International Journal of Advance Research in Science and Engineering
Volume No.07, Special Issue No.04, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

Performance comparison of various TCP variants over Cognitive Radio Adhoc Networks and Mobile Adhoc Networks

Safia Gul[1], Bilal Ahmad Malik [2]

Department of Electronics & Instrumentation technology, University of Kashmir
Department of Electrical & Electronics Engineering , University of Kashmir

ABSTRACT

Cognitive radios are foreseen as the radios capable of using the available licensed band in a very resourceful manner by making efficient utilization of different licensed portions of the spectrum, known as spectrum holes (underutilised spectrum bands), in a dynamic manner. It is these spectrum holes responsible for the evolution of next generation networks or cognitive radio networks. At present, transport layer protocol which mainly deals with flow control and congestion control is an important but not deeply inspected area of research over cognitive radio adhoc networks (CRAHNS). Flow control and congestion control are very important for the realization of dynamic spectrum networks. In this paper performance evaluation of different congestion control TCP variants like TCP Reno, TCP NewReno, and TCP Vegas over mobile adhoc networks (MANETS) and Cognitive radio adhoc networks (CRAHNS) with the help of a cognitive radio cognitive network simulator (CRCN) has been done. The results demonstrate that TCP New Reno improves the average throughput and packet delivery ratio in case of CRAHNS and TCP Vegas offers average throughput and the best packet delivery ratio in case of MANETS.

Keywords: Cognitive Radios, Cognitive Radio Adhoc Networks, Mobile Adhoc Networks, Spectrum Holes, And Transport Control Variants.

I INTRODUCTION

Wireless networks are structured with fixed spectrum assignment strategy, regulated by governmental agencies, so that only licensed users can use them efficiently but according to federal communication commission temporal and geographical variation in the utilization of assigned spectrum ranges from 15% to 85% [1]. This limited spectrum usage results in spectrum holes; it is these spectrum holes responsible for the evolution of next generation networks or cognitive radio networks. Cognitive radio networks use these spectrum holes for improving the spectrum utilization and network capacity. The cognitive radio is a radio that can change its transmitter parameters based on the interaction with the environment, parameters such as : 1) spectrum sensing – determines which portion of spectrum is available and detect the presence of licensed user, 2) spectrum management- select the best available channel, 3) spectrum sharing- coordinate access to this channel with other users, 4) spectrum mobility- vacate the channel when licensed user is detected. [2]

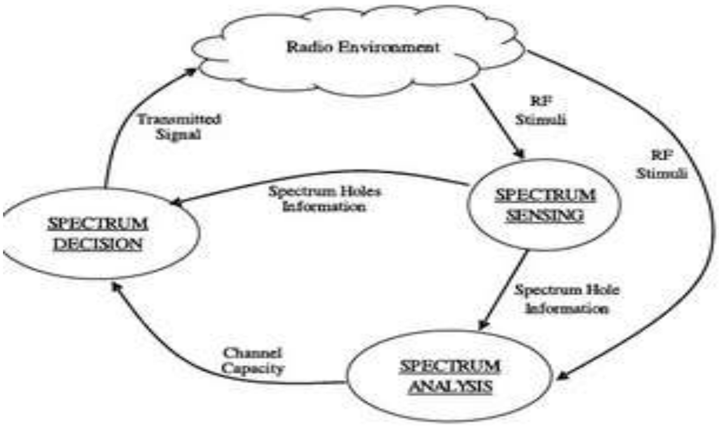

Fig 1. Cognitive cycle [3]

However dynamic use of spectrum causes degradation on the performance of conventional communication protocols. A lot of work has been done on unique challenges in terms of spectrum sensing, spectrum management, spectrum mobility and spectrum sharing however upper layer issues of routing, flow control and congestion control are also important for the realization of dynamic spectrum networks.

In this work congestion control – the transport layer issue has been focussed. Since the performance degradation of TCP arises because of wireless link errors and access delays. However next generation networks or cognitive radio networks impose unique challenges for transport protocols as its performance depends on packet loss probability and the round trip time (RTT). Packet loss probability not only depends on the access technology, but also on the frequency in use, interference level and the available bandwidth.[4] Therefore, based on the frequency of operation, RTT and packet loss probability observed by TCP protocol will vary. The operation frequency of cognitive radio may vary from time to time due to spectrum handoff, this results in finite amount of delay before new frequency can be operational, and this is referred to as spectrum hand off latency. [5] The spectrum handoff latency can increase the RTT, which leads to retransmission timeout (RTO). Conventional transport protocols can perceive this RTO as packet loss and invoke its congestion avoidance mechanism resulting in reduced throughput. Congestion control mechanism will start dealing with the segment loss. [4]

The rest of the paper is organised as follows. First in section 2, congestion control mechanism in TCP is discussed. Next problem identification is described in section 3. After that experimental setup and the measured parameters is carried out in section 4. The results and discussions are presented in section 5 and finally conclusions in section 6 are discussed.

II CONGESTION CONTROL MECHANISM

The predominant example of end-to-end congestion control [6] in use today, implemented by TCP. The essential strategy of TCP is to send packets into the network without a reservation and then to react to observable events that occur. TCP works on FIFO queuing in the network routers, but also works with fair queuing.

2.1 Additive increase / Multiplicative decrease (AIMD)

TCP maintains a new state variable for each connection, called congestion window [3], which is used by the source to limit how much data is allowed to have in transit at a given time. The congestion window is congestion controls counterpart to flow controls advertised window. TCP is modified such that the maximum number of bytes of unacknowledged data allowed is now the minimum of the congestion window and the advertised window.

MaxWindow = MIN (CongestionWindow, AdvertisedWindow)

EffectiveWindow = MaxWindow − (LastByteSent − LastByteAcked).

That is, Max Window replaces Advertised Window in the calculation of Effective Window.

Thus, a TCP source is allowed to send no faster than the slowest component—the network or the destination host—can accommodate. The problem, of course, is how TCP comes to learn an appropriate value for Congestion Window. Unlike the Advertised Window, sent by receiving side of the connection, there is no one to set a suitable Congestion Window to the sending side of TCP.

TCP does not wait for an entire window worth of ACKs to add one packet worth to the congestion window, but instead increments Congestion Window by a little for each ACK that arrives. Specifically, the congestion window is incremented as follows each time an ACK arrives:

Increment = MSS × (MSS/Congestion Window)

Congestion Window + = Increment

That is, rather than incrementing Congestion Window by an entire MSS (maximum segment size) bytes each RTT, we increment it by a fraction of MSS every time an ACK is received. The important concept to understand about AIMD is that the source is willing to reduce its congestion window at a much faster rate than it is willing to increase its congestion window. [6]

2.2 Fast Retransmit and Fast Recovery

The mechanisms described so far were part of the original proposal to add congestion control to TCP. It was soon discovered, however, that the coarse-grained implementation of TCP timeouts led to long periods of time during which the connection went dead while waiting for a timer to expire. Because of this, a new mechanism called fast re-transmit was added to TCP. Fast retransmit is a heuristic that sometimes triggers the retransmission of a dropped packet sooner than the regular timeout mechanism. [6]

III PROBLEM IDENTIFICATION

TCP/IP protocol has been designed for wired networks which provides end to end reliable communication between nodes and assures ordered delivery of packets. But in case of adhoc networks packet losses are due to congestion in the network and due to frequent link failures. So when we adapt TCP to adhoc networks it misinterprets the packet losses due to link failure as packet losses due to congestion and in the instance of a timeout, backing-off its retransmission timeout (RTO). This results in unnecessary reduction of transmission rate because of which throughput of the whole network degrades. Therefore, route changes due to host mobility can have detrimental impact on TCP performance. As a result, TCP is shown to perform poorly over wireless links and several variants of TCP have been proposed for mobile adhoc networks (MANETS).

Transport protocols for MANETS do not consider the cases that may arise in cognitive radio adhoc networks (CRAHNS). In MANETS packets may incur a longer round trip time (RTT) owing to network congestion or due to temporary route outage. In CRAHNS, similar effects on packet RTT may be caused if an intermediate node on the route is engaged in spectrum-sensing and hence, unable to forward packets. Also, sudden appearance of a primary user may force cognitive radio nodes in its vicinity to limit their transmission, leading to an increase in the RTT. So advanced TCP variants in case of CRAHNS are needed to analyse the reliable end-to-end communication with less throughput degradation.

IV EXPERIMENTAL SETUP

In our setup we make use of extended version of NS-2 simulator known as cognitive radio cognitive network simulator (CRCN). We simulated each variant of TCP over different TCP-FTP connections in MANETS and CRAHNS environment and measures how the TCP-variants TCP Reno, TCP NewReno, and TCP Vegas affects the average throughput and packet delivery ratio.

V RESULTS AND DISCUSSIONS

Initially Average throughput for a 10 node mobile and cognitive environments in a 1000m*1000m rectangular region is analysed

Results of TCP variants over 5 TCP-FTP connections as shown in below table:

Simulation time is 200s	**Average throughput(Kbps) for MANETS**	**Average throughput(Kbps) for CRAHNS**
TCP	**374.65**	**233.05**
TCP RENO	**365.35**	**211.08**
TCP NEWRENO	**468.13**	**244.15**
TCP VEGAS	**469.41**	**185.8**

Table 1: For 5 TCP-FTP connections

From the Table 1 results it is evident that in case of MANETS average throughput of TCP Vegas is better than other TCP variants like TCP Reno and TCP New Reno. However, in case of CRAHNS average throughput of TCP Vegas is surprisingly very less as compared to other variants. Variation in case of MANETS is less but variation in case of CRAHNS is very large. New Reno has 24% more average throughput in comparison to TCP Vegas in CRAHNS.

Now result of average throughput for 3 TCP-FTP connections

Simulation time is 200s	**Average throughput(Kbps) for MANETS**	**Average throughput(Kbps) for CRAHNS**
TCP	**399.59**	**171.86**
TCP RENO	**397.1**	**169.9**
TCP NEWRENO	**420.64**	**215.53**
TCP VEGAS	**436.71**	**109.46**

Table 2: For 3 TCP-FTP connections

From the Table 2 we observed that NewReno has 49.2% more average throughput then TCP Vegas in CRAHNS.

Now for 2 TCP-FTP connections average throughput is shown in table 3

Simulation time is 200s	**Average throughput(Kbps) for MANETS**	**Average throughput(Kbps) for CRAHNS**
TCP	**353.09**	**178.07**
TCP RENO	**367.92**	**183.56**
TCP NEWRENO	**388.05**	**218.69**
TCP VEGAS	**427.89**	**92.72**

Table 3: For 2 TCP-FTP connections

Thus from Table 3 for 2 TCP-FTP connections we observed that TCP New Reno has 57.6% more average throughput then TCP Vegas in CRAHNS.

This is because additional RTO events are triggered by sensing induced delay, by the handoff delay in CRAHNS and we have reached to the conclusion that TCP New Reno experiences high throughput then TCP Vegas in CRAHNS and the performance difference increase when more connections are added in the simulation scenario.

PACKET DELIVERY RATIO**:** It is the ratio of packets which are received by the destination node, over the number of packets sent by the source, it provides an indication of delivery capabilities for end-to-end protocols.

Now we analyse packet delivery ratio of TCP variants over different TCP-FTP connections in a 10 node mobile and cognitive environments in a 1000m*1000m rectangular region

NETWORKS	TCP	TCP RENO	TCP NEWRENO	TCP VEGAS
MANETS	99.4	99.51	99.45	99.91
CRAHNS	75.08	75.58	84.89	81.8

Table 4: For 5 TCP-FTP connections

From the above Table 4 it is evident that TCP New reno has got higher packet delivery ratio then TCP Vegas, So TCP NewReno delivers packets more efficiently than TCP Vegas unlike MANETS where the two are almost equal.

NETWORKS	TCP	TCP RENO	TCP NEWRENO	TCP VEGAS
MANETS	99.65	99.66	99.74	99.95
CRAHNS	75.53	75.61	85.28	83.29

Table 5 : For 3 TCP-FTP connections

From Table 5 we can say that TCP New Reno has 1.99% more packet delivery ratio then TCP Vegas in CRAHNS and in case of MANETS there is very less difference between these two TCP variants.

NETWORKS	TCP	TCP RENO	TCP NEWRENO	TCP VEGAS
MANETS	99.67	99.74	99.8	99.94
CRAHNS	75.884	76	85.06	83.37

Table 6 : For 2 TCP-FTP connections

From Table 6 we can conclude that TCP New Reno has 1.69% more packet deliver ratio then TCP Vegas , so NewReno delivers packets more efficiently then TCP Vegas. We can also say that with increased flow performance difference increases.

VI CONCLUSION

In this paper TCP Reno and TCP Vegas in MANETS and CRAHNS is demonstrated, average throughput and packet delivery ratio has been used as metrics for comparison. The results lead to the following conclusions:

1. TCP Vegas is better than other TCP variants like TCP Renoand TCP NewReno for sending data and information in case of MANETS. It provides best Average throughput and packet delivery ratio than TCP New Reno. This is due to the fine tuning of congestion window size by taking into consideration the RTT of a packet.
2. TCP NewReno is better than other TCP variants TCP Vegas and TCP Reno in terms of average throughput and packet delivery ratio in case of CRAHNS.

Unlike MANETS, TCP vegas provides the lowest performance in CRAHNS. This might be because TCP Vegas tries to estimate the available bandwidth to avoid congestion rather than to reach to it. It uses the measured RTT to accurately calculate the number of data packets that a source can send. Due to the primary user activity and sensing time, this available bandwidth estimation might not be correct one leading to low throughput.

Thus this study reveals that TCP New Reno out performs TCP Vegas in case of CRAHNS. So advanced TCP variants in case of CRAHNS are needed to analyse to enhance throughput degradation for reliable end to end communication.

REFERENCES

[1] Xiaoqin Chen, Haley M. Jones, A.D.S Jayalath ―Congestion Aware Routing Protocol for Mobile Adhoc Networks, Department of Information Engineering, CECS, The Australian National University, Canberra.

[2] L.N.T Perera, H.M.V.R.Herath-Review of spectrum sensing in cognitive radio, University of Peradeniya, Peradeniya, Sri Lanka , 6th International conference on industrial information system ,2011

[3] Technologies and issues in cognitive radio networks, computer networks @ SEECS, NUST

[4] *Advanced wireless networks: Technology and Business Model* by Savo G.Glisic, publications by John Wiley and sons

[5] Ian F. Akyildiz ,Won-Yeo Lee, Kaushik R.Chowdhury, CRAHNs: Cognitive radio ad hoc networks, Elsevier Ad Hoc Networks 7 (2009) 810–836

[6] Yuvaraju B. N, Dr.Niranjan N chiplunkar, Scenario Based Performance Analysis of Variants of TCP using NS2-Simulator, International journal of computer applications, *ISSN 097-4860,october 2010*

[7] Laxmi Subedi, Mohamadreza Najiminaini, and Ljiljana Trajković ―Performance Evaluation of TCP Tahoe, Reno, Reno with SACK, and NewReno using OPNET Modeler.

[8] Raju kumar, Ricardo crepaldi,Hosa Rowaihy, Mitigating Performance Degradation in congested sensor networks, IEEE Transaction on mobile computing, *Volume: 7, Issue: 6, 682 - 697* June 2008

[9] Kaushik R. Chowdhury, Marco Di Felice, Ian F. Akyildiz, TP-CRAHN: A Transport Protocol for Cognitive Radio Ad-hoc Networks, *IEEE INFOCOM 2009*.

[10] S.Haykin, cognitive radio:brain empowered wireless communications, IEEE journal on selected areas in communication, Volume: 23, Issue: 2, Feb. 2005