Global Journal of Computer Science and Technology: E
Network, Web & Security
Volume 14 Issue 8 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Energy Consumption of TCP in Ad Hoc Networks

By K. Bhavana & M. Himaja
KL University, India

Abstract- In this paper we study the energy cost (protocol processing and communication cost) and goodput of different flavors of TCP (Transmission Control Protocol) in ad hoc networks.We implemented a tes tbed and measured the actual energy cost as well as goodput of running TCP Reno, Newreno, SACK (Selective ACKnowledgement) and a version that combines Explicit Link Failure Noti_cation (ELFN)and Explicit Congestion Noti_cation (ECN) in Newreno. We see that the use of ECN & ELFN does yield higher good put in most cases with a corresponding lower total energy cost. We see an energy savings of between 20% and 500% depending on the network conditions.

GJCST-E Classification : *C.2.5 C.2.6*

ENERGYCONSUMPTIONOFTCPINADHOCNETWORKS

Strictly as per the compliance and regulations of:

Energy Consumption of TCP in Ad Hoc Networks

K. Bhavana [α] & M. Himaja [σ]

Abstract- In this paper we study the energy cost (protocol processing and communication cost) and goodput of different flavors of TCP (Transmission Control Protocol) in ad hoc networks.We implemented a tes tbed and measured the actual energy cost as well as goodput of running TCP Reno, Newreno, SACK (Selective ACKnowledgement) and a version that combines Explicit Link Failure Noti_cation (ELFN)and Explicit Congestion Noti_cation (ECN) in Newreno. We see that the use of ECN & ELFN does yield higher good put in most cases with a corresponding lower total energy cost. We see an energy savings of between 20% and 500% depending on the network conditions.

I. Introduction

Communication plays a major role in the ad-hoc networks and is used many applications. It account for a large proportion of energy usage. Energy is an important factor in the ad-hoc networks. It is very essential to lower the energy consumption in the adhoc networks. There are many techniques for reducing energy consumption and energy cost in ad-hoc networks. MAC protocols and routing protocols use energy based metrics.

These approaches reduces the energy cost. Aditionally the energy of the TCP also can be reduced as well. There are four variants in saving the energy. The four variants in saving the TCP energy are: Reno ,New Reno ,SACK ,and TCP-ECN-ELFN. SACK means selective acknowledgement. TCP-ECN-ELFN is a combination of ECN AND ELFN. ECN means Explicit Congestion Notification. ELFN is Explicit Link Failure Notification. ECN is a mechanism that enables the senders to respond quickly to the beginning congestion in the network. When the energy cost is measured there is a good throughput for this mechanism. There is a good total energy and idealized energy for this mechanism. The idealized energy is defined as the energy consumed by the sender for transmitting or sending or receiving. The other variant TCP-ECN-ELFN mechanism results in the lower energy consumption when compared to the SACK. The other variants of TCP that is Reno and New Reno also had a good throughput. In this paper we discuss about the energy model and summary of the various TCP variant mechanisms.

Author α σ : Department of Electronics and Computer Engineering KL University. e-mail: tejacool987@gmail.com

II. Related Work

The link is an approach it includes the effect of ARQ AND FEC and the combination of the two in the ad-hoc networks. There are some link layer schemas to improve the energy behavior. The key idea is to discard the packet transmission when channel conditions are worsen. When the channel conditions is good then the packet transmission is resumed. The three implementations of TCP theno ,Reno, New Reno. This mainly focuses on the wired and the wireless environment.

III. Over View of TCP Variants

At present all the TCP implementations depends on tahoe. Various algorithms are incorporated on TCP for slow start ,fast avoidance and fast retransmit and modifications in the formulas for estimation the RTT. RTT means round trip time. The TCP RENO is very much similar to the tahoe but there is a slight difference that is the fast retransmit algorithm this fast retransmit algorithm includes the fast recovery. When a sender receives three duplicate acknowledgment signals then it reduces by half. But as not like a tahoe it becomes the slow start. Thus the RENO increases the congestion rapidly by setting it to the minimum. Here the retransmit timer will turn off and this leads to the congestion and the low throughput.

TCP New Reno overcome the disadvantages of the RENO. A partial acknowledgments infers that there are some unacked packets in the senders window. In RENO a partial acknowledgment gives the sender the fast recovery in a view of the multiple packet losses. When ever the receiver gets a data is out of sequence then that unsequences data creates a hole in the buffer that is present at the receivers end. This is the reason why the reciver generates a duplicate acknowledgment. The receiver includes the starting and ending sequence addresses that is the sequence numbers. These sequence numbers are present in the SACK. The first block in the SACK represents the recently transmitted segment to the reciever. The remaining SACK blocks represents the recently reported blocks. This algorithm is helpful for TCP to recover from multiple segment losses of data with in one round trip time.

When the sender comes to know that there is a loss of the packet then it retransmits and reduces the congestion to half and does fast recovery in RENO and

Global Journal of Computer Science and Technology (E) Volume XIV Issue VIII Version I Year 2014

New RENO. SACK has a variable named pipe it gives the number of packets in the flight. This pipe variable is increased by one that is incremented for the transmission and it is decreased by one that is decremented when it receives a duplicate zed energy cost is high for SACK.

a) TCP-ECN-ELFN

It summarizes the changes made to the operation of TCP to include ECN and ELFN. We note that our implementation goes beyond simply adding ELFN and ECN to TCP - we no longer treat timeouts and triple duplicate ACKs as indications of congestion. Rather, we rely exclusively on ECN to ag network congestion. The table also describes the intuition behind these changes.

i. *Routing Failure*

It describes the interplay between routing failure (due to link outage or propagation of stale routes) and TCP throughput, in detail. Briey, successive route failures (due to link failure) lead to timeouts hence resulting in a small congestion window.. Hence, the throughput of the connection is small. The proposed in and used by us is as follows. A route failure message is propagated back to the TCP sender from the intermediate node that detects the route failure. This message has the effect of freezing TCP's state and initiating the transmission of probe packets. When there is a response to the probe packet (i.e., the route is up), TCP's state is unfrozen and transmission resumes. This solution ensures that there are no timeouts(and hence no unnecessary retransmissions), and that the TCP sender begins sending packets soon after the route is up.

ii. *Out of Order Packets*

Mobility of nodes can cause packets belonging to the same connection to be routed along different routes. This can result in the receiver getting out-of-order packets which causes duplicate ACKs to arrive at the sender. Likewise, packet loss due to link-layer errors can result in triple duplicate ACKs or timeouts. On receiving three duplicate ACKs, the sender reduces its congestion window by a half and retransmits the out-of-sequence packet while in the case of timeouts, the window is reduced to one or two segments. This congestion avoidance behavior has the net e_ect of reducing the throughput of the connection (due to the smaller congestion window) and thus increasing overall energy consumption. We believe that the appropriate _x for this problem is for the TCP sender to retransmit the sending packet but not adjust its congestion window. We made this modification to TCP-ECN-ELFN in our implementation.

iii. *Network Congestion*

A problem with our approach above is that if the triple duplicates (or timeout) were generated as a result of packet drops due to congestion, then the solution of simply retransmitting the packet without reducing the congestion window will have negative consequences (this is the reason why TCP reduces its congestion window). In our design, we rely on explicit congestion notification to signal imminent congestion along a route2. Here, a node whose buffer occupancy. crosses some threshold, sets a bit (the CE bit) in all data packets it sees. Receivers reect this ag back in the ACKs they generate by setting the ECN-ECHO bit. Upon receiving an ACK with the ECN-ECHO bit set, TCP senders enter a recovery phase in which they reduce the congestion win down by a half. The sender sets a CWR (Congestion Window Reduced) bit in new data packets. If the receiver sees an other CE bit set in a future packet and sees that the sender had sent a CWR bit, this indicates that there is still congestion in the network. The receiver again sets the ECN-ECHO bit in new ACKs thus forcing the sender to enter another recovery phase. This can go on until the sender's window has shrunk to one or two segments.

References Références Referencias

1. Ashish Ahuja, Sulabh Agarwal, Jatinder Pal Singh, and Rajeev Shorey. Performance of tcp over different routing protocols in mobile ad-hoc networks. In IEEE Vehicular Technology Conference (VTC 2000), Tokyo, Japan, May 2000.
2. Thomas D. Dyer and Rajendra V. Boppana. A comparison of tcp performance over three routing protocols for mobile ad hoc networks. In ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC), October 2001.
3. Sorav Bansal et.al. Energy eÆciency and through put for tcp traÆc in multi-hop wireless networks. In Proceedings INFOCOM 2002, New York, NY, 2002.
4. K. Fall and S. Floyd. Simulation-based comparison of tahoe, reno, and sack tcp. ACM Computer Communications Review, 26(3):5 { 21, July 1996.
5. S. Floyd. TCP and explicit congestion noti_cation. ACM Computer Communication Review, 24(5):10{23, 1994.
6. M. Gerla, K. Tang, and R. Bagrodia. Tcp performance in wireless multi-hop networks. In IEEE WMCSA'99, (New Orleans, LA), Feb. 1999.
7. Gavin Holland and Nitin H. Vaidya. Analysis of TCP performance over mobile ad hoc networks. In ACM Mobile Computing and Networking (MOBICOM'99), pages 219{230, 1999.
8. M. Srivastava P. Lettieri, C. Schurgers. Adaptive link layer strategies for energy eÆcient wireless networking. In Wireless Networks, volume 5, pages 339 { 355, 1999.
9. L. Rizzo. Issues in the implementation of selective acknowledgements for tcp, 1996.

10. L. Rizzo. Dummy net: a simple approach to the evaluation of network protocols. ACM Computer Communication Review, 27(1), January 1997.

This page is intentionally left blank