT is mainly reflected by the variable RTTVAR, the variable in the downward phase of RTT leads to RTO significantly lagged behind RTT. Thus reducing β, it can reduce the influence of the variables on RTO. The objective of adjusting the weight of α, β is to make the RTO more quickly follow the RTT reduction, can be faster send data to the network, making full use of network bandwidth, improving the efficiency of transmission.

A variable *k* --- RTT's rate of change is required by the following formula.

$$k_{n+1} = \left| \left(RTT_{n+1} - RTT_n \right) / RTT_n \right| \tag{6}$$

According to the results calculated by the formula, if k>1, then k=1.

Calculated the values of α, β, follow the following formula.

$$\alpha_{n+1} = \alpha_0 \left(1 + k_{n+1} \right) \tag{7}$$

$$\beta_{n+1} = \beta_0 \left(1 - k_{n+1} \right) \tag{8}$$

The above α_0, β_0 according to Jacobson, the recommended values is α=1/8 and β=1/4. Obviously, RTT changes in the relatively flat stage, the rate of change of K tends to 0, then the weight of α and β is the Jacobson recommended value, equivalent to the original algorithm. When the RTT is rapidly changing, the rate of change of K is greater than 0, then the value of α increase and β reduction, mixing effect is to make the RTO more quickly follow the changes in RTT, but not in the original algorithm, deviation of RTT quickly. The essence of improved algorithm is to adjust the weights according to the rate of change of RTT size, the weight value with changes of RTT, so RTO can reflect the rapid changes in RTT. The modified α and β value instead of original algorithm in the constant, RTO algorithm will become the following formula shown.

$$SRTT_{n+1} = \left(1 - \alpha_{n+1} \right) * SRTT_n + \alpha_{n+1} * RTT_{n+1} \tag{9}$$

$$RTTVAR_{n+1} = \left(1 - \beta_{n+1} \right) * RTTVAR_n + \beta_{n+1} * \left| SRTT_{n+1} - RTT_{n+1} \right| \tag{10}$$

$$RTO_{n+1} = SRTT_{n+1} + 4 * RTTVAR_{n+1} \tag{11}$$

Simulation Analysis

In order to verify the validity of the improved algorithm, this paper adopts the network simulator NS2, based on the topological structure of the establishment of a single bottleneck link, as shown in Figure 1.

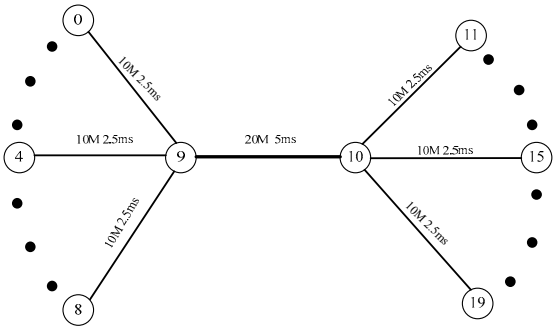

Figure 1 Simulation topology

In Figure 1, node 0-8 as the sending node, node 9-10 as router node, and node 11-19 as the receiving node. Source link, the sending and receiving data-link bandwidth is 10M, the delay is 2.5ms, and the data-link bandwidth between routers is 20 M, delays for 5ms. In the simulation, node 0 and node 11 constitute the data link; node 2and node 12 consists of data link, and so on, node 8 and node 19 to form a data-link. With FTP and CBR (Constant Bit Rate) as a data stream, the packet size is 1000Bytes, the router cache into 20packet size, the router queue management using Drop-Tail packet discard policy. The simulation starts with the FTP and CBR continuing to send 100MB data. In the case of small number of connections, the router data-link relatively smooth, the ACK response will be faster, the RTT value is relatively small. With the increase of number of connections, the router data-link tends to be crowded, the ACK response will slow down significantly, The RTT will increases quickly. By changing the number of connections, you can obtain a series of RTT value.

A series of RTT will be used to the Eq. (3) to Eq. (5) for calculating RTO, and be used to Eq. (6) to Eq. (11) for calculating RTO'. The RTO' reflects the improved algorithm. The corresponding RTO, RTO' and RTT are compared in the same graph, as shown in Figure 2. As can be seen from the graph, The RTO followed RTT changes, follow the situation not only with the RTT value is related, but also with the rate of change of RTT. The RTT changes fast, causing the RTO changes faster. But, can clearly be seen that the RTO' closer to RTT than RTO, meaning that the improved algorithm, make RTO' following better.

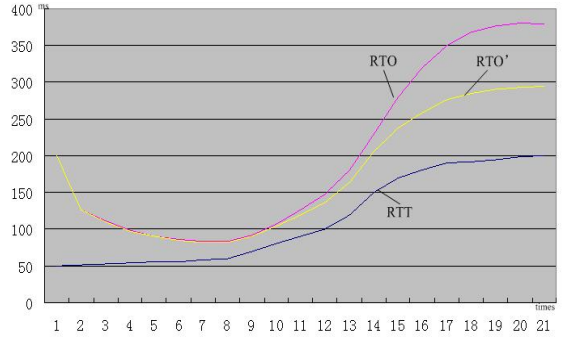

Figure 2 RTT rapid increase situation

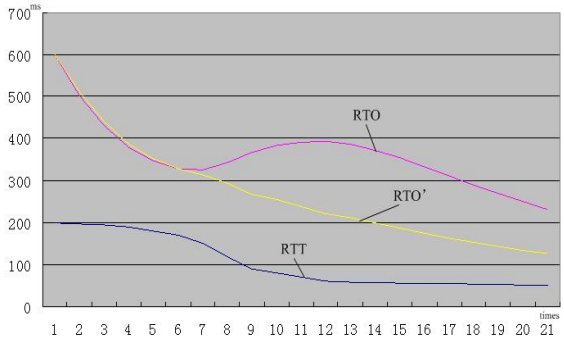

Figure 3 RTT rapid decrease situation

The rapid increase in the number of connections, causes RTT to rapidly increase; in turn, quickly reduce the number of connections, can lead to rapid reduction of RTT. Continuously reduce the connection number can get a series of RTT value. Similarly, according to Eq. (3) to Eq. (5) and Eq. (6) to Eq. (11), a series of RTO and RTO' can be calculated. The RTT, RTO, RTO' drawing in the same diagram, as shown in Figure 3.

Conclusion

It can be observed intuitively the case of RTO following RTT in Figure2 and Figure3. Obviously, the improved algorithm shows better. In order to compare the two kind of algorithms, and to survey the RTO following RTT, a variable γ be introduced, as the percentage of deviation between RTO and RTT. Definition that:

$$\gamma = 100 * \left| \left(RTO - RTT \right) / RTT \right| \tag{12}$$

The RTTs, RTOs and RTO's which are show in Figure2 and Figure3 are used to Eq. (12) for calculating γ, the calculation results are shown in Table 1. The smaller γ means that the RTO follows RTT better.

Table 1 The percentage γ of RTT variety

The sampling sequence (times)	the case of RTT rapid increase		the case of RTT rapid decrease	
	The traditional algorithm (%)	The improved algorithm (%)	The traditional algorithm (%)	The improved algorithm (%)
2	144.23	146.03	153.28	153.75
3	108.61	110.41	120.32	121.13
4	83.29	84.84	96.55	97.38
5	65.42	66.55	78.31	78.92
6	52.84	53.53	75.74	75.46
7	43.33	43.64	79.3	76.71
8	37.89	37.55	105.37	93.55
9	31.09	29.27	173.12	129.38
10	33.76	27.76	249.17	163.08
11	40.19	29.16	303.27	183.48
12	47.62	31.69	363.69	205.32
13	50.17	30.38	435.16	229.83
14	53.13	28.17	525.21	258.81
15	64.53	30.92	520.61	247.32
16	77.27	36.47	506.05	233.91
17	84.51	40.15	485.04	219.71
18	91.79	44.54	460.26	205.43
19	93.97	46.33	444.91	196.86
20	91.87	45.91	418.29	183.85
21	89.29	45.34	391.78	171.61

It can be seen from Table 1 that in the slow changes stage of RTT, two kinds of algorithm of the RTO with good convergence, RTO following RTT closely. In the stage of RTT rapid increase, the traditional algorithm of RTO increased rapidly, the RTO deviation of RTT quickly. The improved algorithm of RTO' growth is slow, the RTO' deviation of RTT slowly. As shown in the figure, The improved algorithm show better following characteristics of RTO'. In the case of RTT rapid decrease, the RTO calculated by traditional algorithm does not follow the RTT decreased, but far away from RTT, and the deviation percentage of γ increased rapidly. The RTO' calculated by the improved algorithm decrease slowly, but decrease speed is slower than RTT, the RTO' can follow the change of RTT in the whole process. Compared with the traditional algorithm, the improved algorithm is better in the convergence.

Through theoretical analysis and simulation calculation, it is prove that the improved algorithm is feasible and effective.

References

[1] Jacobson V. Congestion avoidance and control [J].ACM Computer Communication Review, 1988, 18(4): 314-329.

[2] Brakmo L S .Peterson L L.TCP Vega: end-to-end congestion avoidance on a global Internet [J]. IEEE Journal on Selected Areas in Communication, 8.Otc.1955.13

[3] Jacobson, V., "Modified TCP Congestion Avoidance Algorithm", end 2 end-interest Mailing list, April 30, 1990.

[4] ZHANG Lixia. Why TCP timers don't work well [J].ACMSIGCOMM Computer Communication Review, 1986, 16(3):397-405.

[5] Chen Xiang, Liu Weidong, Ren Fengyuan, Journal of Tsinghua University (NATURAL SCIENCE EDITION) 2007 forty-seventh volume fourth issue of 38/40

[6] Braden, R., "Requirements for Internet Hosts -- Communication Layers", STD 3, RFC 1122, October 1989.

[7] http://datatracker.ietf.org/doc/rfc2988/