Behavior Pattern Recognition of Game Dragon Nest Using Bloom Filter Method

Deris Stiawan[1], Mohd. Yazid Idris[3], Diky Aryandi[1], Ahmad Heryanto[1], Tri Wanda Septian[1], Farkhana Muchtar[2], and Rahmat Budiarto[4]

[1]Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya, Indonesia
[2]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia
[3]Media and Game Innovation Center of Excellence, Institute of Human Centered Engineering, Universiti Teknologi Malaysia
[4]College of Comp. Sc. & IT, Albaha University, Albaha, Saudi Arabia

***Abstract*:** Dragon Nest is one of Massively Multiplayer Online Role-playing Game (MMORPG) online games. It has become the most popular online game played by people around the world. This work observes two examples of the MMORPG online games: the Dragon Nest INA and the Legend DN II. The purpose is to analyze the traffic data of the Dragon Nest to find and discern the patterns of behavior of the Dragon Nest INA and the Legend DN II using Deep Packet Inspection (DPI). A dataset is constructed by capturing traffic data from the testbed environment. Then feature extraction, feature selection, and visualization are performed during the experiments. Experiment results show the traffic data of the Dragon Nest INA is higher than the Legend DN II. It is because of the difference in the number of entries in the game. Then, the Bloom filter method is used as a tool to check the existence of a pattern of the Dragon Nest in the dataset. The false positive rate of matching experiment is 0.399576%.

***Keywords*:** online game, deep packet inspection, Bloom filter, traffic classification, behavior analytics, feature extraction, visualization,

1. Introduction

Online games is a video game played either in part or completely through the Internet [1]. One of the genres of online games is Massively Multiplayer Online Role-playing Game (MMORPG). MMORPG [2] is a virtual world, which is constantly played online by thousands of simultaneous players to access the game's website through a browser.

One of the MMORPG genres in Indonesia is Dragon Nest INA. The Dragon Nest INA is a free online fantasy MMORPG developed by Eyedentity. This game uses a non-target combat system so the players hold full control over every movement of players' character. The players will also get skill and item during the game, in collaboration with thousands of other players in real-time, complete the task to get rewards and compete to get status and power.

Deep Packet Inspection (DPI) is used to get the pattern of behaviors of the Dragon Nest game traffic. The DPI is a way to examine the data packets on the network. It is able to search the contents of the package and discover the existence of certain patterns. The search includes in the header as well as in the payload [3]. Pattern matching, behavior analysis, and flow observation are methods in DPI [4].

This work attempts to seek patterns of the MMORPG games traffic behavior by utilizing the DPI and uses obtained patterns to analyze them. The work limits the observation only on two types of the games, i.e., the Dragon Nest INA and the Legend DN II online games as case studies. Other limitation is only considering three users/machines as users playing the two games due to the machine's availability in the lab. Nevertheless, these limitations do not affect much the observations.

A lack of visual tool for monitoring the traffic has motivated the authors of this paper to come up with a tool that is able to visualize the observations resulted from DPI process.

One advantage of the proposed work is visual assistance to the system/network administrator on monitoring their network traffic against anomalies causing by the users that playing an online game during the working hours.

The paper is organized as follows. Section 2 reviews related work. Section 3 describes the method, the testbed and experiment setup. Section 4 discusses the experimental results, and Section 5 provides a conclusion and potential future works.

2. Related Work

Online games are scattered on a wide range of modern gaming platforms, including PC, console, and mobile devices. It is also in form of different genres, including First-Person Shooter (FPS) and Massively Multiplayer Online (MMO). MMO is a multiplayer video game, which supports a large number of players simultaneously which connected via the Internet. Almost all of the stylish gaming MMO genres have more than one persistent world and it is always updated even while the player is not online. MMO genre game can be found almost in all devices, which support online networks, such as PC, console, and smartphone. Some examples are MMORPG (Massively Multiplayer Online Role-playing Game), MMORTS (Massively Multiplayer Online Real-time Strategy), and MMOFPS (Massively Multiplayer Online First-Person Shooter).

MMORPG is a combination of role-playing video game and a massively multiplayer online game where a large number of players are interacting in a persistent world [2]. Like all games in RPG genre, the player plays a character who has a certain capability, adventuring in the world full of monsters, complete missions, raising the level of the character, as well as virtual money trading [5]. An example of a MMORPG game is Dragon Nest.

MMORTS is a combination genre of real-time strategy and massively multiplayer online games. It is a web-based type of game with access to a large number of players simultaneously in a persistent world. The player usually plays a role, such as a general, a king, or other leader figures which leading the troops into a battle and defend the resources, which needed in the war [5]. The theme of MMORTS game usually science

fiction or fantasy world. An example of an MMORTS game is Clash of Clans.

The client-server architecture is the structure of distributed applications, which have a division of tasks between the providers of the resources or services (servers) and service requestor (client). Host server runs one or more server program which shares resources to the client [6], [7]m [8]. Clients do not share its resources but ask for resources or service function from the server. Therefore, the client starts a communication session with a server, which waits for an incoming request. Server and client exchange message in a request-response pattern: the client sends a request and server give a response.

The client-server architecture is the most used in online game network MMORPG type such as Dragon Nest [9]. The server is in charge of maintaining the ' virtual world ' and controls application logic, while users give feedback process as well as state updates. The client software, which runs on the user side is able to do user input sampling and sends it to the server for processing. Sampling which is sent to the server is also capable of displaying images and sound effects upon state updates to the user. Type of client in Dragon Nest game is a fat client, which are most of the resources such as animation, graphics, sound effects, and the map will be entered into the client to reduce some of the functions of the server. The server and the client continue to communicate in high frequency to maintain the same game condition for all clients. However, due to a limited bandwidth, the server was unable to send out any changes, which occur in the whole of the client side then the server sends a broadcast snapshot condition of the latest games on a regular basis on the whole client.

Deep Packet Inspection (DPI) is a way to examine the data packets on the network. The DPI works by searching and filtering the contents of the packet and discover the existence of certain patterns. The searching includes in the packet header as well as in payload [3]. Most DPI mechanisms use signature analysis to understand and verify the various applications. The signature is a unique sign, which exists in every application. A unique sign, which has been found is kept in a database to be used as a classification engine by comparing traffic which comes in with an existing database. The database should be updated regularly in order to offset new protocols and applications.

There are many signature analysis methods available, some popular methods include:

1. Pattern analysis (Pattern): looking for special patterns (bytes, characters, or strings) in the payload, which allows a classification engine to identify those patterns.
2. Numerical analysis (Numerical): using the characteristic numbers on the packet for identification, such as the size of the payload, number of the response packet, and offset.
3. Behavioral analysis (Behavioral) and Heuristics (Heuristic).
4. Protocol /State Analysis (Protocol/State): in some applications, protocol follows a specific sequence of steps, such as FTP GET, requested by the client usually will be followed by the valid response from the server. The flow like this can be used as a classifier for the corresponding application.

Abdullah and Alhashmi [10] proposes a novel evolving fuzzy system to discriminate anomalies by inspecting the network traffic in time series manner.

Bloom filter is simple random data structures with low storage space requirement to display a set of data, suitable to the order in which support query membership [11]. There is a possibility of a false positive in the bloom filter. However, with the larger size of storage is able to cover this weakness by making this possible mistake as low as possible. Bloom filters usually are used as computing applications [12].

Extracting features from raw data allows us to recognize hidden unusual patterns more effective [13]. Feature extraction can be pursuing a search pattern on large data because very large data use much memory and computational power. By using feature extraction, the pattern is faster and easier to find.

Visualization according to [14] is the disclosure of an idea or the idea of using pictures, writings (words and numbers), graphics, etc. Visualized data are obtained from raw data using certain methods and algorithms of computer graphics. Visualization of the Dragon Nest game traffic represents image pattern (pattern) behavior data packet in the game.

3. Research Method

This work uses Bloom filter to examine whether a pattern is on the incoming data traffic. The mechanism is the administrator should create a 'dictionary' that contains wanted information to be searched in the traffic data. After the dictionary is ready, the examination was undertaken by feeding the traffic data into the Bloom filter for inspection whether the feature is in the dictionary or not. The results of each data value are simply true and false. No false negative in the review, there is only the probability of false positive.

A Bloom filter A is used to display a set $S = \{x_1, x_2 \ldots x_n\}$ with n element consists of an array bit m and initially set to 0. Bloom filter uses k as hash independent function $h_1, \ldots, h_k$ with a span of $\{1, \ldots, m\}$ [12].

Figure 1 shows Bloom filter A begins with a set of arrays with value 0. Each object in set x_1 is hashed as much as k times, a bit on hash will result in value 1. To check if the y element is in the set or not, do a hash as much as k times and see bits which represent it. y_1 element is not in the set because the value of 0 is found in one of its bit, while element $y2$ is within the set but not included before making filter into a filter false positive. The possibility of a false positive is inversely proportional to the length m of the array Bloom filter and given by (1) [15].

$$fp = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k \approx \left(1 - e^{-kn/m}\right)^k \quad (1)$$

To add an element into the filter, insert the element into the hash function k to get the position of the array k. The element, which was inserted into the hash function will have a value of 1 in the position array. While to do a query (testing), insert the element into the hash k function to get the position of the array k. If one of these elements is value 0, then the element must not exist in the set. Reversely, if those elements exist, then all the bits will value 1 as the element is inserted. If an element which is not entered earlier into the filter but the value 1 in its bits when checking, then the case is referred to as a false positive. In simple a Bloom filter is hard to distinguish between false positive and true positive,

however with a more complicated techniques, false positive can be distinguished.

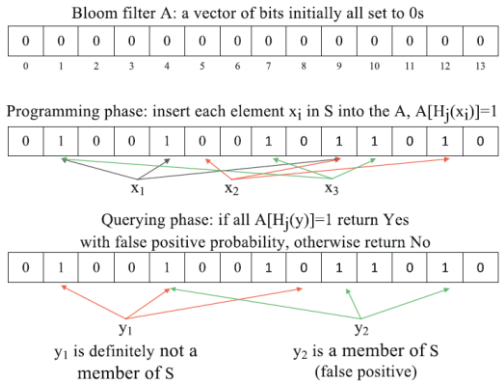


Figure 1. Example of bloom filter [12].

3.1 Experimental Set up

The experiment is conducted in the laboratory of Communication Network and Information Security (COMNETS) Faculty of Computer Science, Sriwijaya University. The experiment uses a manageable switch as a concentrator, which supports port mirroring to set up a star topology. The experiment uses two physical devices in a form of PC as a game clients only and one laptop device as game client and as Packet Analyzer. The machines use Windows operating system. Figure 2 illustrates the topology of the testbed for the experiment. The testbed connects to the game server thru Internet connection.

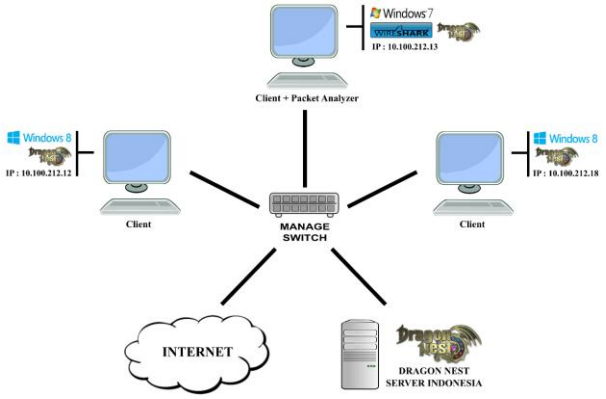


Figure 2. Testbed topology.

3.2 Hardware and Software Specification

Table 1 and Table 2 show respectively the software and hardware specifications of the equipment used in the testbed.

Table 1. Software Specification

Function	Tools/Framework	Description
Sniffer	Wireshark	Version 2.2.4 for Windows
Interpreter	Python	Version 2.7.6
Command Prompt	CMD	Version 6.1.7
Text Editor	Notepad++	Version 7.3.2
Game Client	Dragon Nest INA Client	Launcher Version 277
Game Client	Legend DN II Client	Launcher 252

Table 2. Hardware Specification

Device & OS	Specification	Unit	Description
Manageable Switch	TP-LINK Smart Switch TL-0SG108E 8 port Gigabit Ethernet Interface	1	Port Mirroring
Laptop with MS Windows 7 O/S Ultimate 64-Bit	Intel® Core™ i5-2410M CPU @ 2.30GHz (4 CPUs). ~2.3GHz. 8 GB RAM. 1 TB HDD. Gigabit Ethernet Interface	1	Client + Sniffer
PCs with MS Windows 8 Enterprise 32-Bit	Intel® Core™ i3-4170M CPU @ 3.70GHz (4 CPUs). ~3.7GHz. 4 GB RAM. 1 TB HDD. Gigabit Ethernet Interface	2	Client

3.3 Features Extraction

The captured traffic data in a .pcap file is converted into a .csv file using a utility program. The feature extraction utility program uses two parameters, which are – f as input file '. pcap' and – o as output file '.csv'. Table 3 shows the attributes of the packet data.

Table 3. Features extraction attributes.

#	Attribute	#	Attribute	#	Attribute
1	*Timestamp*	8	*ml*	15	*ack_num*
2	*epoch_time*	9	*length_header*	16	*seq_num*
3	*Protocol*	10	*total_length*	17	*win_num*
4	*ip_src*	11	*id_header*	18	*urg_pointer*
5	*ip_dst*	12	*checksum_header*	19	*checksum_protocol*
6	*port_src*	13	*fragment_off*	20	*Service*
7	*port_dst*	14	*flags*	21	*Payload*

In this work, the use of DPI is to look for behavior or habit pattern of Dragon Nest INA game. Searching and determination of game pattern can be seen from the attributes on that packet, such as IP address, port, TTL, and flags. The existing attribute of the feature extraction program allows the reading and searching pattern/signature become easier. The file with '.csv' extension is normalized to get some attributes, which becomes the main behavior patterns of Dragon Nest game. The normalized data then being visualized with the aim to make the recognition of the behavior patterns of Dragon Nest game easier. Encrypted payload packet complicates the DPI to enter further look into the payload. Thus, to improve the accuracy of the signatures on Dragon Nest INA game traffic data, more number of attributes in the package are considered along with the DPI manual method.

3.4 Visualization

This work uses a parallel coordinate visualization. The visualization system is a lines visualization technique that presents the data in more than one-dimensional or attributes using a different color to distinguish them.

The attributes used for visualization are normalized attributes; they are source IP, source port, length, flags, destination, port, TTL and destination IP. Attribute value exceeding 999 will be normalized by providing a period (.) after the third digit of the value of an attribute in order to form a balanced look visualization.

4. Experiments Result and Analysis

Experiments produced five initial data (raw data) of Dragon Nest game using Wireshark as sniffer and saved as '. pcap' file. The average size of each file is 18 MB with an average of 41,404 data row per file. The dominant protocols obtained in the data are TCP and UDP. Those raw data then are extracted to get attributes, which can be used to determine the behavior pattern of Dragon Nest INA and Legend DN II games.

4.1 Behavior *Pattern*

The user behavior of Dragon Nest INA and Legend DN II game resulted in four patterns: login pattern, main city pattern, transition area patterns, and dungeon pattern.

4.2 Normalization of Behavior Pattern

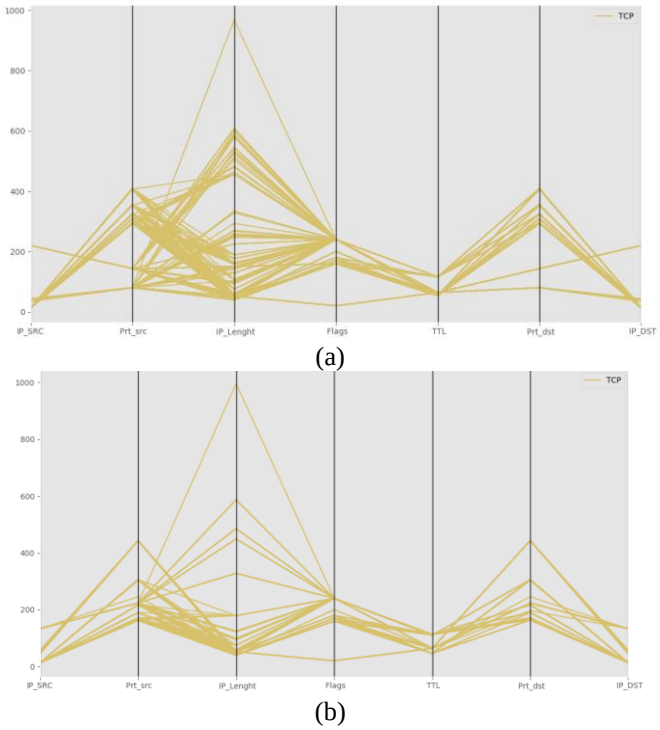
To simplify the visualization process, the captured data are normalized. Attributes, which are taken as normalization attribute are IP_SRC, Prt_src, IP_Length, Flags, TTL, Prt_dst, and IP_DST. On the IP address, the last part of IP is taken as a marker of IP, which used in the normalization. The addition of the period behind the three-digit number on the IP length and port number are used to minimize the range without any mathematical calculations so that the resulting difference lines are not so significant. The value of flags is also modified to generate flags number (F (10), S (20), R (40), P (80), A (160), U (320)). Table 4 shows some examples of normalized data.

Table 4. Examples of data resulted from the normalization process.

#	Pattern type	Normalization result
1	Login	36. 80. 150. 240. 54. 293. 13. TCP
2	Main City	221. 144. 971. 240. 116. 411. 13. TCP
3	Transition region	207. 145. 52. 180. 116. 300. 13. TCP
4	Dungeon	207. 151. 35. 0. 116. 511. 13. UDP

4.3 Visualization of Normalization result of Dragon Nest INA and Legend DN II

Figure 3 shows the visualization of the user login behavior patterns in both games. The pattern in Figure 3 (a) is the pattern when the player login into the Dragon Nest game. The IPs used as the IP source in the Dragon Nest INA game are 49.50.7.36; 49.50.7.43; and 49.50.4.220. Using port 80 for HTTP and dynamic port 143 after normalization from the value of 14300. Length of IP ranges between 40 –1000 after normalized. The value of flags 160 to Ack and 240 to Ack Psh. Destination Port on the client side is dynamic port ranges from 2900 to port 4100. A value will be maintained by the client during a session of the game, and that value will change when the client does Re-login.



(a)

(b)

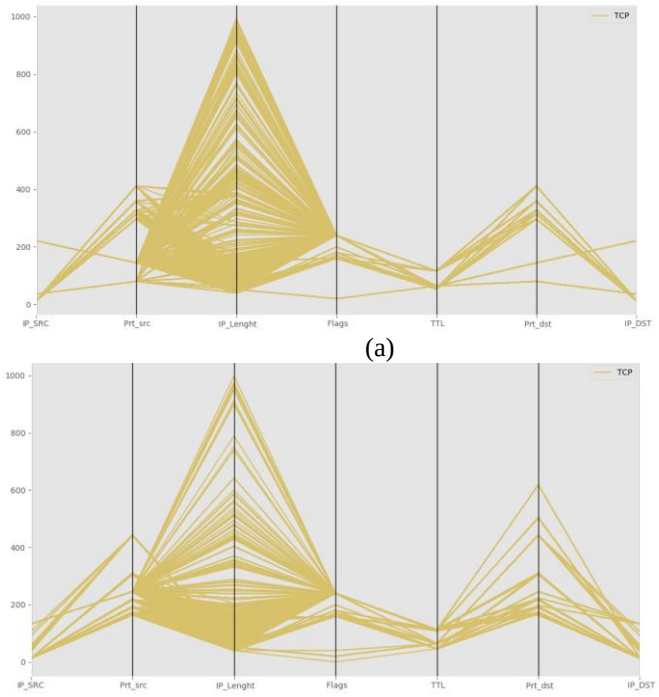
Figure 3. Visualization of *Login Pattern* (a) Dragon Nest INA and (b) Legend DN II.

While in Legend DN II game (Figure 3 (b)), the IP resource is 64.94.100.133 as the primary IP and 62.128.100.45, 62.128.100.47; 62.128.100.53; and 62.128.100.57 are the support IPs appear in the log. Ports and IP data length used by the main IP is the dynamic type, while the port and IP data length for support IP are static, there are port 443 commonly used as an HTTPS port with the IP length between 40 – 52. Dominant flag used in the main IP is AP so its value on visualization is (160 + 80 = 240) and supporting IP flag is A (80).

Figure 4(a) shows the pattern on the main city of the Dragon Nest game. In the Dragon Nest INA game, IP source from the server is 49.50.7.36 and 49.50.4.221. The port number used is HTTP 80 to IP 49.50.7.36 and IP 49.50.4.221 for dynamic port with a number of 144 after normalized from 14402. For the length of the IP is almost the same with the login pattern, but the length of IP from 49.50.4.221 more varied due to a large number of activities which occur at the main town while the length of IP from 49.50.7.36 tend to be more stable with a length of 150 IP after normalized from 1500. Flags which often appears the same as login, which are patterns of 160 to Ack and 240 to Ack Psh. Dynamic destination Port with the range of 2900 – 4200.

While the main city pattern of Legend DN II game, still uses 64.94.100.133 as the main IP, and IP support of 62.128.100.37;62.128.100.43;62.128.100.45; 62.128.100.47; 62.128.100.49;62,128.100.51;62.128.100.53; 62.128.100.55; 62.128.100.57; 62.128.100.92; and 62,128,100,106. IP Support still uses the same port, flags, and IP length: port 443, flags A(80), and the IP length of 40 – 52. When compared to the pattern of Dragon Nest INA main town, the IP average length in the main city of Legend DN II is more dominant in the range of 40 – 200 while Dragon Nest INA on a range of 40 – 991. It is because the state of the main cities on the Legend DN II is a more off guard so that the server

does not always send update circumstances surrounding the player (See Figure 4(b)).



(a)

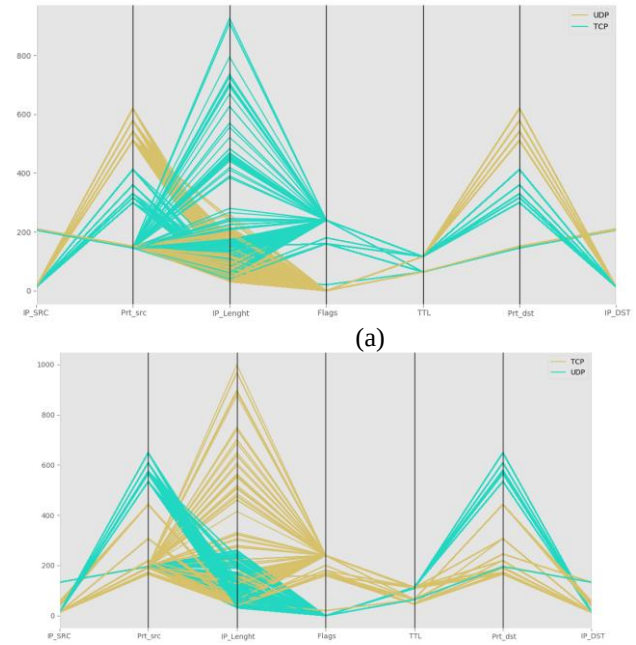
(b)

Figure 4. Visualization of Main City Patterns of Dragon Nest INA (a) and Legend DN II (b).

In the Dragon Nest INA game (Figure 5(a)), the transition area uses two protocols, TCP and UDP. The use of both Protocol aims to maintain the thrill of playing in real time. UDP is used to keep the lag does not occur in the game, and used to set the sent TCP data packet. The transition area is dynamic IP. The IP sources in the transition area of Dragon Nest INA is 49.50.4.206 – 49.50.4.210 with a dynamic port range 14500 – 14600 for TCP and the range of 15000 – 15200 for UDP. The length of the TCP packet range of 40 – 999 with a long IP of 1500 has normalized become 150. As for the length of the UDP, IP is relatively smaller, which is around 30 – 260. Flags remain dominant of 160 and 240 for TCP. Dynamic destination port ranges within 3000 – 4200 for TCP and the range of 51000 – 63000 for UDP.

The transition area on Legend DN II (See Figure 5(b)) similar to Dragon Nest INA. The difference is the IP used. In the Dragon Nest INA, the IP used vary in a particular range, Legend DN II only uses one primary IP, which is 64.94.100.133 and supported some IPs as data transit. IP support on the pattern transition area of Legend DN II are 62.128.100.37; 62.128.100.43; 62.128.100.47; 62.128.100.51; 62.128.100.53; 62,128.100.55; and 62.128.100.57. The protocol used are the same: TCP and UDP. Length of the IP of UDP used in Legend DN II is the same as Dragon Nest INA, which is 30 – 260. The use of destination port is also the same as Dragon Nest INA, which is 51,000 – 63,000 for UDP.

Figure 6 shows patterns in the dungeon. In dungeon, both games use UDP protocol because against monsters need a quick response between input from players and response from the server to keep the sensation of real-time combat. For the visualization, the pattern of dungeon almost equal to the area of transition, since the dungeon was incorporated into a single game session with the transitional area. So the value of the attributes which already exist in the transition area does not change so significantly while entering and exiting the dungeon. The amount of data recorded on the pattern of the dungeon Legend DN II is more compared to the Dragon Nest INA because items drop in the unofficial games is usually more from an official to pamper and ease the players headed for the top position in the game.

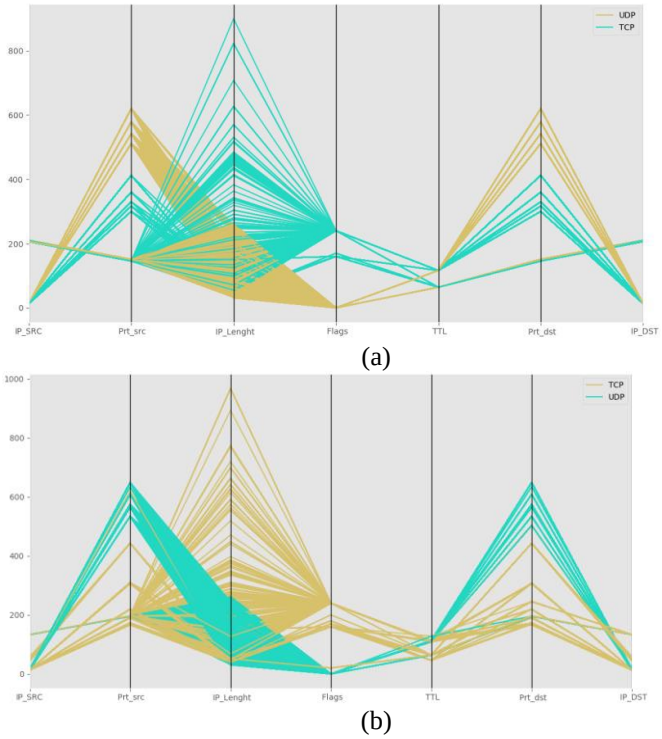


(a)

(b)

Figure 5. Visualization Transition Area Pattern of *Dragon Nest INA (a) and Legend DN II (b).*

Finally, the false positive rate produced by the Bloom filter used in the experiment was 0.399576%. The result of the examination of bloom filter will be inserted into the log data.



(a)

(b)

Figure 6. Visualization of *Dungeon* (a) Dragon Nest INA and (b) Legend DN II.

5. Conclusion and Future Work

The deep packet inspection using Bloom filtering and combined with visualization of traffic attributes help us to recognize and analyze the traffic behavior pattern of the observed online games: the Dragon Nest INA and the Legend DN II. From the observations, in term of the use of IP address, the Dragon Nest INA uses more IP addresses (involve at least eight servers) during the game execution while the Legend DN II uses only one main IP address. The packet length of Dragon Nest INA in the main town more vary compared to the Legend DN II, due to the difference in the number and level of players activity who were in the main city of both games. Furthermore, the difference in the number of items dropped in the dungeon, the players in the Dragon Nest INA need to collect diligently items to strengthen their characters because items obtained during the game is limited. Whereas, in the Legend DN II, items, which obtained when completing a dungeon, are many, so players more easily strengthen their characters during the game. The experiment result showed the false positive rate of the Bloom Filter method was 0.399576%.

Some suggestions for further researches include comparison with other types of online games such as MMOFPS or MMORTS, implementation a real-time visualization of online game traffic patterns, considering security aspect of the Dragon Nest game, and enhancing the Bloom filter through the use of other types of filters to improve the accuracy.

Acknowledgement

This research was supported by the Ministry of Higher Education (MoHE) under grant COE and Research Management Center (RMC) of Universiti Teknologi Malaysia, vote No: Q.J130000.2428.03G94.

References

[1] E. Adams, Fundamentals of Game Design. Pearson Education, 2014.

[2] V. A. Badrinarayanan, J. J. Sierra, and K. M. Martin, “A dual identification framework of online multiplayer video games: The case of massively multiplayer online role-playing games (MMORPGs),” Journal of Business Research , Vol. 68, No. 5, pp. 1045–1052, 2015.

[3] Y. Afek, A. Bremler-Barr, and Y. Koral, “Space-efficient deep packet inspection of compressed web traffic,” Computer Communication, Vol. 35, No. 7, pp. 810–819, 2012.

[4] J. Svoboda, “Network Traffic Analysis with Deep Packet Inspection Method,” Master thesis, Masaryk University Faculty of Informatics, 2014.

[5] X. Che and B. Ip, “Packet-level traffic analysis of online games from the genre characteristics perspective,” Journal of Computer and Computer Applications, Vol. 35, No. 1, pp. 240–252, 2012.

[6] W. Hong-You and Z. San-Ping, “The Predigest Project of TCP/IP Protocol Communication System Based on DSP Technology and Ethernet,” Physics. Procedia, Vol. 25, pp. 1253–1257, 2012.

[7] Linktionary.com, “TCP (Transmission Control Protocol),” 2001.http://www.linktionary.com/t/tcp.html. [Last Accessed: 12-Dec-2016].

[8] B. Smith, “MMO Smart Servers Using Neural Networks for Intelligent, Client-handling Decisions, and Interactions,” Procedia Computer Science, Vol. 95, pp. 201–208, 2016.

[9] R. M. Daniel, E. B. Rajsingh, and S. Silas, “Deriving Practical Applicability of Hierarchical Identity Based Encryption in Massively Multiplayer Online Role-Playing Games,” Procedia Computer Science, Vol. 93, No. 9, pp. 839–846, 2016.

[10] S. Abdulla, A.S. Al Hashmi, “iSEFE: Time Series Evolving Fuzzy Engine for Network Traffic Classification”, International Journal of Communication Networks and Information Security (IJCNIS) Vol. 10, No. 1, pp. 116-124, 2018.

[11] J. Shana and T. Venkatachalam, “An improved method for counting frequent itemsets using bloom filter,” Procedia Computer Science., Vol. 47, No. C, pp. 84–91, 2014.

[12] A. Broder and M. Mitzenmacher, “Network Applications of Bloom Filters: A Survey,” Internet Mathematics, Vol. 1, No. 4, pp. 485–509, 2004.

[13] J. Song, H. Takakura, and Y. Kwon, “A generalized feature extraction scheme to detect 0-day attacks via IDS alerts,” International Symposium on Applications and the Internet, (SAINT), Turku, Finland, pp. 55–61, 2008.

[14] KBBI, “Visualisasi,” 2016, http://kbbi.web.id/visualisasi. [Last Accessed: 09-Feb-2017].

[15] S. Geravand and M. Ahmadi, “Bloom filter applications in network security: A state-of-the-art survey,” Computer Networks, Vol. 57, No. 18, pp. 4047–4064, 2013