JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 8(2), JUNE, 2020
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com

Packet Tracing and Analysis of TCP Traffic on Transport Layer of Peer to Peer Networks

Ahmad Musa
Department of Software Engineering,
Bayero University Kano.

ABSTRACT
Peer Networks (P2P) is presenting a lot of fascinating capabilities to the internet since its inception, which has made and is still making it gather so much interest. As a result, it is being used in many domains, particularly in the transfer of a large amount of data, which is essential for modern computing needs. A P2P network contains a diverse number of independent nodes to form a highly distributed system. These nodes are used to exchange all kinds of files without the use of a single server as in a Client-Server architecture. Such types of files make the network highly vulnerable to malicious attackers. Nevertheless, P2P systems have become susceptible to different malicious attacks due to its widespread usage, including the threat of sharing malware and other harmful programs, which can be significantly damaging and harmful. In this study, we approach the harmful exchange of malicious programs with network monitoring and traffic analysis using Wireshark to secure the network. We analysed how TCP traffic in the transport layer can be tracked to ensure accurate analytics and processing. We also focused on network traffic analysis of a sample file of interest to monitor, collect, and analyze the traffic on the network's transport layer.

ARTICLE INFO
Article History
Received: February, 2020
Received in revised form: April, 2020
Accepted: April, 2020
Published online: June, 2020

KEYWORDS
TCP, Wireshark, Peer to Peer, Traffic, Network

INTRODUCTION

In 2020, Cisco estimated that P2P file sharing accounted for 3,807 petabytes of global internet file sharing, or 59.6% of the total usage. Cisco forecast that P2P traffic would increase from 3807 petabytes to 3858 petabytes in 2021, showing a steady increase (Cisco, 2017). Cisco notes that the large portion of the P2P traffic due to the exchange are video files. The total view of the impact of video on the network counts as P2P video traffic and the traffic count in the Internet video-to-TV and Internet video-to-PC categories. While the volume of P2P traffic is set to keep increasing, its proportion of total internet traffic is set to decrease due to the rising popularity of content streaming services such as Netflix, Apple music, Amazon Prime, etcetera. BitTorrent is the most popular P2P protocol used worldwide and accounts for the biggest proportion of Internet traffic when compared to other P2P protocols (Musa et al., 2019).

The BitTorrent network provides the ability to transfer files among many users quickly and efficiently with minimal load on the original file source. The uploading user (the seed) makes one copy of the file available, which is then broken up into chunks by the application. Different chunks are sent out to the multiple computers (BitTorrent user), trying to download a copy of the file by other users (leeches) (Neuner, 2016). Each user uploads their portion of the file to other users while simultaneously downloading bits of the file they don't have from other users. Due to the BitTorrent ease of use, decentralized nature, and minimal bandwidth

Corresponding author: Musa, A. asmusa.se@buk.edu.ng Dept. of software Engineering, Bayero University, Kano, Nigeria.

requirements, it lends itself as a suitable platform for the unauthorized distribution of malware and copyrighted material, which typically commences with a single source user sharing large sized files to many downloaders (Yu, 2010).

In computer networks, data communication, working on the 4th layer of the Open Systems Interconnect (OSI) reference model, takes place through Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) in the form of packets (Wararkar et al., 2016). UDP is a connectionless protocol and, as such, is preferred in real-time data transmissions such as voice and video transmissions over WANs and chat applications, just like with skype (Venckauskas et al., 2015). However, UDP lacks some essential security features that are supported by the connection-oriented protocol, TCP. Because of UDP's aim of increasing the speed of data transfer, the data transmission time is reduced by not performing connection establishment, retransmissions mechanism, and flow control (Liberatore et al., 2010). When data transmission over a public network is completed, capturing network communication packets can be used to reveal a lot of information, such as the applications used by the senders, web address, and location (Kim et al., 2010). Nevertheless, the accuracy of the information is variable and depends on the methodology used for the packet analysis and the third-party software used by the sender/receiver. Hence, there is a need to measure the accuracy of the information revealed through network traffic analysis (Alhazmi et al., 2017).

Network traffic analysis is a valuable tool for network administrators that are set out to investigate BitTorrent traffic. Previous works have shown that BitTorrent traffic is often not only encrypted by a Virtual Private Network (VPN) but also mixed with other types of network traffic (including video streaming and web traffic) (Shi & Biswas, 2017). This makes the identification of a said file more challenging. Packet sniffing and capture can be a useful tool for pre-filtering and the actual classification of the traffic (Musa et al., 2019). Another function of a packet sniffer is the ability to measure data that may be useful in modelling p2p systems and identifying design issues of a BitTorrent file that has the right mechanisms to attract a large user's attention (Balakrishnan et al., 2005). Researchers have also found that the BitTorrent network protocol offers anonymity through which malicious files can be exchanged (Bin & Jiangtao, 2016).

Understanding the existing anonymous techniques of the protocol is a crucial step in developing countermeasures (Leong et al., 2010). The detection of the users participating in the network and enabling malicious activities are challenging but need to be addressed to ensure the security of BitTorrent networks (Schafer, 2008). Therefore, this article investigates the usability and reliability of Wireshark for deep packet analysis on the transport layer of p2p networks. The technique of packet analysis has been used on other types of networks but to the best of our knowledge, this is the first time we are testing its usability on p2p networks.

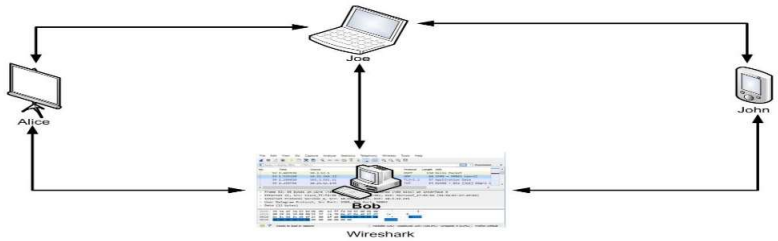

Fig. 1: Internet Traffic Capture Model on P2P networks

Corresponding author: Musa, A. asmusa.se@buk.edu.ng *Dept. of software Engineering, Bayero University, Kano, Nigeria.*

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 8(2), JUNE, 2020
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com

EXPERIMENTAL PROCEDURE

For this experiment we secured a Laptop PC running Windows 10 Home 64-bit Operating System, with Installed memory (RAM) of 8.00 GB, 1 TB Hard Disk Drive and Processor details: Intel (R) Core (TM) i5- 3230M CPU @ 2.60GHz. We also obtained the latest version of Wireshark available v2.6.8 (Wireshark, 2016) and uTorrent v1.8.8. uTorrent is a BitTorrent client that allows the sharing and downloading of torrent files (uTorrent, 2018). We chose uTorrent because it is the most famous and widely used torrent that is responsible for saving metadata used for BitTorrent.

To illustrate how the model presented in Fig. 1 works, we performed an experiment to allow us to monitor and capture p2p traffic for analysis. At first, we downloaded and installed the latest versions available of Wireshark and uTorrent programs. When a Network Interface Card (NIC) receives a packet, it first compares the MAC address of the packet to its own if it matches, then it accepts the packet otherwise filters it. This is due to a non-promiscuous operation that sets the network card to discard all the packets that do not carry its MAC address, which means that each network card is only reading the frames directed to it. NIC then has to be set in the promiscuous mode in order to capture packets (Dabir & Matrawy, 2007).

Wireshark does sniffing by setting the NIC card of its system to promiscuous mode and, hence, receiving all available packets communicated on the network. So, Wireshark captures the packets by setting the NIC into promiscuous mode. A packet passes through many intermediate nodes when the packets are sent from one node to another in the network (Bauer et al., 2009). A node whose NIC is set in the promiscuous mode then receives the packet. In the transport layer of the OSI reference model, data communication is done through nodes using TCP and UDP protocols (Kun & Lu, 2013). Hence, we set up our NIC to especially capture TCP packets.

RESULTS AND DISCUSSION

The communication behaviour of uTorrent traffic is very intricate. We used Wireshark to sniff and capture uTorrent traffic by enabling the NIC on our laptop to be in promiscuous mode for TCP filtering. The NIC checks the MAC address of every incoming packet against its own. The filtered traffic is then saved on the memory of our laptop as one of the nodes participating in the data exchange, as illustrated in Fig. 1. The accepted packets from the physical layer, data link layer, and the transport layer are captured in a packet capture (.pcap) format (Musa et al., 2019). Wireshark communicates with the system to initiate the capture of all the applications running on the system, which is why we made sure only our experiment was running. TCP stands to be the core protocol of the OSI reference model because of its connection-oriented ability (Leong et al., 2010). This ability allows it to establish connections using a three-way handshake, alongside error and flow control mechanisms.

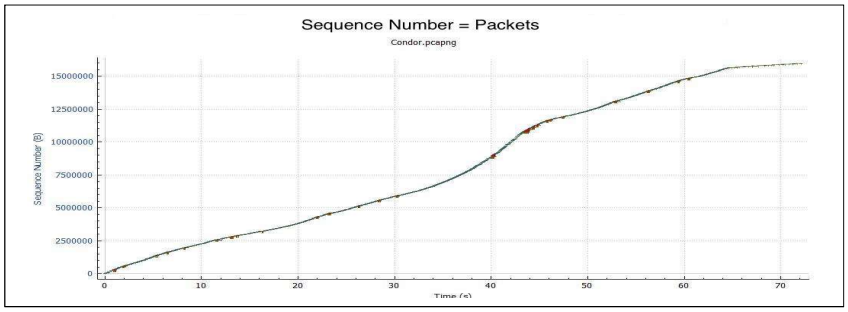

Fig. 2: TCP Time Sequence Graph

Corresponding author: Musa, A. asmusa.se@buk.edu.ng *Dept. of software Engineering, Bayero University, Kano, Nigeria.*

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 8(2), JUNE, 2020
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com

Wireshark records TCP sequence numbers over time and is represented by a simple graph known as TCP time sequence graph. During handshake, the bytes that gets exchanged are counted as TCP sequence numbers. These bytes are packets represented various applications that includes application headers sent from one side to another. The behaviour of the packets is visible in the graph in Fig. 2. The data seems to be in a perfect diagonal with no visible break anywhere on the line and a high gradient, thereby indicating no interruptions and a successful transfer of data (Das & Tuna, 2017). A diagonal line with a high gradient in a time sequence graph is an indication of a fast data transfer.

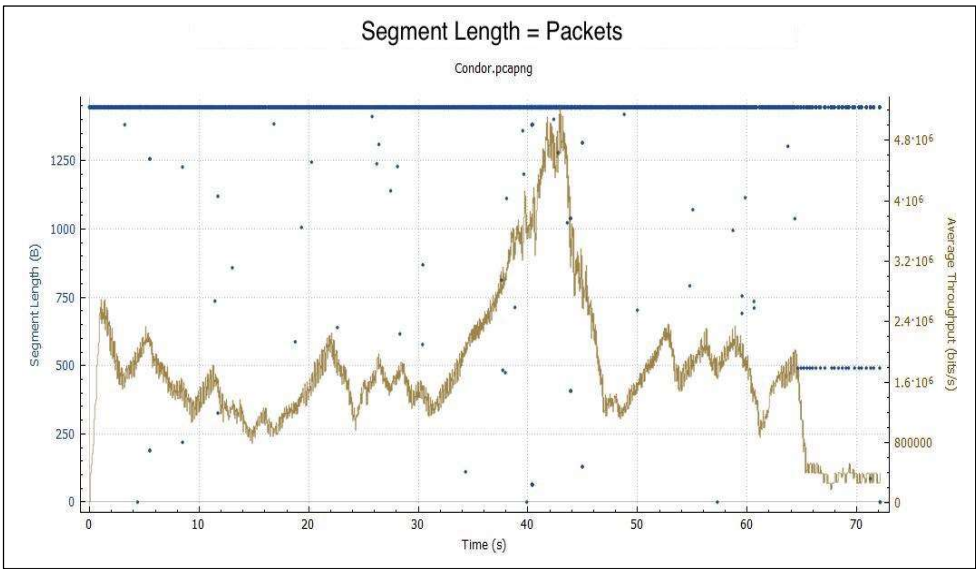

Fig. 3: TCP Throughput Graph

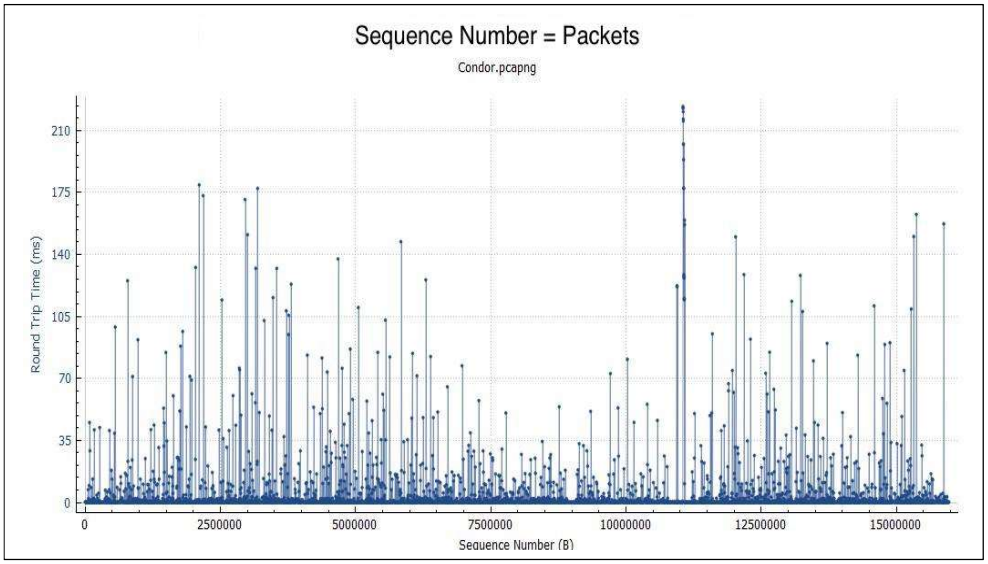

Fig. 4: TCP round trip time graph performance

Corresponding author: Musa, A. asmusa.se@buk.edu.ng Dept. of software Engineering, Bayero University, Kano, Nigeria.

The TCP round trip time graph performance, visible in fig. 4, represents the round trip between packets sent and the time they were acknowledged. In the graph, it can be observed that most of the sequence numbers were acknowledged (ACK) in a very short time and consistently shows the stability of the sequence that influences the TCP performance (Gao & Zhai, 2016). The result of the graph shows the period when a packet is sent, and the ACK received. The graph might illustrate instability, but this does not mean there is a problem because the time of communication is presented in milliseconds and also represents how the uTorrent application works. A typical example is when a node begins participating in the network but then drops or fluctuates due to network connection or another problem (Liberatore et al., 2010). The roundtrip time of that node will be more than the average expected roundtrip time.

CONCLUSION

Accurate monitoring of P2P traffic has grown into an essential aspect of Internet traffic modelling. In this study, we demonstrated how network communication is established using TCP between source and destination ports through the reliable three-way handshake. We also showed various ways to ascertain if the transmission was error-free and stable through network traffic analysis with the help of the Wireshark network sniffer. The sniffer helped us analyze the time sequence graph, throughput graph, and the roundtrip time graph of the completed data communication. Through these graphs, a case study of a p2p session was realized and found to be reliable and secure. A further avenue of research would be to perform a comprehensive analysis by comparing the results of TCP and UDP reliabilities and securities in the transport layer.

REFERENCES

Alhazmi, A., Maciá-Fernández, G., Camacho, J., & Salah, S. (2017). *Torrent Forensics: Are your Files Being Shared in the BitTorrent Network?* The Second International Conference on Cyber-Technologies and Cyber-Systems.

Balakrishnan, M., Haridasan, M., Linga, P., Liu, H., Ramasubramanian, V., Rhea, S., Singh, M., Venkatraman, V., Vishnumurthy, V., Walsh, K., Wong, B., & Zhong, M. (2005). *The bittorrent P2P file-sharing system: Measurements and analysis* (pp. 1–12). https://doi.org/10.1007/11558989_1

Bauer, K., McCoy, D., Grunwald, D., & Sicker, D. (2009, December 1). *BitStalker: Accurately and efficiently monitoring bittorrent traffic*. IEEE Xplore. https://doi.org/10.1109/WIFS.2009.5386457

Bin, G., & Jiangtao, Z. (2016). A survey of covert channels in BitTorrent network. *IJISET - International Journal of Innovative Science, Engineering & Technology*, *3*, 102–106. http://ijiset.com/vol3/v3s9/IJISET{_}V3{_}I9{_}16.pdf

Cisco, W. P. (2017). *Cisco visual networking index: Forecast and methodology cisco visual networking index: Cisco visual networking index: Forecast and methodology* (pp. 1–17). https://www.reinvention.be/webbhdfs/v1/docs/complete-white-paper-c11-481360.pdf

Dabir, A., & Matrawy, A. (2007, November 1). *Bottleneck Analysis of Traffic Monitoring using Wireshark*. IEEE Xplore. https://doi.org/10.1109/IIT.2007.4430446

Corresponding author: Musa, A. asmusa.se@buk.edu.ng Dept. of software Engineering, Bayero University, Kano, Nigeria.

Das, R., & Tuna, G. (2017). Packet tracing and analysis of network cameras with Wireshark. *2017 5th International Symposium on Digital Forensic and Security (ISDFS)*. https://doi.org/10.1109/isdfs.2017.7916510

Gao, B., & Zhai, J. (2016). A Survey of Covert Channels in BitTorrent Network. *IJISET -International Journal of Innovative Science, Engineering & Technology*, *3*(9).

Gomes, J. V. P., Inacio, P. R. M., Freire, M. M., Pereira, M., & Monteiro, P. P. (2008, December 1). *Analysis of Peer-to-Peer Traffic Using a Behavioural Method Based on Entropy*. IEEE Xplore. https://doi.org/10.1109/PCCC.2008.4745138

Kim, S., Wang, X., Kim, H., Kwon, T., & Choi, Y. (2010). Measurement and Analysis of BitTorrent Traffic in Mobile WiMAX Networks. *2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P)*. https://doi.org/10.1109/p2p.2010.5569997

Kun, H., & Lu, W. (2013). Research of Trust Model Based on Peer-to-Peer Network Security. *IEEE Xplore*, 126–129. https://doi.org/10.1109/ITA.2013.35

Leong, R. S. C., Lai, P. K. Y., Chow, K. P., Kwan, M. Y. K., & Law, F. Y. W. (2010). *Forensic Investigation of Peer-to-Peer Networks*. Handbook of Research on Computational Forensics, Digital Crime, and Investigation: Methods and Solutions. https://www.igi-global.com/chapter/forensic-investigation-peer-peer-networks/39225

Liberatore, M., Erdely, R., Kerle, T., Levine, B. N., & Shields, C. (2010). Forensic investigation of peer-to-peer file sharing networks. *Digital Investigation*, *7*, S95–S103. https://doi.org/10.1016/j.diin.2010.05.012

Musa, A., Abubakar, A., Gimba, U. A., & Rasheed, R. A. (2019). An Investigation into Peer-to-Peer Network Security Using Wireshark. *2019 15th International Conference on Electronics, Computer and Computation (ICECCO)*. https://doi.org/10.1109/icecco48375.2019.9043236

Musa, A., Almohannadi, H., & Alhamar, J. (2018, August 1). *Malware Propagation Modelling in Peer-to-Peer Networks: A Review*. IEEE Xplore. https://doi.org/10.1109/W-FiCloud.2018.00038

Neuner, S., Schmiedecker, M., & Weippl, E. R. (2016). PeekaTorrent: Leveraging P2P hash values for digital forensics. *Digital Investigation*, *18*, S149–S156. https://doi.org/10.1016/j.diin.2016.04.011

Schäfer, J., Malinka, K., & Hanácek, P. (2008, June 1). *Peer-to-Peer Networks Security*. IEEE Xplore. https://doi.org/10.1109/ICIMP.2008.26

Shi, Y., & Biswas, S. (2017). *Using traffic analysis for simultaneous detection of BitTorrent and streaming video traffic sources*. 79–86. https://doi.org/10.1109/COMSNETS.2017.7945361

uTorrent. (2018). *µTorrent - a BitTorrent client*. Utorrent.Com. https://www.utorrent.com/

Venčkauskas, A., Damaševičius, R., Jusas, N., Jusas, V., Maciulevičius, S., Marcinkevičius, R., Paulikas, K., & Toldinas, J. (2015). Investigation of Artefacts Left by BitTorrent Client in Windows 8 Registry. *Information Security*

Corresponding author: Musa, A. asmusa.se@buk.edu.ng Dept. of software Engineering, Bayero University, Kano, Nigeria.

and Computer Fraud, *3*(2), 25–31. https://doi.org/10.12691/iscf-3-2-1

Wararkar, P., Kapil, N., Rehani, V., Mehra, Y., & Bhatnagar, Y. (2016). Resolving problems based on peer to peer network security issues. *Physics Procedia*, *78*, 652–659. https://doi.org/10.1016/j.procs.2016.02.113

Wireshark. (2016). Wireshark.Org. https://www.wireshark.org/

Yu, B. (2010). Based on the network sniffer implement network monitoring. *IEEE Xplore*, *7*, V7–1-V7-3. https://doi.org/10.1109/ICCASM.2010.5620505

Corresponding author: Musa, A. asmusa.se@buk.edu.ng Dept. of software Engineering, Bayero University, Kano, Nigeria.