IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, **p**-ISSN: 2278-8719
Vol. 3, Issue 10 (October. 2013), ||V2|| PP 56-59

Reliable New Transport Protocol

Firas Sabah Alturaihe

Abstract: - We need a new protocol which is expected to be easily deployed and easily integrated with the applications, in addition to utilizing the bandwidth efficiently and fairly.

I. INTRODUCTION

As the spread of TCP/IP networks, many kinds of applications that work over a TCP/IP network have appeared. So the diversity of these applications demands various qualities of services from a network. However, there exist only two transport layer protocols. TCP and UDP, in TCP/IP protocol suite. These protocols do not always meet these various demands. In this paper, we propose a new Application layer protocol over IP/UDP networks that has variable reliability for transferring data and satisfies both the objectives of efficiency and fairness (including intra-protocol fairness, RTT fairness, and TCP friendliness).It can be used in the emerging distributed data intensive applications such as grid computing, where a small number of data flows share the abundant optical network bandwidth. It is at the application level and it is easy to deploy without the need for administrative privileges.

II. PREVIOUS WORK

1- TCP Modifications

Researchers have continually worked to improve TCP. A straightforward approach is to use a larger increase parameter and smaller decrease factor in the AIMD algorithm than those used in the standard TCP algorithm. Scalable TCP [2]and High Speed TCP [1] are the two typical examples of this class. Scalable TCP increases its sending rate proportional to the current value, whereas it only decreases the sending rate by 1/8 when there is packet loss. HighSpeed TCP uses logarithmic increase and decreases functions based on the current sending rates. Both of the two TCP variants have better bandwidth utilization, but suffer from serious fairness problems. The MIMD (multiplicative increase multiplicative decrease) algorithm used in Scalable TCP may not converge to fairness equilibrium, whereas HighSpeed TCP converges very slowly. BiC TCP [3] uses a similar strategy but proposes a more complicated method to increase the sending rate. Achieving good bandwidth utilization, BiC TCP also has a better fairness characteristic than Scalable and HighSpeed TCP. Unfortunately, none of the above three TCP variants address the RTT bias problem; instead, the problem becomes more serious in these three TCP versions, especially for Scalable TCP and HighSpeed TCP. In addition, BiC TCP has an upper limit on the increase parameter, thus it is less scalable. TCP Westwood [5] tries to estimate the network situation (available bandwidth) and then tunes the increase parameter accordingly. The estimation is made through the timestamps of acknowledgments. This strategy demonstrates a good idea for using a bandwidth estimation technique in end-to-end congestion control algorithms. However, the Westwood method may be seriously damaged by the impact of ACK compression [4], which can occur at the existence of reverse traffic or NIC interrupt coalescence. Other recently proposed loss-based TCP control algorithms also include Layered TCP (L-TCP) [8] and Hamilton TCP (HTCP) [7]. L-TCP uses a similar strategy as HighSpeed TCP by simulating the performance of multiple TCP connections to realize higher bandwidth utilization. H-TCP tunes the increase parameter and the decrease factor according to the elapsed time since the last rate decrease. Delay-based approaches have also been investigated. The most well known TCP variant of this kind is probably the TCP Vegas algorithm. TCP Vegas compares the current packet delay with the minimum packet delay that has been observed. If the current packet delay is greater, then it means that in some place the queue is filling up, which indicates network congestion. Recently, a new method that follows the Vegas' strategy called FAST TCP was proposed. FAST uses an equation-based approach in order to react to the network situation faster. Although there has been much theoretical work on Vegas and FAST, many of their performance characteristics on real networks are yet to be investigated. In particular, the delay information needed by these algorithms can be heavily affected by reverse traffic. As a consequence, the performance of the two protocols is very vulnerable to the existence of reverse traffic.

2- XCP

XCP [6], which adds explicit feedback from routers, is a more radical change to the current Internet transport protocol. While those TCP variants mentioned in sub-section 1.3.1 tried many methods to estimate the network situation, XCP takes advantage of explicit information from the routers. As an XCP data packet passes each router, the router calculates an increase parameter or a decrease factor and updates the related information in the data packet header. After the data packet reaches its destination, the receiver sends the information back through acknowledgments.

An XCP router uses an MIMD efficiency controller to tune the aggregate data rate according to the current available bandwidth at the bottleneck node. Meanwhile, it still uses an AIMD fairness controller to distribute the bandwidth fairly among all concurrent flows. XCP demonstrates very good performance characteristics. However, it suffers more serious deployment problems than the TCP variants because it requires changes in the routers, in addition to the operating systems of end hosts. In addition, recent work showed that gradual deployment (to update the Internet routers gradually) has a significant performance drop [13].

III. APPLICATION LEVEL SOLUTIONS

While TCP variants and new protocols such as XCP suffer from deployment difficulties, application level solutions emerge as a favorite timely solution. A common approach is to use parallel TCP, such as PSockets [13] and Grid FTP [2]. Using multiple TCP flows may utilize the network more efficiently, but this is not guaranteed. Performance of parallel TCP relies on many factors from end hosts to networks. For example, the number of parallel flows and the buffer sizes of each flow have significant impact on the performance. The optimal values vary on specific networks and end hosts and are hard to tune. In addition, parallel TCP inherits the RTT fairness problem of TCP. Using rate-based UDP has also been proposed as a scheme for high performance data transfer to overcome TCP's inefficiency. There is some ongoing work including SABUL [10], FOBS [9], RBUDP [12], FRTP [11], and Hurricane [14]. All of these protocols are designed for private or QoS-enabled networks. They have no congestion control algorithm or have algorithms only for the purpose of high utilization of bandwidth.

1- SABUL

SABUL (Simple Available Bandwidth Utilization Library) was our prototype for UDT. The experiences obtained from SABUL encouraged us to develop a new protocol with better protocol design and congestion control algorithm.

SABUL is an application level protocol that uses UDP to transfer data and TCP to transfer control information. SABUL has a rate-based congestion control algorithm as well as a reliability control mechanism to provide efficient and reliable data transport service. The first prototype of SABUL is a bulk data transfer protocol that sends data block by block over UDP, and sends an acknowledgment after each block is completely received. SABUL uses an MIMD congestion control algorithm, which tunes the packet-sending period according to the current sending rate. The rate control interval is constant in order to alleviate the RTT bias problem. Later we removed the concept of block to allow applications to send data of any size. Accordingly, the acknowledgment is not triggered on the receipt of a data block, but is based on a constant time interval. Our further investigation on the SABUL implementation encourages us to re-implement it from scratch with a new protocol design. Another reason for the redesign is the use of TCP in SABUL. TCP was used for the simplicity of design and implementation. However, TCP's own reliability and congestion control mechanism can cause unnecessary delay of control information in other protocols that have their own reliability and congestion control as well. The in-order delivery of control packets is unnecessary in SABUL, but the TCP reordering can delay control information. During congestion, this delay can be even longer due to TCP's congestion control.

IV. PROPOSED PROTOCOL: NEW TRANSPORT PROTOCOL (NTP)

New transport protocol is the data transport protocol proposed by this proposal is to support the distributed data intensive applications in wide area high-speed networks. It aims to address the solution by investigating two orthogonal research problems:

1) the design and implementation of transport protocols with respect to throughput and CPU usage; and,

2) the Internet congestion control algorithm with respect to efficiency, fairness, and stability.

NTP is an application level, end-to-end, unicast, reliable, connection-oriented streaming data transport protocol. This protocol is completely at user space above UDP, i.e., it uses UDP to transfer user data and protocol control information. UDT uses packet-based sequencing to check packet loss and guarantee data reliability. It is specially designed for high-speed bulk data transfer by aiming to remove or reduce the overhead of memory copy, loss information processing, acknowledging, etc. It provides reliable streaming data transfer service, similar to TCP. The NTP protocol supports a large variety of control algorithms. Moreover, it supports congestion control algorithms to be configured at run time, thus each NTP flow can have its own control algorithm and it can change the algorithm at any time. The built-in (default) NTP congestion control algorithm is proposed to utilize high bandwidth efficiently and fairly. The NTP algorithm uses a loss-based AIMD mechanism. Bandwidth estimation technique is used to optimize its increase parameter dynamically. A random decrease factor is used to remove the negative effect of loss synchronization. It is not used to replace TCP on the Internet where the bottleneck bandwidth is relatively small and there are large amounts of multiplexed short life flows. However, when coexisting with TCP flows, it is designed not to occupy more bandwidth than TCP does unless the TCP flows fail to utilize their fair share due to TCP's efficiency problems in high bandwidth-delay product (BDP) environments. This design goal is due to the fact that TCP will still be used in these high BDP networks, and application that uses NTP may sometimes run on public networks. It distinguishes itself from the related work described in previous section in this major aspect:

- It is at the application level. This promotes a much better deployment method than in-kernel protocols including XCP and TCP variants. This also addresses many different research problems, especially in implementations.

The layered architecture of NTP is declared below, In this layered architecture, the NTP layer is completely in user space above the network transport layer of UDP, whereas the NTP layer itself provides transport functionalities to applications.

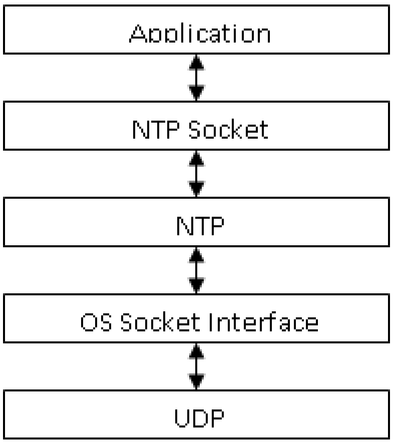

Figure : Layered architecture of NTP .

V. OBJECTIVES

I would like to design and implement a NTP, which recognized and addressed numerous research problems in data transport protocols. It aims to achieve the following specific contributions.

- It provides a timely and practical solution to the problem of transferring bulk data in high-speed wide area networks. It should be easily deployable. This is important as there are only four versions of TCP that get widely deployed in the past three decades because of the long time lag of standardization, implementation, and deployment of kernel space protocols.
  In addition, we aim to use bandwidth estimation techniques in NTP congestion control mechanism such that there is no need for manual tuning of the control parameters.
- Our work systematically investigated the design and implementation issues of high performance data transport protocol at the application level [15]. Although they were often neglected, protocol design and implementation have a significant impact on efficiency.
- We would like to use congestion control algorithm addresses both efficiency and fairness objectives [16].
- Finally, we would like to developed a productivity quality open source NTP library that can be used in real world applications and research work [108].

REFERENCES

[1] S. Floyd: HighSpeed TCP for large congestion windows. IETF, RFC 3649, Experimental Standard, Dec. 2003.

[2] T. Kelly: Scalable TCP: Improving performance in highspeed wide area networks. ACM Computer Communication Review, Apr. 2003.

[3] L. Xu, K. Harfoush, and I. Rhee: Binary increase congestion control for fast long-distance networks. IEEE Infocom '04, Hongkong, China, Mar. 2004.

[4] Lixia Zhang, Scott Shenker, David D. Clark: Observations on the dynamics of a congestion control algorithm: The Effects of two-way traffic. ACK SIGCOMM 1991, pp. 133-147.

[5] M. Gerla, M. Y. Sanadidi, R. Wang, A. Zanella, C. Casetti, and S. Mascolo: TCP Westwood: Congestion window control using bandwidth estimation. IEEE Globecom 2001, Volume: 3, pp 1698-1702.

[6] D. Katabi, M. Hardley, and C. Rohrs: Internet congestion control for future high bandwidth-delay product environments.ACM SIGCOMM '02, Pittsburgh, PA, Aug. 19 - 23, 2002

[7] R. N. Shorten, D. J. Leith: H-TCP: TCP for high-speed and long-distance networks. Proc. PFLDNet 2004, Argonne, IL, 2004.

[8] Sumitha Bhandarkar, Saurabh Jain, and A. L. Narasimha Reddy: Improving TCP performance in high bandwidth high RTT links using layered congestion control. Proc. PFLDNet 2005 Workshop, February 2005.

[9] Phillip M. Dickens: FOBS: A lightweight communication protocol for grid computing. Euro-Par 2003: 938-946.

[10] Yunhong Gu and R. L. Grossman: SABUL: A transport protocol for grid computing. Journal of Grid Computing, 2003, Volume 1, Issue 4, pp. 377-386.

[11] X. Zheng, A. P. Mudambi, and M. Veeraraghavan: FRTP: Fixed rate transport protocol -- A modified version of SABUL for end-to-end circuits. Pathnets2004 on Broadnet2004, Sept. 2004, San Jose, CA.

[12] E. He, J. Leigh, O. Yu, T. A. DeFanti: Reliable Blast UDP: Predictable high performance bulk data transfer. IEEE Cluster Computing 2002, Chicago, IL 09/01/2002.

[13] H. Sivakumar, S. Bailey, R. L. Grossman. PSockets: The case for application-level network striping for data intensive applications using high speed wide area networks. SC '00, Dallas, TX, Nov. 2000.

[14] Qishi Wu, Nageswara S. V. Rao: Protocol for high-speed data transport over dedicated channels. Third International Workshop on Protocols for Long-Distance Networks (PFLDnet 2005), Lyon, France, Feb. 2005.

[15] Yunhong Gu, Xinwei Hong, and Robert Grossman: Experiences in design and implementation of a high performance transport protocol. SC 2004, Nov 6 - 12, Pittsburgh, PA, USA.

[16] Yunhong Gu and R. Grossman. UDT: A transport protocol for data intensive applications. Internet Draft, work in progress.