2018 4th International Conference on Systems, Computing, and Big Data (ICSCBD 2018)

Study on Performance of TCP in Wireless Networks

Yankui Cai

Nanjing University of Science and Technology, School of Electronic and Optical Engineering, Nanjing 210094, China

Corresponding author: caiyankui@njust.edu.cn

Keywords: TCP; wireless network; NS2.

Abstract: The AIP Proceedings article template has many predefined paragraph styles for you to use/apply as you write your paper. To format your abstract, use the Microsoft Word template style: *Abstract*. Each paper *must* include an abstract. Begin the abstract with the word “Abstract” followed by a period in bold font, and then continue with a normal 9 point font.Wireless networks are widely used in Internet of Things (IoT) , Ad-hoc networks, Sensor networks. An important problem need to be solved in wireless networks is there is no reliable transport layer protocol at present. TCP (Transport Control Protocol) is the that provides reliable end to end data delivery. It was originally designed primarily for wired networks. Buffer queue overflow at the bottleneck nodes (also known as congestion) is the main reason for packet loss and retransmission timer timeout in wired networks. The main problems are high BER, precarious wireless channel and user mobility in wireless networks. Therefore, traditional TCP performs badly in the wireless network. But study on traditional TCP is very important. We can promote TCP algorithm in wireless networks through the study of traditional TCP. In this paper, we designed a series of experiments based on the NS2. We focus on the different performance of the traditional TCP used in the wireless networks and wired networks. We tried to analyze the reasons for the relevant data results and put forward our own opinions on how to design TCP mechanisms.

1. Introducton

TCP (Transport Control Protocol) is the most widely used transport layer protocol at present. TCP contains the following functions: dividing the data received from the application into many smaller data sections, providing connection-oriented services, providing reliable service, providing flow control and congestion control. In order to use bandwidth equally with other users, we need transport control mechanism when the application has data to send. Traditional TCP has several versions: Tahoe, Reno, NewReno, SACK and Vegas. These mechanisms are different in congestion control algorithm.

In our paper, we will analyze the differences between different versions of TCP algorithms. Later versions of TCP tend to be refinements from earlier versions. Some of these mechanisms are used on a large scale in wired networks. The design of TCP in wireless networks usually based on the traditional TCP versions [1][2]. We need to study TCP mechanism in depth in order to came up with better design of TCP mechanism in wireless networks.We will explain the implementation details of the TCP mechanism. We will conduct a series of experiments based on the NS2. NS2 (Network Simulator version 2) is a kind of open source code, free software simulation platform for network technology. Researchers can easily develop network technology by using it, and its modules cover almost all aspects of network technology today. Therefore, NS2 has become a kind of network simulation software widely used in the academic circle. Our experiments were designed for observation the performance of the traditional TCP used in the wireless networks, mobile wireless network and wired networks. We tried to analyze the reasons for these results and explained the impact of wireless environment on TCP performance. At last, we will put forward our own opinions on how to design TCP mechanism based on our experiments and study.

In the second chapter, we will introduce the implementation of traditional TCP (including Tahoe,

  DOI: 10.25236/icscbd.2018.021

Reno, NewReno, SACK and Vegas) and conduct experiments in wired networks based on the NS2. Then in the third chapter, we will give our simulated data and graph in wireless network based on the NS2. In the conclusion part, we will summarize the points of this paper and give the points and directions of congestion control in wireless networks.

2. Congestion Control Mechanisms

2.1 Basic Method of Congestion Control

In general, TCP congestion control methods can be divided into five phases Slow-start, Congestion Avoidance, Fast Retransmission, Fast Recovery and Timeout Retransmission. TCP uses ACK (Acknowledgement from destination nodes) to detect network status and provide reliable service. TCP distinguish Slow-start and Congestion Avoidance by Slow-start threshold (ssthresh) value and cwnd value (send window) when adjusting the sending speed of the sending end, as shown in equation (1).

$$cwnd(t+t_r) = \begin{cases} slow\ start\ \ phase: \\ cwnd(t)+1, \qquad\qquad if\ \ cwnd(t) < ssthresh; \\ congestion\ avoidance\ \ phase: \\ cwnd(t)+\dfrac{1}{cwnd(t)}, \qquad if\ \ cwnd(t) \geq ssthresh; \end{cases} \tag{1}$$

As shown in equation (1), TCP is in Slow-start phase when cwnd is less than ssthresh. For each ACK received, the value of cwnd is incremented by 1 in this phase. So every RTT (Round-Trip time), the value of cwnd becomes twice the value of the previous RTT. The value of cwnd increases exponentially in this phase. TCP is in Congestion Avoidance phase when cwnd is greater than ssthresh. The value of cwnd increases in a linear fashion. Every RTT (Round-Trip time), the value of cwnd is incremented by 1 than the previous RTT to avoid packet loss caused by sending data too quickly. Sender uses ACK to confirm whether the packet is received by the receiver. If a discontinuous packet is received, the receiver returns a duplicate ACK. TCP uses ACK and retransmission to secure reliable delivery.

2.2 TCP Versions and TCP Performance Simulation in Wired Networks

TCP Tahoe: The earliest version of TCP is called Tahoe. Tahoe has the basic structure of TCP, including Slow-start, Congestion Avoidance and duplicate ACK. In addition, Tahoe added Fast Retransmit. Fast Retransmit mechanism will immediately resend the missing packet after receiving three repeated ACK (Three repeated ACK means this packet has lost). Then TCP sets the value of ssthresh to 1/2 of cwnd and resets the value of cwnd to 1.

TCP Reno: TCP Reno is the most widely used TCP versions. Reno adds Fast Recovery algorithm based on Tahoe. TCP resend the missing packet and sets the value of ssthresh and the value of cwnd to 1/2 of cwnd after receiving three repeated ACK. Reno can resume sending speed faster than Tahoe.

As shown in FIGURE 1, Where R0 and R1 represent routers on the network. FTP Source is the source node. FTP Sink is the destination node. Delay and bandwidth are shown in the FIGURE 1. We used TCP Tahoe and TCP Reno to send data. We can see different cwnd change from FIGURE 2.

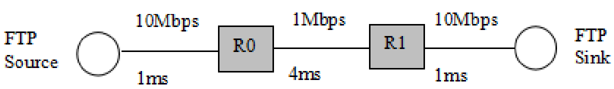

Figure 1. wired network for Tahoe and Reno.

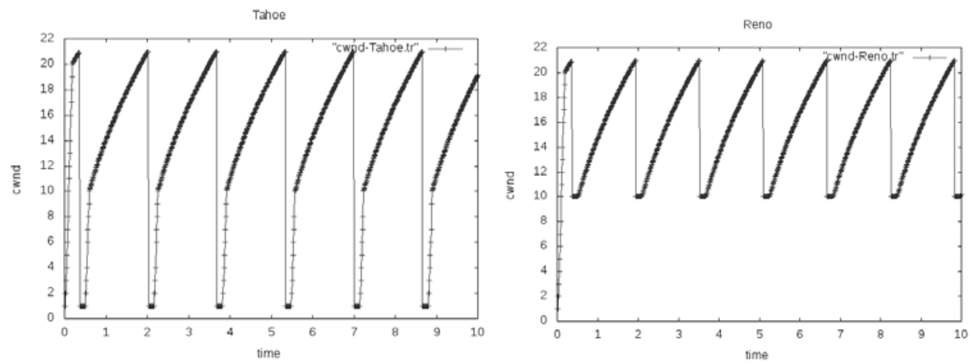

Figure 2. cwnd of Tahoe and Reno in wired network.

TCP NewReno: NewReno is the improved algorithm of Reno. The main difference between NewReno and Reno is Fast Recovery algorithm. After receiving three repeated ACKs, NewReno will resend all packets that the source node has already sent out but not receive its ACK. NewReno can handle a case of a lot of packets loss meanwhile Reno can only handle a case of one packet loss

TCP SACK: TCP SACK is another improved algorithm of Reno. When packet loss happened, destination node give details about which packets have been received and which packets have been lost. The SACK mechanism is more efficient at retransmission.

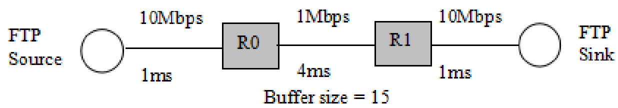

Figure 3. wired network for Reno, NewReno and SACK.

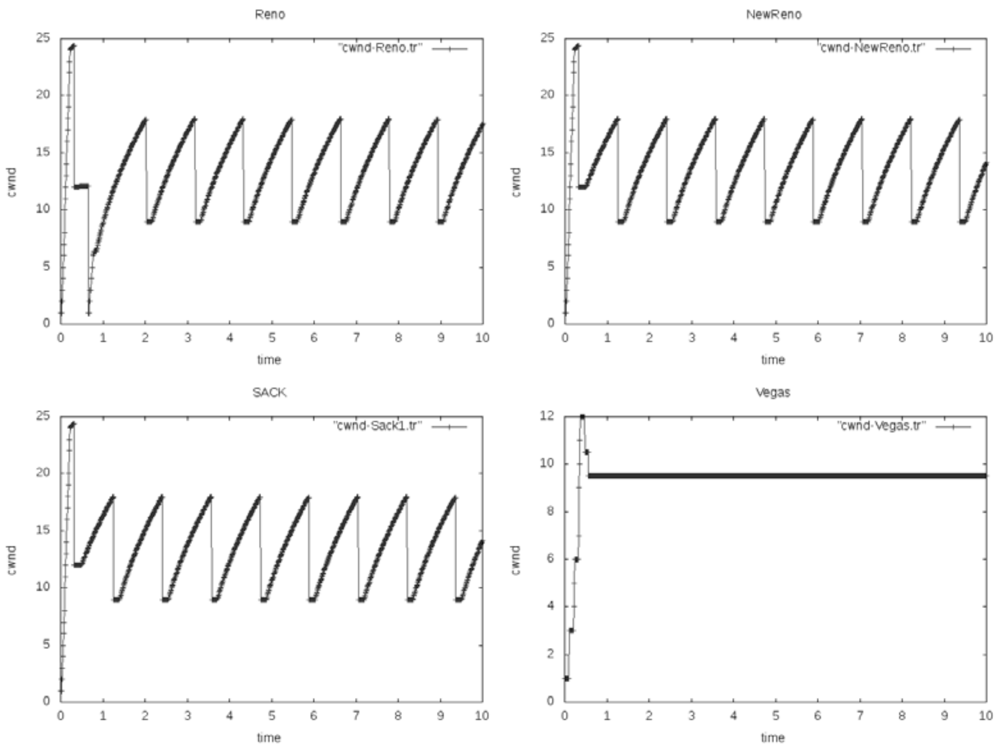

Figure 4. cwnd of Reno, NewReno and SACK in wired network(packet loss), cwnd of Vegas.

As shown in FIGURE 3, we use the similar network in the FIGURE 1. We deliberately set the cache capacity between R1 and R2 to be small to simulate the case of packet loss. We used Reno, NewReno and SACK to send data. We can see different cwnd change from FIGURE 4. NewReno and SACK could deal with packet loss at 0.7s very well while Reno could not. In fact, SACK is more efficient than NewReno at retransmission. This is not shown in the FIGURE 4.

TCP Vegas: TCP Vegas is a mechanism different from the above methods. When the RTT time measured is shorter than a set constant value, the transmission rate is accelerated (cwnd increases). When the RTT time measured is longer than another set constant value, the transmission rate will decrease (cwnd reduced). When the RTT time measured is between this two value, the sending rate will remain the same (cwnd stays the same). TCP Vegas is a less aggressive algorithm.

As shown in FIGURE 4, we use the same network in the FIGURE 1. We use Vegas to send data. We can know cwnd of Vegas will tends to be stable after short time. When multiple TCP connections coexist, Vegas may get into trouble because of less aggressive algorithm.

3. Tcp Performance Simulation In Wireless Networks

We will conduct experiment in low speed mobile wireless networks based on the NS2 in this section.

TABLE 1. Parameters in wireless network experiment.

Parameters Types	**Channel Type**	**radio-propagation mode**	**network interface type**	**MAC type**	**interface queue type**	**link layer type**
Parameters Values	Channel/WirelessChannel	Propagation/TwoRayGround	Phy/WirelessPhy	Mac/802_11	Queue/DropTail/PriQueue	LL
Parameters Types	**antenna model**	**max packet in ifq**	**number of mobile nodes**	**routing protocol**	**scene size**	**simulation time**
Values	Antenna	18	10	DSDV	1000m*300m	50s

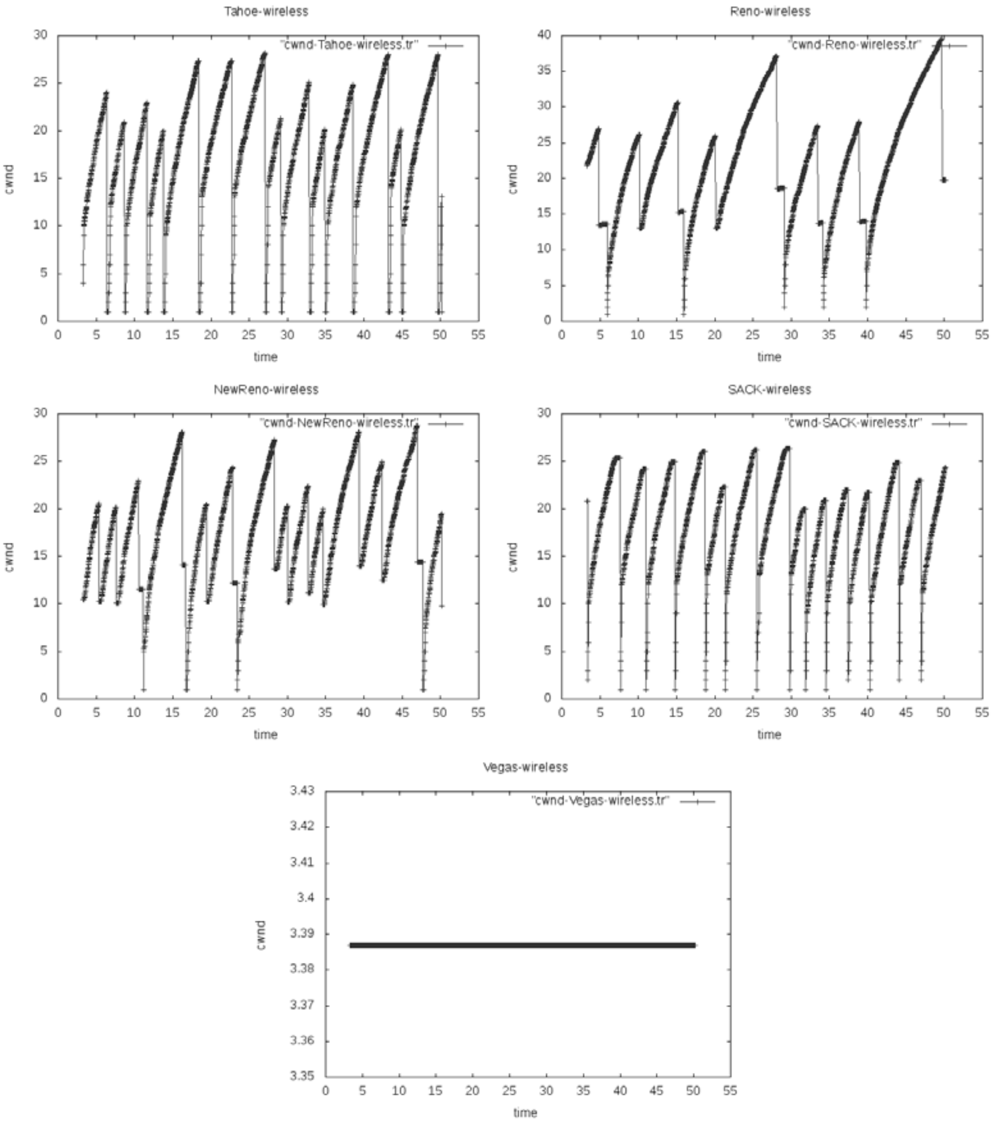

Figure 5. cwnd of different TCP versions in wireless network.

We conduct experiment in low speed mobile wireless networks as the parameters shown in TABLE 1. We use different TCP versions (one TCP connection) and get the following cwnd curve in FIGURE 5.

As shown in FIGURE 5, we can seen performance declines for almost all TCP versions in wireless network. Vegas looks pretty stable because we just one TCP connection. If we add multiple TCP connections or increase mobility of nodes, performance of Vegas will degrade very quickly. Moreover mechanism of Vegas is not universal, so we don't talk about it very much. Performance of Tahoe is similar to its performance in wired networks but the situation of is packet loss is more frequent. Cwnd values of Tahoe are frequently unnecessarily reduced. As for Reno, we know it can deal with single packet loss. In wireless network, error code and packet loss occur frequently and multiple packets are often lost, so Reno is not performing well. SACK and NewReno are two improved algorithms of Reno. SACK have many problems in wireless networks because of too much RTO. Relatively, the performance of NewReno is optimal. This is largely due to ability to handle multiple packet loss and NewReno no longer takes time like SACK to figure out which packets have been lost.

4. Conclusion

We compared the performance of different TCP versions in wireless network with in wired network in our paper. We came to the conclusion: the wireless network environment is more complex than the wired network. The traditional TCP needs to be improved in order to work in wireless network. Specifically, the disadvantages in wireless networks include limited bandwidth, long Round Trip Times, high bit error rate, user mobility and so on [3]. Today Many TCP mechanisms (For example, ADTCP for Ad-Hoc [4]) in wireless networks have been proposed but Their effect was unsatisfactory [5]. This study is the first step to improve TCP in wireless networks.

References

[1] Rajaboina, Raja Sekhar, P. C. Reddy, and R. A. Kumar, "Performance comparison of TCP, UDP and TFRC in static wireless environment," in *International Journal of Information and Computer Security*8.2(2016), pp: 158-180.

[2] Bajeja, Sunil, and A. Gosai, "Performance evaluation of traditional TCP variants in wireless multihop networks," in *International Conference on Computing for Sustainable Global Development* IEEE, 2016.

[3] Zouhair and El-Bazzal, "Improving the performance of TCP over wireless networks," in *International Conference on Digital Information, Networking, & Wireless Communications* IEEE, 2018.

[4] Fu Zhenghua, "Design and Implementation of a TCP-Friendly Transport Protocol for Ad Hoc Wireless Networks," in *IEEE International Conference on Network Protocols* IEEE Computer Society, 2002, pp: 216-225.

[5] Ong.Kevin, "Experimental evaluation of less-than-best-effort TCP over 802.11 wireless networks," in *Communications* IEEE, 2018, pp: 1-6.