RESEARCH ARTICLE OPEN ACCESS

Optimizing TCP Congestion Control techniques for modern Wireless Network

K. Vasudha Rani *, Dr. G.A. Ramachandra **
** Research Scholar, RAYALASEEMA UNIVERSITY, Kurnool*
(Email: vasudharani.ru@gmail.com)
*** (Department of Computer Science & Technology, SK University, Ananthapuram*

ABSTRACT
Transmission Control Protocol (TCP) intended to convey consistent and solid end-to-end information exchange crosswise over untrustworthy systems works perfectly well in wired condition. Indeed, TCP bears the 90% of Internet movement, so execution of Internet is to a great extent in view of the execution of TCP. In any case, end-to-end throughput in TCP corrupts strikingly when worked in remote systems. In remote systems, because of high piece mistake rate and changing level of congestion, retransmission timeouts for bundles lost in transmission is unavoidable. TCP misinterprets these irregular parcel misfortunes, because of the unusual idea of remote condition, and the resulting bundle reordering as blockage and conjures clog control by triggering quick retransmission and quick recuperation, prompting under-usage of the system assets and influencing TCP execution fundamentally. This postulation audits existing methodologies, suitable elements two proposed frameworks for better dealing with in systems with irregular misfortune and deferral. Assessment of the proposed frameworks is led utilizing NS2 test system by contrasting against standard TCP variations and shifting number of bounces.

Keywords: TCP, NS2,Congestion Control

Date of Submission: 24-11-2017 Date of acceptance: 06-12-2017

I. INTRODUCTION

The Transmission Control Protocol (TCP) is the most mainstream transport Protocol on the Internet supporting the World Wide Web, email and records exchange, and is in this way a basic part of the Internet. TCP gives a dependable byte stream benefit from an application on one host to an application on another host over the Internet. A standout amongst the most critical components of TCP with respect to its execution attributes is congestion control. congestion control is about utilizing the system as proficiently as could reasonably be expected. Now a day, systems are regularly over provisioned, what's more, the hidden question has moved from 'how to take out blockage' to 'how to productively utilize all the accessible limit'. Utilizing the system productively implies noting both these inquiries in the meantime; this is the thing that great blockage control systems do [1]. There is numerous usage of TCP, each working somewhat diversely and even some with huge issues. There are quantities of variations of TCP that are presently sent, Such as Tahoe, Reno, New Reno, Sack, Vegas, Westwood, Fack and Veno. In this paper we will assess the execution of two protocols of TCP that is Reno, Vegas along with the proposed method.

II. TCP OVERVIEW

This paper will explore the execution correlations of these previously mentioned adaptations of TCP and discover which one is better in which cases. TCP is one a player in two understood protocol principles normally alluded to as TCP/IP. TCP sits on top of the IP layer and passes fragments onto the IP layer for further preparing. These portions are then passed onto the lower level layers and in the long run onto the system. TCP was authoritatively received as a standard in RFC793 [2] in 1981 and was intended to manage message stream control and mistake amendment, guaranteeing solid conveyance of a message from a source application to a goal application. IP was too formally embraced as a standard in RFC791 [3] in 1981 Various schemes [4, 5, 6, 7] have been proposed in wireless networks to improve

TCP throughput and to handle congestion indication in such a way that TCP throughput is retained high.

TCP is a bi-directional, dependable, end-to-end Protocol for controlling information transmission.TCP sources break messages from higher protocol layers into datagram's that are embodied in packets which are then transmitted over the organize. These packets are reassembled by the

TCP beneficiary into the first message and passed onto the larger amount Protocol layers. For each bundle sent on the system by a source an affirmation (ACK) is required to be transmitted over from the goal. This ACK (or lack thereof) is utilized by the source to figure out whether the recognized parcel was effectively gotten at the goal.

2.1. TCP Congestion Control

The basic methodology of TCP is to send packets into the arrange without a reservation and after that to respond to recognizable occasions that happen. TCP expect just FIFO lining in the system's switches, additionally works with reasonable lining. The Internet was experiencing blockage crumple—has would send their bundles into the Internet as quick as the publicized window would permit, blockage would happen at some switch (making bundles be dropped), and the hosts would time out and retransmit their bundles, bringing about even more Congestion. Extensively, the possibility of TCP Congestion control is for every source to decide how much limit is accessible in the system, with the goal that it knows what number of bundles it can securely have in travel. Once a given source has this numerous bundles in travel, it utilizes the entry of an ACK as a flag that one of its bundles has left the system and that it is subsequently sheltered to embed another bundle into the system without adding to the level of blockage. By utilizing ACKs to pace the transmission of packets, TCP is said to act naturally timing. Of course, deciding the accessible limit in any case is no simple undertaking. To aggravate matters, on the grounds that other associations travel every which way, the accessible data transmission changes after some time, implying that any given source must have the capacity to alter the quantity of bundles it has in travel. This segment depicts the calculations utilized by TCP to address these and other issues. Take note of that, in spite of the fact that we portray the TCP blockage control instruments each one in turn, along these lines giving the feeling that we are discussing three free systems, it is as it were when they are taken all in all that we have TCP blockage control. Additionally, while we will start here with the variation of TCP Congestion control regularly alluded to as standard TCP, we will see that there are entirely a couple of variations of TCP blockage control being used today, and analysts proceed to investigate new ways to deal with tending to this issue.

2.2. TCP VARIANTS

They are many variants of TCP protocol (BIC, TCP Compound CUBIC, H-TCP, TCP-HYBLA, New Reno, Scalable TCP, Vegas, Westwood, High-speed TCP, TCP Veno, TCP Low-Priority). Tahoe alludes to the TCP blockage control calculation which was proposed by Van Jacobson. TCP depends on a rule of "protection of bundles", i.e. on the off chance that the association is running at the accessible transmission capacity limit then a parcel is definitely not infused into the system unless a parcel is taken out too. TCP actualizes this rule by utilizing the affirmations to clock active bundles in light of the fact that an affirmation implies that a parcel was removed the wire by the beneficiary. It moreover keeps up a blockage window CWND to mirror the system limit [8].

2.2.1 TCP Reno

This Reno holds the fundamental guideline of Tahoe, for example, moderate begins and the coarse grain re-transmit clock. In any case it includes some knowledge over it so that lost bundles are recognized prior and the pipeline is not purged each time a parcel is lost. Reno requires that we get quick affirmation at whatever point a fragment is gotten. The rationale behind this is at whatever point we get a copy affirmation, then his copy affirmation could have been gotten if the following fragment in arrangement expected, has been postponed in the system furthermore, the portions came to there out of request or else that the bundle is lost. In the event that we get various copy affirmations then that implies that adequate time have passed and regardless of the possibility that the section had taken a more extended way, it ought to have gotten to the recipient at this point. There is a high likelihood that it was lost. So Reno recommends a calculation called 'Quick Re-Transmit'. At whatever point we get duplicate ACK's we take it as a sign that the section was lost, so we re-transmit the fragment without sitting tight for timeout. In this way we figure out how to re-transmit the portion with the pipe full. Another change that RENO makes is in that after a bundle misfortune, it doesn't decrease the Congestion window to 1. Since this purges the pipe. It goes into a calculation which we call 'Quick Re-Transmit' [9]. The fundamental calculation is exhibited as:

- Each time we get 3 copy ACK's we take that to imply that the portion was lost and we re-transmit the portion promptly and enter 'Quick Recovery'.
- Set ssthresh to a large portion of the present window measure and furthermore set CWND to a similar esteem.
- For every copy ACK get increment CWND by one. On the off chance that the expansion CWND is more prominent than the sum of information in the pipe then transmit another fragment else hold up. On the off chance that there are "w" fragments in the window and one is lost, we will get (w-1) copy ACK's. Since CWND is diminished to W/2, thusly a large

portion of a window of information is recognized before we can send another portion. When we retransmit a fragment, we would need to sit tight for no less than one RTT before we would get a new affirmation. At whatever point we get a new ACK we decrease the CWND to SSthresh. In the event that we had already gotten (w-1) copy ACK's then now we ought to have precisely w/2 sections in the pipe which is equivalent to what we set the CWND to beat the finish of quick recuperation. In this manner we don't vacant the pipe, we simply decrease the stream. We proceed with blockage shirking period of Tahoe after that.,

2.2.2 TCP Vegas

Vegas is a TCP execution which is a change of Reno. It expands on the way that proactive measure to experience Congestion is substantially more productive than receptive ones. It attempted to get around the issue of coarse grain timeouts by recommending a calculation which checks for timeouts at an exceptionally proficient plan. Additionally, it conquers the issue of requiring enough copy affirmations to distinguish a parcel misfortune, and it moreover recommends an adjusted moderate begin calculation which counteracts it from blocking the system. It doesn't depend exclusively on bundle misfortune as an indication of Congestion. It identifies blockage some time recently the bundle packet losses happen. Be that as it may regardless it holds the other component of Reno and Tahoe, and a bundle misfortune can at present be distinguished by the coarse grain timeout of alternate systems fall flat. The three noteworthy changes actuated by Vegas are:

2.2.2.1 New Re-Transmission Mechanism:

Vegas stretch out on the retransmission component of RENO. It monitors when every section was sent and it additionally ascertains a gauge of the RTT by monitoring to what extent it takes for the affirmation to get back.

2.2.2.2 Congestion avoidance:

TCP Vegas is not quite the same as all the other execution in its conduct amid Congestion evasion. It doesn't utilize the loss of portion to flag that there is Congestion. It decides Congestion by a reduction in sending rate when contrasted with the normal rate, as aftereffect of extensive lines developing in the switches. It utilizes a variety of Wang what's more, crow croft's Tri-S plot.

2.2.2.3 Modified Slow-start:

TCP Vegas varies from the other calculations amid its moderate begin stage. The explanation behind this adjustment is that when an association first begins it has no clue of the accessible transmission capacity and it is conceivable that amid exponential increment it over shoots the transmission capacity by a major sum and in this way instigates blockage. To this end Vegas increments exponentially just every other RTT, between that it figures the real sending through put to the normal and at the point when the distinction goes over a specific limit it exits moderate begin and enters the blockage shirking stage

III. PROPOSED APPROACH

The main notion of the proposed mechanism is to keep the congestion window as high as possible during congestion control. There are mainly two scenarios when congestion window is reduced. One is during a retransmission timeout (RTO) and the
other is when the TCP sender receives a threshold number (usually set to three) of duplicate ACKs.

Scenario 1: Pseudocode for Retransmission of packets

```
if (retransmission timeout Occurs)
{
ssthresh = (send_max -
send_unacked) / 2;
if (ssthresh <= 2.0 * send_mss)
{
ssthresh = 2.0 * snd_mss;
}
cwnd = ssthresh;
}
```

where *send_max* is sequence number of the latest packet sent, *send_unacked* is the sequence number of first unacknowledged segment and *send_mss* is the maximum segment size for outgoing segments.

Scenarios 2 : Pseudo code for Handling Duplicate ACK's

```
if (threshold duplicate acks
received)
{
ssthresh = (send_max -
send_unacked)* 3 / 4;
if (ssthresh <= 2.0*snd_mss)
{
ssthresh = 2.0*snd_mss;
}
cwnd = ssthresh;
}
```

where *send_max* is sequence number of the latest packet sent, *send_unacked* is the sequence number of first unacknowledged segment and *send_mss* is the maximum segment size for outgoing segments.

The proposed approach experimented into ways, one is single-hop scenario and other is multip-hop. In single hop scenario, two nodes are talking to each other over wireless medium directly with congestion introduced at the sender side. FTP traffic is sent from sender to receiver using both the standard TCP Reno and our proposed TCP as transport layer protocol and compared against each other. From the results we observed that the average throughput is increased when the congestion window is kept at values nearer to where it was before congestion occurred. By using the proposed mechanism, congestion window is retained at higher values and thereby higher TCP throughput is

achieved. We used a multi hop chain topology with four wireless nodes with congestion introduced at the sender side. Again, FTP is used as the application protocol with TCP Reno, Vegas and proposed TCP at the transport layer. All the nodes where using same TCP version during simulation. The proposed mechanism aids keeping the congestion window at higher values and thereby higher TCP throughput is achieved. From the results we observed that although the average throughput is increased it is considerably less than single hop scenario.

IV. SIMULATION RESULTS

Figure 1 and Figure 2 show the TCP throughput and congestion window comparison, respectively, between TCP Reno, TCP Vegas and TCP with proposed changes, in single hop scenario. TCP throughput obtained using our proposed change is considerably healthier and during our simulations we observed that on an average 20 – 25 % throughput increase is achieved. Figure 4 shows that when proposed TCP is used the congestion window is retained very close to the values before congestion started and simulation results show that on an average, congestion window size is around 30 % higher than that achieved while using TCP Reno.

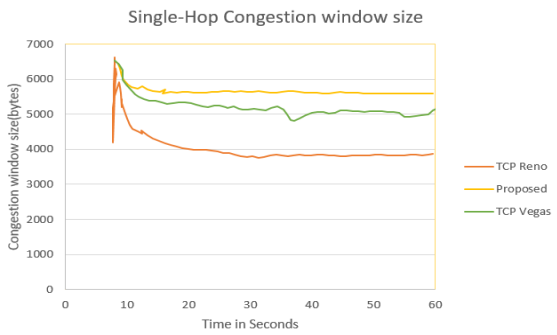

Fig.1. Comparision of Single-hop Congestion window size

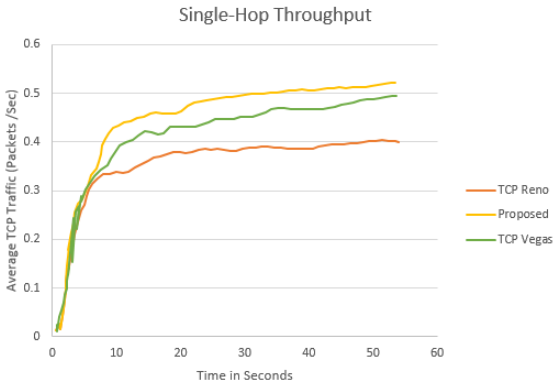

Fig.2. Comparision of Single-hop Throughput.

Figures 3 and 4 show the TCP throughput and congestion window comparison, respectively, between TCP Reno, TCP Vegas and TCP with proposed changes, in multi hop scenario. During our simulations, proposed TCP in multi hop scenario achieved on an average 35 – 40 % higher throughput than TCP Reno and congestion window is around 45– 50 % higher than observed with TCP Reno.

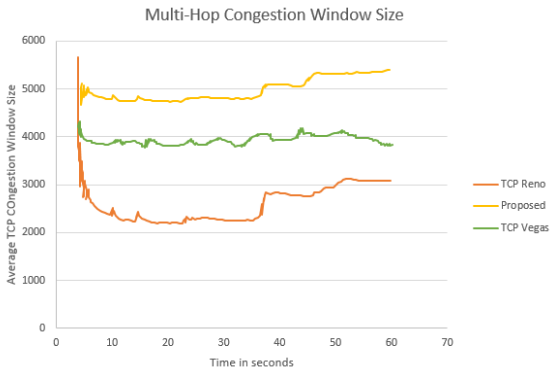

Fig.3. Comparision of Multi-hop Congestion window size

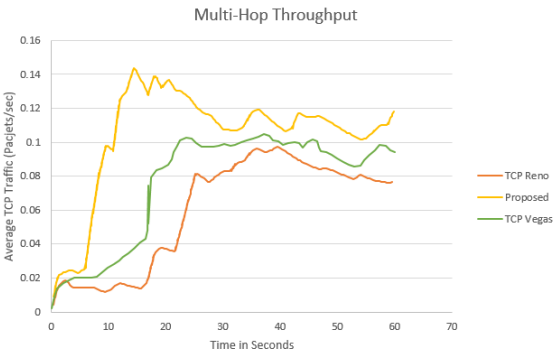

Fig.4. Comparision of Multi-hop Throughput

V. CONCLUSION

In this research paper, we have proposed sender side TCP alterations to enhance TCP execution performance in remote systems by implementing two scenarios. We have evaluated the TCP performance with proposed scenarios using

NS2. we compared proposed scenarios with TCP Reno & Vegas, which are good to handle duplicate packets ACK's.

The proposed approach dynamically calculates congestion window size during retransmission and acknowledging duplicate packets, handles high dropping packets in wireless networks. Proposed scenarios achieved high throughput other than TCP variants by maintaining congestion window size high during dropping of packets.

REFERENCES

[1]. M. Welzl, Network congestion control. Chi Chester, West Sussex, England: J. Wiley, 2005.
[2]. J. Postel, DOD standard transmission control protocol. Marina del Rey, Calif.: Information Sciences Institute, University of Southern California, 1980.
[3]. J. Postel, DoD Standard Internet Protocol. Ft. Belvoir: Defense Technical Information Center, 1980.
[4]. Elrakabawy, S.M., Klemm, A., Lindemann, C.: TCP with adaptive pacing for multihop wireless networks. In: Proc. 6th ACM Int. Symp. Mobile Ad Hoc Networking & Computing, pp. 288–299 (2005).
[5]. Long, W., Zhenkai, W.: Performance Analysis of Improved TCP over Wireless Networks. In: Proc. 2nd Int. Conference on Computer Modeling and Simulation, pp. 239–242 (2010).
[6]. Prasanthi, S., Chung, S.: An Efficient Algorithm for the Performance of TCP over Multihop Wireless Mesh Networks. In: Proc. 7th Int. Conference on Information Technology, pp. 816–821 (2010).
[7]. Rai, I.A., Hellen, T.: On improving the TCP performance in asymmetric wireless mesh networks. In: Proc. Int. Conference on Communications, Computing and Control Applications, pp. 1–6 (2011).
[8]. Jacobson, "Congestion avoidance and control", ACM SIGCOMM Computer Communication Review, vol. 18, no. 4, pp. 314-329, 1988.
[9]. M. Welzl, Network congestion control. Chichester, West Sussex, England: J. Wiley, 2005.

International Journal of Engineering Research and Applications (IJERA) is **UGC approved** Journal with Sl. No. 4525, Journal no. 47088. Indexed in Cross Ref, Index Copernicus (ICV 80.82), NASA, Ads, Researcher Id Thomson Reuters, DOAJ.

K. Vasudha Rani "Optimizing TCP Congestion Control techniques for modern Wireless Network." International Journal of Engineering Research and Applications (IJERA) , vol. 7, no. 12, 2017, pp. 58-62.