

Comparative Study on Various Congestion Control Protocols: TCP, XCP and RCP

Mehta Ishani*, Nikul Virpariya** & Dr. Atul M Gonsai***

*PG Student, Department of Computer Science, Gardi Vidhyapith, Rajkot, Gujarat, INDIA. E-Mail: ishani.mehta91{at}gmail{dot}com

**Assistant Professor, Department of Computer Science, Gardi Vidhyapith, Rajkot, Gujarat, INDIA.

E-Mail: ngvirpariya{at}gardividypith{dot}ac{dot}in

***Associate Professor, Department of Computer Science, Saurashtra University, Rajkot, Gujarat, INDIA.

E-Mail: atul.gosai{at}gmail{dot}com

Abstract—Today's modern era of high speed multimedia transmission is demanding faster data transmission of huge amount of data. As a result data traffic is growing faster than capacity of existing network. So it becomes inescapable to deal with Congestion control. Various congestion protocols are used for avoiding congestion in network. TCP is the most popular congestion protocol in the Internet but it has some limitations, such as fail to utilize network link optimally in high bandwidth-delay product networks, and come out with high load and overhead in the network. To nullify these limitations, several congestion protocols like explicit Control Protocol and Rate Control Protocol have been proposed. These protocols have been proposed previously to work in wired and wireless networks and environments and also have pros and cons. Here we present evaluation study of these protocols that can be taken as a base to building an enhanced version of existing protocol.

Keywords—Congestion Control; Flow Completion Time; Processor Sharing; RCP; TCP; XCP.

Abbreviations—Explicit Control Protocol (XCP); Rate Control Protocol (RCP); Transmission Control Protocol (TCP).

I. INTRODUCTION

CONGESTION control is an important issue of computer network. The motivation of this research is to address this issue and trying to solve the Problem of large data transfer in high bandwidth delay networks. This paper aims with providing comparison between existing protocols which will help in building a new approach for this.

Congestion is a problem that occurs on shared networks when multiple users contend for access to the same resources (bandwidth, buffers, and queues). When two packets arrive at the same time then receiver node can only transmit one and either buffer or drop the other. If many packets arrive in a short period of time, the node cannot keep up with the arriving traffic and the buffer may eventually overflow. Now according to researchers Koushik Reddy Ramasani et al., (2014) congestion means, "Increase in network load results in a decrease of useful work done". Here flow control means preventing fast sender from overriding slow receiver and congestion control means preventing set of senders from overloading network. Moving to factors that cause congestion are [Dr. Atul M Gosai et al., 2013]:

- Packet arrival rate can't cope up with the outgoing link capacity.
- Insufficient buffer memory to store arriving packets
- Busy traffic
- Slow processor

Next section covers protocols for congestion control mechanism like TCP, XCP and RCP and discusses their positives and negatives. In section III we concluded with the performance evaluation of these established protocols.

II. CONGESTION CONTROL PROTOCOL

2.1. Transmission Control Protocol

According to Nandita Dukkupati (2007) thesis tcp fulfills 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? Congestion control mechanism given in Nandita Dukkupati (2007) thesis is shown in figure 1. Each TCP connection

initiate with a pre-configured small congestion window no larger than 4 Maximum Segment Size (MSS).

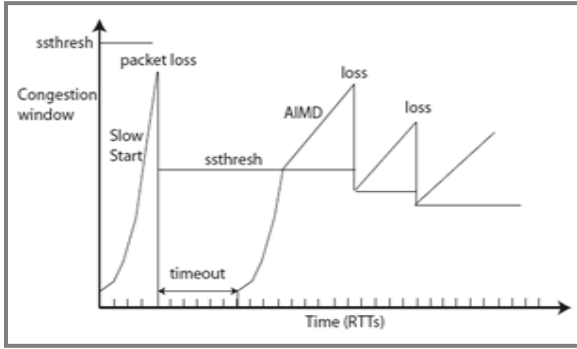


Figure 1: TCP's Congestion Control Mechanism

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 [Manjeet Kaur Bedi et al., 2012]. 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, $RTT \log_2 W$ (where RTT stands for round-trip time) time is required to reach a window of size W . 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,

$$cwnd = cwnd + 1 \quad (1)$$

on receiving each acknowledgment packet). Lu's Barreto & Susana Sargento (2010) paper found the problems with TCP in wired network are as following :

- Can't deal with today's increasing bandwidth demands
- Unable to utilize network link optimally
- Time consuming in finding fair transmission rate

Lu's Barreto & Susana Sargento (2010) paper found the problems with tcp In wireless network are as following:

- Doesn't respond well to packet loss due to bit errors and handoff

2.2. Explicit Control Protocol

The researcher of paper Magnus Proebster et al., (2009) had given the objective of XCP is efficiently uses the available link capacity, and also distributes the available bandwidth fairly between concurrent connections. These objectives can be achieved by Congestion controller (CC) and a fairness controller (FC), respectively. While the CC periodically

computes an aggregate feedback, denoting the required change of the congestion window of all flows, the FC distributes this aggregate feedback fairly between packets, without keeping any per flow state.

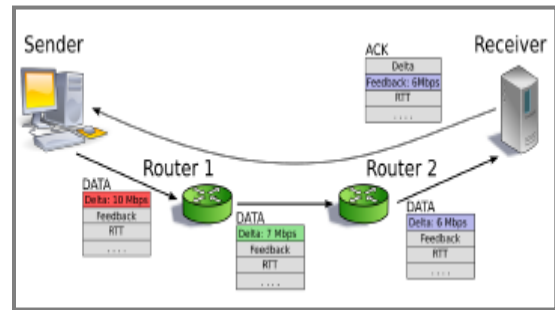


Figure 2: Principle Function of Explicit Congestion Control

CC matches input traffic to link capacity and drains the queue by using MIMD mechanism which gives equation for aggregate traffic changes by following equation,

$$\Delta = \delta(spare) - \delta(queue) \quad (2)$$

FC divides Δ between flows by using AIMD mechanism. If $\Delta > 0$ then divide Δ equally between flows. So equation for feedback is,

$$F = \left[\alpha (C - y(t)) - \beta \frac{q(t)}{d_0} \right] \quad (3)$$

Where, d_0 = a moving average of the RTT measured across all flows,

C = link capacity,

$y(t)$ = measured input traffic rate during the last update interval (d_0 in this case),

$q(t)$ = instantaneous queue size,

α, β = parameters chosen for stability and performance.

According to Magnus Proebster et al., (2009), benefits of XCP congestion control mechanism are:

- Use MIMD (Multiplicative Increase Multiplicative Decrease) for congestion control to grab/release large bandwidth quickly.
- Use AIMD (Additive Increase Multiplicative Decrease) to ensure fair bandwidth allocation

Cons of XCP congestion control mechanism are:

- Difficult to deploy
- Time consuming
- Computation overhead in routers

2.3. Rate Control Protocol

According to Magnus Proebster et al., (2009), the design objective of RCP protocol is to achieve very short flow completion time, i.e., all flows are completed as fast as possible. In order to fulfil this goal RCP requires the processor sharing scheduling algorithm when distributing the link capacity among all flows.

Magnus Proebster et al., (2009), stated that all router-assisted congestion control schemes require additional packet processing in the end-systems and also in all routers. This result in deployment challenges.

However, it is proved that router-assisted congestion control can indeed be implemented at high link speeds. The results show that XCP needs very complex processing

procedure in routers, whereas RCP have lower processing requirements. Besides that, the performance of XCP capable routers might be limited due to synchronization issues. RCP does not have this drawback.

According to Nandita Dukkipati & Nick McKeown (2007), 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.

There are three main reasons why TCP and XCP come out with such long FCTs:

2.3.1. *Extending Flows to last many Round Trip Times (RTT) in spite of they are Capable of Finishing within one/few RTTs*

In tcp flows that have entered AIMD phase have long FCTs because AIMD increases the window size s even slowly. While in xcp, it gradually reduces the window size of existing flows and increases the window size of new flows, making sure the traffic is never over-subscribed. It takes multiple RTTs for most flows to reach their fair share rate (which is changing as new flows arrive). Many flows complete before they reach their fair share rate.

2.3.2. *Filling up Buffers*

TCP will try to fill the buffers at the bottleneck link – until a packet is dropped. While this leads to high utilization of the bottleneck link, it increases queuing delay, RTT and the FCT.

2.3.3. *Timeouts and Retransmissions due to Packet Losses (TCP)*

Flows experience packet losses when the buffer overflows. Part of the link utilization is contributed by the retransmitted packets. We recently described a simple congestion control algorithm called the Rate Control Protocol (RCP) that greatly reduces FCTs for a broad range for network and traffic characteristics.

Nandita Dukkipati & Nick McKeown (2007A), emulate processor sharing in 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. This is clearly shown in figure 3.

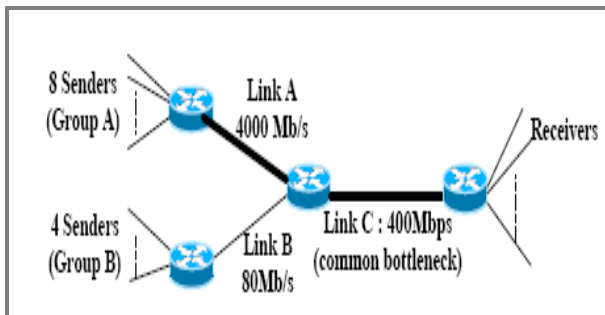


Figure 3: Processor Sharing in RCP

Following three main characteristics make RCP simple and practical:

- The flow rate is picked by the routers and it is based on very little information such as the current queue occupancy and the aggregate input traffic rate.
- Each router assigns a single rate, $R(t)$, through which all flows pass.
- The router doesn't require any per-flow state or per-packet calculations.

Nandita Dukkipati & Nick McKeown have been proposed an algorithm for RCP

- 1) Every router maintains a single fair-share rate, $R(t)$, that it offers to all flows. It updates $R(t)$ approximately once per RTT.
- 2) Every packet header carries a rate field, R_p . When transmitted by the source, $R_p = \infty$. When a router receives a packet, if $R(t)$ at the router is smaller than R_p , then $R_p \leftarrow R(t)$; otherwise it is unchanged. The destination copies R_p into the acknowledgment packets, so as to notify the source. The packet header also carries an RTT field, RTT_p , where RTT_p is the source's current estimate of the RTT for the flow. When a router receives a packet it uses RTT_p to update its moving average of the RTT of flows passing through it, d_0 .
- 3) The source transmits at rate R_p , which corresponds to the smallest offered rate along the path.
- 4) Each router periodically updates its local $R(t)$ value according to Equation (4) below.

$$R(t) = R(t - d_0) + \frac{\alpha(C - y(t)) - \beta \frac{q(t)}{d_0}}{N(t)} \quad (4)$$

where,

d_0 = a moving average of the RTT measured across all flows,

$R(t-d_0)$ = last updated rate,

C = link capacity,

$y(t)$ = measured input traffic rate during the last update interval (d_0 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.

III. COMPARATIVE STUDY

Magnus Proebster et al., (2009) stated the comparison of RCP with XCP and TCP. The main difference between XCP and RCP is in the kind of feedback that flows receive. XCP gives a window increment or decrement over the current window size of the flow (which is small for all newly starting flows).

At any time XCP flows could have different window sizes and RTTs and therefore different rates. XCP continuously tries to meet the point where all flows have the fair-share rate, by slowly reducing the window sizes of the

flows with rates greater than fair-share and increasing windows of the flows with rates less than fair-share (while avoiding over-subscription). New flows start with a small window, and the convergence could take several RTTs especially if there is little or no spare capacity. If the flows arrive as a Poisson process with heavy-tailed flow sizes, then most of the flows finish by the time they reach their fair share.

In RCP, all flows (new and old) receive the same rate feedback which is their equilibrium rate. This helps flows finish quickly. XCP is computationally more complex than RCP since it gives different feedback values to each flow, and involves multiplications and additions for every packet. RCP maintains a single rate for all flows and involves no per-packet computation. Router-assisted congestion control is a promising solution for the Future Internet.

According to Dr. Atul Gonsai et al., (2013), the new protocols XCP and RCP can utilize links more efficiently and fairer than state-of-the-art TCP mechanisms, since they do not only rely on packet losses to signal congestion. However, they require some limited additional processing in routers.

Table 1: Comparison for the Congestion Control Protocol

Points	TCP	XCP	RCP
Mechanism	AIMD, Slow Start	feedback rate equally divided among all flow	divided equilibrium rate among short lived and long lived flows
Router's Involvement	No	Yes	Yes
Per Packet Calculations	No	Yes	Yes
Flows	Not specific	Faster flow competition time but less than RCP	Short flow competition time
Bandwidth Utilization	Low	Average	Complete
Congestion Control	Low	Average	High
Problem of large data transfer in high bandwidth delay networks	Unsolved	Partially solved	Near to complete solution

Following figure shows the comparison of performance of TCP, XCP and RCP by taking parameters as Flow completion time and flow length. These results confirm that RCP is indeed better suited, since RCP provides a fast start for short flows. Its performance comes closest to the minimum possible FCT. With the chosen parameters, RCP approximates processor sharing (PS). An interesting effect is that TCP has a shorter FCT for very long flows. But this comes at the cost of larger delays for short flows which are limited by slow-start and results in a longer average FCT.

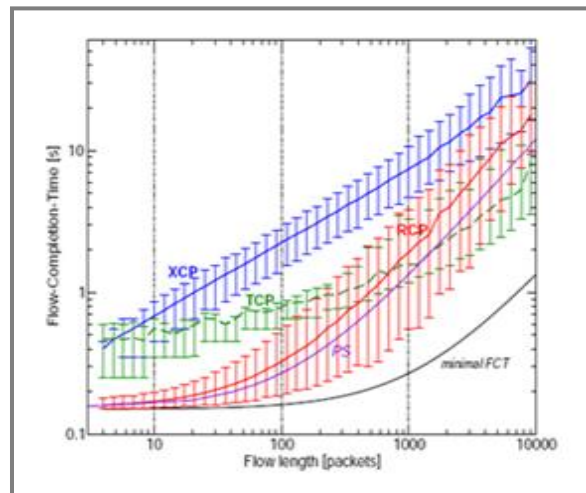


Figure 5: Comparing Flow Completion Time and Processor Sharing of TCP, XCP and RCP

Chia-Hui Tai et al., (2008) found that RCP too have some limitations regarding adapting into a network include co-existing with non RCP traffic and non RCP bottlenecks.

IV. CONCLUSION

In this paper we have reviewed mechanism of various congestion control protocol TCP, XCP and RCP. Also present the performance comparison of this protocol by taking different parameters like window size, flow length, and flow completion time.

As a result RCP and TCP perform well with minimal flow completion time. TCP come up with minimal FCT at cost of large delays and RCP also have problem regarding co-existing with non RCP traffic and non RCP bottlenecks. To solve this we are going to propose a new enhances version of protocol RCP. The future work will be concentrated on the implementation of enhanced RCP protocol.

REFERENCES

- [1] Nandita Dukkupati (2007), "Rate Control Protocol (RCP): Congestion Control to make Flows Complete Quickly", *Ph.D. Thesis, Stanford University*, Stanford, California.
- [2] Nandita Dukkupati & Nick McKeown (2007), "Why Flow-Completion Time is the Right Metric for Congestion Control and why this means we need New Algorithms", *Stanford HPNG Technical Report, Stanford University*, Stanford, California.
- [3] Nandita Dukkupati & Nick McKeown (2007A), "Processor Sharing Flows in the Internet", *Stanford HPNG Technical Report, Stanford University*, Stanford, California.
- [4] Chia-Hui Tai, Jiang Zhu & Nandita Dukkupati (2008), "Making Large Scale Deployment of RCP Practical for Real Networks", *IEEE Infocom*.
- [5] Magnus Proebster, Michael Scharf & Simon Hauger (2009), "Performance Comparison of Router Assisted Congestion Control Protocols: XCP vs. RCP", *QoSSim 2009, Rome, Italy Copyright 2009 ICST, ISBN 978-963-9799-45-5*.
- [6] Lu's Barreto & Susana Sargento (2010), "TCP, XCP and RCP in Wireless Mesh Networks: An Evaluation Study", *IEEE*, 978-1-4244-7755-5/10/\$26.00

- [7] Manjeet Kaur Bedi, Kendar Pratap & Raj Kumari (2012), "A New Approach for Congestion Control in Wireless Network and Analyze it for TCP", *International Journal of Application or Innovation in Engineering & Management (IJAIEEM)*, Vol. 1, No. 2.
- [8] Dr. Atul Gosai, Bhargavi Goswami & Uditnarayan Kar (2013), "Design of Congestion Control Protocol for Wireless Networks with Small Flow Completion Time", *Proceedings on National Conference on Emerging Trends in Information & Communication Technology*.
- [9] Dr. Atul M Gosai, Bhargavi H Goswami & Uditnarayan Kar (2013), "Evolution of Congestion Control Mechanisms for TCP and Non TCP Protocols", *Matrix Academic International Online Journal Of Engineering And Technology*.
- [10] Koushik Reddy Ramasani, Dr. Syed Umar & Seelam Eswar (2014), "The Survey of Congestion Control Protocols for Smart Transport System", *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 2, No. 7.
- [11] Nandita Dukkipati & Nick McKeown "Rate Control Protocol (RCP) Home Page", URL: <http://yuba.stanford.edu/rcp/>.



Ishani Mehta. Completed her B.E. in computer Science from Nobel Engineering College Junagadh, Gujarat (affiliated with Gujarat Technological University) in 2012 and pursuing M.E. in computer science from Gardi Vidhyapith Rajkot, Gujarat (affiliated by Gujarat Technological University). Now working on dissertation subject congestion control protocols.



Prof. Nikul Virpariya, M.Tech. [ICT], B.E.[computer] is an assistant professor, department of computer science engineering at B.H. Gardi college of engineering and technology, Rajkot. He has completed Master of Technology [M.Tech.] from gyan vihar university, Jaipur in July-2011. He has also completed Bachelor of Engineering [B.E.] from Atmiya Institute of Technology and Science, Rajkot in June-2009. His research area is networking protocol analysis, implementation and design. He is teaching subjects like computer network, Advanced Computer Network, Data structure, Database Management system, operating system, operating system design.



Dr. AtulGosai, Ph.D., MCA, BBA is an Associate Professor, Department of Computer Science, Saurashtra University, Rajkot. AtulGosai has received his Ph. D degree in the field of Computer Science from the same University. He has also completed Master of Computer Applications (MCA) from the same Department in April 2000. He has total teaching experience of Fourteen years. He has been awarded the "CAREER AWARD FOR YOUNG TEACHERS" from ALL INDIA COUNCIL FOR TECHNICAL EDUCATION (AICTE) New Delhi with amount of Rs. 10.5 lakhs. He has been teaching subjects like UNIX/ Linux Concepts and Shell Programming, C Programming and Data Structure, Networking and Advanced Networking, Network Programming and Java. He is the author of Seven books and over 69 research papers related to computer science and computer networking in International and National Journals and conferences. His research interest encompasses performance tuning of computer networks, wireless networks, High performance networking etc.