IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 2, Ver. V (Mar – Apr. 2015), PP 129-138
www.iosrjournals.org

Recital Study of Various Congestion Control Protocols in wireless network

Mehta Ishani[1], UditNarayan Kar[2], Dr. Atul Gonsai[3]
[1](PG Student, Computer Science, GardiVidyapith,Rajkot,Gujarat, India)
[2](Research Scholar,Computer Science, Saurastra University,Rajkot, Gujarat, India)
[3](Associate Professor, Computer Science, Saurastra University,Rajkot, Gujarat, India)

Abstract: *Congestion control in packet switching networks became a high priority in network design and research due to ever-growing network bandwidth and intensive network applications. Dozens of various congestion control strategies have been proposed, and more are forthcoming. Based on control theory concepts congestion control can be viewed as control policy to achieve prescribed goals (e.g., round-trip delay or throughput) in a distributed network environment. This paper discusses the advantages, disadvantages and the applications of various congestion control protocols for wireless networks. It explores the motivation behind the design of congestion control protocols which is suitable to large scale wireless network and abolish drawbacks of the most widely accepted two algorithms of TCP and RCP.*

Keywords: *TCP- Transmission Control Protocol, RCP- Rate Control Protocol*

I. Introduction

The use and demand of wireless network has increased significantly in past years. Wireless environment is more flexible, easy to deploy and scalable than traditional wired setup. Besides that in wireless network air is used as access medium it is also more sensitive to inferences and to congestion. The developed congestion control protocols like Transmission Control Protocol (TCP) and Rate Control Protocols (RCP) do not take into account the problems and particularities of wireless network. TCP is the most widely accepted and used congestion protocol in internet. It used window based Additive Increase Multiplicative Decrease (AIMD) strategy for congestion control [1]. However, TCP proved to be unsuitable for dynamic environment of wireless network where nodes are mobile and frequently participate and departure from the flow. RCP is built on rate based congestion control strategy. It relay on network interaction modules such as routers for congestion control. Also it is proved that RCP performs better than TCP [6]. The above described protocols are also proved heavy for implementation into light devices with the reference of OS, Processing, and Memory Requirement and Power consumption. They perform complex computation which is consuming power, requiring more memory for storing per packet states, heavy for Hot Spot Tethering and might get hang with heavy processing [3]. Some of these performance problems led to the development of new congestion control protocols. In this paper we present performance comparison of TCP, RCP+ and our proposed model Enhanced RCP in wireless network. The paper organizes as follows. The next section II, III and IV describes the mechanism of different congestion control protocols, TCP, RCP and RCP+.Section V briefly introduced our proposed approach and section VI presents implementation and evaluation results. Finally section VII provides conclusion.

II. TCP

The TCP achieves two important functions: (1) Reliable and ordered delivery and (2) Congestion control. Congestion can be detected by Packet loss or Packet delay. Solution of this problem is limiting sender's transmission rate. Now question arise, at what rate should the data be sent for the current network path? [5]. Congestion control mechanism given in [1] is shown in figure 1.

Congestion control mechanism given in [1] is shown in Fig 1. Each TCP connection initiate with a pre-configured small congestion window no larger than 4 Maximum Segment Size (MSS).The goal of Slow-Start is to keep a new sender from overflowing network buffers, while at the same time increasing the congestion window fast enough and avoiding performance loss while the connection is operating with a small window.Slow-Start increases the congestion window by one MSS for each new acknowledgment received, which results in the window doubling after each window's worth of data is acknowledged. With this exponential increase, RTT log2W (where RTT stands for round-trip time) time is required to reach a window of size W.

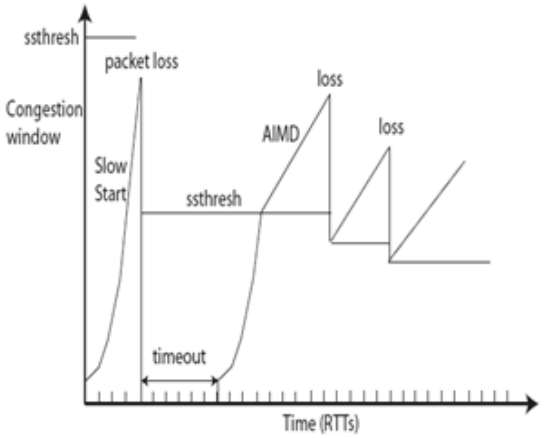


Figure 1: TCP's congestion control mechanism [1]

A connection enters Slow-Start on newly starting or on experiencing a packet retransmission timeout, and exits Slow-Start when it detects a packet loss or when the congestion window has reached a dynamically computed threshold, ssthresh. More specifically, ssthresh is set to half of the current congestion window when packet loss was detected. TCP exits Slow-Start to enter the Congestion Avoidance phase, where it continues to probe for available bandwidth, but more cautiously than in Slow-Start. During periods when no packet losses are observed, TCP performs an Additive Increase of the window size, by 1 MSS each time a full window is acknowledged (i.e., increases the congestion window as:

$$\text{cwnd} = \text{cwnd} + 1 \tag{1}$$

on receiving each acknowledgment packet). And when Congestion occurs it decreases the window size by half as given in equation below

$$\text{cwnd} = \text{cwnd} * \frac{1}{2} \tag{2}$$

III. RCP

Researchers of paper [6, 7] have proposed Rate Control Protocol (RCP). In RCP, a router assigns a single rate, R (t), through which all flows pass. Here, the basic idea is: If there is spare capacity available, then share it equally among all flows. Furthermore if there is a queue building up, then the link capacity is insufficient and the flow rate is decreased evenly. they have tried to minimize Flow completion time (FCT). FCT is time from when the first packet of a flow is sent (in TCP, this is the SYN packet) until the last packet is received. To minimize FCT for each router a well-known method is to use processor-sharing (PS) i.e. a router divides outgoing link bandwidth equally among ongoing flows. Also they have proposed an equation for RCP:

$$R(t) = R(t - d0) + \frac{\left[\alpha(C - y(t)) - \beta\frac{q(t)}{d0}\right]}{N(t)} \tag{3}$$

where,
d0 = a moving average of the RTT measured across all flows,
R(t−d0) = last updated rate,
C = link capacity,
y(t) = measured input traffic rate during the last update interval (d0 in this case),
q(t) = instantaneous queue size,
N(t) = router's estimate of the number of ongoing flows (i.e., number of flows actively sending traffic) at time t
α, β = parameters chosen for stability and performance.

Simulation results show that RCP performs better than TCP in terms of FCT and link utilization in wired network [9]. It is also taken account that RCP possess routing overhead on network also due to having complex computation and involving routers. Researchers of paper [8] have introduced two limitations of RCP: (1) RCP will need to operate alongside existing non-RCP traffic, such as TCP and UDP, without adversely affecting or being affected by the other traffic; and (2) RCP will need to operate in a network where some routers are not RCP-enabled.

IV. RCP+

Authors [2] described a simple congestion control algorithm called RCP+ which reduces flow completion time for diverse flow types to large extend for broad range of traffic conditions and network situations. In RCP+, unlike XCP, RCP they are not following feedback mechanism but they are adapting the ancient congestion window based congestion control mechanism. The reason behind sticking to the congestion window for congestion control is that, while each second new flow are entering and moving out of network, it is tough to obtain exact number of flows at particular RTT. This inspired us to stay with congestion window based mechanism instead of feedback based mechanism. We set the _cwnd' value completely based on the rate computation, but that doesn't impose any overhead over the router. In addition, we are setting the value of _maxcwnd' to obtain the better start for our optimum data rate calculation at the initial stage.

RCP+ wins upon existing congestion control algorithm with three characteristics, 1) RCP+ is more flexible to be implemented on wireless networks in comparison of RCP. 2) It allows multiple variants of TCP to co-exist in the same networks with RCP+ and work with each other smoothly. 3) And it performs well in Large scale wireless networking scenario too.The equation of RCP+ can be given as:

$$N(t) * R(t) = (\alpha*C - \alpha*y(t) - (\beta*q(t)/d)) \quad (4)$$

Where,
d = moving average of RTT per interval,
R(t) = last updated rate,
y(t) = existing traffic observed in network,
q(t) = the instantaneous queue size,
C = link capacity and
N(t) = number of flows.
α (alpha) is a stability parameter and β (beta) is a performance parameter added to the equation to make the rate stable and not aggressive.

Thus, the equation gives us the desired aggregate rate change in presence of traffic in the next interval. Here, Rate kept same for each flow. "cwnd" congestion window value is having proportionate relationship with the rate and so, "cwnd" value is set on the bases of the Rate computed. RCP+ is very acute to packet loss. This equation was obtained by manipulating Rate equation of RCP [5]

V. Proposed Approach

Our proposed approach is based Improved AIMD and RCP+ algorithm. In our proposed approach we use congestion window mechanism of Improved AIMD algorithm to use the spare capacity of congestion window after occurrence of congestion event. The limitation of AIMD is that the algorithm does not include the different arrival time of flows.[1]. So we have used modified equation of RCP+ algorithm.

RCP+ algorithm is implemented based on the theory of RCP. RCP+ is having the added advantage of coexistence with other wired and wireless TCP, XCP, RCP and DCCP protocols. RCP+ is flexible like TCP and so is expected to have wide implementation over current demands of Internet. [2]. In the theory of RCP there is a concept of queue. RCP was originally implemented in wired network. In wireless network the essential point is traffic is in bursty nature. Hence it is difficult to use queue concept while going to practical simulation. So we modified the equation of rate change.

Here we come up with a new proposal of congestion control scheme enhanced RCP. Here we use Improved AIMD mechanism as well as modified rate change equation of RCP+. Our aim is to gain benefits of both schemes by eliminating each other's demerits.

Initially we defined congestion window size 4 MSS and start data transfer. If acknowledgment is received means no congestion and increase congestion windows size by +1 and transfer data

If acknowledgment is not received then congestion would likely to be occurred then decrease congestion window size by equation:

$$Cwnd = (cwnd *1/2) + K \quad (5)$$

Where, k is the increment in congestion window size (w) in t cycles or epoch

Calculate the rate change by equation:

$$Rate = (\alpha*C - \alpha*y(t))/N(t) \quad (6)$$

Again check if rate is greater then congestion window size then increase congestion window size otherwise transfer data.

Algorithm

1) Initially set congestion window size 4 MSS
2) Data transfer
3) If acknowledgement is received increase congestion window size by cwnd = cwnd +1 and repeat step 2
4) Otherwise decrease congestion window size by equation Cwnd = (cwnd *1/2) + K
5) Calculate the rate change by equation Rate = (α*C - α*y(t))/N(t)
6) If rate >cwnd then go to step 2
7) Otherwise transfer data

Below figure shows the flow chart for the same.

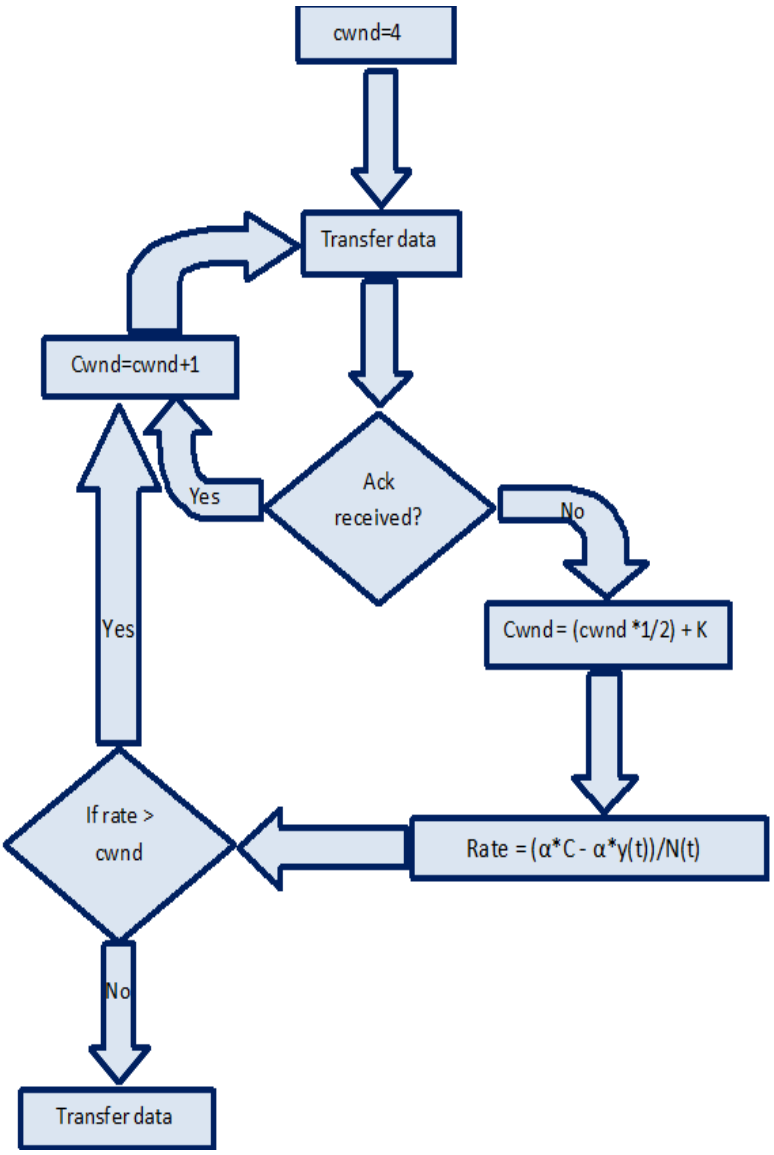


Figure2 Flow chart for proposed algorithm

VI. Implementation

We have implemented proposed model in NS-2 simulator and version is 2.35 [10].Following are the steps for modification we have done in C++ file of TCP to implement RCP++

Step 1: Go to the location where tcp.cc is located. There, under the class definition of Tcp agent, bind two variables for its usability under the procedure of bind.

```
bind("bw_",&bw_);
bind("flow_",&flow_);
```

Step 2: Modify TcpAgent::window procedure

```
if (frto_ == 2)
{
return (force_wnd(2) <wnd_ ?
force_wnd(2) : (int)wnd_+4);
}
```

Step 3: Modify TcpAgent::slowdown procedure

```
double win, halfwin, decreasewin, k;
k = (windowd() / 2) -1;
if (cwnd_ <ssthresh_)
        slowstart = 1;
if (precision_reduce_)
        {
        halfwin = (windowd() / 2)+k;
        }
        else
        {
        int temp;
        temp = (int)((window() / 2)+k);
        halfwin = (double) temp;
        }
```

Step 4: Modify TcpAgent::processQuickStart procedure

Define following variables

```
intapp_rate, bw_=400000, flow_=4;
float alpha = 0.1,  yt = 0.005;
```

Step 5: Add following lines to the procedure of processQuickStart

```
qs_requested_ = 0;
qs_approved_ = 0;

if (qsh->flag() == QS_RESPONSE &&qsh->ttl() == ttl_diff_ &&qsh->rate() > 0) {

app_rate = (int)((alpha*TcpAgent::bw_ - alpha*yt) )/TcpAgent::flow);
```

Step 6: Give the computed rate to qs_cwnd defined under the procedure of processQuickStart

```
if (app_rate>initial_window())
{
qs_cwnd_ = app_rate;
qs_approved_ = 1;
}
else
{ // Quick Start rejected
}
```

Step 7: Once the changes in tcp.cc are made, save and exit the editor. Next step is to open the terminal and under super user, type following commands one by one.

```
./configure
Make clean
Make
Make Install
```

Once you have done this your TCP agent will become RCP enabled now run the tcl scripts and obtained the results.Following table 1.1 present the configuration details of the simulation. We have created a wireless network and

Table 1 Configuration Table

Layer	Parameter	Values
Application	FTP	FTP over TCP agent and RCP+
Configuration	No of nodes	10 to 50 incremental

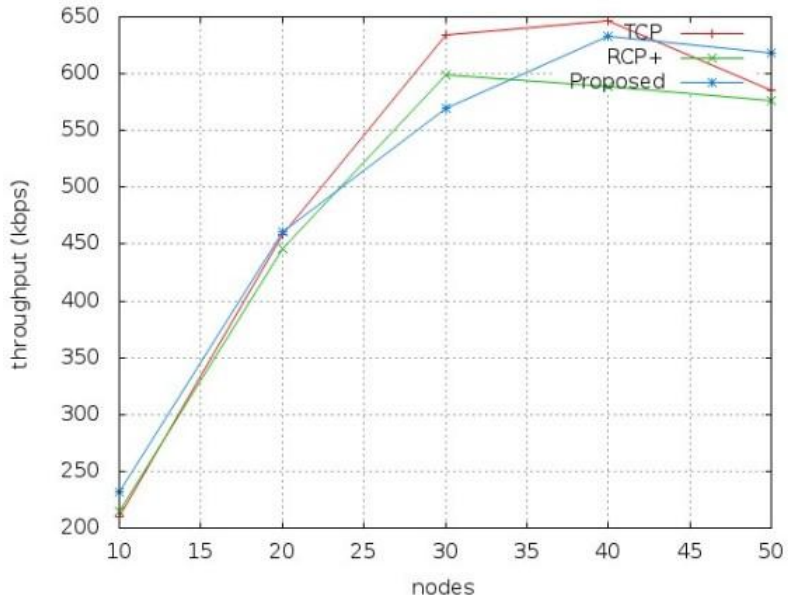
Mobility	Maximum Speed	10 Mbps
	Pause time	2 s
	Simulation time	500 s
Traffic	Type	TCP/CBR
	Rate	4.0 Mbps
Routing	Protocol	AODV
MAC	Mac	802_11
PHY	Propagation model	Two ray ground
	Antenna	Omni
System	OS	Ubuntu 12.10
	Processor	Intel(R) Core(TM) i5

The values of throughput, packet delivery ratio and routing load for TCP, RCP+ and proposed model are given in tables bellow. From these values graphs are plotted and performance of each protocol can be evaluated.

Throughput: In general terms, throughput is the rate of production or the rate at which something can be processed. When used in the context of communication networks, throughput or network throughput is the rate of successful message delivery over a communication channel. The data these messages belong to may be delivered over a physical or logical link, or it can pass through a certain network node. Throughput is usually measured in bits per second (bit/s or bps), and sometimes in data packets per second or data packets per time slot.It is clearly shown that after 20 nodes the packer delivery ratio stated decreasing so that is the bottleneck area and after that congestion occurs.

Table 2 Throughput Statistics

Throughput			
Nodes	TCP	RCP+	Proposed
10	210.17	215.22	232.22
20	458.35	446.3	460.95
30	633.91	599.13	569.76
40	646.97	588.9	633.41
50	585.45	576.54	618.02

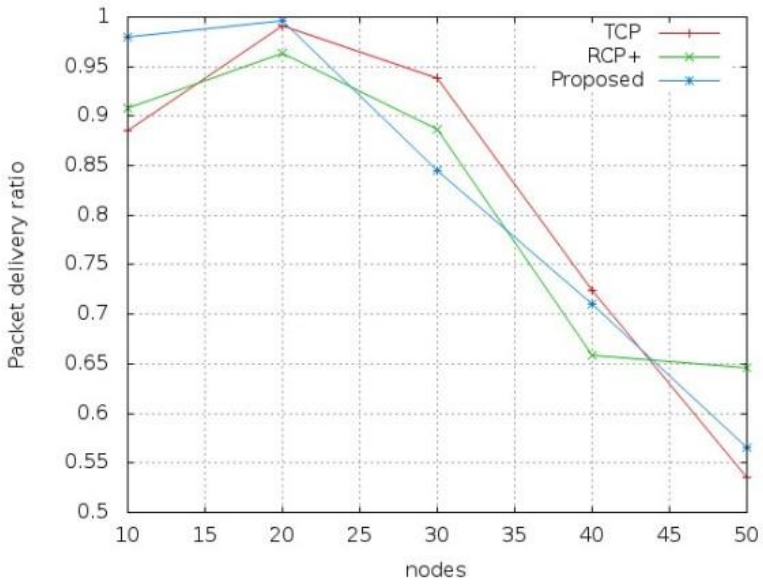


Graph 1 Throughput graph

Packet Delivery Ratio: The ratio of the number of delivered data packet to the destination. ∑ Number of packet receive / ∑ Number of packet send. This illustrates the level of delivered data to the destination. The greater value of packet delivery ratio means the better performance of the protocol. It is observed that proposed model have greater values of Packet Delivery Ratio than existing protocols.

Table 3 Packet Delivery Ratio Statistics

Packet Delivery Ratio			
Nodes	TCP	RCP+	Proposed
10	0.8858	0.908	0.9803
20	0.991	0.9636	0.9961
30	0.9386	0.8871	0.8453
40	0.7242	0.6587	0.7106
50	0.5357	0.6457	0.5654

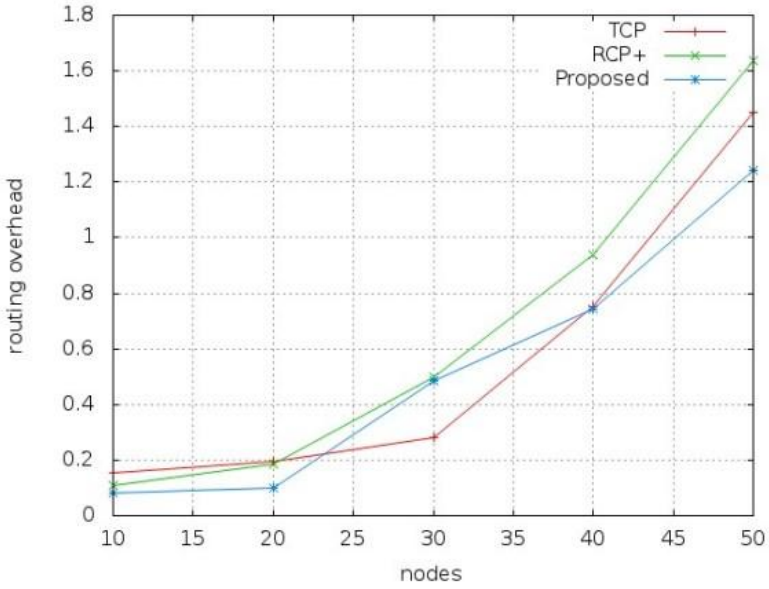


Graph 2 Packet Delivery Ratio graph

Routing load: Routing load is the ratio of data packets or receiving packets and routing packets. Greater value of routing load indicates more impact of router overhead on the performance of the protocol. From below statistics Routing load of proposed model is less than other protocols.

Table 4 Routing load Statistics

Routing load			
nodes	TCP	RCP+	Proposed
10	0.155	0.11	0.08
20	0.194	0.184	0.102
30	0.282	0.501	0.484
40	0.753	0.939	0.744
50	1.453	1.637	1.244

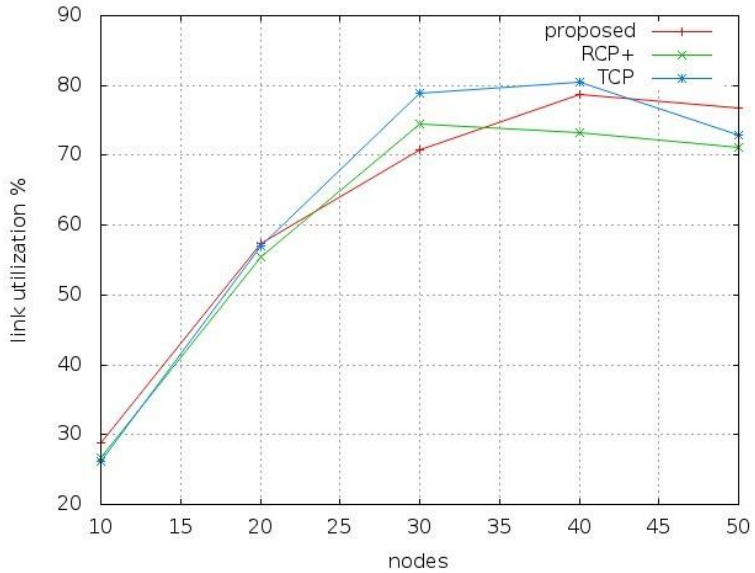


Graph 3 Routing load graph

Link Utilization: Link utilization is an important parameter to evaluate in congestion control protocols. It can be given as portion of link is used in transmitting and receiving packets. We have also obtained the percentage of link is utilized with respect to node and with respect to time.

Table 5 link utilization with respect to nodes

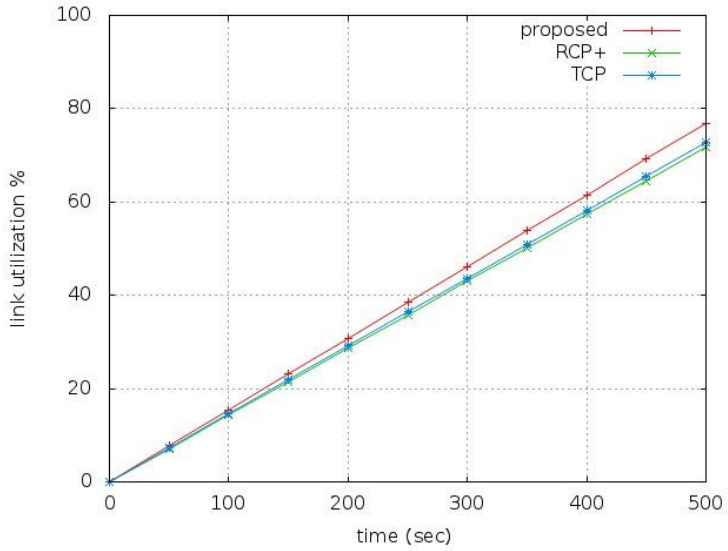
link utilization %			
nodes	TCP	RCP+	Proposed
10	26.13	26.76	28.88
20	57	55.5	57.32
30	78.83	74.51	70.85
40	80.46	73.23	78.77
50	72.81	71.7	76.86



Graph 4 Link utilization with respect to nodes

Table 6 link utilization with respect to time

link utilization %			
Time	TCP	RCP+	Proposed
0	0	0	0
50	7.28	7.17	7.69
100	14.56	14.34	15.37
150	21.84	21.51	23.06
200	29.12	28.68	30.74
250	36.4	35.85	38.43
300	43.68	43.02	46.11
350	50.96	50.19	53.8
400	58.25	57.36	61.49
450	65.53	64.53	69.17
500	72.81	71.7	76.86



Graph 5 Link utilization with respect to simulation time

VII. Conclusion

In this paper various protocols used for congestion control in wired and wireless network are presented and discussed their pros and cons. We have proposed a model and mechanism for congestion control, Enhanced RCP, which is window based strategy that has its fundamentals associated with RCP and TCP. We also done simulation of proposed model and evaluate it's performance with existing protocols. From the results we conclude that proposed model performs best in wireless network.

Acknowledgements

We would like to thank University of Grant Commission(UGC) for providing us the infrastructure on which we implemented our experimental study.

References

Journal Papers:

[1]. S. Floyd, "HighSpeed TCP for Large Congestion Windows," RFC 3649 ,http://www.icir.org/floyd/hstcp.html, December 2003.

[2]. Dr.AtulGonsai, BhargaviGoswami&UditnarayanKar, "Newly developed Algorithm RCP+ for Congestion Control on large scale Wireless Networks", International Journal of Innovations & Advancement in Computer Science IJIACS ISSN 2347 – 8616 Volume 3, Issue 2Apri 2014

[3]. Dr.AtulGonsai, BhargaviGoswami, UditnarayanKar, "Evolution of Congestion Control Mechanisms for TCP and Non TCP Protocols" Matrix Academic International Online Journal Of Engineering And Technology © MAIOJET. 2013

[4]. NanditaDukkipati& Nick McKeown, "Why Flow-Completion Time is the Right Metric for Congestion Control and why this means we need New Algorithms", ACM SIGCOMM Computer Communication Review, Volume 36,2006

[5]. Chia-Hui Tai, Jiang Zhu, NanditaDukkipati, "Making large scale deployment of RCP practical for real networks" IEEE INFOCOM, 2008

[6]. Mehta Ishani, UditnarayanKar, Dr.AtulGonsai, "Why RCP is better than TCP" International Journal for Scientific Research & Development (IJSRD),Vol. 2, Issue 12, 2015

Proceedings Papers:

[7]. Dr.AtulGonsai, BhargaviGoswami, UditnarayanKar, "Design of Congestion Control Protocol for Wireless Networks with Small Flow Completion Time" Proceedings on National Conference on Emerging Trends in Information & Communication Technology. 2013

Thesis:

[8]. NanditaDukkipati, "Rate Control Protocol (RCP): Congestion Control to make Flows Complete Quickly", Ph.D. Thesis, Stanford University, Stanford, California. 2007

Technical Reports:

[9]. NanditaDukkipati& Nick McKeown, "Processor Sharing Flows in the Internet", Stanford HPNG Technical Report, Stanford University, Stanford, California. 2007.

Books:

[10]. Book "Introduction to Network Simulator NS2" springer, 2009