Middle-East Journal of Scientific Research 23 (5): 940-943, 2015
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2015.23.05.22233

Efficient Use of Network Property in IEEE 802.11 Using RACC Mechanism

[1]*J. Subash Chandra Bose,* [2]*C. Saravana Kumar,*
[1]*G. Anandhi,* [1]*C. Willson Joseph and* [1]*M. Murugeswari*

[1]Associate Professor and Head, Department of CSE,
Professional Group of Institutions, Coimbatore, India
[2]P.G. Scholoar Department of CSE, Professional Group of Institutions, Coimbatore, India

Abstract: Many applications would require fast data transfer in high speed wireless networks. In transmission control protocol (TCP) cannot effectively utilize the network capacity in wireless networks. In this paper, We propose a RACC in which sender performs loss based control while receiver is performing delay based control. The receiver measures the network bandwidth based on the packet inter arrival interval and uses it to compute a congestion window. After receiving the advertised value feedback from the receiver, the sender then uses the additive increase and multiplicative decrease (AIMD) mechanism to compute the correct congestion window size to be used.

Key words: Wireless network · Delay based control · Packet · Congestion window

INTRODUCTION

IEEE 802.11-It is a set of Media Access Control and physical layer specification for implementing a wireless local area network (WLAN) communication.

Network-It is a telecommunication network that allows computers to exchange the data.

RACC Mechanism-Receiver Assisted Congestion Control Mechanism-Sender performs loss based control while receiver is performing delay-based control.

Many applications would require fast data transfer in high-speed wireless networks nowadays. However, due to its conservative congestion control algorithm, Transmission Control Protocol (TCP) cannot effectively utilize the network capacity in lossy wireless networks [1-5]. Here a receiver-assisted congestion control mechanism (RACC) in which the sender performs loss-based control, while the receiver is performing delay-based control. The receiver measures the network bandwidth based on the packet inter arrival interval and uses it to compute a congestion window size deemed appropriate for the sender. After receiving the advertised value feedback from the receiver, the sender then uses the additive increase and multiplicative decrease (AIMD) mechanism to compute the correct congestion window size to be used [5]. By integrating the loss-based and the delay-based congestion controls, this mechanism can mitigate the effect of wireless losses, alleviate the timeout effect and therefore make better use of network bandwidth. Here a congestion window determines the number of bytes that can be outgoing at any time. The size of this window is calculated by estimating how much congestion there is between the two places.

Receiver-Assisted Congestion Control: Transmitting bulk data over high-speed links is a requirement for many applications. Some times it won□t be satisfied with an ordinary network and it is better to migrated to the wireless networks [6]. However it is not satisfied with the delay-throughput performance because the Transmission Control Protocol (TCP) used in the bulk data transfer suffers from significant throughput degradation and very high interactive delay through the wireless networks. In addition, there is an increasing interest and demand to access the Internet via high-bandwidth wireless networks anytime and anywhere.

Thus, it is very desirable to improve TCP performance in wireless networks because TCP traffic is accounting for about 90% of all Internet traffic nowadays [7]. To understand the poor performance of TCP in

Corresponding Author: Dr. J. Subash Chandra Bose, Associate Professor and Head, Department of CSE, Professional Group of Institutions, Coimbatore, India.

wireless networks, one needs to understand how TCP operates [8]. Upon a timeout or receiving some continuous amount of duplicate acknowledges (ACKs), the duplication of these acknowledges are taken as a packet loss and reduces its congestion window. This technique is very efficient in a traditional wired network. Unfortunately, packet loss in a wireless network may also be due to transmission problems such as a high link error probability, fading and interference. Therefore, packet loss is no longer an appropriate indication for network congestion. With the wrong constructed information, TCP may reduce its congestion window unnecessarily, resulting in poor performance of wireless networks [7].

Another problem of TCP is its poor capability to utilize the network bandwidth efficiently, especially in networks with a high bandwidth delay product (BDP)[8]. Bandwidth-delay product refers to the product of a data link's capacity (in bits per second) and its end-to-end delay (in seconds) [9]. The result, an amount of data measured in bits (or bytes), is equivalent to the maximum amount of data on the network circuit at any given time, i.e. data that has been transmitted but not yet received [9]. Sometimes it is calculated as the data link's capacity times its round trip time. The standard TCP congestion avoidance algorithm employs an additive increase and multiplicative decrease (AIMD) scheme. When there is no packet loss detected, the congestion window (candy) is increased by one maximum segment size (MSS) every round-trip time (RTT). Otherwise (if a packet loss is detected), the TCP sender reduces candy by half if the packet loss is detected by three duplicate ACKs or reduces candy to one if the packet loss is detected by timeout. In a high-speed network with a large RTT, TCP requires a very large window (equal to at least the BDP) to efficiently utilize the network resource. From the above description, one can see that standard TCP takes a very conservative approach to update its window in the congestion avoidance stage. What is worse, upon a retransmission timeout, the sender has to wait for time duration before sending any new packets and this waiting time greatly reduces the TCP throughput.

Unlike regular TCP where the receiver only performs flow control, here it is allowed the receiver to participate in congestion control. A timer is used at the receiver to time the arrival of the next packet and to therefore detect a packet drop if timeout occurs [10]. Since the receiver always detects a packet drop earlier than the sender, it can send an ACK earlier to inform the sender about the timeout the sender is going to see. This would greatly reduce the waiting time of the sender to retransmit a lost packet [1]. The receiver can then estimate the rate the sender should adopt in order to make the best use of this measured bandwidth. The rate is advertised to the sender by embedding the rate information in the ACKs returned to the sender. The sender can then adjust its congestion window based on both the receiver advertised sending rate and the AIMD mechanism [4]. Through this rate-based control, our mechanism can make better use of the available bandwidth and through the window-based control, our mechanism can make good use of the network buffer. Consequently, the afore-described mechanism can improve the throughput performance in loss high-speed wireless networks [4]. Simulation and preliminary experimental results suggest that our mechanism is a promising algorithm to achieve high link utilization while remaining friendly with regular TCP [2].

Relative Work

Evaluation of TCP Vegas: Emulation and Experiment: TCP Vegas detects congestion at an incipient stage based on increasing Round-Trip Time (RTT) values of the packets in the connection. Which detect congestion only after it has actually happened via packet drops? [4, 7]

The effects of packet loss:

- In data, packet loss produces errors.
- In video conference environments it can create jitter.

Binary Increase Congestion control-BIC TCP: Binary Increase Congestion control is one of the congestion control algorithms that can be used for TCP. BIC is optimized for high speed networks with high latency: so-called "long fat networks" [6]. BIC has a unique congestion window (cwnd) algorithm. This algorithm tries to find the maximum where to keep the window at for a long period of time, by using a binary search algorithm. BIC TCP is implemented and used by default in Linux kernels 2.6.18 and above [7].

Exsiting System: The main task of TCP congestion control is to adjust the sending rate of the source in accordance with the state of the network. For this purpose, TCP limits the amount of outstanding data. The congestion window (cwnd) represents the maximum amount of data a sender can have sent and for which no acknowledgment was yet received [7].

Due to network congestion, traffic load balancing or other UN predictable network behavior, IP packets can be lost, duplicated, or delivered out of order. TCP is not satisfied with the delay throughput performance because the TCP used in the bulk data transfer [7].

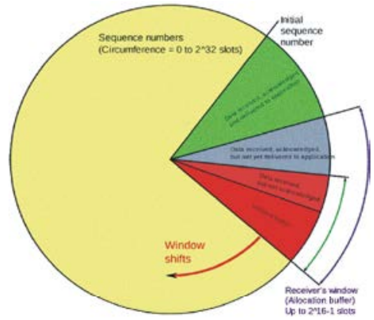
Timeout or TCP receiving three duplicate acknowledges, TCP treat as an indication of a packet loss and reduces its congestion window. In TCP receiver only performs the flow of control.

Proposed System: In RACC (Receiver assisted congestion control) receiver participate in congestion control. Sender and receiver combined and integrating the congestion control mechanism.

The receiver estimates a congestion window deemed to be appropriate from the measured bandwidth and RTT and then advertises the window size (feeds this information back) to the sender. The sender then adjusts its congestion window according to the advertised window of the receiver [10].

In the RACC (Receiver assisted congestion control) method, the sender can increase the congestion window quickly to the available bandwidth, thus improving the network bandwidth [11, 12].

The sender uses the additive increase and multiplicative decrease (AIMD) mechanism to compute the correct congestion window size to be used by integrating the loss based congestion controls and delay based congestion control [5].



Transmission Control Protocol

Network Topology used

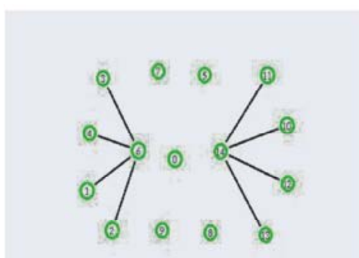


Fig 7.1, network topology for simulation

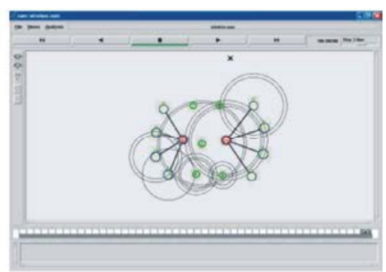


Fig 7.2 Simulation of the Network

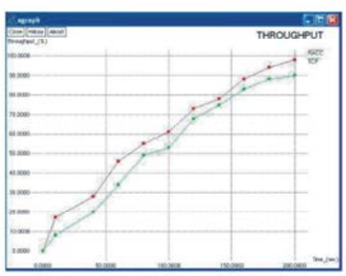


Fig 7.3 Throughput Analysis

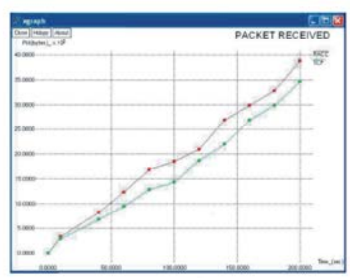


Fig 7.4 Data Packet Reception

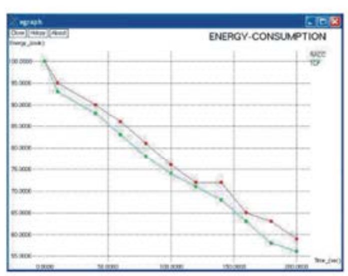


Fig 7.5 Energy Consumption

CONCLUSION

In receiver congestion control mechanism, the sender can increase the congestion window quickly to the available bandwidth, thus improving the network utilization. Receiver end protocol is modified without measurements changing the sender functions; it can still improve the network throughput and fairness performance. The implementation of RACC in the Windows XP OS and test bed confirm the RACC can improve the throughput considerably. Screen Shot

REFERENCES

1. Casetti, C., 2001. TCPWestwood: Bandwidth estimation for enhanced transport over wireless links," in Proc. 7th ACM Annu. Int. Conf. Mobile Comput. Netw., Rome, Italy, pp: 287-297.
2. Wei, D.X., C. Jin, S.H. Low and S. Hedge, 2006. FAST TCP: Motivation, architecture, algorithm and performance, IEEE/ACM Trans. Netw., 14(6): 1246-1259.
3. Hsieh, H., 2003. A receiver-centric transport protocol for mobile hosts with heterogeneous wireless interfaces, in Proc. 9th ACM Annu. Int. Conf. Mobile Comput. Netw., San Diego, CA, pp: 1-15.
4. Ahn, J.S., P.B. Danzig, Z. Liu and L. Yan, 1995. Evaluation of TCP Vegas: Emulation and experiment," Comput. Commun. Rev., 25(4): 185-195.
5. Shi Kai, Yantai Shu, Oliver Yang and Jiarong Luo, 2010. Receiver-Assisted Congestion Control to Achieve High Throughput in Lossy Wireless Networks", IEEE Trans. nuclear science, 57(2).
6. Xu, L., K. Harfoush and I. Rhee, 2004. Binary increase congestion control (BIC) for fast long-distance networks, in Proc. 23rd IEEE INFOCOM, Hong Kong, China, pp: 2514-2524.
7. Brakmo, L.S. and L.L. Peterson, 1995. TCP vegas: End to end congestion avoidance on a global internet, IEEE J. Sel. Areas Commun., 13(8): 1465-1480, Oct.
8. Paxson, V. and M. Allman, 2000. Computing TCP?s retransmission timer, IETF, RFC 2988.
9. Stevens W., 1997. TCP slow start, congestion avoidance, fast retransmit, IETF, RFC 2001.
10. Hong, Y. and O. Yang, 2007. Design of adaptive PI rate controller for best-effort traffic in the internet based on phase margin, IEEE Trans. Parallel Distrib. Syst, 18(4): 550-561.
11. Shu, Y., 2008. Mobile-host-centric transport protocol for EAST experiment, IEEE Trans. Nucl. Sci., 55(1): 209-216.
12. Shu, Y., L. Zhang, W. Zhao, H. Chen and J. Luo, P2P-based data system.