Analysis of Business Intelligence Systems Transmission Session

Abdul Shabir Ahmadzai, Qurban Yazdani, and Adamu Abubakar [1]

[1] Department of Computer Science, International Islamic University Malaysia, Kuala Lumpur, Malaysia. Email: adamu@iium.edu.my

https://doi.org/10.20428/jst.v27i1.1984

Abstract:

Information and Communication Technology (ICT) is the central part of Intelligent systems. Business Intelligence (BI) utilizes ICT resources and capabilities to enhance decision-making from the available information. One of the core components of ICT resources is the network. Hypertext Transfer Protocol (HTTP) is an application layer protocol in a network model responsible for handling "requests" in a transmission session. Transmission Control Protocol (TCP) establishes connections session and keeps track of the state of the communication session. Unfortunately, a request from a critical online transaction over TCP in a BI environment is bound to face some uncertainty. There is a lack of studies on the impact of communication over transmission sessions on BI operations. This paper presented an experimental evaluation of the BI transaction transmission session. Various transactions were performed to monitor and analyze the request made with HTTP over a TCP transmission session. Both the HTTP and TCP features for all transactions were captured. The finding indicates that time is highly associated with less traffic congestion and transaction overhead.

Keywords: Business Intelligence; ICT resources; TCP; HTTP

https://doi.org/10.20428/jst.v27i1.1984

1. Introduction

Business intelligence (BI) refers to the combinations of ICT resources and ICT capabilities necessary for analyzing information to improve and optimize decisions and performance. BI gives organizations better access to information for decision-making purposes [1]. ICT resources are a combination of hardware, software, data, and network. On the other hand, ICT capabilities refer to human capabilities necessary to operate the ICT resources. Since "network" is an essential component of ICT resources required for BI operation, this paper analyzes BI transmission sessions to evaluate the behavior of the operation of the BI system on a network.

Capturing and analyzing packets over a BI transmission session is very important. The justification of the work lies with the main challenge faced in processing the data stream on-the-fly as it arrives from one BI operation to the other. The key factor involved while adopting computing those data for accessing its path and location is the protocol. BI offers long-term data analytics for wide connectivity. These connections are real-time and require robust synchronization and coordination. Hence, the major protocols governing the connection and transmission (HTTP/TCP) through each BI operation should analyze the data at rest and in motion. That is attributed to the added information in their header, as there is much essential information in the HTTP header [2-4]. That gives many researchers an edge in analyzing the header components in various network transaction environments. In general, HTTP header components can be classified by using a machine-learning algorithm [2]. The Content-Type field is one of the components of the HTTP header. It is usually found in the first several inbound packets of a new HTTP flow, so it is possible, based on condition, to differentiate different HTTP streams. Different transmission sessions on various transactions could show different behavior of this component. The relationship between Content-Type and traffic type has been investigated with the C5.0 algorithm to classify the traffic. An integrated analysis method on HTTP traffic, in which utilizing the downsides of previous HTTP traffic analysis studies and attends analysis for each site at the HTTP server side, as well as for application levels at the client side, has been presented in [4].

Some of the HTTP messages are encrypted. Thus, the HTTP messages often do not include a User-Agent field, or HTTP headers cannot be extracted. To improve the analysis performance, the study [5] developed a method that uses the server-client correlation. When User-Agent is not usually extracted, it compares with the information on the flow that was 1 minute ago. For instance, if the 5-tuple information of the current flow is consistent with the 5-tuple information created a minute ago, the two flows were formed in the same application that correspondingly saved as the same User-Agent. Likewise, to evaluate web server performance and analyze the server, it is crucial to describe the characteristics of HTTP traffic. It is a design with an automatic scheme that is used to classify sessions in HTTP traffic into several categories based on their features and sources as presented in [3]. Results indicate that, in one specific server, the sessions within the same category with highly unchanged characteristics are examined, even though such sessions might appear from different periods. On the other hand, sessions from different categories are totally different in characteristics and performance implications. Nevertheless, the result is interesting, but the transaction on which the experiment where undertaken has not been presented.

The variation characteristics of HTTP traffic have been analyzed in [6]. The research records the transmission over a day, including the difference in accessing the Internet between workdays and weekends. Essential to that is the distribution of the record number and types of content through examining HTML response packet length and the number of HTTP headers. Yet again, the specific transaction under the transmission has not been stated. The results show a linear relationship between the HTML response packet length and the number of HTTP headers. In a different approach as [7], they investigate the dissemination of certain characteristics of retrieved objects when Web pages of the most popular websites are accessed. The dissemination is based on object size, type occurrence, and type size. Consequently, the dissemination data can be used as a template for Web-traffic modelling. An intriguing result is the fact that, on average, 5.7% of HTTP traffic from Web servers to its clients is due to either spacer objects or non-existing files.

https://doi.org/10.20428/jst.v27i1.1984

Following some approaches and techniques used to analyze the transmission session from the studies mentioned above, and considering that most of the studies are without regard to the specific transmission session transactions details, this research proposed to evaluate the transmission session of business intelligence systems.

This paper is organized as follows: Apart from the present section provides the overview of this study. Next, Sections 2 and 3 represent the existing related work and the experimental technique respectively. Section 4 examines and discusses the result of this study respectively. Meanwhile, section 5 is provided for the conclusion of the paper.

2. Related Work

BI systems transmission session is an in-depth perception of the operations involved in analyzing information for better decision-making. Previous research disregards an evaluation of BI operation on a specific transaction. Furthermore, critical analysis of network transmission sessions for BI operation is lacking, although many research studies analyze network transmission sessions.

BI operation in a network environment directly works and correlates to the server's condition. Alongside the rapid growth and emergence of the abundance of the Web server, computer hardware should also develop to meet the requirements of the server's operation. Unfortunately, it is still difficult to attain the desired performance requirements without understanding transmission session transactions. A failure of a server in BI network environment is the failure of the entire BI system. A Model Based on the Response Time used to evaluate the maximum load of a server in the specified configuration has been performed in [8]. The study evaluates the server's performance by measuring the number of different loads and the response time of the Web server. However, the study does not provide the transaction operations in which the analyses were performed.

Furthermore, there is an attempt to optimize the performance of Web servers based on TCP connection management [9]. The authors examine the TCP connection control based on the response delay. Therefore, they propose a forward neural network to predict the response delay. Yet again, the transmission session transaction has not been presented.

Dynamic optimization for the Apache server to optimize server performance has been presented in [10]. The finding in this research unveils a negative relationship between CPU and keep-alive. At the same time, there is a positive connection between memory use and the maximum number of users based on cybernetics and MIMO models. Next, a combination of synthetically response time, throughput rate, and reject rate for the analyses of Web server performance has been introduced in [11]. However, it is still that the transaction on the transmission session has not been performed. Besides being unaware of the transaction details, security is also an issue in unrecognized malicious activity [12]. The aho-Corasick-based algorithm on compressed HTTP to extract HTTP traffic has been presented in [13]. It improves the string-matching algorithm of decompressed data behavior in a transmission session.

Besides, the current structure of web traffic on the Internet-based has been performed in [14]. Hence, this paper focuses on a reliable and secure connection based on TCP and HTTP analysis. Considering previous research techniques for analyzing transmission sessions, almost all the studies in this research do not treat specific details of transmission session transactions [15-16]. As a result, this research is willing to fill this gap and evaluate BI transaction transmission sessions.

3. Experimental Setup

A BI transaction was initiated, and the transmission was captured by a sniffing tool. Many tools, such as cables, taps, hubs, and switches, attach a sniffer to a network. In this experiment, to attach a sniffer to a network, a laptop running Wireshark is used to see the flow of packets going through the network. The packets captured by Wireshark capture various protocols running in the network. That helps to detect and resolve problems in the network. Two protocols detected from transactions using Wireshark are TCP and HTTP.

The HTTP protocol is based on a server/client that is used to transfer web pages over the network. TCP is a stack of protocols having different protocols on both layers 3 and 4. The TCP protocol is in layer four, and it allows bi-directional communication. Meanwhile, IP in the layer three protocol establishes communication over the network by providing the addressing system.

https://doi.org/10.20428/jst.v27i1.1984

TCP is used to accomplish the connection between the local host/client and server. It is a three-way handshake process. SYN/ACK message is sent and received between the local host/client and server at the start. SYN packet segment number is 0. Figure 1 shows that the sequence number was 0 in packet number 130, and in the following packet number 131, it gets acknowledged by the destination, and the sequence number changed from 0 to 1. When the same source calls upon the same destination, the sequence number changes from 1 to 0.

HTTP request and transmission: - It is the state where the actual request and transmission on the web page will be seen now. To view web pages, it comprises HTTP and TCP protocols seen by the GET command, and the OK message response comes after that. When packets are to be transmitted, FIN/ACK messages are sent and received.

No.	Time	Source	Destination	Protocol	Length	Info
126	1.319712	172.20.10.2	172.20.10.1	DNS	77	Standard query 0x5b08 A plus.l.g
127	1.375300	172.217.27.4	172.20.10.2	TCP	66	443→58440 [ACK] Seq=67190 Ack=94
128	1.385697	172.20.10.1	172.20.10.2	DNS	93	Standard query response 0x5b08 A
129	1.386222	172.20.10.2	216.58.221.46	TCP	78	58454→443 [SYN] Seq=0 Win=65535
130	1.444106	216.58.221.46	172.20.10.2	TCP	74	443→58454 [SYN, ACK] Seq=0 Ack=1
131	1.444190	172.20.10.2	216.58.221.46	TCP	66	58454→443 [ACK] Seq=1 Ack=1 Win=
132	1.444431	172.20.10.2	216.58.221.46	TLSv1.2	583	Client Hello
133	1.517613	216.58.221.46	172.20.10.2	TCP	66	443→58454 [ACK] Seq=1 Ack=518 Wi
134	1.517744	216.58.221.46	172.20.10.2	TLSv1.2	222	Server Hello, Change Cipher Spec
135	1.517795	172.20.10.2	216.58.221.46	TCP	66	58454→443 [ACK] Seq=518 Ack=157
136	1.518143	172.20.10.2	216.58.221.46	TLSv1.2	117	Change Cipher Spec, Hello Reques
137	1.518318	172.20.10.2	216.58.221.46	TLSv1.2	223	Application Data
138	1.596800	216.58.221.46	172.20.10.2	TLSv1.2	122	Application Data
139	1.596909	172.20.10.2	216.58.221.46	TCP	66	58454→443 [ACK] Seq=726 Ack=213
140	1.597004	216.58.221.46	172.20.10.2	TLSv1.2	108	Application Data

Figure 1: HTTP request and transmission

4. Results

About twenty-two samples of transactions are analyzed. An IP address is a unique address that distinguishes a device over the Internet or across a local network. It allows a system to be identified by other systems connected through the internet protocol [17]. TCP provides a reliable transport layer in which the reliability is provided by utilizing a timeout when it sends the data. Hence, it will retransmit the data if the data is not recognized when the timeout expires. It can be seen from Figure 2 that the timer expires from 40 sec to 50 sec when the packet is retransmitted, and it gets its highest peak in the Figure. This attribute of TCP protocol ensures a reliable connection.

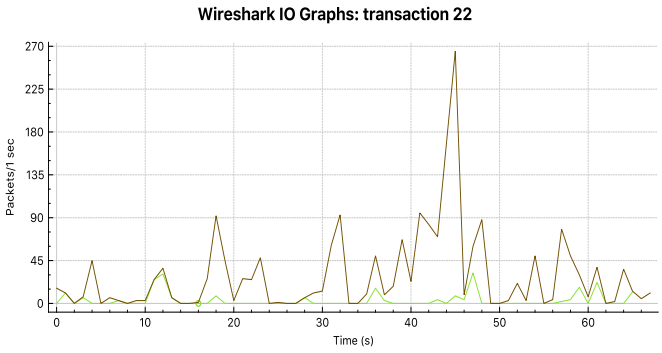

Figure 2: Total Transaction session

Figure 3 indicates the time packets take vs. the packet's length from the HTTP protocol. HTTP is a transfer protocol the World Wide Web uses to retrieve information from the distributed server. Information is always broken down into packets while it is sent or received over the network. The time and length of the packets in the HTTP protocol are the focus explained in this paper. The client performs connections with a remote server and then issues a request. Next, the client sends a connection request to the server's HTTP port, and the server will respond to the request with a connection response. After that, the client will get the connect response, and send the first 536 bytes of the request. The server acknowledges the first part of the request.

https://doi.org/10.20428/jst.v27i1.1984

After that, the client sends the second part, followed immediately and without further delay by the third and last part of the request. The server transmits a packet comprising the first 512 bytes of the response and recognition of the second and third parts of the request. It is then followed by another packet that contains the second 512 bytes.

Furthermore, the client sends a message confirming the first two response packets. The server will next send off the third and fourth response packet. In addition, the fourth packet also contains a flag indicating that the connection is being closed. Next, the client recognizes the data and sends a message to close the connection. Lastly, the server acknowledges the close. Figure 3 shows the time and length of each request and response between the client and the server.

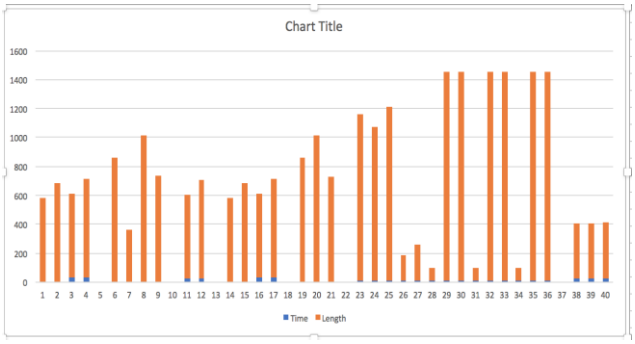

Figure 3: HTTP Packet Length

Each query/transmission is associated with a specific protocol and length. The length of the query is called packet length. In other words, the captured packet of a network transaction can be from more than one protocol, as each of these protocols contains several associated packets. Thus, each packet has a specific length (see Figure 4). The length of every packet can be in different sizes as it depends on the protocol type itself. For example, the length of the HTTP protocol is usually known to be more than others. The packet's length is often given in bytes, as seen in the status bar of Wireshark.

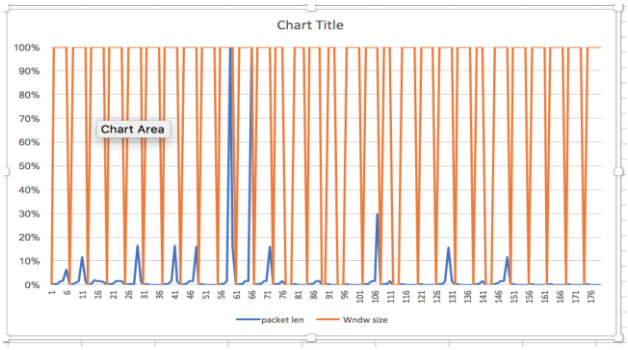

Figure 4: Packet length and windows size for all transactions

Unlike packet length, which is owned by all protocols, the window size is used by TCP protocol only. The initial window size is agreed upon when the TCP session is established during the three-way handshake. One point more,

Figure 4 shows the flow rate of packet lengths vs. window sizes of 30 different captures being done in different stages of online banking transaction sessions. As seen from the graph, the size of 'window size' is kept almost the same throughout all captured transactions, but the size of 'packet length' has changed accordingly based on the amount of transactions' payload. In any transaction session, when the amount of data load increases, the packet length trend is also increased. That can usually happen during HTTP file transactions more than other data transactions. Concluding the explanation of this graph with one important security message, that is, when the session has entered an HTTP transaction, it is crucially important to the bank organization to use a secure protocol like SSL to make HTTP an HTTPS transaction over STL.

Time: Since the start of the transaction session, the time for each packet transaction is going to keep itself tracking ahead. This time record can tell us two important things. One thing is to show us the time each packet is captured in the transaction session. The second thing is to compare the time record of each packet with its previous and subsequent time-sequence records. We also can understand how long each packet takes to reach its destination device if it is a request or how long it takes to receive back to its source device if it is a reply. Sequence Number, as the name implies, keeps track-number of each data transmission flowing between the source and destination devices from the start of a transaction session

https://doi.org/10.20428/jst.v27i1.1984

to the end of this transaction session. In the first stage, when the source device broadcasts to its destination, the sequence number is set to 0 because no data byte has been sent yet. After the three-way handshake is established, the sequence number will keep tracking all data bytes sent from the source to the destination accumulatively. Now for analyzing time vs. sequence number, in general, the trends for both patterns will increase accordingly as the same as a session time. Since this graph is captured from different transaction sessions, the trends are not increased gradually as the general definition says.

Nevertheless, the important point of this graph is to tell us about a reliable connection by showing less or more gaps between the trend of time and sequence number. It means that when there are many gaps between the trend of time and sequence number, the network connection is established with less traffic congestion. Therefore, the probability of packet loss and transaction overhead is decreased.

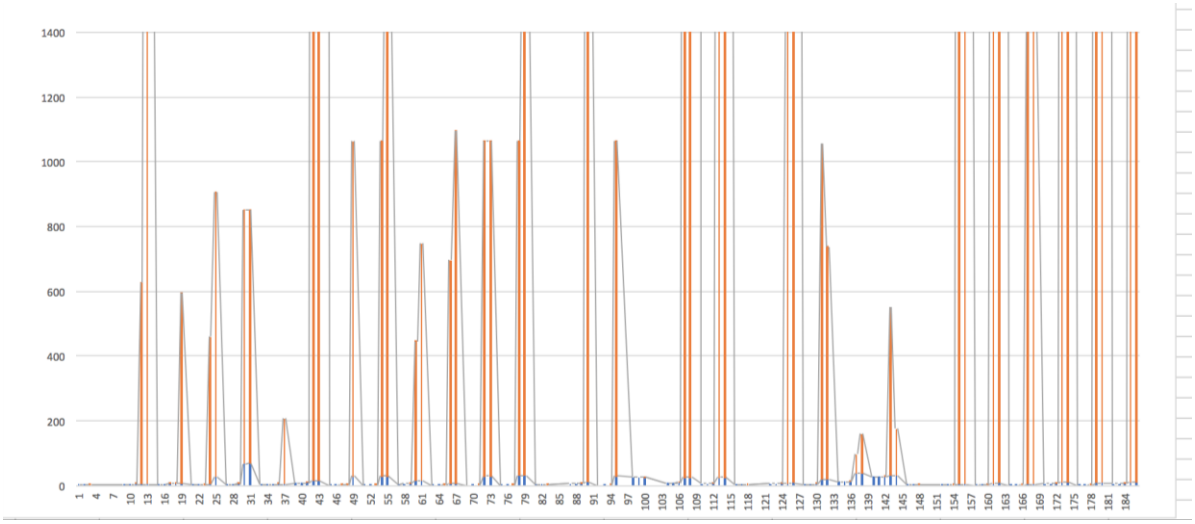

Figure 5: Time vs. sequence number from all 30 captured transactions.

As the window size is defined in part b of section A, it is the number of bytes that the destination device of a TCP session can accept and process at one time. It also is said that the initial window size is agreed upon when the TCP session is established during the three-way handshake. After the establishment of a three-way handshake, the window size will remain the same with the destination host.

However, from the source host, it will change to take a different size because the source host is the device that can limit and fit the number of bytes requested by the destination according to the buffer availability of the destination host. We conclude that the TCP session is the best mechanism for a reliable connection because the data transmission between the source and destination hosts flows without any loss. Even if a piece of data is lost due to a flowing problem, a TCP session also offers a mechanism of retransmission.

5. Conclusion

In this paper, we have presented network traffic from a live network involving BI transactions. Statistics and detailed analysis are based on various traces and conversations between two endpoints. Several transmission sessions are captured, and other attractive features are shown using Wireshark, an excellent network analysis tool. Results generated from captured provide details of network dynamics, secure and reliable connection between two endpoints, and problems that can lead to network slowness. The evaluation of BI transaction transmission sessions has shown that HTTP over TCP transmission sessions in BI transactions could be affected by traffic congestion and transaction overhead.

6. Acknowledgment

This paper was supported under FRGS15-226-0467, International Islamic University of Malaysia (IIUM).

https://doi.org/10.20428/jst.v27i1.1984

7. References

[1] H.H. Huang, Z. Wang, and W. Chung, "Efficient parameter selection for SVM: The case of business intelligence categorization," IEEE International Conference on In Intelligence and Security Informatics (ISI),pp. 158-160, 2017.

[2] Tomasz Bujlow, Tahir Riaz, and Jens Myrup Pedersen. "Classification of HTTP traffic based on C5.0 Machine Learning Algorithm," In 2012 IEEE Symposium on Computers and Communications (ISCC), Cappadocia, pp. 882-887. IEEE, July 2012.

[3] Xiaozhu Lin, Lin Quan, and Haiyan Wu. "An Automatic Scheme to Categorize User sessions in Modern HTTP Traffic," In 2008 IEEE Global Telecommunications Conference, New Orleans, LO, pp. 1-6. IEEE, November 2008.

[4] Chang-Gyu Jin and Mi-Jung Choi. "Integrated analysis method on HTTP traffic," In 2012 14th Asia-Pacific Network Operations and Management Symposium (APNOMS), Seoul, pp. 1-6. IEEE, September 2012.

[5] Hwan-Hee Kim and Mi-Jung Choi. "Improvement of application-level analysis of HTTP traffic through Server-Client Correlation," In International Conference on ICT Convergence (ICTC), Jeju, pp. 992- 996. IEEE, October 2013.

[6] Pengcheng Jiang, Fang Liu, Huan Wang, and Chenyu Li. "Characterizing HTTP Traffic of Mobile Internet Services in Provincial Network," In the 6th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, vol. 1, pp. 78- 81. IEEE, August 2014.

[7] Y. C. Chehadeh, A. Z. Hatahet, and A. E. Agamy. "Investigating Distribution of Data of HTTP Traffic: An Empirical Study," In 2006 Innovations in Information Technology, Dubai, pp. 1-5. IEEE, November 2006.

[8] Xianghua Xu, Tingting Xu, Yuyu Yin, and Jian Wan. "Performance evaluation model of Web servers based on response time," Conference Anthology, China, pp. 1-5. IEEE, January 2013.

[9] Yu Guofang and Tan Xianglu. "QoS Control of Web Server Based on Active TCP Connection Management and Delay Prediction," In 2010. International Conference on Intelligent Computation Technology and Automation (ICICTA), Changsha, vol. 2, pp. 955-958. IEEE, May 2010

[10] Jun Li and Menghan Lu. "The performance optimization and modeling analysis based on the Apache Web Server," In the 32nd Chinese Control Conference (CCC), Xi'an, pp. 1712-1716. IEEE, July 2013.

[11] Zhaoyang Qu, Wei Wang, and Zhiqian Li. "Web Server Optimization Model Based on Performance Analysis," In the 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), Chengdu, pp. 1-4. IEEE, September 2010.

[12] Martin Husák, Petr Velan, and Jan Vykopal. "Security Monitoring of HTTP Traffic Using Extended Flows," In the 10th International Conference on Availability, Reliability and Security (ARES), Toulouse, pp. 258-265. IEEE, August 2015.

[13] Anat Bremler-Barr, and Yaron Koral. "Accelerating Multipattern Matching on Compressed HTTP Traffic," IEEE/ACM Transactions on Networking, vol. 20, pp. 970-983, December 2011.

[14] Chen, J., & Cheng, W. (2016, September). "Analysis of web traffic based on HTTP protocol," In 2016 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM) (pp. 1-5). IEEE.

[15] Ibrahim, A. A., Salsabil, S. I., & Lawal, I. A. (2022). "Effective Utilization of An Unused Bandwidth in IEEE 802.16 Network," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 14(2), 15-22.

[16] Ibrahim, A. A., Atayee, A. R., & Lawal, I. A. (2022). "SDN Multi-Domain Supervisory Controller with Enhanced Computational Security Count," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 14(2), 23-29.

[17] Abubakar, A., & Wan Husin, W. N. I. (2022). "The effective utilization of internet bandwidth in organizational demand services and applications," Journal of Information and Knowledge Management (JIKM), 12(1), 140-160.

https://doi.org/10.20428/jst.v27i1.1984