Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 5, No. 3, August 2020, Pp. 169-178

Visualization of granblue fantasy game traffic pattern using deep packet inspection method

Deris Stiawan[*1], Christian Prabowo[2], Ahmad Heryanto[3], Nurul Afifah[4], Agus Eko Minarno[5], Rahmat Budiarto[6]
Universitas Sriwijaya, Indonesia[1,2,3,4]
Universitas Muhammadiyah Malang, Indonesia[5]
Albaha University, Saudi Arabia[6]

Article Info
Keywords:
Granblue Fantasy, RPG, Cygames
Deep Packet Inspection, Data Traffic
Feature Extraction, Gacha, Solo Raid
Casino, Multiraid, TTL

Article history:
Received 06 May 2020
Revised 31 May 2020
Accepted 18 July 2020
Published 31 August 2020

Cite:
Stiawan, D., Prabowo, C., Heryanto, A., Afifah, N., Minarno, A., & Budiarto, R. (2020). Visualization of Granblue Fantasy Game Traffic Pattern Using Deep Packet Inspection Method. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 5(3). doi:https://doi.org/10.22219/kinetik.v5i3.1073

*Corresponding author.
Deris Stiawan
E-mail address:
deris@unsri.ac.id

Abstract
Granblue Fantasy is one of Role Playing Games (RPG developed by Cygames. This research observes the Gandblue Fantasy Game with the purpose is to analyze its traffic data to find patterns trough Deep Packet Inspection (DPI) technique. The stages involve in analysis process are construction of the dataset by capturing the traffic data, features extraction, features selection and lastly pattern visualization. The Patterns are Gacha, Solo Raid, Casino and Multiraid. Experiment results show that the Multiraid battle uses a lot of data than the other patterns with 1400 Ip Length.

1. Introduction

Role Playing Game (RPG) is a type of game which prioritizes the storyline and the development of the main characters to determine the course of the game. One type of RPG gamea that so popular is Granblue Fantasy [1]. Granblue Fantasy is a video role-playing game developed by Cygames. This game uses a turn-based system where players and opponents take turn actions such as attacking monsters to win the game. Players can invite other players from all over the world to help fight monsters that are almost impossible to fight alone to win with teamwork. On the other hand, the monster that is impossible to defeat alone actually gives a challenge to some players, then comes the players who can defeat the monster alone and it becomes special satisfaction for these players. Then the developer takes the initiative to create a game for players who can complete a solo battle against. The monsters are impossible to defeat themselves. When players already have had great equipments and characters, they tend to do e-sports. E-sport is a predicate called the Most Valuable Player (MVP). Deep Packet Inspection (DPI) is a method for checking data packages in internet traffic [2]. This method is usually used to find a hidden pattern in data packet that is usually contained in headers, protocols, payloads and others [3]. DPI is also useful for classifying traffic data based on the port number and protocol used [3]. This work selects the DPI because it can quickly and efficiently find desired patterns. Before checking the data packet, the data packet is captured by using WireShark, an open source sniffer. According research work in [4] the genre or type of online game has different traffic patterns, because there are different parameters in the traffic patterns generated by each different type of games. Those parameters will be used for searching patterns in the traffics that have been captured. Research carried out by Kim et al. [5] analyzes data traffic on an MMORPG Line Age II game. Some parameters are examined such as data packet size, bandwidth size, inter-arrival time and Round Trip Time (RTT). The study concludes that the characteristics of the game traffic are small packet size, large bandwidth, fast inter-arrival time, and fast round trip time to avoid delay or lag.

Previous research in [6] discusses the pattern recognition in the online game Dragon Nest. The work explains how to find patterns in Dragon Nest game traffic and analyze the player's habit patterns. Research work in [7] uses the DPI method to identify traffic data. Data traffic is captured using WireShark, then extracted using feature extraction, and then analyzed to see whether there is a pattern in the traffic. Research in [7] uses the Correlation based Feature Selection (CFS). CFS was implemented to reduce the raw data size and improve the accuracy. Stiawan et al. [8] use

selection feature process in obtaining key features such as port destination, flags, ip length and packet length, and report that the accuracy of traffic analysis was improved. Winanto et al. [9] state that the small number of features can complicate the pattern recognition process. Thus, selecting features from raw data is more effective. Feature selection can reduce the dimensionality of the data. Large data consumes a lot of memory and power resource. By using feature selection, pattern visualization is faster and easier to construct. The main contribution of this study is a selection of key features of Granblue Fantasy game traffic using CFS, and its visualization using the DPI method.

This paper is organized as follows, Section I gives introduction, Section II describes the data and material along with the methodology used in this research. Section III discusses the obtained result and its analysis. Lastly, Section IV conludes the study.

2. Research Method

In this paper, DPI is used to find the patterns in the Granblue Fantasy game traffic. Patterns can be found by looking at the feature of the traffic [10][11], e.g.: IP addresses, payloads, protocols and others.

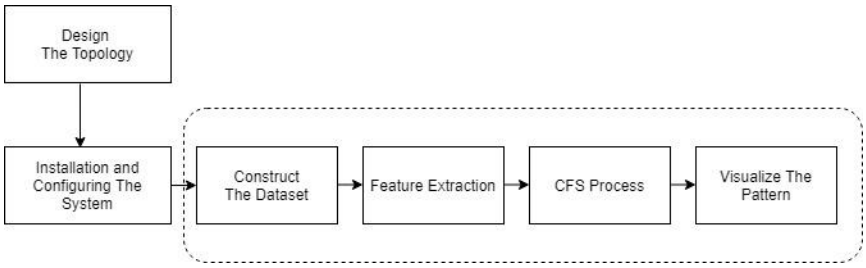

Figure 1. Research Methodology

Figure 1 shows the research steps diagram. The first step is to build the topology and then followed by installation and configuration of the wireShark program and the Chrome browser on PC. The next step is dataset creation by sniffing the network traffic using WireShark. The sniffed traffic is store in the dataset in the form of *.Pcap* then is processed and extracted using feature extraction. The following step is feature selection process using CFS and DPI to get the patterns. Finally, visualize the patterns.

2.1 Experimental Set Up

Figure 2 depicts the topology design for the experiments. The client-server topology is chosen. Four scenarios are prepared for the experements.

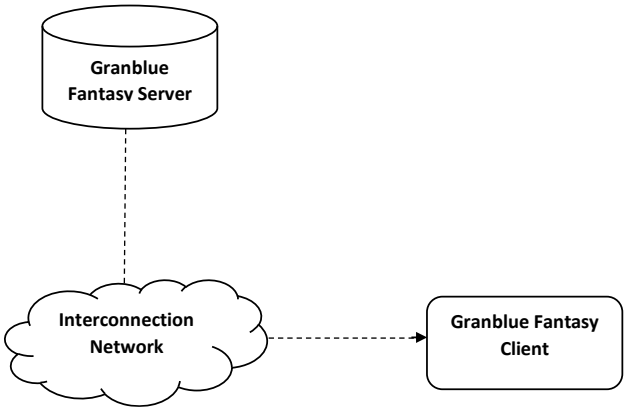

Figure 2. Topology Design

2.2 Hardware and Software Specification

Hardware specifications used in this work is PC as a client running Microsoft windows 7, 32-bit and attached to a network with access to the internet through a router. Hardware Specifications are presented in Table 1.

Table 1. Hardware Specifications

Function	Specifications
Client	*Microsoft Windows* 7 32-bit
Network	Smartfren Andromax M3Y

Softwares used for the expereiments include WireShark, Python programming language, and internet browser. The WireShark as a traffic sniffer [12][13]. The python is used to implement feature extraction and CFS process. The last is Chrome as the internet browser. Software specification is presented in Table 2.

Table 2. Software Specification

Tool	Function	Description
WireShark	Traffic Sniffer	WireShark 2.6.6 Windows
Python	Feature extraction	Python 3
Chrome Browser	Browser	Version 73.0.3683.103

2.3 Features Extraction

The detail of the feature extraction process is explained in Figure 3. The extracted features are then selected appropriately and will be used for visualization purpose. All These features run at the Transport Layer.

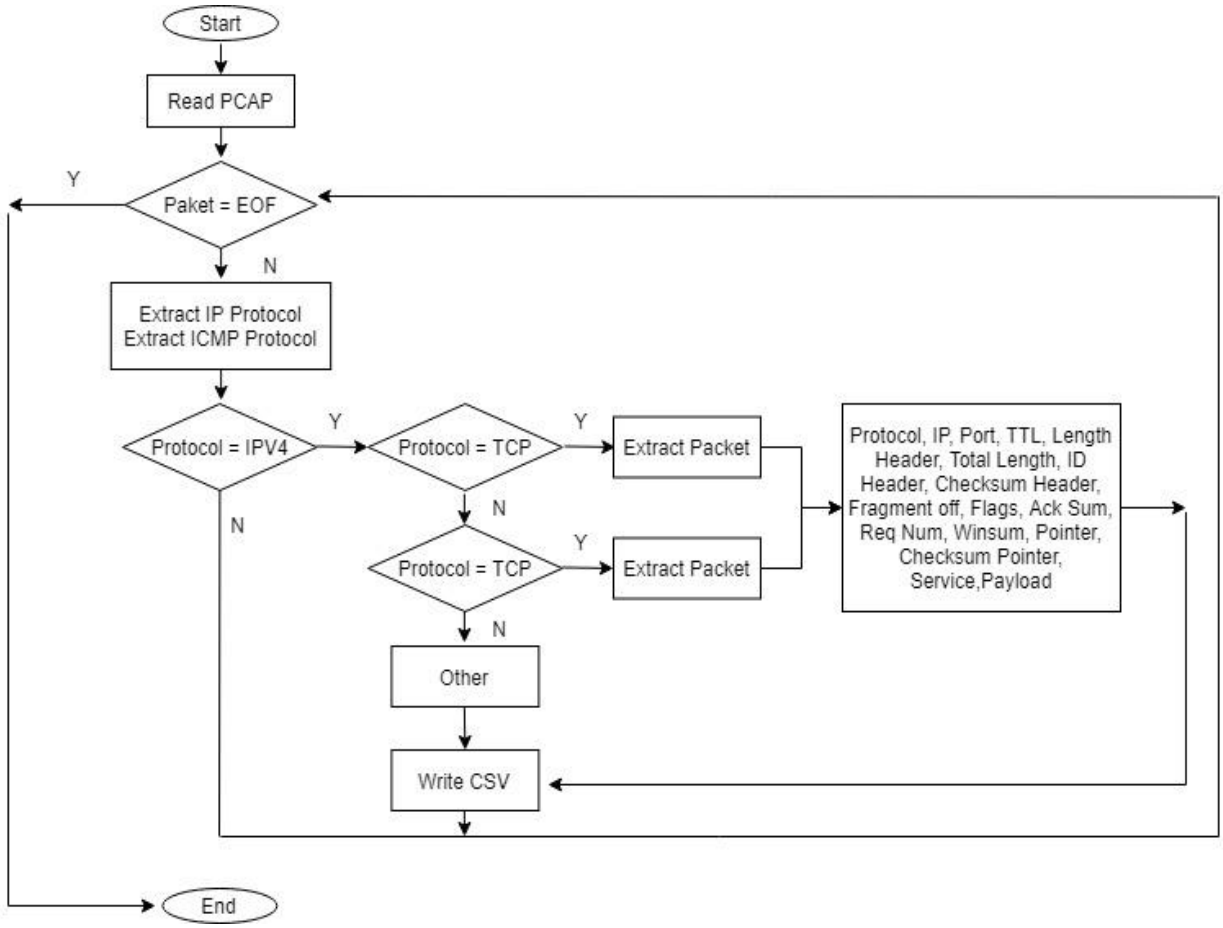

Figure 3. Feature Extraction Process

Traffic that has been captured will produce raw data, the data is difficult to read and understand because the header is unique and hidden in the protocol [14]. The features extracted from the raw data are used as parameters that represent the whole traffic data [15] and can simplify and speed up the traffic classification/detection process. Table 3 shows the features names and their ID numbers.

Table 3. Feature List

Feature ID#	Feature Name
1	Packet Number
2	Tempstems
3	Service
4	Ip source
5	Ip destination

6	Port source
7	Port destination
8	Sequence
9	Acknowledge
10	Windows
11	Flags
12	TTL
13	Ip length
14	Ip checksum
15	Ip_id
16	Ip_off
17	Packet length
18	Protokol
19	Payload

2.4 Correlation based Feature Selection (CFS)

This work uses the Correlation based Feature Selection (CFS) method. It belongs to the heuristic method by looking each field used for inter class predictions field with correlations between levels. Formulae for Correlation-based Feature Selection (CFS) are as follow.

$$r_{sk} = \frac{kr_{cf}}{\sqrt{k + k\,(k-1)r_{ff}}} \quad (1)$$

$$CFS = max_{sk}\left[\frac{r_{cf1} + r_{cf2} \ldots + r_{cfk}}{\sqrt{k + k\,(k-1)r_{ff}}}\right] \quad (2)$$

r_{sk} is relation between the fields. *K* is the number of field, r_{cf} is average the field and r_{ff} is correlation between the field. The CFS value will be computed using Equation 1 and Equation 2.

2.5 Deep Packet Inspection

Deep Packet Inspection (DPI) is a method for viewing the contents of data packets on internet traffic [16], in other words DPI filters the interconnection network. DPI detects the data (packets contents) and signatures (packet ID). The DPI filters the network by analyzing the signature payload on the packet using the string maching algorithm or using the expression matching algorithm.

DPI analyzes the signatures to understand data packages in an application. Signature is a unique sign contained in an application. The signature is stored into a database, so the detection engine can classify the traffic. This research uses DPI to identify/recognize patterns by analyzing signatures on traffic payloads [17].

2.5.1 Pattern Based Signature Algorithm

Pattern based signature is the way of DPI for identifying the network traffic. In the network traffic, the signatures were captured using WireShark. From all of the features, only seven features were selected by CFS. The selected features were classified using DPI method to get the patterns. The considered patterns are Gacha, Solo Raid, Kasino and Multiraid. The filtering process is as follow.

$$\text{Pattern} = \begin{cases} \text{Gacha } if\ doing\ the\ battles\ in\ the\ first\ time \\ \text{Solo Raid } if\ doing\ the\ raid\ battle\ independently\ in\ a\ day\ for\ three\ times \\ \text{Kasino } if\ doing\ the\ casino\ poker\ for\ five\ minutes\ in\ five\ times \\ \text{Multiraid } if\ doing\ the\ raid\ battles\ with\ many\ players\ and\ do\ it\ in\ three\ times \end{cases}$$

2.6 Visualization

Visualization is a method to delivering information of the traffic data to get the image form with the aim to understand the information provided. In this work, visualization of the Granblue Fantasy game traffic [18] patterns is done using parallel coordinate graphs [19][20]. A parallel coordinate graph is a graph with coordinate lines that present data dimensionality. The used features will be distinguished by different colors. The CFS features that will be used for visualization are Ip Src, Ip Lenght, Ip Dest, Port Src, Port Dest, Flags and TTL.

3. Results and Discussion

3.1 Sniffing Dataset

In the experiment, up to 20 MB of raw data of Granblue Fantasy traffic were captured by using WireShark in form of (.pcap).and contains up to 29,341 rows of data. TCP and UDP traffics dominate the protocols that appear in traffic data. The traffic data percentage is calculated as the number of data packets divided by the total data packets and multiplied by one hundred percent. The total of all data packets is 83,787 (100%), with total TCP packets was 81906 (98%), total UDP packets was 943 (1%) and other packages was 938 (1%). The entire TCP packet comes from the online game Granblue Fantasy, while the UDP comes from Domain Name Server (DNS) traffic. Table 4 presents the captured data for each scenario.

Table 4. Sniffing Dataset

Scenario	Trial	Protocol			Total
		TCP (%)	UDP (%)	Etc (%)	
1	1	5376 (98%)	62 (1%)	32 (1%)	5470 (100%)
	2	5313 (98%)	58 (1%)	12 (1%)	5383 (100%)
	3	4733 (97%)	82 (2%)	29 (1%)	4844 (100%)
2	1	10234 (98%)	68 (1%)	103 (1%)	10405 (100%)
	2	2748 (97%)	30 (1%)	58 (2%)	2836 (100%)
	3	3118 (96%)	62 (2%)	72 (2%)	3252 (100%)
3	1	1635 (95%)	47 (3%)	28 (2%)	1710 (100%)
	2	2044 (98%)	24 (1%)	29 (1%)	2097 (100%)
	3	2286 (96%)	50 (2%)	44 (2%)	2380 (100%)
4	1	28752 (98%)	250 (1%)	339 (1%)	29341 (100%)
	2	7720 (98%)	108 (1%)	95 (1%)	7923 (100%)
	3	7947 (98%)	102 (1%)	97 (1%)	8146 (100%)
Total		81906 (98%)	943 (1%)	938 (1%)	83787 (100%)

3.2 Feature Extraction Result

The traffic packets were captured on April 19th 2019, for duration of 2 hours 58 minutes and 50 seconds. Figure 4 depicts a screenshot of the packet capturing in the experiment and displays IP source of 203.104.248.7 and IP destination of 192.168.1.103. The source port is using port 80 and destination port is using port 57467. The number of bytes that can be sent was 4080. The Flags used were SYN and ACK. The TTL value was 238. IP length was 48. The checksum value in decimal is 19920 and Data Reader was 4DD0. The value of packet identification in decimal was 47479 and Data Reader is B977. TCP length was 28.

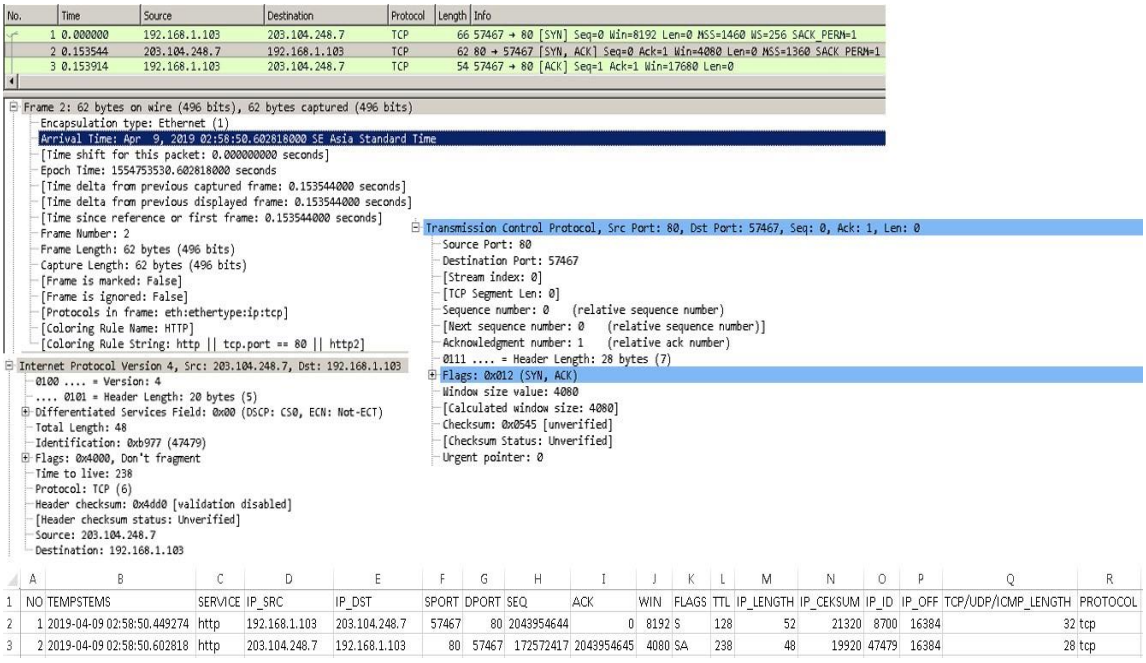

Figure 4. Screenshot of Feature Extraction Result

3.3 Correlation-based Feature Selection (CFS) Result

Feature selection in CFS process filters only seven features, i.e.: features with ID: 4, 5, 6, 7, 11, 12 and 13 out of 19 features. They are IP Src, IP Lenght, IP Dest, Port Src, Port Dest, Flags and TTL.

3.4 The DPI Result

The seven features selected from CFS process will be used by the DPI method to classify the traffic in the dataset. Table 5 shows the classification results. It is observed that Multiraid traffic pattern uses the most of the data with 1400 IP Length.

Table 5. Feature in CFS Result

Pattern	IP Src	Port Src	IP Dest	Port Dest	Flags	TTL	IP Length
Gacha	120	443	103	56175	10	43	1100
Solo Raid	168	443	103	57193	110	236	1001
Kasino	7	80	103	56683	210	238	1103
Multiraid	5	112	103	49906	160	237	1400

3.5 Data Visualization of the Granblue Fantasy Game Traffic

The visualization technique used in this work is parallel coordinates. Parallel coordinates is a visualization technique that describes dimensional data and attributes that are used by more than one type of attributes by using different colors for each type of attribute to distinguish them. The attributes used for visualization are the normalized results displayed in Table 6, namely ip src, src port, ip dest, port etc., flags, ttl and ip length, which will be displayed in the visualization as IP Src, IP Lenght, IP Dest, Port Src, Port Dest, Flags and TTL.

Figure 5 shows a pattern when a player performs a gacha. Each attribute has its own place called coordinate which is shown by pulling the line from top to bottom, with values ranging from 0 to 300. The IP used is 192.168.1.101 as the player's IP. IPs for the servers (destination IPs) are: 203.104.248.7, 103.23.4.120, 103.23.4.168, 202.166.185.168, 202.166.185.169. The port used by the player is a dynamic port with a value range between 40,000 and 70,000, while the server uses a static port with a value range of 80 to 443. Then the flags are 10 for ACK, 50 for FIN, 100 for PSH, 150 for RST, 200 for SYN, 60 for FIN ACK, 110 for PSH ACK, 160 for RST ACK and 210 for SYN ACK. The length of the IP from the server side varies between 40 to 1400, the player side also varies between 40 to 1400.

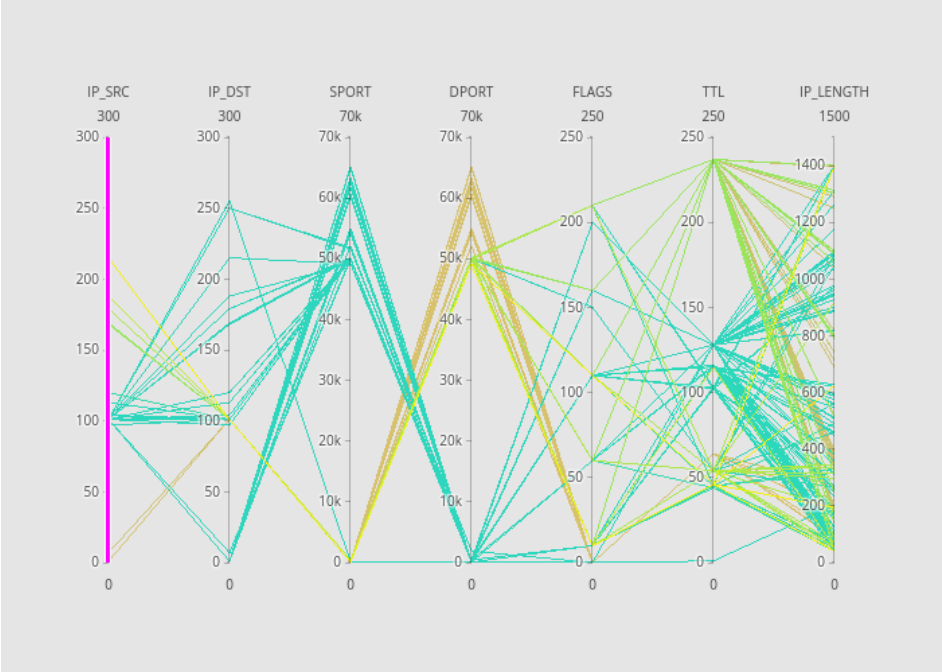

Figure 5. Gacha Traffic Pattern Visualization

The TTL from the server side is 44, 45 and 237, while from the player side is 128. The length of the IP from the player side is more numerous and dense than the server side because the player makes a lot of data requests from the server to display gacha goods data that will be received by the player.

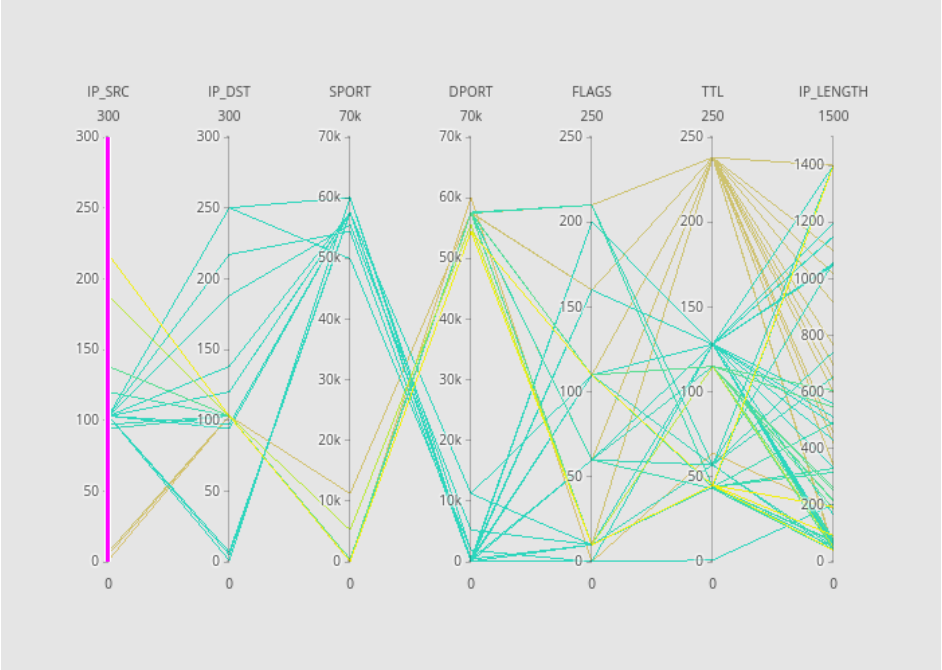

Figure 6. Solo Raid Pattern Visualization

Figure 6 shows the Solo Raid traffic pattern. The IP used by the player is 192.168.1.103. While the IP used by the servers are 203.104.248.5, 203.104.248.7, 103.23.4.120, 103.23.4.168, 202.166.185.168, 202.166.185.169. The port used by the player is a dynamic port with a value range between 40,000 to 65,000, while the server uses a static port with a value range of 80, 443 and 11230. The flags are 10 for ACK, 50 for FIN, 100 for PSH, 150 for RST, 200 for SYN, 60 for FIN ACK, 110 for PSH ACK, 160 for RST ACK and 210 for SYN ACK. The server's length IP varies between 40 and 1400, the player side also varies between 40 and 1400. The server sides' TTL are 43, 44, 236 and 238, while the player side is 128. The IP length in Solo Raid traffic pattern is a bit lower compared to the Gache pattern because the player makes a data request that is slightly directly proportional to the action of the player that is done a little while doing a Solo Raid, while the IP length of the server is quite less than the length of the player's IP. Besides, a lot of data were cut off during data normalization,

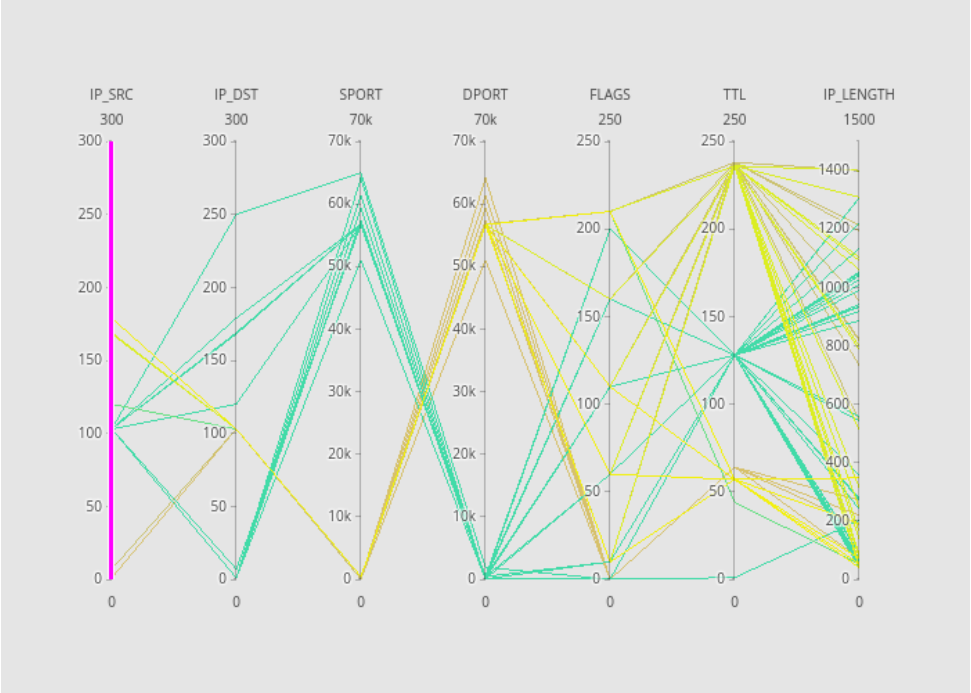

Figure 7. Casino Traffic Pattern Visualization

Figure 7 shows the traffic pattern when a user plays Casinos. The IP used by the player is 192.168.1.103. While the IP used by the servers are 203.104.248.7, 103.23.4.120, 103.23.4.168, 202.166.185.168, 202.166.185.169. Ports used by players are dynamic ports ranging from 50,000 to 70,000, while servers use static ports with ranges of 80 and 443. Flags are 10 for ACK, 50 for FIN, 100 for PSH, 150 for RST, 200 for SYN, 60 for FIN ACK, 110 for PSH ACK, 160 for RST ACK and 210 for SYN ACK. The length of the IP from the server side varies between 40 and 1400; the player side also varies between 40 and 1400. The TTL from the server side is 43, 44, 236 and 238, while the player side is 128. The length of the IP from the player side looks more centered on an area in the picture, because the data requested by the player is done repeatedly.

The last pattern is a Multiraid pattern. The pattern when a user plays Multiraid is shown in Figure 8. The IP used by the player is 192.168.1.101. While the IP used by the server is 203.104.248.5, 203.104.248.7, 103.23.4.120, 103.23.4.168, 202.166.185.168, 202.166.185.169. The port used by the player is a dynamic port with a range between 40,000 and 70,000, while the server used is a static port with a range of 80, 443 and 11230. The flags are 10 for ACK, 50 for FIN, 100 for PSH, 150 for RST, 200 for SYN, 60 for FIN ACK, 110 for PSH ACK, 160 for RST ACK and 210 for SYN ACK. The length of the IP from the server side varies between 40 and 1400; the player side also varies between 40 and 1400. The TTL from the server side is 44, 45 and 237, while the player side is 128. The IP length of the Multiraid pattern looks higher and denser than Solo Raid, because many players are in one raid space so the data generated by the server is more.

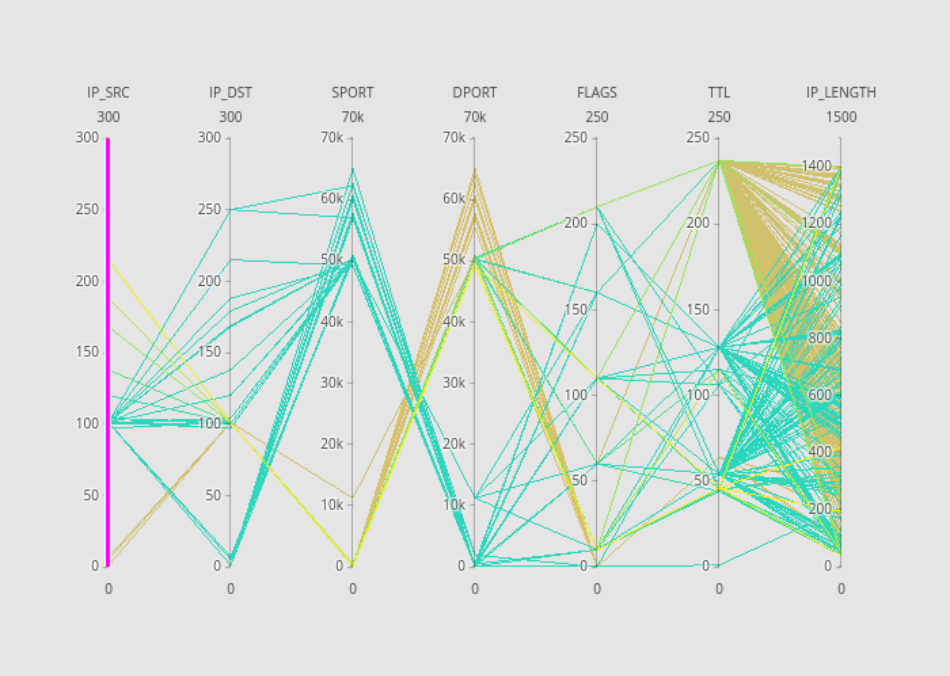

Figure 8. Multiraid Traffic Pattern Visualization

4. Conclusion

In this paper, the authors have proposed a new scheme for visualizing traffic patterns of the Granblue Fantasy Game. To reduce the size of raw data and memory, this work uses CFS as it has a good performance to get the key features. The CFS filters seven features for classifying the traffic using DPI. The experiment result shows that Multiraid battle uses a lot of data compared to other games because many players are in one raid battle. Contrary, the Casino game uses a little data in one scenario session. This work also shows that network traffic visualization makes the traffic analysis process easier. As for future work, the authors consider security level in Granblue Fantasy Game against the hacker.

Acknowledgement

This research is supported by Ministry of Education and Culture of the Republic of Indonesia under research grand from Directorate of Research and Community Services, Universitas Sriwijaya.

References

[1] App Annie, “Top grossing apps and download statistics google play | app annie.”
[2] J. Svoboda, “Network traffic analysis with deep packet inspection method,” Fac. Informatics Masaryk Univ., no. Master’s Thesis, 2014.

[3] Y. Afek, A. Bremler-barr, and Y. Koral, "Space Efficient Deep Packet Inspection of Compressed Web Traffic," No. 259085, Pp. 1–14, 2012.
[4] X. Che and B. Ip, "Packet-level traffic analysis of online games from the genre characteristics perspective," J. Netw. Comput. Appl., Vol. 35, No. 1, Pp. 240–252, 2012. https://doi.org/10.1016/j.jnca.2011.08.005
[5] J. Kim, J. Choi, D. Chang, T. Kwon, Y. Choi, and E. Yuk, "Traffic characteristics of a massively multi-player online role-playing game," Pp. 1, 2006. https://doi.org/10.1145/1103599.1103619
[6] D. Aryandi, "Pengenalan Pola Behavior Game Dragon Nest Menggunakan Metode Bloom Filter," 2017.
[7] T. Sasut, A. Valianta, and D. Stiawan, "Klasifikasi Trafik Terenkripsi Menggunakan Metode Deep Packet Inspection (Dpi)," Vol. 2, No. 1, Pp. 424–429, 2016.
[8] D. Stiawan, A. H. Abdullah, and M. Y. Idris, "Classification of habitual activities in behavior- based network detection," Journal of Computing, vol. 2, no. 8, 2010.
[9] E. A. Winanto, A. Heryanto, and D. Stiawan, "Visualisasi Serangan Remote to Local (R2L) Dengan Clustering K-Means," Annu. Res. Semin. 2016, Vol. 2, No. 1, Pp. 359–362, 2016.
[10] C. Carter, A. El Rhalibi, M. Merabti, and M. Price, "Homura and net-homura: The creation and web-based deployment of cross-platform 3D games," 2009 Int. Conf. Ultra Mod. Telecommun. Work., 2009. https://doi.org/10.1109/ICUMT.2009.5345337
[11] Cisco, "Traffic Classification," WAN Appl. Optim. Solut. Guid., Pp. 1–12, 2008.
[12] M. Finsterbusch, C. Richter, E. Rocha, J. A. Müller, and K. Hänßgen, "A survey of payload-based traffic classification approaches," IEEE Commun. Surv. Tutorials, Vol. 16, No. 2, Pp. 1135–1156, 2014. https://doi.org/10.1109/SURV.2013.100613.00161
[13] T. T. T. Nguyen and G. Armitage, "A Survey of Techniques for Internet Traffic Classification using ML," IEEE Comst, Vol. 10, No. 4, Pp. 56–76, 2008. https://doi.org/10.1109/SURV.2008.080406
[14] Z. Cao, G. Xiong, Y. Zhao, Z. Li, and L. Guo, "A Survey on Encrypted Traffic Classification," in Applications and Techniques in Information Security, 2014, Pp. 73–81. https://doi.org/10.1007/978-3-662-45670-5_8
[15] 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. https://doi.org/10.1016/j.jbusres.2014.10.006
[16] 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. https://doi.org/10.1016/j.phpro.2012.03.229
[17] 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. https://doi.org/10.1016/j.procs.2016.07.250
[18] Linktionary.com, "TCP (Transmission Control Protocol)," 2001.
[19] 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.
[20] Shi, Lei et al, "Scalable network traffic visualization using compressed graphs" 2013 IEEE International Conference on Big Data, Big Data 2013. https://doi.org/10.1109/BigData.2013.6691629