IJCSN International Journal of Computer Science and Network, Volume 2, Issue 4, August 2013
ISSN (Online) : 2277-5420 www.ijcsn.org

A Dynamic Performance-Based Flow Control Method for High-Speed Data Transfer

[1] Umme Gousia, [2] Dr.Mohd.Abdul.Waheed, [3] Syed Shah Md Saifullah Hussaini

[1] P.G. Student, Department of Computer Science and Engineering
Khaja Banda Nawaz College of Engineering. Gulbarga, Karnataka, India

[2] Assistant Professor, Dept of Computer Science and Engineering
Khaja Banda Nawaz College Of Engineering, Gulbarga, Karnataka, India

[3] Software Engineer, GoldMan Sachs, Bangalore, Karnataka, India

Abstract

This paper develops a protocol, Performance Adaptive UDP (henceforth PA-UDP),which aims to dynamically and autonomously maximize performance under different systems. A mathematical model and related algorithms are proposed to describe the theoretical basis behind effective buffer and CPU management. A novel delay-based rate throttling model is also demonstrated to be very accurate under diverse system latencies. Based on these models, we implemented prototype under Linux, and the experimental results demonstrate that PA-UDP outperforms other existing high-speed protocols on commodity hardware in terms of throughput, packet loss, and CPU utilization. PA-UDP is efficient not only for high-speed research networks, but also for reliable high-performance bulk data transfer over dedicated local area networks where congestion and fairness are typically not a concern.

Keywords: *Flow control, high-speed protocol, reliable UDP, bulk transfer.*

1. Introduction

New types of specialized network applications are being created that need to be able to transmit large amounts of data across dedicated network links. TCP fails to be a suitable method of bulk data transfer in many of these applications, giving rise to new classes of protocols designed to circumvent TCP's shortcomings. It is typical in these high-performance applications, however, that the system hardware is simply incapable of saturating the bandwidths supported by the network infrastructure. When the bottleneck for data transfer occurs in the system itself and not in the network, it is critical that the protocol scales gracefully to prevent buffer overflow and packet loss. It is therefore necessary to build a high-speed protocol adaptive to the performance of each system by including a dynamic performance-based flow control. This paper develops such a protocol, Performance Adaptive UDP (henceforth PA-UDP), which aims to dynamically and autonomously maximize performance under different systems. A mathematical model and related algorithms are proposed to describe the theoretical basis behind effective buffer and CPU management. A novel delay-based rate throttling model is also demonstrated to be very accurate under diverse system latencies. Based on these models, we implemented a prototype under Linux, and the experimental results demonstrate that PA-UDP outperforms other existing high-speed protocols on commodity hardware in terms of throughput, packet loss, and CPU utilization. PA-UDP is efficient not only for high-speed research networks, but also for reliable high-performance bulk data transfer over dedicated local area networks where congestion and fairness are typically not a concern.

2. Related Work

The default implementations of Transmission Control Protocol (TCP) and UserDatagram Protocol (UDP) do not adequately meet these requirements. While several Internet backbone links havebeen upgraded to OC-192 and 10GigE WAN PHY, end users have not experienced proportional throughput increases. The weekly traffic measurements reported in [41] reveal that most of bulk TCP traffic carrying more than 10 MB of data on Internet2 only experiences throughput of 5 Mbps or less. For control applications, TCP may result in jittery dynamics on lossy links [37].

Currently, there are two approaches to transport protocol design: TCP enhancements and UDP-based transport with non-Additive Increase Multiplicative Decrease (AIMD) control. In the recent years, many changes to TCP have been introduced to improve its performance for high-speed networks. Efforts by Kelly have resulted in a TCP variant Called Scalable TCP [32]. High-Speed TCP Low Priority (HSTCP-LP) is a TCP-LP version with an aggressive window increase policy targeted toward high-bandwidth and long-distance networks.

The Fast Active-Queue-Management. PA-UDP falls under the class of reliable UDP-based protocols and like the others is implemented at the application layer. PA-UDP differentiates itself from the other high-speed reliable UDP protocols by intelligent buffer management based on dynamic system profiling

3. Architecture and Implementation

We discuss a generic architecture which takes advantage of the considerations related in the previous section. In the next three sections, a real-life implementation is presented and its performance is analyzed and compared to other existing high-speed protocols

3.1 Rate Control Algorithms

An optimum rate can be calculated so that the receiver will not run out of memory during the transfer. Thus, a target rate can be negotiated at connection time. We propose a simple three-way handshake protocol where the first SYN packet from the sender asks for a rate. The sender may be restricted to 500 Mbps, for instance. The receiver then checks its system parameters rðdiskÞ, rðrecvÞ, and m, and either accepts the supplied rate, or throttles the rate down to the maximum allowed by the system.

3.2 Processing Packets

Multithreading is an indispensable stepto decouple other processes which have no sequential liability with one another. Minimizing I/O and system call sand appropriately using murexes can contribute to overall efficiency. Thread priorities can often guarantee CPU attentiveness on certain kernel scheduler implementations. Also, libraries exist which guarantee high-performance, low-latency threads. Regardless of the measures mentioned above to curb latency, great care must be made to keep the CPU attentive to the receiving portion of theprogram. Even the resulting latencies from a single print statement inline with the receiving algorithm may cause the buildup and eventual overflow of the UDP buffer

4. Performance Analysis

4.1 Assumptions

We compared PA-UDP to three UDP-based protocols Tsunami, Hurricane, and UDT (UDT4).Five trials were conducted at each file size for both protocols using the same parameters for buffers and speeds. We used buffers 750 MB large for each protocol and generated test data both on-the-.fly and from the disk. The average throughputs and packetloss percentages are given in Tables 1 and 2, respectively, for the case when data were generated dynamically. The results are very similar for disk-to-disk transfers

Table 1: Throughput Averages

	Average Throughput / Std. Dev. (Mbps)			
File Size (MB)	PA-UDP	Tsunami	Hurricane	UDT
100	947.15 / 0.07	374.30 / 16.65	452.63 / 11.47	235.14 / 29.98
400	953.42 / 0.99	608.88 / 1.60	200.22 / 44.79	273.05 / 18.21
800	948.66 / 1.32	341.99 / 19.92	157.4 / 44.05	282.25 / 13.84
1000	938.51 / 10.94	294.96 / 11.51	145.08 / 71.49	295.90 / 11.57
2000	450.20 / 2.28	294.03 / 83.65	124.21 / 30.48	295.94 / 9.14
3000	371.01 / 10.43	timeout	87.11 / 4.27	246.78 / 6.46
5000	331.96 / 7.73	timeout	93.12 / 19.33	269.50 / 17.95

Table 2: Packet Loss Averages

	Average Packet Loss / Std. Dev. (%)			
File Size (MB)	PA-UDP	Tsunami	Hurricane	UDT
100	0.00 / 0.00	0.00 / 0.00	18.67 / 12.90	2.10 / 2.62
400	0.00 / 0.00	0.00 / 0.00	75.47 / 19.04	1.41 / 0.77
800	0.03 / 0.04	34.03 / 26.46	180.93 / 84.41	1.44 / 0.28
1000	0.01 / 0.02	41.48 / 21.40	122.18 / 107.95	0.46 / 0.57
2000	0.00 / 0.00	50.71 / 60.76	135.53 / 51.60	0.87 / 0.72
3000	0.00 / 0.00	time-out	230.83 / 31.95	0.00 / 0.00
5000	0.00 / 0.00	time-out	213.87 / 57.14	0.79 / 0.97

4.2 CPU Performance

One of the primary benefits of our flow control method is its low CPU utilization. The flow control limits the transfer speeds to the optimal range for the current hardware profile of the host. Other protocols without this type of flow control essentially have to “discover” the hardware-imposed maximum by running at a unsustainable rate, and then, reactively curbing throughput when packet loss occurs. In contrast to other high-speed protocols, PAUD maintains a more stable and more efficient rate.

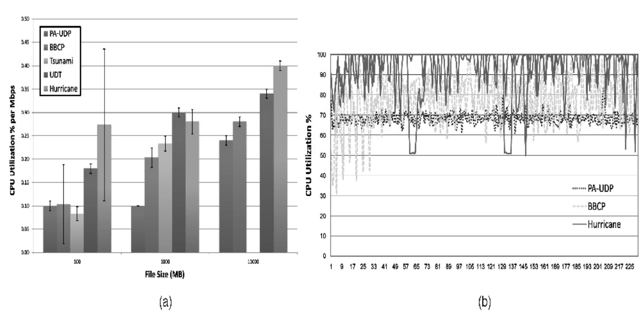

Fig.1. (a) Percentage CPU utilization per megabits per second for three file sizes: 100, 1,000, and 10,000 MB. PA-UDP can drive data faster at a consistently lower computational cost. Note that we could not get UDT or Tsunami to successfully complete a 10 GB transfer, so the bars are notshown. (b) A section of a CPU trace for three transfers of a 10 GB file using PA-UDP, Hurricane, and BBCP. PA-UDP not only incurs the lowest CPU utilization, but it is also the most stable.

5. Conclusion and Future Work

The protocol based on the ideas in this paper has shown that transfer protocols designed for high-speed networks should not only rely on good theoretical performance but also be intimately tied to the system hardware on which they run. Thus, a high-performance protocol should adapt in different environments to ensure maximum performance, and transfer rates should be set appropriately to proactively curb packet loss. If this relationship is properly understood, optimal transfer rates can be achieved over high-speed, high-latency networks at all times without excessive amounts of user customization and parameter guesswork.

In addition to low packet loss and high throughput, PAUDP has shown to be computationally efficient in terms of processing power per throughput. The adaptive nature of PA-UDP shows that it can scale computationally, given different hardware constraints. PA-UDP was tested against many other high-speed reliable UDP protocols, and also against BBCP, a high-speed TCP variant. Among all protocols tested, PA-UDP consistently outperformed the other protocols in CPU utilization efficiency.

The procedure presented in this paper is computationally inexpensive and can be added into existing protocols without much recoding as long as the protocol supports rate control via interpacket delay. Additionally, these techniques can be used to maximize throughput for bulk transfer on Gigabit LANs, where disk performance is a limiting factor. Our preliminary results are very promising, with PA-UDP matching the predicted maximum performance.

References

[1] N.S.V. Rao, W.R. Wing, S.M. Carter, and Q. Wu, "Ultrascience Net: Network Testbed for Large-Scale Science Applications," IEEE Comm. Magazine, vol. 43, no. 11, pp. S12-S17, Nov. 2005.
[2] X. Zheng, M. Veeraraghavan, N.S.V. Rao, Q. Wu, and M. Zhu, "CHEETAH: Circuit-Switched High-Speed End-to-End Transport Architecture Testbed," IEEE Comm. Magazine, vol. 43, no. 8, pp. 11- 17, Aug. 2005.
[3] On-Demand Secure Circuits and Advance Reservation System, http://www.es.net/oscars, 2009.
[4] User Controlled LightPath Provisioning, http://phi.badlab.crc. ca/uclp, 2009.
[5] Enlightened Computing, www.enlightenedcomputing.org, 2009.
[6] Dynamic Resource Allocation via GMPLS Optical Networks, http://dragon.maxgigapop.net, 2009.
[7] JGN II: Advanced Network Testbed for Research and Development, tp://www.jgn.nict.go.jp, 2009.
[8] Geant2, http://www.geant2.net, 2009.
[9] Hybrid Optical and Packet Infrastructure, http://networks.internet2.edu/hopi, 2009.
[10] Z.-L. Zhang, "Decoupling QoS Control from Core Routers:ANovel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services," Proc. ACM SIGCOMM '00, pp. 71-83, 2000.
[11] M.Khlaif, M.Talb, "Digital Data Security and Copyright Protection Using Cellular Automata", arXiv:1307.0082.
[12] N.S.V. Rao, Q. Wu, S. Ding, S.M. Carter, W.R. Wing, A. Banerjee, D. Ghosal, and B. Mukherjee, "Control Plane for Advance Bandwidth Scheduling in Ultra High-Speed Networks," Proc. IEEE INFOCOM, 2006. [12] K. Wehrle, F. Pahlke, H. Ritter, D. Muller, and M. Bechler, Linux Network Architecture. Prentice-Hall, Inc., 2004.
[13] S. Floyd, "RFC 2914: Congestion Control Principles," Category: Best Current Practise, ftp://ftp.isi.edu/in-notes/rfc2914.txt, Sept. 2000.
[14] V. Jacobson, R. Braden, and D. Borman, "RFC 2647: Tcp Extensions for High Performance," United States, http:// www.ietf.org/rfc/rfc1323.txt, 1992.
[15] A. Hanushevsky, "Peer-to-Peer Computing for Secure High Performance Data Cop," http://www.osti.gov/servlets/purl/ 826702-5UdHlZ/native/, Apr. 2007.
[16] R.L. Grossman, M. Mazzucco, H. Sivakumar, Y. Pan, and Q. Zhang, "Simple Available Bandwidth Utilization Library for High-Speed Wide Area Networks," J. Supercomputing, vol. 34, no. 3pp. 231-242, 2005.
[17] Y. Gu and R.L. Grossman, "UDT: UDP-Based Data Transfer for High-Speed Wide Area Networks," Computer Networks, vol. 51, no. 7, pp. 1777-1799, 2007.
[18] E. He, J. Leigh, O.T. Yu, and T.A. DeFanti, "Reliable Blast UDP: Predictable High Performance Bulk Data Transfer," Proc. IEEE Int'l Conf. Cluster Computing, pp. 317-324, http://csdl.computer.org/, 2002.
[19] M. Meiss, "Tsunami: A High-Speed Rate-Controlled Protocol for File Transfer," www.evl.uic.edu/eric/atp/TSUNAMI.pdf/, 2009.
[20] M. Goutelle, Y. Gu, and E. He, "A Survey of Transport Protocols Other than Standard tcp," citeseer.ist.psu.edu/he05survey.html, 2004.
[21] D. Newman, "RFC 2647: Benchmarking Terminology for Firewall Performance," www.ietf.org/rfc/rfc2647.txt, 1999.
[22] Y. Gu and R.L. Grossman, "Optimizing udp-Based Protocol Implementations," Proc. Third Int'l Workshop Protocols for Fast Long-Distance Networks (PFLDnet), 2005.

Umme Gousia - B.E in ISE in 2008, pursuing Mtech in Software Engineering, Worked as Lecturer prior to Mtech at K.C.T Polytechnic gulbarga. Research interest is in network and security.

Dr Mohd Abdul Waheed – Ph.d in Computer Science with specialization in Ad hoc networks. Working as Assistant Professor at Khaja banda Nawaz college of Engineering Gulbarga.

Syed Shah Md Saifullah Hussaini- B.E in ISE, currently working as Software Engineer at Goldman Sachs. Prior to this he had been with Hewlett Packard as Software test Engineer. Research interest is in Microsoft SharePoint technologies.