IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 23, Issue 5, Ser. II (Sep. –Oct. 2021), PP 14-17
www.iosrjournals.org

New Protocol to Avoid Congestion in Wireless Network

Dr. Ahlal H.Montaser. Mohamed
Faculty of IT University of Tripoli, Libya,

Abstract:
TCP Reno and TCP Vegas work well in wire network, but they do not work properly in wireless network, because they take packet losses or timeout as the signal of congestion. However, in wireless network, temporary link outages or fading-induced bit errors will result in packet losses, so TCP will decrease CWND improperly. Some differences of TCP have been developed to improve the transport protocol performance for wireless network. It is to hide any reduction not related to congestion from the sender, so that the sender can only detect the packet losses due to congestion, such as I-TCP or snoopy protocol. However these protocols are not fit for Ad Hoc network in which there are not middle base stations between wire link and wireless link. Thus, the other direction, Explicit Congestion Notification (ECN), is a promising direction in wireless network. This paper consists, a new protocol based on expanded ECN mechanism to respond packet losses in wireless environment properly. The paper gives the idea to construct the new mechanism and analyses the stability of the new mechanism. By using Matlab Simulator, the paper compares the new mechanism on throughput, drop ratio and delay time with older mechanism.

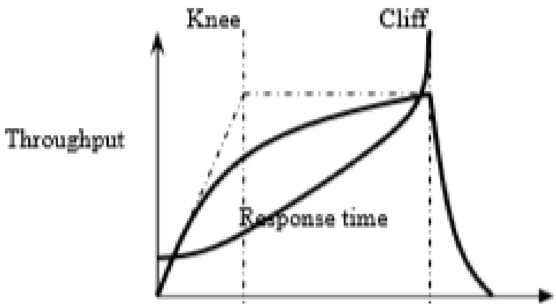
Date of Submission: 08-10-2021 Date of Acceptance: 23-10-2021

I. Introduction

Communication over the wireless network is increased these days and increases the traffic in the wireless network also. This paper consists to reduce the traffic in the network. These times the communication technology connect more and more wireless devices and wireless channels are added to Internet and they give the Internet many new properties, consequently, causing many new problems to be solved, such as, significant bit-error rates (BER), high and variable latency, even variable link bandwidth and short-duration sessions. The key factors affecting TCP/IP performance in wireless environment are that the TCP end hosts cannot detect the congestion by conventional method used in wire line network such as sensing congestion by packet losses or the duplicate Acknowledgements, so the TCP end hosts cannot find the available bandwidth efficiency. It is important by relating the link level states to the TCP session and using of finer congestion control in wireless network. It is a very common idea to explicitly notify the congestion occurring to the end host, rather than let the end hosts apply complex method to predict the congestion occurring. This is a very promising direction to improve the performance of transport protocol over wireless network especially in Ad Hoc environment. One method to explicitly find congestion is to use AMPP packets. The advantage of AMPP is that it can notify the sender timely, but AMPP can consume the network resources and AMPP packets may loss themselves at that time if there are other routers in the path. ECN can be a more preferable mechanism to adapt to the wireless properties. ECN mechanism is one mechanism of Active Queue Management (AQM) described in. It informs the sender of congestion by setting the Congestion Experienced (CE) codepoint in the IP header of packets. IP header has two bits in ECN field. ECN signals consist of four ECN binary codepoints, '00', '01', '10', '11'. The codepoint '00' called not-ECT indicates a packet without using ECN. The code points '10' and '01' called ECT(0) and ECT(1) are set by a data sender to indicate that it is ECN-capable Transport(ECT). Although they have different names, they have the same meaning to routers. The senders can use either ECT(0) or ECT(1). The CE codepoint '11' is set by a router to indicate congestion occurring. This paper expand ECN mechanism by using the two bits to display the congestion level, so the ECT(0) and ECT(1) will have different meaning in our mechanism, and also, the senders have more fine information about network state and can respond the report more efficiently.

IMPROVED PROTOCOL

Network performance versus network load has shown in figure. It seen that the throughput of the network will decline rapidly when the load increases beyond the “cliff”. The traditional older algorithm of TCP Reno lets the senders continues to probe the “cliff” point by linearly increasing the CWND so that the senders can close to the “cliff”. Beyond the “cliff”, the senders will halve the CWND that is, decreasing the sending rate exponentially.

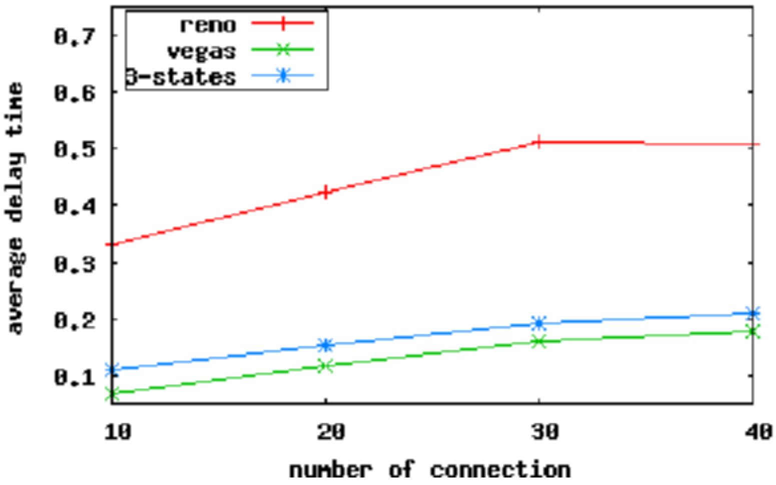


Older mechanism of TCP Reno is not stable from the point of view of control when the product of the delay time and capacity increases, such as in high-speed network. Also, the sending rate usually oscillates with large amplitude and results in retransmission. The best performance of a network should close to “knee” rather than “cliff” when considering the increase of response time. One good congestion control mechanism should drive the sending rate to converge to “knee” and make the flows share the bandwidth fairly. TCP Vegas modifies TCP Reno such that the rate can oscillate near the rate when beginning to connect, which can be thought as the allowed rate since congestion does not occur at that time. However, TCP Vegas does not compete with TCP Reno. To solve these problems, the sender need more clear information about network states. Then, we call the network in available state, or A-state. Although P-state phase looks like the slow-start phase of TCP Reno, it differs from the slow-start phase essentially. The slow-start phase based on sender action is the period when the sender begins to send data or timeout occurs, while the P-state phase describing network load is independent of the sender and relative to the whole network. The mechanism adopted by the sender in the slow-start phase of TCP Reno is also to increase its sending rate as soon as possible, which would be implemented in P-state. The reason of defining the slow-start phase by sender behavior is that the sender does not know the network state in TCP Reno. When the load increases between “knee” and “cliff” in the throughput increases little while the response time increases rapidly. In this phase, the performance decreases actually. Because in the phase the packet queue in the router has built up, we call the network in busy state, or Q-state. We should stabilize the network in Q-state as possible as we can, and drive it close to “knee” to get the best performance. Similarly, Q-state phase looks like the congestion avoidance phase in TCP Reno, but it differs from the congestion avoidance phase. The mechanism in the congestion voidance phase of TCP Reno is to increase its sending rate cautiously, which will often result in congestion finally. This is very important probing mechanism to let the sender know the available bandwidth currently. Otherwise, the sender is not able to use enough the link resources because of the property of “black box” of the network. In fact, if the router can explicitly inform the available resources to the sender, we can use better control mechanism to avoid the congestion, and consequently, to avoid unnecessary retransmission. So when the network is in Q-state, the sending rate should be adjusted continuously instead of increasing in one direction, so that the entire network converges to “knee” finally.

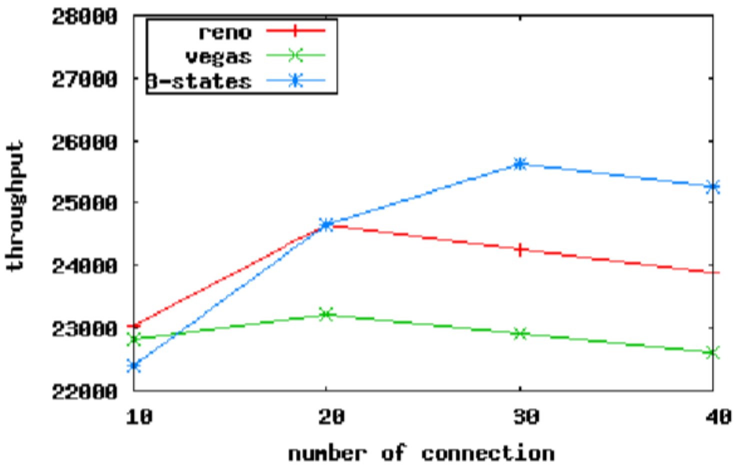
R-state phase is similar to the retransmit and recovery phase in TCP Reno, in which the packet losses can be used as the signal that the network load outgoes the “cliff” and the network enters into congested state. The mechanism in the retransmit and recovery phase of TCP Reno is to halve the CWND and ssthresh. However, the packet losses will result in retransmission, so using packet losses as congestion signal is not a satisfactory method. Further in wireless network, packet losses can results from other causes such as link errors. Thus it is necessary to differentiate packet losses due to congestion from packet losses due to other reasons. ECN can be a convenient method without introducing extra traffic in the network and can give congestion signal in advance, that is, before the network enters into R-state.

II. Simulation Comparison

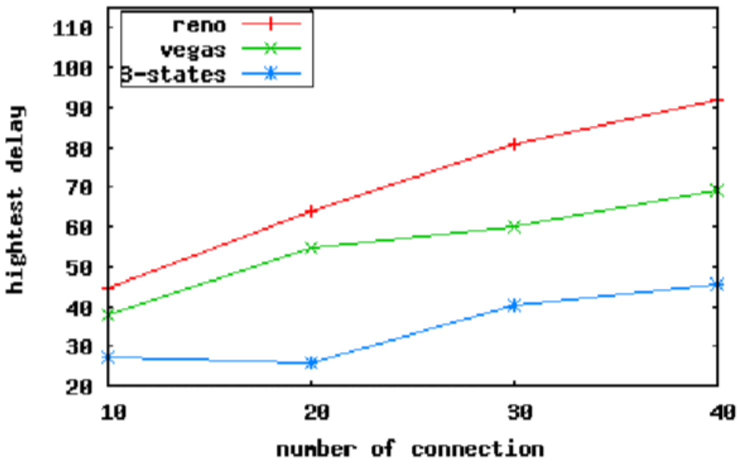
Matlab is used to generate the mobile scene, which has 60 nodes moving within 450m×450m flat topology, in which all of the connections use long ftp connection. The number of the connection is set to 10, 20, 30 and 40 respectively. The run time is 300 sec. The ad hoc routing selects the AODV protocol and all the other wireless parameters adopt default setting. In the simulation, the nodes can use TCP Reno, TCP Vegas and TCP 3-states/ECN2 protocols. For convenience of comparison, in each simulation, all of the nodes use the same transport protocol. For a given connection number and transport protocol, it run 10 tests to calculate the average ofthe results. The 10 tests have different mobile scenes and traffic models individually. The simulation compares the throughput, drop ratio, average delay time and highest delay time versus the number of the connections. When the connections increase, from the above test condition, the net load will increase, so according to the experimental results, we can see the protocol performance when the traffic load increases.



When traffic increases, the throughput of TCP Vegas does not increase correspondingly, while the throughput of another two TCP protocol increases much. Thus it is obvious that TCP Vegas does not utilize net resource adequately. We also see that if the traffic load increases more, TCP Reno does not improve the throughput more. Because it does not estimate the available bandwidth rightly and blocks the buffer often, which will delay the sender receiving ACK.? Even under high load condition, TCP 3-states mechanism can still reach a high throughput. It displays the change of average delay time of the three protocols. From the figure we can see that the average delay time of TCP Reno is much higher than TCP Vegas and TCP 3-states, while TCP Vegas has the lowest average delay time.



It displays the change of average delay time of the three protocols. From the figure below it can see that the average delay time of TCP Reno is much higher than TCP Vegas and TCP 3-states, while TCP Vegas has the lowest average delay time.



It displays the drop ratio comparison of the three protocols. From the figure, we can see that due to the short queue length, TCP Vegas and TCP 3-states have lower drop ratio. It displays the highest delay time comparison of the three protocols. From the figure, we can see that TCP 3-states have a lowest highest delay time, which is an excellent property when it is used for interactive TCP flows. Based on the above discussion, we can see that in the wireless environment, the performance of TCP 3-states mechanism will exceed the traditional TCP version, such as TCP Reno or TCP Vegas.

III. Conclusion

In this paper, to improve the congestion control in wireless environment, we modify ECN mechanism to provide tripe feedback to senders, which we call ECN2 since it uses the two bits in ECN field. We also improve the traditional mechanism to respond ECN2 feedback information, which we called TCP 3-states because it has three different control schemes according to different ECN2 feedback. Based on a flow model we analyze the stability of TCP 3-states. Then by simulation on Matlab platform, it can see that TCP 3-states/ECN2 mechanism can utilize the network resources more efficiently with low drop ratio and low delay time.

References

[1]. K. Kar, S. Sarkar, and L. Tassiulas, A Simple Rate Control Algorithm for Maximizing Total User Utility, IEEE INFOCOM, April 2001.

[2]. L.S. Brakmo and L.L. Peterson, TCP Vegas: End to End Congestion Avoidance on a Global Internet, IEEE Journal on Selected Areas in Communications. I3(8):, October 1995, 1465-80.

[3]. T. Kelly, Scalable TCP: Improving Performance in High-speed Wide Area Networks, First International Workshop on Protocols for Fast Long-Distance Networks, February 2003.

[4]. A. B. M. Alim AI Islam and Vijay Raghunathan, "Endto-End Congestion Control in Wireless Mesh Networks using a Neural Network", IEEE WCNC, West Lafayette, 2011.

[5]. Dr. T.Karthikeyan and B. Subramani, "A New Congestion Control Algorithm Based on Novel AQM" , T.Karthikeyan et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 4440-4443, 2014.

[6]. K. A. Yadav and S. Kumar, "A review of congestion control mechanisms for wireless networks," 2017 2nd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, 2017, pp. 109-115. doi: 10.1109/CESYS.2017.8321245.\

[7]. G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," Proc. ACM Mobicom '99, Seattle, WA, 1999

[8]. K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A Feedback based scheme for improving TCP performance in Ad-Hoc wireless networks," in Proc. of the International Conference on Distributed Computing System (ICDCS 98), Amsterdam, Netherlands, May 1998.

[9]. J. Liu and S. Singh, "ATCP: TCP for mobile ad hoc networks," IEEE J-SAC, vol. 19, no. 7, pp. 1300–1315, 2001.

Ahlal H. Montaser. Mohamed .PhD in the Department of Computing at University of Bradford, UK(**2004 – 2008**). **BSc** at Department of control engineering, Faculty of Electronic Engineering, Baniwalid, Libya (**1994-1999**). **MSc** Computing at at University of Bradford, UK (2002-2003). **staff Member in Faculty IT University of Tripoli(2016).**
a.montaser@uot.edu.ly