Journal of Engineering and Applied Sciences 12 (Special Issue 6): 7755-7758, 2017
ISSN: 1816-949X
© Medwell Journals, 2017

Balancing between TCP and UDP to Improve Network Performance

Wameed Deyah and Wesam Bhaya
College of Information, University of Babylon, Technology, Babil, Iraq

Abstract: Now a days, Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) work as the backbone for almost everything can be done online in the internet. The main advantages of TCP are always guarantees three things which are the data arrives its destination correctly, the data reaches there in the correct order and the data arrives there without loss or duplication. The main advantage of UDP is that UDP is not a connection based communication model, thus UDP is much faster than TCP. This study proposes a new protocol which combines the advantages of TCP and UDP. In other words, we will avoid the disadvantages of TCP and UDP by using the proposal protocol and as a result improving the performance of the network from the side of data transmission speed and integrity. Therefore, ensuring the arrival of packets in the correct order and avoiding the unnecessary connection between source and destination before any transmission. We fine transmission parameters such as correct delay time before any transmission depending on the efficiency of the path between source and destination. Thus, we got that the proposed protocol has a significant impact on the network traffic compared with standard TCP and UDP.

Key words: Transmission Control Protocol (TCP), UDP, disadvantages, network, significant, Iraq

INTRODUCTION

Most of the conversations today on the internet use Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) as the main transport protocols. TCP provides many important services such as reliable data transfer and congestion control (2017). Unfortunately, TCP is not appropriate for the short conversations. The overhead of creating and finalize TCP state effects on these connections. Moreover, there are a big number of TCP connections and this will effect on the travelling of the data (Mahmood *et al*., 2010).

UDP is another important protocol for short transfers, while TCP is used for all other transfers (Yadav *et al*., 2014). In this study, we designed a proposed scheme to address these issues. In other words, we will exploit the advantages of using TCP such as reliability with correct order of data in the destination and the advantages of using UDP such as the speed of sending data.

The proposed protocol will be used for sending messages (text or images) after converting them to bytes. We will show how to ensure the reliability without unnecessary connections which can be used in the TCP protocol and how to ensure that there are no lost or dropped packets. The important aspect in proposed protocol is how to provide the required delay before sending any packet depending on the quality of path between the source and destination. The proposed protocol will use the UDP protocol actually to ensure the speed in sending packets and using the advantages of TCP such as reliability and avoiding the disadvantages of using TCP such as the big number of connections or unnecessary delay.

Transmission Control Protocol (TCP): TCP forms the backbone for almost everything can be done on the internet from web browsing to email (Hashemi, 2015). If the TCP socket is used, then this is means that it is a connection based protocol which provide the reliability aspect. A connection must establish between two computers and then it can send the data between the two computers (Comer, 2000).

TCP connection must be reliable and ordered, it means that all the transmitted data is guaranteed to reach at the destination in the same order that you wrote it at the source side. These data is also a stream of data, this means that TCP responsible for splitting up the data into packets and sending them across the network (Comer, 2000).

MATERIALS AND METHODS

User Datagram Protocol (UDP): UDP is a protocol built on the top of Internet Protocol (IP), it is like TCP but instead of inserting many features and complexity, UDP is a very thin layer over IP (GGNPC, 2008).

With UDP, it can transmit a packet to a destination with the IP address (e.g., 123.123.123.123) and with the port (51976) and the data will pass from node to node until it reaches at the destination or it is lost or dropped along the way (Forouzan and Fegan, 2010).

Corresponding Author: Wameed Deyah, College of Information, University of Babylon, Technology, Babil, Iraq

On the receiver or destination side, we will listen for a specific port (e.g., 51976) and when a packet reaches from any source (note that there are no connections), we will notify the address and port of the computer that sent the packet, the packet's size and will read the data packet. UDP is not a reliable protocol. In practice, most packets that are sent will passed but usually have around 1-5% packet loss (Forouzan and Fegan, 2010).

There is no guarantee of packet's ordering. For example we could send 4 packets in the order 1, 2, 3, 4 and they could arrive at the destination completely out of order like 3, 1, 2, 4 (Hashemi, 2015).

TCP vs. UDP: The main advantages of TCP are: TCP can guarantee three important things which are the data arrives its destination, it arrives there in time and it arrives there without duplication (Yadav and Bansal, 2014). The main disadvantages of TCP are:

- We cannot use TCP for broadcast and multicast connections
- TCP is slower in working than UDP

The main advantages of UDP are:

- The connections of broadcast and multicast are available with UDP which is not available in TCP (Cidon *et al*., 1999)
- Does not require the existence of connections while using UDP (Cidon *et al*., 1999)
- UDP much faster than TCP

The main disadvantages of UDP are there are no guarantees with UDP. It is possible that a one or more than one packets may not be delivered at the destination side or delivered twice or more than twice or delivered not in time (Yadav and Bansal, 2014). UDP must manually break the data into packets (Yadav and Bansal, 2014).

Proposal work: The internet today is a packet-based network and highly decentralized network. The design of the internet today is aiming to minimize the amount of higher layer information and the number of connections which need to be kept between the sources and destination. When coupled with the decentralized structure of the internet, this has created major challenges for network managers of the IP networks (Narayan, 2014). To overcome these challenges, we propose a new protocol uses the same work mechanism of UPD and the advantages of TCP. By using the proposed protocol in this paper we will guarantee the arrival of data at the destination in the correct order and without any loss

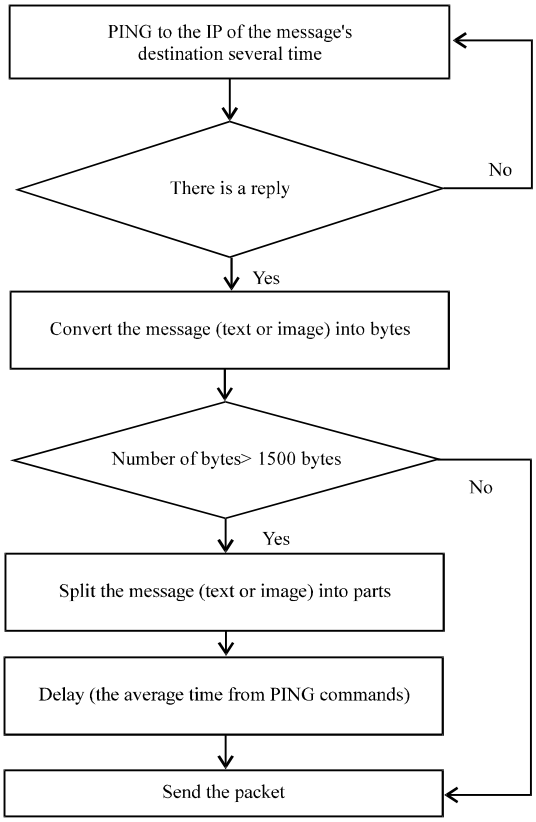

Fig. 1: The proposal work flowchart

(advantages of TCP) without creating any connection between the source and the destination. In other words, this protocol will send the data with the required speed (advantages of UDP) and the delay before sending any data will be depending on the efficiency of the path between the source and the destination.

One of the most important problems in the network is routing (Sugeng *et al*., 2015). By using the proposed protocol we will achieve better performance by providing better routing for the data and higher efficiency in today's complex network. Figure 1 shows the work's steps of the proposal protocol which can be summarized as follow.

Step 1: Make a ping command to the IP of the message's destination several times (not >5 times) because usually there is no replay in the first transmitted packets.

Step 2: If there is a replay by using ping command, get the time spent to send and receive the packet in ping command as main parameter otherwise repeat the ping command.

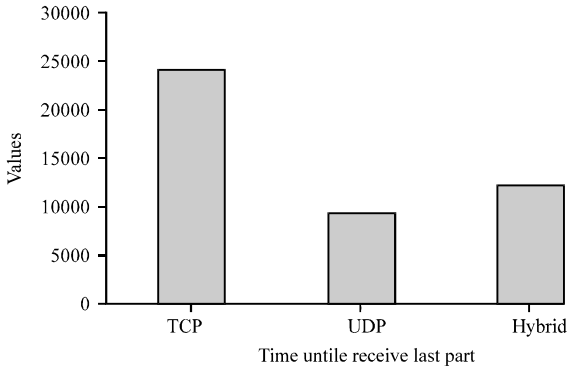

Fig. 2: An illustration of the time consumed to send image with the size 162 kb using TCP, UDP and the proposal protocol

Step 3: Convert the message (text or image) into bytes.

Step 4: If the number of bytes is >1500 bytes, the proposed protocol will split the message (text or image) into parts (the size of each part must not exceed 1500 bytes) because the smaller message is sent faster than the bigger message. And then insert the current part number after converting it to bytes to the end of the part, in other words, insert for example (2 of 12) in the end of the part to ensure that this part is part 2 of the total number of parts 12 (this is an important step to provide the reliability). Otherwise send the message without any splitting.

Step 5: Using the time parameter (the average time from ping command) which always be in milliseconds as the amount of delay before sending any packet. This step is very important to ensure the arriving of the message or the part of the message to the destination and avoiding the data overflow problem. This time is the expected time which can be used by the packet to reach to the destination.

RESULTS AND DISCUSSION

To compare between the results, we sent an image with the size 162 kb using TCP, UDP and the proposed protocol. After sending the image we can notice that the time consumed to send the parts of the message (image) from he first part to the last part is 24000 msec using TCP, 9000 msec using UDP and 12000 msec using the propose protocol (Fig. 2).

From the side of reliability, we can notice that the received parts of the message (image) is 120 parts using TCP, 118 parts using UDP and 120 parts using the proposed protocol (the proposed protocol) of the total

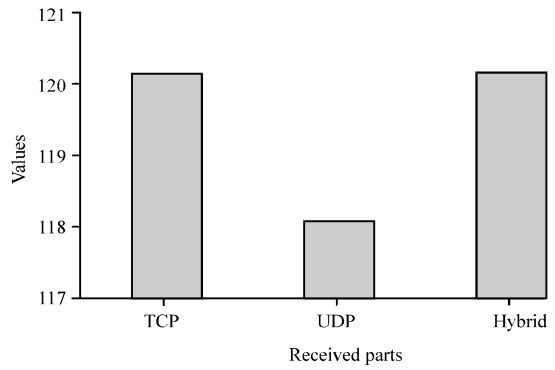

Fig. 3: An illustration of the received parts of an image with the size 162 kb using TCP, UDP and the proposed protocol

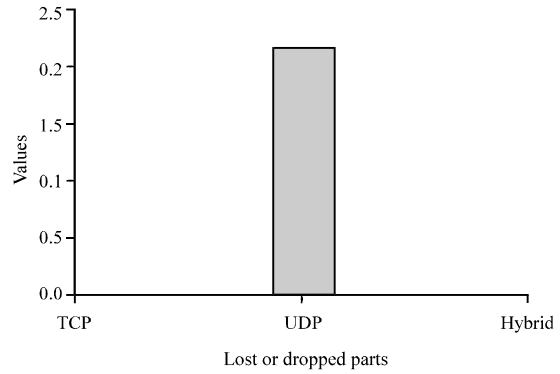

Fig. 4: An illustration of the lost or dropped parts of an image with the size 162 kb using TCP, UDP and the proposed protocol

parts of the message (image) which is 120 parts as in Fig. 3. In other words, we can notice that the lost or dropped parts of the message (image) is 0 parts using TCP, 2 parts using UDP and 0 parts using the proposed protocol (the proposed protocol) as in Fig. 4.

CONCLUSION

The most commonly protocols used in the internet today are TCP and UDP, each of them has advantages which can provide flexible network traffic and disadvantages which can effect on the network traffic. In this study, we summarized these advantages and disadvantages and compare between these protocols. This study proposed a new protocol using the advantages of TCP and UDP and as a result this proposed protocol can provide flexible network traffic. In other words, the proposed protocol can improve the performance of the network.

REFERENCES

Cidon, I., A. Gupta, R. Rom and C. Schuba, 1999. Hybrid TCP-UDP transport for Web traffic. Sun Microsystems, Mountain View, California. http://dl.acm.org/citation.cfm?id=974975

Comer, D.E., 2000. Internetworking with TCP/IP. 4th Edn., Prentice Hall, New Jersey, USA.,.

Forouzan, B.A. and S.C. Fegan, 2010. TCP-IP Protocol Suite. McGraw-Hill Education, New York, USA., ISBN:9780256241662, Pages: 836.

GGNPC., 2008. UDP vs. TCP: Which protocol is best for games?. Gaffer On Games, Network Protocol Company, Santa Monica, California. https://gafferongames.com/post/udp_vs_tcp/.

Hashemi, S.M., 2015. Performance evaluation of network-on-chip routing deterministic and adaptive algorithms. Intl. J. Comput. Sci. Netw. Solutions, 3: 21-25.

Mahmood, A.N., C. Leckie, J. Hu, Z. Tari and M. Atiquzzaman, 2010. Network Traffic Analysis and SCADA Security. In: Handbook of Information and Communication Security, Stavroulakis, P. and M. Stamp (Eds.). Springer, Berlin, Germany, ISBN:978-3-642-04116-7, pp: 383-405.

Narayan, S., 2014. Improving network performance: An evaluation of TCP-UDP on networks. MS Thesis, Unitec Institute of Technology, Auckland, New Zealand.

Sugeng, W., J.E. Istiyanto, K. Mustofa and A. Ashari, 2015. The impact of QoS changes towards network performance. Intl. J. Comput. Netw. Commun. Secur., 3: 48-53.

Yadav, S. and V. Bansal, 2014. Hybrid TCP-IP and UDP: A review article. Intl. J. Adv. Res. Comput. Sci. Software Eng., 4: 953-957.

Yadav, S., S. Yogi and R. Yadav, 2014. Proxy server for Hybrid TCP-IP and UDP. Intl. J. Comput. Sci. Mobile Comput., 3: 825-829.