UDT: AN APPLICATION LAYER DATA TRANSPORT PROTOCOL

[1]ANSHU DALAL, [2]DEEPIKA

[1] Department of Computer Science & Engineering, IGI Sonepat, DCRUST Murthal, Sonepat, Haryana
[2] Assistant Professor, Department of Computer Science & Engineering, IGI Jhundpur, Sonepat

[1]anshudalal@gmail.com, [2]dhulldeepika11@gmail.com

ABSTRACT: *As high-speed wide area networks rapidly increase, new distributed applications come out and require new control mechanisms and large network bandwidth in data transport services to support them. In various data-intensive applications over high speed network such as computational grids TCP becomes inefficient due to need of large network bandwidth and delays in transmission. So we need a new transport protocol as UDT (UDP-based Data Transport Protocol) to support these requirements. In this paper I have covered the overview of UDT protocol, its architecture, key features, usage, applications and conclusion.*

KEY WORDS *: UDT, UDP, TCP, Transport Protocols, Congestion Control.*

1. INTRODUCTION

UDP-based Data Transfer Protocol (UDT), is a high performance data transfer protocol mainly designed for transferring large volumetric datasets over high speed WAN. UDT is an application level, connection oriented, duplex protocol that supports both reliable data streaming and partial reliable messaging. [8] UDT is built on top of UDP (i.e. User Datagram Protocol) by adding congestion control and reliability control mechanisms.

UDT's congestion control mechanism maintains efficiency, fairness and stability, and its application-level nature enables it to be deployed at the lowest cost, without any changes in the network infrastructure or operating systems.[4] UDT is at the application level and it is easy to deploy without the need for administrative privileges.UDT can support data transfer at very high speeds (e.g., 1 GB/s and above) with a single flow while sharing the network resources fairly with concurrent flows. [11] This is not feasible by using existing Internet data transport protocols (e.g., TCP). UDT 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. [13]

UDT has an open source implementation which can be found on Source Forge. It is one of the most popular solutions for supporting high speed data transfer and is part of many research projects and commercial products. [3]

2. THE UDT PROTOCOL

2.1 OVERVIEW

UDT is an application level, end-to-end, unicast, reliable, connection-oriented streaming data transport protocol.[11]The protocol is specially designed for efficient high-speed data transfer. UDT satisfies both the objectives of efficiency and fairness (including intra-protocol fairness, RTT fairness, and TCP friendliness). In addition, the UDT protocol is designed to be able to accommodate a large variety of control algorithms such that it can also be used as a protocol framework for the implementation and evaluation of new control algorithms.[8] Finally, UDT's default congestion control algorithm enables UDT to be used in shared network environments while realizing high throughput.

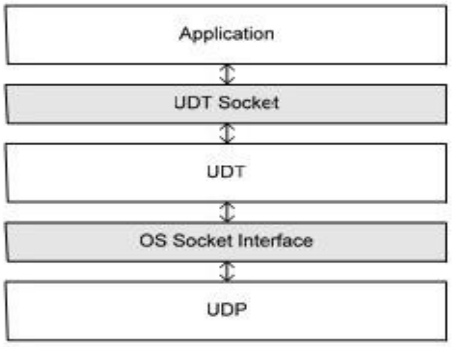

Fig 1: UDT in the Layer Architecture. *UDT is in the application layer above UDP. Application exchanges its data through UDT socket, which then uses UDP socket to send or receive the data. [11]*

2.2 ARCHITECTURE

UDT introduces a new three-layer protocol architecture that is composed of a connection flow multiplexer, enhanced congestion control, and resource management. UDT adapts itself into the layered network protocol architecture. [8]

The new design allows protocol to be shared by parallel connections and to be used by future connections. It improves congestion control and reduces connection set-up time. UDT provides better usability by supporting a variety of network environments and application scenarios. It addresses TCP's limitations by reducing the overhead required to send and receive streams of data. [2]

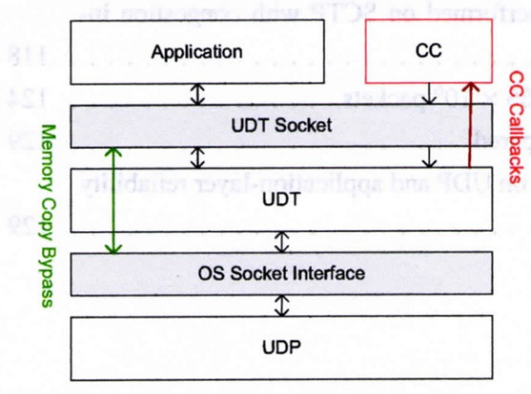

Fig 2: UDT/CCC Architecture. *In this layered architecture, the UDT layer is completely in user space above the network transport layer of UDP, whereas the UDT layer itself provides transport for applications*. [11]

UDT uses UDP through the socket interface provided by operating systems. Meanwhile, it provides a UDT socket interface to applications. Applications can call the UDT socket API in the same way they call the system socket API.[8] An application can also provide a congestion control class instance (CC in Figure 1) for UDT to process the control events, thus a customized congestion control scheme will be sued, and otherwise the default congestion control algorithm of UDT will be used.

Meanwhile, applications provide optional control handlers to UDT as callbacks**.** Memory copy is bypassed between UDT socket and UDP socket, in order to reduce processing time. Application can also provides a customized control scheme (CC). [11]

2.3 DESIGN

UDT is a connection-oriented duplex protocol, while it supports both reliable data streaming and partial reliable messaging. [11]

Figure 3 describes the relationship between the UDT sender and the receiver. In Figure 3, the UDT entity A sends application data to the UDT entity B. The data is sent from A's sender to B's receiver, whereas the control flow is exchanged between the two receivers.

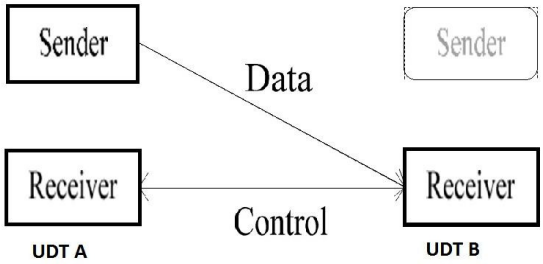

Fig 3: Relationship between UDT sender and receiver.

All UDT entities have the same architectures, each having both a sender and a receiver. This figure demonstrates the situation when a UDT entity A sends data to another UDT entity B. Data is transferred from A's sender to B's receiver, whereas control information is exchanged between the two receivers.[8]

The receiver is also responsible for triggering and processing all control events, including congestion control and reliability control, and their related mechanisms as well.

UDT uses rate-based congestion control (rate control) and window-based flow control to regulate the outgoing data traffic. Rate control updates the packet-sending period every constant interval, whereas flow control updates the flow window size each time and acknowledgment packet is received. There are two kinds of packets in UDT: the data packets and the control packets. [4]

A UDT data packet contains a packet-based sequence number, a message sequence number, and a relative timestamp (which starts counting once the connection is set up) in the resolution of microseconds in addition to the UDP header information. [11]

3. UDT CHARACTERISTICS

- High-speed: **-** It is designed for very high speed networks and it has been used to support global data transfer of terabyte sized data sets. [3]
- Fair and Friendly**: -** Concurrent UDT flows can share the available bandwidth fairly, while UDT also leaves enough bandwidth for TCP.[3]
- Simple to Use:- As UDT resides completely at the application level. Users can simply download the software and start to use it. No kernel reconfiguration is needed. In addition, UDT's API is very similar to the traditional socket API so that existing applications can be easily modified. [3,6]
- Highly Configurable: - UDT supports user defined congestion control algorithms with a simple configuration. Users may also modify UDT to suit various situations. This feature can also be used by students and researchers to investigate new control algorithms.[3,4]
- Firewall Friendly**: -** UDT is completely based on UDP, which makes it easier to traverse the firewall. In addition, multiple UDT flows can share a single UDP port, thus a firewall can open only one UDP port for all UDT connections. UDT also supports rendezvous connection setup.[3]

4. USAGE AND APPLICATIONS

- UDT is widely used in high performance computing area to support high speed data transfer over optical networks.[11]
- UDT has been used in many commercial products for fast file transfer using WAN, over the commodity Internet.
- This protocol is used in Cloud/Grid computing.
- It can transfer a data buffer from one single byte to multiple tera-bytes, as long as your system can provide enough resources.
- It is used as an Alternative Transport Protocol for GridFTP.
- It is used to achieve higher throughout rates for file transfers on high-speed networks with latency.
- UDT has been thoroughly tested for the disk-disk transfer.[3]
- UDT does work between different platforms even with different byte orders.[3]
- UDP-based protocol is one of the solutions to bulk data transfer in high BDP networks.
- It is also used for Peer-to-Peer applications, video/audio communication and many others.

5. CONCLUSION

UDT is widely used in high performance computing area to support high speed data transfer over optical networks. As UDT is purely based on UDP; it has also been used in many situations where TCP is at a disadvantage to UDP. It's a new generation protocol used for high speed networks as required by today's data-intensive applications.

REFERENCES

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

[2] S.Kishore,R.Chandra and D.Ganesh: Exploring Configurable Congestion Control Feature of UDT Protocol, International Journal of Computer Science and Telecommunications, Volume 3, Issue 5, May 2012

[3] http://udt.sourceforge.net.

[4] Yunhong Gu, Robert L. Grossman: Supporting Configurable Congestion Control in Data Transport Services, IEEE Computer Society,Nov 2005 ,Proceedings of the 2005 ACM/IEEE conference on Supercomputing

[5] T. Faber, A. Falk, J. Bannister, A. Chien, R. Grossman, and J. Leigh: Transport Protocols for High Performance: Whither TCP, Communications of the Association for Computing Machinery (CACM), 47(11), November 2003.

[6] Xinran (Ryan) Wu, Andrew A. Chien, Matti A. Hiltunen, Richard D. Schlichting and Subhabrata Sen.: A High Performance Configurable Transport Protocol for Grid Computing, In Proceedings of the 5th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2005).

[7] Yee-Ting Li: An Investigation into Transport Protocols and Data Transport Applications Over High Performance Networks, University College London ,JUL 2006

[8] Y.Gu and R.L. Grossman.UDT: A transport protocol for data intensive applications. The

International Journal of Computer and Telecommunications Networking, Volume 51 Issue 7,May 2007

[9] Yongmao Ren, Haina Tang, Jun Li and Hualin Qian, 2009: Performance Comparison of UDP-based Protocols Over Fast Long Distance Network. Information Technology Journal, 8: 600-604.

[10] Sunae Shin ,Kaustubh Dhondge, Baek-Young Choi: Understanding the Performance of TCP and UDP-based Data Transfer Protocols using EMULAB. The First GENI Research and Educational Experiment Workshop (GREE), Jointly with the 13th GENI Engineering Conference (GEC 13), Los Angeles, CA, March 2012.

[11] 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,

[12] Danilo Valeros ,Bernardo and Doan B Hoang: Securing Data transfer in the CLOUD through identification packet and UDT –authentication option field: a characterization, CoRR (2010),

[13] Y. Gu, RL Grossman: Udt: An application level transport protocol for grid computing. Second International Workshop on Protocols for Fast Long-Distance Networks,2003