International Journal of Technology Research and Management
ISSN (Online): 2348-9006
Vol 2 Issue 12 December 2015

A Survey on Techniques to Enhance Performance of TCP Data type and its variants for Wireless adhoc Network

Pinky Singh[1], Mrs. Margi Patel[2]
Pursuing M.E.[1], Asst. Professor[2]
Pinkising0823@gmail.com[1]

ABSTRACT: *Mobile Ad-Hoc Network is an autonomous group of mobile users that communicate using wireless links with no support from any pre-existing infrastructure network and used as a highly reliable end-to-end protocol for transporting applications.*

In this paper analysis of the four TCP variants (New Reno, SACK, TCP TAHOE and VEGAS) under a variety of network conditions. The simulations results reveal that out of the four, the SACK variant can adapt relatively well to the changing network sizes while the VEGAS performs most robustly in different node density scenarios. On the other hand, the research asserts the fact of superiority of, reactive protocol over proactive protocol when routing the same traffic in the network. Nonetheless, among the reactive protocols AODV performance (in the presence of a high mobility) has been found to be remarkable.

Keywords: *MANET, Routing Protocol, TCP variants (NEWRENO, SACK, TCP TAHOE and VEGAS), ns-2.35.*

1. Introduction:

The advent of ubiquitous computing and the proliferation of portable computing devices have raised the importance of mobile and wireless networking. A mobile ad hoc network is an autonomous collection of mobile nodes forming a dynamic network and communicating over wireless links. Ad hoc communication concept allows users to communicate with each other in a multi-hop fashion without any fixed infrastructure and centralized administration. Due to their capability of handling node failures and fast topology changes, such networks are needed in situations where temporary network connectivity is required, such as in battlefields, disaster areas, and large meeting places. Such networks provide mobile users with ubiquitous communication capability and information access regardless of location. TCP has gained its place as the most popular transmission protocol due to its wide compatibility to almost all today's applications. However, TCP as it exists nowadays may not well fit in mobile ad hoc networks since it was designed for wire-line networks where the channel Bit Error Rate (BER) is very low and network congestion is the primary cause of packet loss. On the contrary of wired links, wireless radio channels are affected by many factors that may lead to different levels of BER. Wireless channel behavior cannot be predictable, but in many cases, such channels are having a high BER that cannot be neglected when studying ad hoc networks. Furthermore, node's mobility can also affect TCP sessions in ad hoc networks, which is obviously not the case of wired networks. Indeed, when nodes move, link can be broken and TCP sessions using that links can lose packets. Hence, TCP does not have the capability to recognize whether the packet loss is due to network congestion or channel errors.

2. Motivation:

In the last few years, many researchers have studied TCP performance in terms of energy consumption and average good put within wireless mobile networks [2][3][4]. Due to the specific issues related to wireless ad hoc networks, it is expected that the performance of TCP will be affected considerably in these environments. In wireless ad hoc networks, reasons for packet losses are more sophisticated than traditional wireless (cellular) networks. Those reasons include the unpredictable wireless channel characteristics due to fading and interference (implying a high BER), the vulnerable shared media access due to random access collision, the hidden and exposed terminal problems, path asymmetry, multi-path routing, and so on. Undoubtedly, all of these pose great challenges on TCP to provide reliable end-to-end communications in such environment.

3. Objective:

Thesis contains the objective which follows:-

- The focal point of this thesis study and analyze of TCP Variants.
- Creating scenario for different node density which are 50 nodes, 75 nodes and 100 nodes.
- Study and analyze the effects of energy, packet delivery ratio, throughput and end to end delay in wirelessscenario with different- different node density.
- The results of both Proactive and Reactive protocols compare to analyze which of these two types of protocols gives better performance.

4. Transmission Control Protocol (TCP) :

TCP is a reliable connection oriented end-to-end protocol. It contains within itself, mechanisms for ensuring reliability by requiring the receiver acknowledge the segments that it receives. The network is not perfect and a small percentage of packets are lost en route, either due to network error or due to the fact that there is congestion in the network and the routers are dropping packets. We shall assume that packet losses due to network loss are minimal and most of the packet losses are due to buffer overflows at the router [5]. Thus it becomes increasingly important for TCP to react to a packet loss and take action to reduce congestion.

In our work we compare the four TCP variants which are as follows-

- TCP Tahoe
- New RENO
- SACK
- Vegas

A. TCP Tahoe:

Tahoe refers to the TCP congestion control algorithm which was suggested by Van Jacobson in his paper [5]. TCP is based on a principle of ‘conservation of packets’, i.e. if the connection is running at the available bandwidth capacity then a packet is not injected into the network unless a packet is taken out as well. TCP implements this principle by using the acknowledgements to clock outgoing packets because an acknowledgement means that a packet was taken off the wire by the receiver. It also maintains a congestion window CWD to reflect the network capacity [5]. However there are certain issues, which need to be resolved to ensure this equilibrium.

1. Determination of the available bandwidth.
2. Ensuring that equilibrium is maintained.
3. How to react to congestion.

Slow Start: TCP packet transmissions are clocked by the incoming acknowledgements. However there is a problem when a connection first starts up cause to have acknowledgements you need to have data in the network and to put data in the network you need acknowledgements. To get around this circularity Tahoe suggests that whenever a TCP connection starts or re-starts after a packet loss it should go through a procedure called ‘slow-start’. The reason for this procedure is that an initial burst might overwhelm the network and the connection might never get started. Slow starts suggest that the sender set the congestion window to 1 and then for each ACK received it increase the CWD by 1. So in the first round trip time (RTT) we send 1 packet, in the second we send 2 and in the third we send 4. Thus we increase exponentially until we lose a packet which is a sign of congestion. When we encounter congestion we decreases our sending rate and we reduce congestion window to one. And start over again.

The important thing is that Tahoe detects packet losses by timeouts. In usual implementations, repeated interrupts are expensive so we have coarse grain time-outs which occasionally checks for time outs. Thus it might be some time before we notice a packet loss and then re-transmit that packet.

Congestion Avoidance: For congestion avoidance Tahoe uses ‘Additive Increase Multiplicative Decrease’. A packet loss is taken as a sign of congestion and Tahoe saves the half of the current window as a threshold. value. It then set CWD to one and starts slow start until it reaches the threshold value. After that it increments linearly until it encounters a packet loss. Thus it increase it window slowly as it approaches the bandwidth capacity.

B. NEWRENO:

New RENO is able to detect multiple packet losses and thus is much more efficient that RENO in the event of multiple packet losses. Like Reno, New-Reno also enters into fast-retransmit when it receives multiple duplicate packets, however it differs from RENO in that it doesn't exit fast-recovery until all the data which was out standing at the time it entered fast- recovery is acknowledged [4]. Thus it overcomes the problem faced by Reno of reducing the CWD multiples times.

C. SACK:

TCP with 'Selective Acknowledgments' is an extension of TCP RENO and it works around the problems face by TCP RENO and TCP New-Reno, namely detection of multiple lost packets, and re-transmission of more than one lost packet per RTT. SACK retains the slow-start and fast- retransmits parts of RENO. It also has the coarse grained timeout of Tahoe to fall back on; increase a packet loss is not detected by the modified algorithm.

SACK TCP requires that segments not be acknowledged cumulatively but should be acknowledged selectively. Thus each ACK has a block which describes which segments are being acknowledged. Thus the sender has a picture of which segments have been acknowledged and which are still outstanding. Whenever the sender enters fast recovery, it initializes a variable pipe which is an estimate of how much data is outstanding in the network, and it also set CWND to half the current size. Every time it receives an ACK it reduces the pipe by 1 and every time it re-transmits a segment it increments it by 1. Whenever the pipe goes smaller than the CWD window it checks which segments are unreceived and send them. If there are no such segments outstanding then it sends a new packet [6]. Thus more than one lost segment can be sent in one RTT.

D. VEGAS:

Vegas is a TCP implementation which is a modification of Reno. It builds on the fact that proactive measures to encounter congestion are much more efficient than reactive ones. It tried to get around the problem of coarse grain timeouts by suggesting an algorithm which checks for timeouts at a very efficient schedule. Also it overcomes the problem of requiring enough duplicate acknowledgements to detect a packet loss, and it also suggest a modified slow start algorithm which prevent it from congesting the network. It does not depend solely on packet loss as a sign of congestion. It detects congestion before the packet losses occur. However it still retains the other mechanism of Reno and Tahoe, and a packet loss can still be detected by the coarse grain timeout of the other mechanisms fail.

5. Routing in Ad Hoc Network:

The routing protocols for ad hoc wireless network should be capable to handle a very large number of hosts with limited resources, such as bandwidth and energy. The main challenge for the routing protocols is that they must also deal with node density, meaning that nodes can appear and disappear in various scenarios. Thus, all nodes of the ad hoc network act as routers and must participate in the route discovery and maintenance of the routes to the other nodes. For ad hoc routing protocols it is essential to reduce routing messages overhead despite the increasing number of nodes and their mobility. Keeping the routing table small is another important issue, because the increase of the routing table will affect the control packets sent in the network and this in turn will cause large link overheads [9]. In this paper we are working on AODV and DSDV protocols.

A. AODV:

The Ad Hoc On-Demand Distance Vector (AODV) routing protocol described in [14] builds on the DSDV algorithm previously described. AODV is an improvement on DSDV because it typically minimizes the number of required broadcasts by creating routes on a demand basis, as opposed to maintaining a complete list of routes as in the DSDV algorithm. The authors of AODV classify it as a pure on demand route acquisition system, since nodes that are not on a selected path do not maintain routing information or participate in routing table exchanges [14]. When a source node desires to send a message to some destination node and does not already have a valid route to that destination, it initiates a path discovery process to locate the other node. It broadcasts a route request (RREQ) packet to its neighbors, which then forward the request to their neighbors, and so on, until either the destination or an intermediate node with a " fresh enough" route to the destination is located. Figure 2.3.a illustrates the propagation of the broadcast RREQs across the network. AODV utilizes destination sequence numbers to ensure all routes are loop-free and contain the most recent route information. Each node maintains its own sequence number, as well as a broadcast ID. The broadcast ID is incremented for every RREQ the node initiates, and together with the node's IP address, uniquely identifies an RREQ.

B. DSDV:

The Destination-Sequenced Distance-Vector Routing protocol (DSDV) described in [12] is a table-driven algorithm based on the classical Bellman-Ford routing mechanism [13]. The improvements made to the Bellman-Ford algorithm include freedom from loops in routing tables. Every mobile node in the network maintains a routing table in which all of the possible destinations within the network and the number of hops to each destination are recorded. Each entry is marked with a sequence number assigned by the destination node. The sequence numbers enable the mobile nodes to distinguish stale routes from new ones, thereby avoiding the formation of routing loops. Routing table updates are periodically transmitted throughout the network in order to maintain table consistency. To help alleviate the potentially large amount of network traffic that such updates can generate, route updates can employ two possible types of packets. The first is known as a full dump. This type of packet carries all available routing information and can require multiple network protocol data units (NPDUs).

6. Conclusion:

From our study it is concluded that the TCP variant SACK is best between these four variants along with VEGAS is burst variant in terms of previous result. When we analyze Protocols we cannot analyze clearly that which one is better.

References:

[1] S. Senouci, and G. Pujolle, "Energy efficient consumption in wireless ad hoc networks" IEEE ICC 2004 (International conference on communications), Paris JUNE 2004.

[2] M. Zorzi and R. Rao, "Energy Efficiency of TCP in a local wireless environment"mobile networks and applications,, vol. 6, no. 3, July 2001.

[3] S. Agrawal and S. Singh, "An Experimental Study of TCP' s Energy Consumption over a Wireless Link" 4th European personal Mobile Communications Conference , Feb 20-22, 2001, Vienna Austia.

[4] H. Singh and S. Singh, "Energy consumption of TCP Reno, New Reno, and SACK in multi-hop wireless networks", in ACM SIGMETRICS 2002, June 15-19 2002.

[5] V. Jacobson., "Congestion avoidance and control", SIGCOMM symposium on communications architectures and protocols, pages 314-329,1988. An updated version is available via ftp://ftp.ee.lbl.gov/papers/congavoid.ps.z.

[6] K. Fall and S. Floyd., "Simulation-based comparison of Tahoe, Reno, and sack TCP", in ACM computer communications review, july 1996.

[7] Thomas Clausen, "Comparative Study of Routing Protocols for Mobile Ad-Hoc NETworks", INRIA, March 2004.

[8] L. Brakmo and L. Peterson, "TCP Vegas: end-to-end congestion avoidance on a global internet," IEEE Journal on Selected Areas in Communication, vol. 13, pp. 1465-1480, Oct. 1995.

[9] A. Huhtonen, "Comparing AODV and OLSR Routing Protocols", session on Internetworking, April 2004.

[10] A. Aaron and J. Weng, "Performance Comparison of Ad-hoc Routing Protocols for Networks with Node Energy Constraints", IEEE INFOCOM, Hong Kong, 2004.

[11] Elizabeth M. Royer, Chai-Keong Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks", IEEE Personal Communications, April 1999.

[12] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers", Comp. Comm. Rev., Oct. 1994, pp. 234-44.

[13] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu and J. Jetcheva, "A Performance comparison of multi-hop wireless ad hoc network routing protocols," in Proc. ACM/IEEE Int. Conf. on Mobile Computing and Networking, pp. 85-97, October 1998.

[14] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing", Proc. 2nd IEEE Wksp. Mobile Comp. Sys. And Apps., Feb. 1999, pp. 90-100.