Identifying Service Contexts for QoS Support in IoT Service Oriented Software Defined Networks

Hong Jin Kim, Moon Yong Jung, Won Sang Chin, and Ju Wook Jang(✉)

Department of Electronics Engineering, Sogang University, Seoul 04107, Korea
{chii92,myjung,mokey82,jjang}@sogang.ac.kr

Abstract. An important challenge for supporting variety of applications in the Internet of Things is the network traffic engineering and virtual network technologies such as SDN (Software Defined Network). To assign virtual network, it require service context (QoS) however, identifying service context is not easy. For that reason, the proliferation of new applications use port numbers already known (e.g. HTTP = 80). In addition, the encrypted packets (e.g. HTTPS) make it difficult to identify service contexts. This paper presents an identifying scheme for service contexts from real network traffic to support service-oriented IoT network. We use statistical properties of network traffic such as mean packet length, mean interpacket arrival time, and standard deviation interpacket arrival time to identify service contexts (e.g. Video Streaming, Video Conference, File Transfer Service). The contribution of our approach is in identifying services which have not been identified by previous methods. We devise a scheme which incrementally add dimensions to separate services until all services are identified. For example, Video Streaming and FTP shows identical statistical properties when we examine by two dimensions (MPL: Mean Packet Length, MIAT: Mean Inter-Arrival Time), hence not separable. However, if we add one more dimension (SDIAT: Standard Deviation of Inter-Arrival Time), the two services can be clearly separated. Our scheme can be used to find out which traffic needs what QoS in combined traffics, which can be used for traffic engineering in SDN.

Keywords: IoT · Network context · Service context · Statistical property · MPL · MIAT · SDIAT

1 Introduction

The number of sensors deployed around the world is growing at a swift speed. Naturally, large amount of data is being gathered from the devices, hence using data approach has been enlarged. In addition, the more data emerged, the more we need to define the data. For this, towards moving to the internet of things, context is considered to be extremely important. Perera et al. [1] claims, context-aware computing allows us to store context information linked to sensor data, therefore the interpretation can be done easily and more meaningfully. Furthermore, understanding context makes it easier to fulfill

S. Bouzefrane et al. (Eds.): MSPN 2017, LNCS 10566, pp. 99–108, 2017.
DOI: 10.1007/978-3-319-67807-8_8

machine to machine (M2M) communication, as it is a core element in IoT vision. Accordingly, there have been several surveys conducted in relation to this field.

In identifying context, Figo et al. [2] claims that device can understand user's performance, such as walking or running, with analyzing accelerometer data. It is one example of identifying user context from device context. Eisenman et al. [3] presents BikeNet, a mobile sensing system for mapping the cyclists' experience. They claim that users could gain empirical knowledge of important factors, such as exposure to air and noise pollution, and danger due to car density with BikeNet. This can be explained by identification of service context from user and device contexts. To identify service context either way, we can use network context.

Our study initiated from building service oriented platforms, similar to Paganelli [4]. In this paper, as a first step, we identified service contexts from network contexts to lay the ground work for Software defined network (Fig. 1).

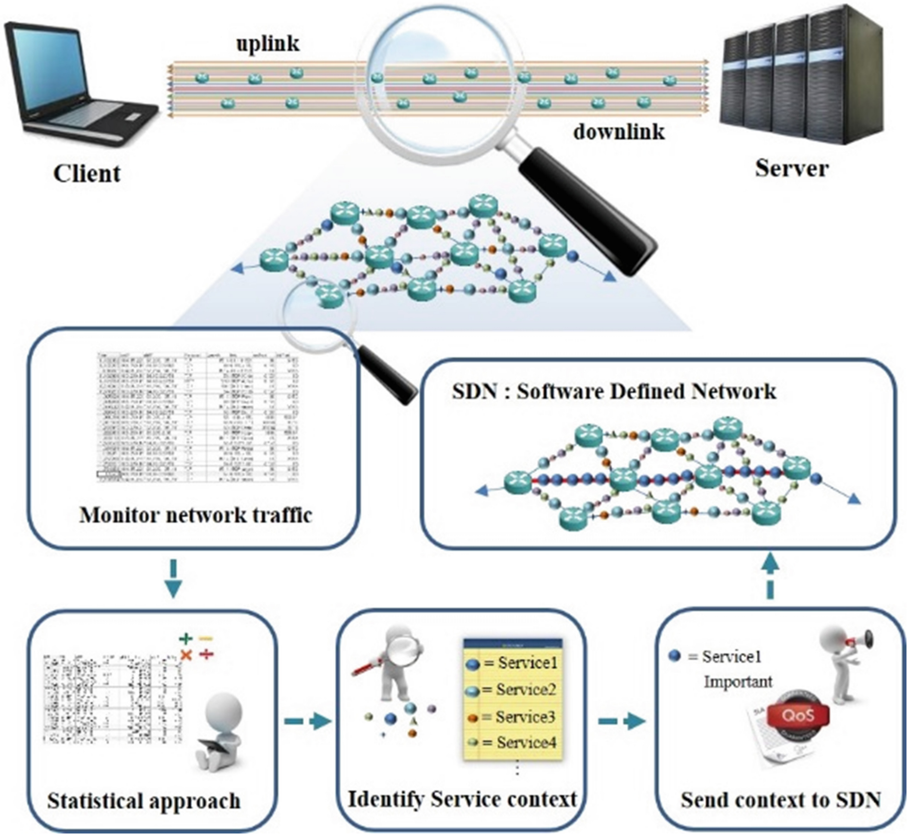


Fig. 1. Overall architecture

The remainder of this paper is arranged as followed. The different traffic classification methods are reviewed in Sect. 2. Section 3 presents our methodology and outline experimental results, respectively. Section 4 presents our conclusions.

2 Related Works

Due to its fundamental nature and its basis in other techniques, the field of traffic classification (to identify service) has maintained everlasting interest. For instance, Port-based approach is the most common technique for identifying internet network applications. For now, however, it's not easy to classify the network as well as identify service contexts due to the proliferation of new applications. Several new applications have no IANA registered ports, but instead use ports that are already registered (e.g. HTTP = 80, HTTPS = 443). In addition, application developers and users use assigned ports to disguise their traffic and circumvent filtering or firewalls. Furthermore, pervasive deployment of network and port address translation make it hard to classify traffic (e.g. several physical servers may offer services through the same public IP address but on different ports) [5].

As applications and user behaviors appeared on port-based flow classification undependable, payload-based approaches emerged. Payload-based approach, sometimes called deep packet inspection (DPI), relies on specific application data. This method can further divide into two parts which are protocol decoding - where the application protocol data has been used, and signature-based identification - where a search will be carried out to identify application's specific byte sequence in packet payload [6].

Nonetheless, it is easily circumvented by encryption, protocol obfuscation or encapsulation (e.g. tunneling traffic in HTTP), and prohibitively computationally expensive for general use on high-bandwidth links. These concerns with payload based techniques have motivated researchers to seek new discriminating properties of traffic classes and other classification techniques.

In other way, Erman et al. [7] claims, using a clustering approach is called clustering for the network traffic identification problem. We pursue this clustering approach, and in particular by using network statistical properties.

Classification involves two stages; sets of features with known traffic classes (creating "rules"), and applying these rules to classify unknown traffic and identify service contexts.

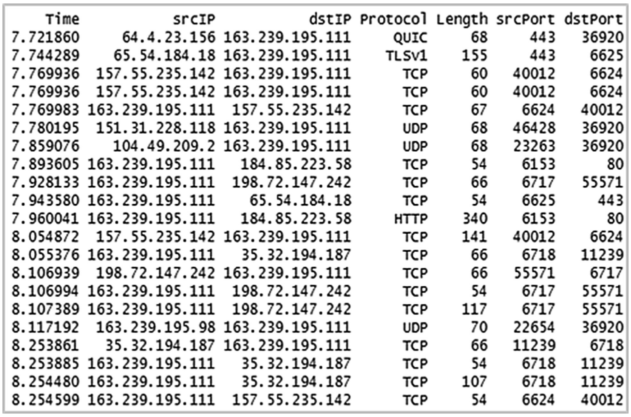
3 Acquiring Service Context Through Traffic Classification

3.1 Target Services

In this paper, our target service contexts are Video streaming, Video conference and File transfer service. Since these three applications need QoS based routing, in extreme cases, Video streaming needs less than 1% loss, less than 30 ms jitter, and less than 150 ms latency for their services [8]. YouTube and Netflix are examples of these kinds of Video streaming. Skype and Google Hangouts are included in Video conference. For File transfer service, various web services that use HTTP (Port 80) or HTTPs (Port 443) are examples.

3.2 Data Acquisition and Pre-processing

Data acquisition was carried out using the Wireshark packet sniffer. This network packet analyzer is able to capture network packets and tries to display that packet data as detailed as possible. We collected data packets from the client's side on Window OS environment. Figure 2 shows collected packet dissection data as a CSV file from Wireshark. Then we read it through R program, which is a statistical analysis software for data pre-processing. Dataset consists of *Timestamp*, *Source IP*, *Destination IP, Protocol, Packet Length, Source Port Number*, and *Destination Port Number.*



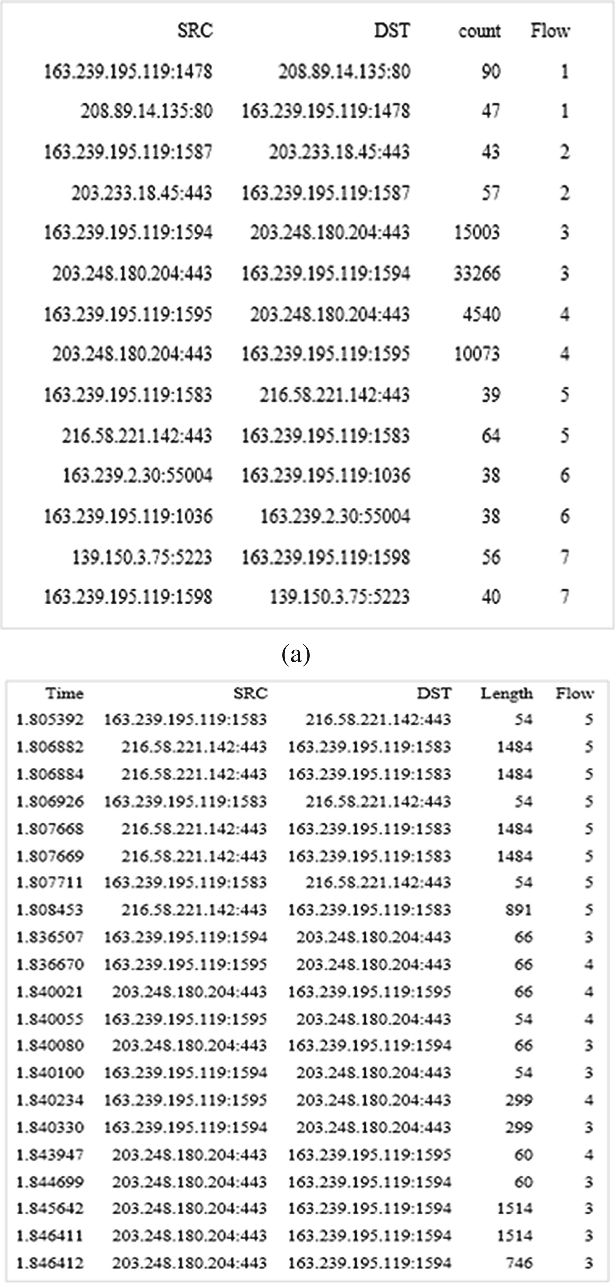
```
    Time           srcIP           dstIP Protocol Length srcPort dstPort
7.721860     64.4.23.156 163.239.195.111     QUIC     68     443   36920
7.744289    65.54.184.18 163.239.195.111    TLSv1    155     443    6625
7.769936  157.55.235.142 163.239.195.111      TCP     60   40012    6624
7.769936  157.55.235.142 163.239.195.111      TCP     60   40012    6624
7.769983 163.239.195.111  157.55.235.142      TCP     67    6624   40012
7.780195  151.31.228.118 163.239.195.111      UDP     68   46428   36920
7.859076    104.49.209.2 163.239.195.111      UDP     68   23263   36920
7.893605 163.239.195.111    184.85.223.58     TCP     54    6153      80
7.928133 163.239.195.111  198.72.147.242      TCP     66    6717   55571
7.943580 163.239.195.111    65.54.184.18      TCP     54    6625     443
7.960041 163.239.195.111    184.85.223.58    HTTP    340    6153      80
8.054872  157.55.235.142 163.239.195.111      TCP    141   40012    6624
8.055376 163.239.195.111    35.32.194.187     TCP     66    6718   11239
8.106939  198.72.147.242 163.239.195.111      TCP     66   55571    6717
8.106994 163.239.195.111  198.72.147.242      TCP     54    6717   55571
8.107389 163.239.195.111  198.72.147.242      TCP    117    6717   55571
8.117192  163.239.195.98 163.239.195.111      UDP     70   22654   36920
8.253861   35.32.194.187 163.239.195.111      TCP     66   11239    6718
8.253885 163.239.195.111    35.32.194.187     TCP     54    6718   11239
8.254480 163.239.195.111    35.32.194.187     TCP    107    6718   11239
8.254599 163.239.195.111  157.55.235.142      TCP     54    6624   40012
```

Fig. 2. A CSV dataset imported from Wireshark

To obtain meaningful dataset, we constructed a flow table based on *SRC* (srcIP + srcPort), *DST* (dstIP + dstPort) 2 tuple (Fig. 3a). We set the value of K (count threshold) as 25, since we want to focus on large flows as previously stated(hence excluded DNS, SNMP, NBNS, and other mice flows) Fig. 3(b), shows the result of preprocessing. It consists of *Time, SRC, DST, packet lengths* and *flow*. Flows are classified by SRC, DST pair and seven bidirectional flows are identified.

3.3 Identifying Flows Based on Legacy Method

Legacy port-based classification method could be applied on our dataset. In this case, we can only identify HTTP and HTTPs among the classes. In other words, the majority of flows use same port (e.g. class 2, 3, 4, 5) that makes it difficult to interpret its application (Table 1). Therefore, we can find out that port based classification is not an efficient way to identify dataset that were collected from web services. Also in payload based classification case, as mentioned above, HTTPs traffics are encrypted so that it is



SRC	DST	count	Flow
163.239.195.119:1478	208.89.14.135:80	90	1
208.89.14.135:80	163.239.195.119:1478	47	1
163.239.195.119:1587	203.233.18.45:443	43	2
203.233.18.45:443	163.239.195.119:1587	57	2
163.239.195.119:1594	203.248.180.204:443	15003	3
203.248.180.204:443	163.239.195.119:1594	33266	3
163.239.195.119:1595	203.248.180.204:443	4540	4
203.248.180.204:443	163.239.195.119:1595	10073	4
163.239.195.119:1583	216.58.221.142:443	39	5
216.58.221.142:443	163.239.195.119:1583	64	5
163.239.2.30:55004	163.239.195.119:1036	38	6
163.239.195.119:1036	163.239.2.30:55004	38	6
139.150.3.75:5223	163.239.195.119:1598	56	7
163.239.195.119:1598	139.150.3.75:5223	40	7

(a)

Time	SRC	DST	Length	Flow
1.805392	163.239.195.119:1583	216.58.221.142:443	54	5
1.806882	216.58.221.142:443	163.239.195.119:1583	1484	5
1.806884	216.58.221.142:443	163.239.195.119:1583	1484	5
1.806926	163.239.195.119:1583	216.58.221.142:443	54	5
1.807668	216.58.221.142:443	163.239.195.119:1583	1484	5
1.807669	216.58.221.142:443	163.239.195.119:1583	1484	5
1.807711	163.239.195.119:1583	216.58.221.142:443	54	5
1.808453	216.58.221.142:443	163.239.195.119:1583	891	5
1.836507	163.239.195.119:1594	203.248.180.204:443	66	3
1.836670	163.239.195.119:1595	203.248.180.204:443	66	4
1.840021	203.248.180.204:443	163.239.195.119:1595	66	4
1.840055	163.239.195.119:1595	203.248.180.204:443	54	4
1.840080	203.248.180.204:443	163.239.195.119:1594	66	3
1.840100	163.239.195.119:1594	203.248.180.204:443	54	3
1.840234	163.239.195.119:1595	203.248.180.204:443	299	4
1.840330	163.239.195.119:1594	203.248.180.204:443	299	3
1.843947	203.248.180.204:443	163.239.195.119:1595	60	4
1.844699	203.248.180.204:443	163.239.195.119:1594	60	3
1.845642	203.248.180.204:443	163.239.195.119:1594	1514	3
1.846411	203.248.180.204:443	163.239.195.119:1594	1514	3
1.846412	203.248.180.204:443	163.239.195.119:1594	746	3

(b)

Fig. 3. Flow table (a), Pre-processed packet dissection dataset (b)

also difficult to find applications. We applied this method on our dataset which we obtained by wireshark (Sect. 3.2). However, 57% of the traffic were encrypted, making identifying service contexts hard. To solve these problems, we use statistical properties approach to identify service contexts.

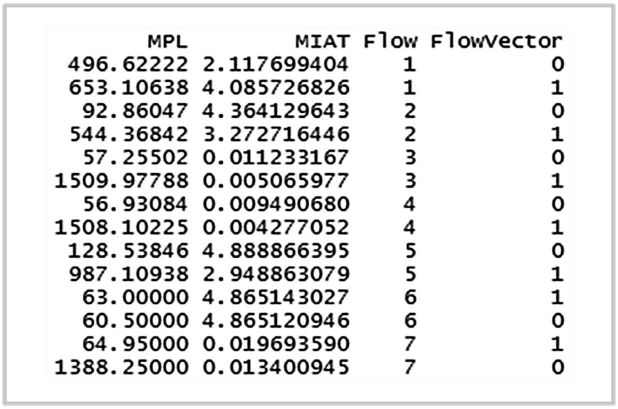
Table 1. Port-based classification

Port	Class	Protocol
80	1	HTTP
443	2,3,4,5	HTTP
55004	6	Unregistered
5223	7	XMPP

3.4 Identification Based on Statistical Properties Flows

To use statistical classification, selecting a feature is the most significant way. Accordingly, there have been a lot of works conducted in relation to this field [9]. Roughan et al. [10] claims that average packet length and flow duration are the most important features to classify network traffic data. However, Roughan et al. [10] do not separately identify uplink traffic and downlink traffic. We refine identification of service contexts by considering uplink traffic and downlink traffic separately. We improved their method by introducing directions of flows (Client to Server or Server to Client), since we want to divide unidirectional transmission and duplex transmission.

Firstly, mean packet length (MPL) and mean interpacket arrival time(MIAT) are used. Figure 4 shows the result of flows direction with downlink (server to client) and uplink (client to server) state by mean packet length and mean interpacket arrival time property. We represented it on two dimension spaces for easy understanding. As Fig. 5, we notice that each of the flows indicate different statistical properties.



```
       MPL        MIAT Flow FlowVector
 496.62222 2.117699404    1          0
 653.10638 4.085726826    1          1
  92.86047 4.364129643    2          0
 544.36842 3.272716446    2          1
  57.25502 0.011233167    3          0
1509.97788 0.005065977    3          1
  56.93084 0.009490680    4          0
1508.10225 0.004277052    4          1
 128.53846 4.888866395    5          0
 987.10938 2.948863079    5          1
  63.00000 4.865143027    6          1
  60.50000 4.865120946    6          0
  64.95000 0.019693590    7          1
1388.25000 0.013400945    7          0
```

Fig. 4. Classification of dataset flows with direction

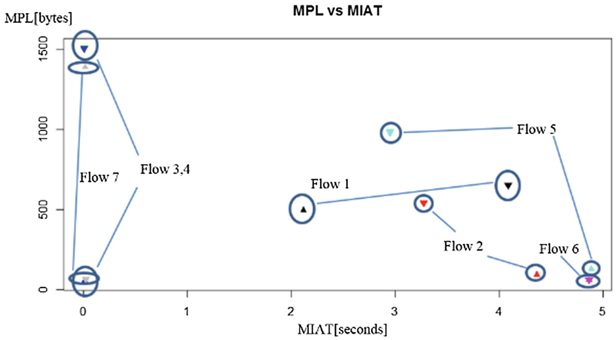


Fig. 5. Classification of dataset flows (2-dimension)

Subsequently, to identify our target services (Streaming, Video conference, File transfer service), we collect the representative traffics with packet sniffer. The clustering result was obtained through 10 experiments.

Figure 6 shows that result of services' mean of µ-packet length, mean of µ-interpacket arrival time. From this we can see the apparent differences of uplink and downlink statistical characteristics (especially the uplink MPL) between Video streaming flow and Video conference flow.

In contrast, Video streaming and File transfer service have similar features (MPL, MIAT). Still, using only two features does not seem to be enough to identify their services. (e.g. In Fig. 5 we try to identify flow 3 and 4, yet we do not know which is Steaming or File transfer). To solve this problem, the service is identified through protocol analysis and by adding features as followed.

3.5 MPL, MIAT and Standard Deviation Interpacket Arrival Time (SDIAT)

We now need a new dimension which separates MPEG-DASH [12] from FTP. We choose Standard Deviation Interpacket Arrival Time (SDIAT) as a new dimension based on the following observation: MPEG-DASH [12] is an adaptive bitrate streaming technique [Wikipedia]. MPEG-DASH works by breaking the content into a sequence of small HTTP-based file segments, each segment containing a short interval of playback time of content that is potentially many hours in duration, such as a movie or the live broadcast of a sports event. The content is made available at a variety of different bit rates, i.e., alternative segments encoded at different bit rates covering aligned short intervals of play back time are made available. While the content is being played back by an MPEG-DASH client, the client automatically selects from the alternatives the next segment to download and play back based on current network conditions. The client selects the segment with the highest bit rate possible that can be downloaded in time for play back without causing stalls or re-buffering events in the playback. Thus, an MPEG-DASH client can seamlessly adapt to changing network conditions.

Our reasoning is that since MPEG-DASH adjusts its transmission rate to available bandwidth and buffer space for each client, its traffic may exhibit higher standard deviation of inter-packet arrival time(SDIAT) than FTP. Both MPEG-DASH and FTP use TCP to adapt to network conditions in transport layer. However, MPEG-DASH adds adaptation in application layer, i.e. client program chooses segments of different bit-rate. Based on this reasoning we add SDIAT to differentiate MPEG-DASH from FTP. Figure 7 shows a three dimensional classification where MPL, MIAT and SDIAT serve as coordinates. Note that adding SDIAT as a new coordinate separates MPEG-DASH from FTP. Recall that MPEG-DASH and FTP are inseparable in a two dimensional classification as in Fig. 6.

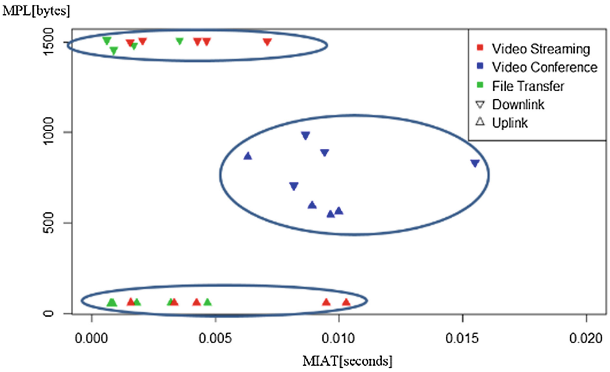


Fig. 6. Classification representative traffics (YouTube, Skype, File transfer service)

Figure 7 shows Video streaming and File transfer service in 3D graph, which have similar properties. FTP services and Video Streaming send almost the same packet length near 1500. However, in downlink SDIAT, video streaming service has long SDIAT than File transfer service as we conjectured in the above.

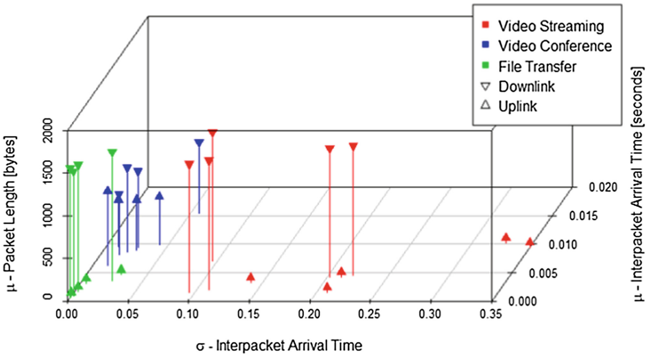


Fig. 7. Classification of File transfer service and DASH video stream

In this sense, we could identify both service contexts that have similar network properties. Thus, we can identify our target contexts (that were using same ports or not registered on IANA, which were so difficult to identify by legacy method) through MPL, MIAT and SDIAT.

Lastly, we applied the above method on our dataset (Fig. 8). Based on video streaming characteristics in the graph, video streaming flows are identified as flow 3 and 4.

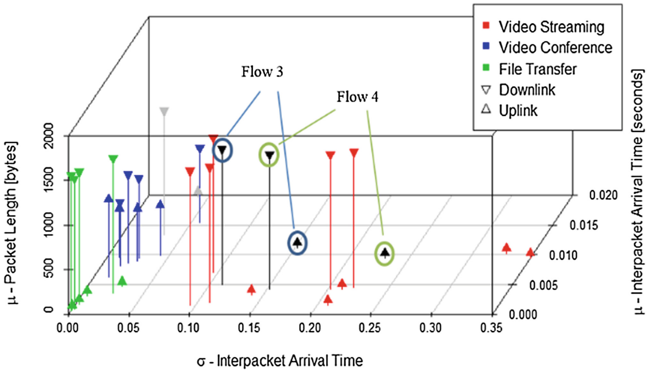


Fig. 8. Classification our dataset in 3D to identify video streaming

4 Conclusion

In this paper, we have introduced identification of service contexts from network contexts as Table 2. Port based classification scheme is used to infer the traffic service, but several ports are already used in same number (e.g. HTTP = 80, HTTPs = 443) so it is hard to classify. The other method, Payload based classification is also difficult to classify, since deep packet inspection method increases system complexity and processing load. In order to derive service contexts from network, we proposed network statistical properties approach. We assumed the traffic flows have statistical properties. To obtain meaningful flow classes, we utilized traffic packet lengths, interpacket arrival time and etc. Also, we introduced our traffic identification method in three stages. First, we demonstrated legacy port based classification. Second, we used traffic packet lengths and interpacket arrival time to identify services (e.g. YouTube and Skype). Finally, to classify analogous properties, we added standard deviation (e.g. DASH

Table 2. Identified services by statistical property method

Methods	Protocol
Legacy (Port based)	None
Statistical property (MPL, MIAT)	Video conference
Statistical property (MPL, MIAT, SDIAT)	Video streaming Video conference File transfer

Video stream and File transfer service). Future works will be carried out to verify new traffics with machine learning methodology and also place a priority on captured traffic to guarantee the network QoS.

Acknowledgement. This research was supported by the Institute for Information & Communications Technology Promotion(IITP) grant funded by the Korean government(MSIP) (No. 2017-0-00325, Block chain based IoT Reliability and Management System Development).

References

1. Perera, C., et al.: Context aware computing for the Internet of Things: a survey. Commun. Surv. Tutor. IEEE **16**(1), 414–454 (2014)
2. Figo, D., et al.: Preprocessing techniques for context recognition from accelerometer data. Pers. Ubiquitous Comput. **14**(7), 645–662 (2010)
3. Eiseman, S.B., et al.: BikeNet: a mobile sensing system for cyclist experience mapping. ACM Trans. Sens. Netw. (TOSN) **6**(1), 6 (2009)
4. Paganelli, F., Ulema, M., Martini, B.: Context-aware service composition and delivery in NGSONs over SDN. Commun. Mag. IEEE **52**(8), 97–105 (2014)
5. Dainotti, A., Pescape, A., Claffy, K.C.: Issues and future directions in traffic classification. Netw. IEEE **26**(1), 35–40 (2012)
6. Maldeniya, S.L., Atukorale, A.S., Vithanage, W.W.: Network data classification using graph partition. In: 2013 19th IEEE International Conference on Networks (ICON). IEEE (2013)
7. Erman, J., Arlitt, M., Mahanti, A.: Traffic classification using clustering algorithms. In: Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data, pp. 281–286. ACM (2006)
8. Szigeti, T., et al.: End-to-End QoS Network Design: Quality of Service for Rich-Media & Cloud Networks, pp. 170–172. Cisco Press (2013)
9. Nguyen, T.T.T., Armitage, G.: A survey of techniques for internet traffic classification using machine learning. Commun. Surv. Tutor. IEEE **10**(4), 56–76 (2008)
10. Roughan, M., Sen, S., Spatscheck, O., Duffield, N.: Class-of-service mapping for QoS a statistical signature based approach to IP traffic classification. In: Proceedings of the 4th ACM/SIGCOMM Conference on Internet Measurement, Taormina, pp. 135–148. ACM (2004)
11. Seufert, M., et al.: A survey on quality of experience of HTTP adaptive streaming. Commun. Surv. Tutor. IEEE **17**(1), 469–492 (2014)
12. MPEG: Dynamic Adaptive Streaming over HTTP (DASH), ISO/IEC 23009 (2012)