OPTIMIZATION IN CONGESTION AT TRANSPORT LAYER

Richa Yadav[1], Dr. R. K. Mehrotra[2]

[1]Student, Electronics and Communication Department, AKG Engineering College, (India)

[2]Professor, Electronics and Communication Department, AKG Engineering College, (India)

ABSTRACT

Existing networks with transport protocols such as TCP and UDP have limitations when fitting into new technologies, like increased heterogeneity and mobility. In order to solve this problem, our attempt is to design a self-adaptive transport protocol, which can fit into the network condition dynamically according to the parameters given by upper layer and the network layer in (wired/wireless) NGN architecture.

This paper analyzes the optimization in congestion control mechanisms of nowadays transmission protocols, and does comparison among them in different aspect. We first describe the congestion control mechanism of traditional TCP, and then give out an overview and list the congestion function of transport protocols (such as TCP, UDP, DCCP, SCTP). In the end, we do a simple comparison among all the protocols above, and point out the possible advantages and limitations of these protocols under different transport performances (RTT Fairness, TCP Fairness, Utilization Ratio, Packet Loss Rate, Smoothness of throughput, Convergence Time, etc.).

Future work will focus on, to implement the next-generation gateway, optimized for broadband multi-media communications over satellites. It will draw on the full power of the DVB-RCS protocol, and will merge technical advances on transport layer optimization, Quality of Service (QoS) provisioning and dynamic bandwidth allocation.

Keywords: Congestion Algorithm, DCCP, ECN, NGN, RTT Fairness, SCTP, TCP Fairness.

I INTRODUCTION

Recently, the rapid evolution and successful deployment of various emerging wireless technologies,e.g., IEEE 802.11 a/b/g, WiMax, etc., has pushed into a strong demand to integrate numerous wirelesslocal area networks (WLANs) with the existing cellular network infrastructure. The typical example involves the integration of WLAN with Global System for Mobile Communications/General Packet Radio Service (GSM/GPRS), third-generation (3G) Universal Mobile Telecommunications System (UMTS), or cdma2000 networks. The inter-working of such heterogeneous, packet-based radio access networks (RANs), also referred to as *next generation* or *beyond 3G (B3G)* mobile data networks[1], poses many technical challenges with mobility managementthat can guarantee service continuity and IP connectivity provision for wireless multi-mode mobile terminals like cellular phones, personal digital assistants (PDAs), and notebook computers, being one of the most important .

Earlier works on the mobility management problem in heterogeneous networks discussed solutions in various layers of the International Organization for Standardization Open Systems Interconnection Basic Reference Model (ISO/OSI model) of the protocol stack where the mobility can be handled: application, transport, and network and data-link layer, respectively. In terms of challenges present in heterogeneous networks, transport layer seems a feasible candidate to host seamless mobility management. Nevertheless, the vast majority of the new mobility-related proposals seem to follow most of the existing schemes and stick to mobility handled at the network layer.

In the last few years several research efforts have been devoted to study the performance of both TCP and UDP over wired, wireless, and heterogeneous network scenarios. Recently, the Stream Control Transmission Protocol (SCTP) and the Datagram Congestion Control Protocol (DCCP) have attracted the attention of the research and industry communities [2]. SCTP was developed to support signaling in Voice Over IP like applications. Considering that many applications may gain advantages from its peculiarities, SCTP is now considered as a general purpose transport protocol. Therefore, there is an increasing interest about the performance of SCTP over real networks. DCCP is a fairly new unreliable datagrams transport protocol that provides a congestion-controlled flow. DCCP should make easy to deploy delay-sensitive application, without risking congestion collapse.Datagram Congestion Control Protocol (DCCP) is an un-reliable transport layer protocol that provides congestion control on datagrams innetwork.It has support for Explicit Congestion Notification (ECN) [3]. ECN supports for the notification of end-to-end network congestion. Traditional transport protocols that operate over unreliable battlefield networks provide an "all-or-nothing" choice for transport Quality of Service (QoS); either total order and reliability (e.g., TCP) or no guarantees at all (e.g., UDP).

In this paper, our main objective is to optimize in the different congestion control mechanisms of the existing protocols and finding out how they affect the performance in different aspects. In section 2, the traditional transmission control protocol (TCP) would be presented as an example to explain how congestion control mechanism works. Section 3 to section 4, list some other transport protocols used nowadays and describes their main features, including the motivation, congestion algorithm, advantages and limitations. The future research and conclusion is given in Section 5.

II CONGESTION CONTROL MECHANISM OF TCP

The transport layer flow control as any scheme by which the transport sender limits the rate at which data assent over the network [10]. The goals of flow control may include one or both of the following:

(a) preventing a transport sender from sending data for which there is no available buffer space at the transport receiver, or (b) preventing too much traffic in the underlying network.

Flow control for (b) also is called congestion control, or congestion avoidance. Congestion is essentially a network layer problem, and dozens of schemes are discussed and classified. In TCP, there is a congestion window (*cwnd*) which determines the number of bytes that can be sent out at any time. This is used to prevent the physical link between two end points from getting overloaded with too much traffic. The sending rate should always be under the size of the congestion window. TCP uses a slow-start mechanism at the beginning when association established. For initializing the connection, every Route Trip Time, *cwnd*= *cwnd*+ 1, which is exponential growth. Although it's

called slow-start, actually it is not slow anymore. As we can see, through slow-start, the congestion window can reach to a large value quickly, but obviously it cannot keep growing [4]. TCP set a threshold for the slow-start session.

When the size of congestion window has reached the value, it comes into the congestion control session. In this session, if all segments are received and the acknowledgments reach the sender on time, the window size will increase steadily but slowly. The window keeps growing linearly until a timeout occurs or the receiver reaches its limit. If a timeout occurs, the window size will drop dramatically. For congestion control algorithm, TCP use:

Ack:cwnd= cwnd+ a/cwnd

Loss :cwnd= cwnd - b×cwnd

It is additive increase or multiplicative decrease (AIMD) algorithm [1]. This congestion control algorithm works well in low-RTT network. But with the development of long distance network, if RTT is high, the utilization of the whole network is quite low. Another important requirement is RTT fairness. It means, in a certain bandwidth network environment, different connections with different RTT should share approximately the same bandwidth utilization ratio, but TCP is not a good solution. Nowadays, people are already trying to find a solution to improve TCP.

III DATAGRAM CONGESTION CONTROL PROTOCOL (DCCP)

DCCP is a connection oriented and un-reliable transport layer protocol. Many other transport protocols appear in order to fit into different usages and vary with network physical conditions, especially for the high bandwidth delay product network. In the following sections we analyze some of the most well-known protocols so as to do the comparison in the discussion part.

3.1 Motivation

Fast-growing Internet applications including streaming media, telephony and interactive games need new requirements of network protocols. Most of them prefer timeliness to reliability. One special requirement of those applications is that they are extremely sensitive to delay and quality fluctuation. On the other hand, losing a certain number of packages would not affect the quality of service [5]. This special characteristic of real-time application decides that TCP is not suitable for them, because TCP rather focuses on ensuring data transmission. In this case, retransmission of data packets is not need, and so does the order of packets' arrival. Most of these applications currently use UDP. Through the analysis of UDP traffic, UDP's lack of explicit connection setup and teardown presents unpleasant difficulties to network address translators and firewalls. Furthermore, because of UDP's lacking of congestion control, competing traffic problem would be caused. In this circumstance, DatagramCongestion Control Protocol appears.

3.2 Overview

DCCP is a unicast, connection-oriented transport protocol with bidirectional data flow. It provides built-in congestion control, including Explicit Congestion Notification (ECN) support [12]. DCCP offers a choice of modular congestion control mechanisms among a set of standardized algorithms for real-time applications. For the moment, two mechanisms are currently specified, TCP-like and TCP-Friendly Rate Control (TFRC) congestion control. These algorithms aim different applications. For instance, on-line games which want to make quick use of any available bandwidth might use TCP-Like; while streaming media applications trade off this responsiveness for a steadier, less busty rate might useTFRC.

Different from UDP, DCCP connections start and end with three-way handshakes, and the 16-byte generic header which datagrams begin with is shown in Fig. 1. The Type field gives the type of packet. Even the acknowledgement number is optional, potentially reducing header overhead for unidirectional flows of data. Normally sequence and acknowledgement numbers are 48 bits long, but end points can set it to 24 bits while in the session negotiation period.

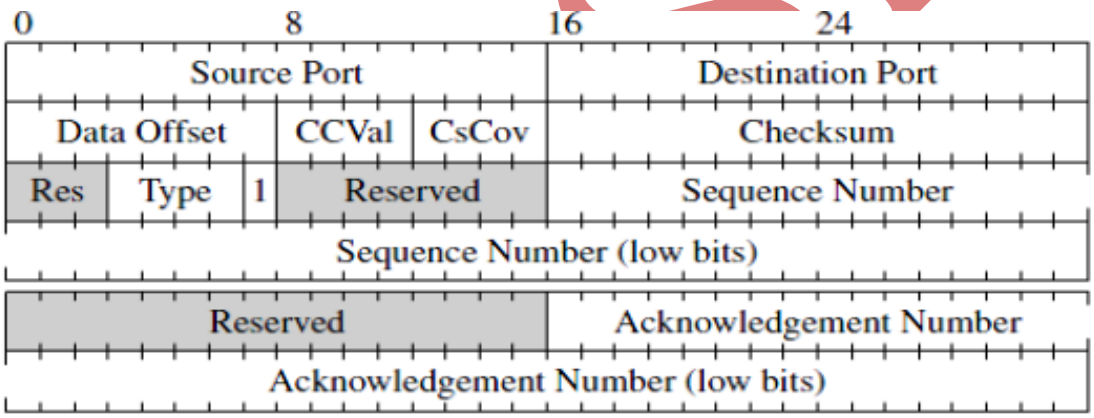

Figure 1: DCCP Header

3.3 Congestion Control Mechanism

For congestion control algorithm, DCCP gives the application some choices of congestion control mechanisms. The choice is made via Congestion Control IDs (CCIDs). A connection's CCIDs can be negotiated when establishing the connection. DCCP's CCID2 provides a TCP-like congestion control mechanism. Its congestion control algorithms are quite similarwith TCP: a congestion window *cwnd*, a slow-start threshold, and an estimate of the number of data packets outstanding [14]. To reduce Ack load, it is set to at least two for a congestion window of four or more packets. However, to ensure that feedback is sufficiently timely, it is capped at *cwnd*=2, rounded up. Within these constraints, the sender changes Ack Ratio as follows. Let *R* equals the current Ack Ratio.

(1) For each congestion window of data where at least one of the corresponding Acks was lost or marked, *R* is doubled;

(2) For each *cwnd*=(*R*2¡*R*) consecutive congestion windows of data whose Acks were not lost or marked, *R* is decreased by 1.

The second formula is used to increase the number of Acks per congestion window, namely *cwnd=R*, by one for every congestion-free window that passes. However, since *R*is an integer, we instead find a *k* so that, after *k* congestionfree windows, *cwnd=R + k = cwnd=*(*R ¡* 1).

TFRC congestion control in DCCP's CCID3 uses a different approach. Instead of a congestion window, a TFRC sender uses a sending rate. The receiver sends feedback to the sender roughly once per round trip time (RTT) reporting the loss event rate. The sender uses this loss event rate to determine its sending rate; if no feedback is received for several round-trip times, the sender halves its rate [7], [8]. Regulate Sending rate is set by a Markov Model. The model is described as: *X* is the transmit rate in bytes/second s is the packet size in bytes *p* is the loss event rate *T*0 is the TCP retransmission time in seconds.

$$X(p) = \frac{s}{RTT\sqrt{2bp/3} + T_0(3\sqrt{3bp/8})p(1+32p^2)} \quad (1)$$

This is reasonably straightforward, and does not require reliable delivery of feedback packets, as long as the sender trusts the receiver's reports of the loss event rate. However, a mere loss event rate is ripe for abuse by misbehaving receivers.

3.4 Advantages

DCCP is a nice solution for real-time application. It avoids Internet congestion caused by package loss like UDP usually do. With the unreliable feature of UDP, it also wastes the traffic of networks. DCCP is a low-expense, unreliable congestion control protocol.

DCCP can send data messages simultaneously. The header of DCCP is changeable, the most common one only use 12 bytes. DCCP have two kinds of congestion control mechanism, and may adding new ones now. DCCP have 9 kinds of packets, more than TCP and UDP. This increases its flexibility and expands ability. Such as the DCCP-Mov packets helps it to be adapted to mobile devices [4].

3.5 Limitations

Until now, DCCP also has some problems. DCCP should support both IPv4 and IPv6 at the same time. Whether DCCP is secure enough is still under consideration. Furthermore, applications generally do not want to implement TCP-friendly congestion control themselves. This is not only because congestion control can constrain performance, but also because properly implementing congestion control is very hard, as the long history of buggy TCP implementations makes clear [4]. Applications might be willing to subject themselves to congestion control, not least for the good of the network, as long as it was easy to use and met their needs. A modular congestion control framework would also make it easier to develop new applications, and to deploy congestion control advances across many applications at once.

IV STREAM CONTROL TRANSMISSION PROTOCOL (SCTP)

4.1 Motivation

It is necessary to transfer signaling messages over it. But TCP and UDP is not a good solution for those telecommunication network applications. SCTP is a general purpose unicast transport protocol for IP network data communications, which has been recently standardized by the IETF [6]. It was initially introduced as a means to transport telephony signaling messages in commercial systems, but has since evolved for more general use to satisfy the needs of applications that require a message-oriented protocol with all the necessary TCP-like mechanisms. Traditional TCP protocol has the problems of Head Of Line blocking, bad real time support, vulnerable to Denial of Service attack and so on [13]. SCTP is a better solution under this circumstance.

4.2 Overview

SCTP provides sequencing, flow control, reliability and full duplex data transfer like TCP. However, it also enhances a set of capabilities not in TCP that make applications less susceptible to loss. Like UDP, SCTP is message-oriented and supports the framing of application data. Meanwhile like TCP,

SCTP is session-oriented and communicates by establishing an association between two endpoints. Different with TCP, in SCTP, it is possible to have multiple logical streams within an association where each is an independent stream of messages and delivered in-order. Some of the important feature of SCTP is: Multi-homing, Multi-streaming, Initiation protection, Message framing, Configurable unordered delivery and Graceful shutdown. Normally, the upper layer user of SCTP would be Switched Circuit Network (SCN) signaling adaptable module, and the lower layer would be IP network.

4.3 Congestion control Algorithm

Comparing with traditional transport protocol, the most important feature of SCTP is multi-homing and multi-streaming. One of the most important change of SCTP is its support of multi-homed nodes, which means a server can be reached by several different IP addresses.

If packages from one node to another travels on physically different paths, and also different destination IP address are used, the connection becomes tolerant against physical network failures and otherproblems of that kind. As shown in Fig. 4.1, server has two IP addresses which are available for both Ethernet and wireless network connection. Client can connect to server by multi-homing, therefore if one connection break, client can still maintain data exchanging with server by another connection.

Another important feature of SCTP is, it supports multiple streams within an association. In SCTP, each stream represents a sequence of messages within a single association, and they use their own sequence number just like different TCP sessions. Both stream identifiers and sequence numbers are included in the data package [9], [11]. This means that there would be no unnecessary head-of-line blocking between independent streams of messages in case of loss in one stream. All the streams within an association are independent but related to the association as shown in Fig. 4.2 for the congestion control function, STCP uses a more aggressive method.

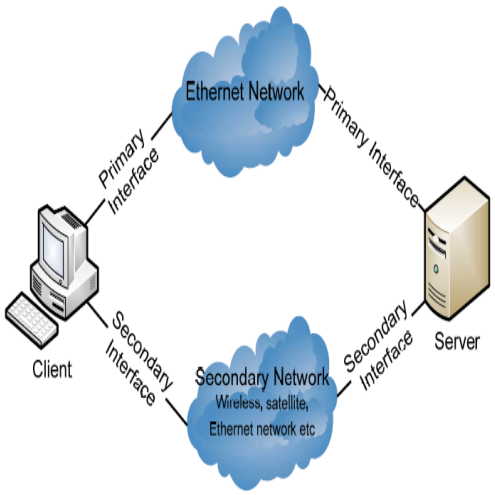

Figure 4.1: Multi-Homing Mechanism

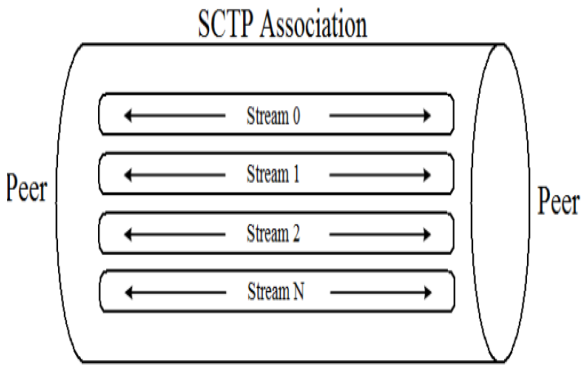

Figure 4.2: Multi-Streaming Mechanism

Ack:cwnd= cwnd+ a

Loss :cwnd= cwnd - b × cwnd

In this case, every time congestion window decrease, only a certain number of RTT is needed for recovering the original size of the congestion window, which is not relevant with the size of window now. This means, SCTP uses multiplicative increase and multiplicative decrease algorithm for congestion control.

4.4 Advantages

Though SCTP has TCP-like congestion and flow control mechanisms targeted for bulk data transfer, we argue that SCTP's feature-set makes it a better web transport than TCP. Performance-wise, SCTP's multistreaming avoids TCP's HOL blocking problem when transferring independent web objects, and facilitates aggregate congestion control and loss recovery [11]. Functionality-wise, SCTP's multihoming provides fault-tolerance and scope for load balancing, and a built-in cookie mechanism in SCTP's association establishment phase provides protection against SYN attacks.

4.5 Limitations

The major limitation of SCTP might be RTT fairness and TCP friendly. As SCTP use a quite aggressive congestion control function, it may occupy most of the bandwidth while working with other TCP connection. Furthermore, the aggressive algorithm would surely cause high package loss and lead to the multiplicative decreasing of TCP connections.

V CONCLUSION

In this paper, we did a survey on the most popular Transport protocols nowadays. TCP, DCCP, SCTP, and UDP are listed and analyzed. Especially, we focused on the congestion avoidance control algorithm of them. During the analysis, we find the most important feature of TCP nowadays is to balance efficiency with fairness. In different network conditions, fairness, convergence time, packet loss rates, link utilization, RTT fairness, TCP friendliness, and stability of throughput are considered to evaluate these protocols. Congestion control algorithms might directly decide the performance of these protocols. DCCP is based on UDP and added congestion control for real time application. SCTP use multihoming and multi streaming, to increase the utilization ratio. And my analysis is just used as a reference and needs experimental proof**.**

REFERENCES

[1] K. Tan, J. Song, Q. Zhang, and M. Sridharan. Compound tcp: "*A scalable and tcp-friendly congestion control for high-speed networks*", In In Proc. of PFLDnet, 2006.

[2] Aydin, I., &Shen, C.-C. (2009), "*Performance Evaluation of Concurrent Multipath Transfer Using SCTP Multihoming in Multihop Wireless Networks*",Proceedings 8th IEEEInternational Symposium on Network Computing and Applications, pp. 234-241.

[3] E. Kohler, M. Handley, and S. Floyd, "*Datagram congestion control protocol (DCCP)*", IETF Internet Draft, draft-ietf-dccp-spec-11.txt.

[4] G. A. Abed, M. Ismail, and K. Jumari, "*A Survey on Performance of Congestion Control Mechanisms for Standard TCP Versions*," Australian Journal of Basic and Applied Sciences, vol. 5, pp. 1345- 1352, 2011.

[5] M. C. Chuah and Q. Zhang, "*Design and Performance of 3G Wireless Networks and wireless LANS"*: Springer, 2006.

[6] D. Kliazovich, F. Granelli, S. Redana, and N. Riato, "*Cross-layer error control optimization in 3G LTE*," 2007, pp. 2525-2529.

[7] G. A. Abed, M. Ismail, and K. Jumari, "*A Comparison and Analysis of Congestion Window for HS-TCP, Full-TCP, and TCP-Linux in Long Term Evolution System Model*," in Open Systems (ICOS), 2011 IEEE Conference on, 2011, pp. 358-362.

[8] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "*Stream control transmission protocol*", RFC 2960, Oct. '07

[9] Cheng, R.-S., D.-J., Chao, H.-C., & Chen, W.-E. (2010), "*An Adaptive Bandwidth Estimation Mechanism for SCTP over Wireless Networks"*, Proceedings 5th InternationalConference on Future Information Technology, pp. 1-5.

[10] S. Iren, P. D. Amer, and P. T. Conrad, "*The transport layer: tutorial and survey*", ACM Comput. Surv.,31(4):360–404, 2005.

[11] H. Kamal, B. Penoff, and A. Wagner. "*Sctp versus tcp for mpi*". In *SC* '05: Proceedings of the 2005ACM/IEEE conference on Supercomputing, page 30, Washington, DC, USA, 2005. IEEE Computer Society.

[12] E. Kohler, M. Handley, and S. Floyd. Designing, "*DCCP: congestion control without reliability*", In SIGCOMM'06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols forcomputer communications, pages 27–38, New York, NY, USA, 2006. ACM.

[13] P. Natarajan, J. R. Iyengar, P. D. Amer, and R. Stewart. "*Sctp: An innovative transport layer protocol for the web*". In WWW '06: Proceedings of the 15th international conference on World Wide Web, pages 615–624, New York, NY, USA, 2006.

[14] F. Nivor. "*Experimental study of SCTP and DCCP for multimedia applications*", In CoNEXT '05: Proceedings of the2005 ACM conference on Emerging network experiment and technology, pages 272–273, New York,USA, 2005.