Master's Thesis

Title

Hierarchical Clustering Approach Based on Swarm Intelligence for Real-time Flow Classification

Supervisor

Professor Masayuki Murata

Author

Takumi Sue

February 12th, 2016

Department of Information Networking

Graduate School of Information Science and Technology

Osaka University

Master's Thesis

Hierarchical Clustering Approach Based on Swarm Intelligence for Real-time Flow Classification

Takumi Sue

Abstract

As the Internet has become playing an important role in our society, the number of types of the applications provided through the Internet increases. The requirements of the applications depend on the type of applications. The interactive applications such as online game require the communication with low latency. On the other hand, the file transfers for the cloud storages require the communication with a large bandwidth. The network should accommodate the traffic of such applications so as to satisfy the requirements of the applications.

The classification of the flows based on their corresponding application is necessary to accommodate the traffic with different applications, satisfying their requirements. The traditional classification of the traffic uses the port numbers. However, in recent years, a large number of types of the applications, such as YouTube and network game, have become provided through HTTP or HTTPS. All of these applications use 80 or 443 port. As a result, the flow classification using the port numbers is no longer applicable in the Internet.

Therefore, new methods to classify the traffic without using the port numbers have been developed. The clustering is one of the most commonly used approaches. By using the clustering, the clusters are constructed so that each cluster includes the flows with the similar features. The flow classification methods based on the clustering first construct the clusters offline by using the traffic traces. Then, the newly monitored flows are classified by comparing their features with the constructed clusters.

However, these methods have the following two problems. The first problem is that these methods do not consider the case that the new applications emerges. One approach to handling the new applications is to periodically run the clustering methods to learn the features of the new applications. However, this approach cannot classify the traffic of the new application before running the clustering methods.

Another problem is that these methods assume that the accurate features of the flow are obtained before classifying the traffic. However, the accurate features may not be able to be obtained before the flow is completed. One approach to classifying the flows before the flow is completed is to obtain the features when a predefined number of packets are monitored. By setting the required number of packets to the small value, we can classify the flow soon after the flow arrives. However, the features obtained by monitoring a small number of packets may be inaccurate, and some applications may be classified into wrong cluster.

In this thesis, we propose a new hierarchical clustering method for real-time flow classification. Our method constructs the hierarchical cluster, updating the features of the traffic flows; the features are updated every time a packet arrives, and the cluster which a flow belongs to in each layer is determined after the features become accurate enough. If the features of the flow are not accurate enough to determine the cluster the flow belongs to, our method waits another packet to improve the accuracy of the features.

The hierarchical clusters are useful to estimate the characteristics of the flow when the flow is related to an unknown application. In the hierarchical cluster, the flows are clustered into a small number of groups in the upper layer, and the flows belonging to the same group in the upper layers are clustered into several groups in the lower layer. If flows of a new application construct a new group in the lower layer, but they are classified in an existing group in the upper layer, the characteristic of the new application can be estimated from the characteristic of the existing group.

We evaluated our clustering method using the actual traffic traces. The results indicate that our method construct the similar cluster as the method that uses the accurate features; our method achieves the accuracy of 88% when 3 clusters are constructed, and achieves the accuracy of 74% when 5 clusters are constructed. In addition, the results demonstrated that our method classifies the flows by monitoring only a small number of packets; 81 % of the flows are classified into 3 clusters before 30 packets arrive, and 99 % of the flows are classified into 5 clusters before 40 packets arrive.

Keywords

Flow Classification

Hierarchical Clustering

Contents

List of Figures

1 Introduction

As the Internet has become playing an important role in our society, the number of types of the applications provided through the Internet increases; the various new applications such as online games, video streaming services and cloud storage services have been deployed over the Internet. The requirements of the applications depend on the type of applications. The interactive applications such as online game require the communication with low latency. On the other hand, the file transfers for the cloud storages requires the communication with a large bandwidth. The network should accommodate the traffic of such applications, satisfying the requirements of the applications.

Future network architectures that can accommodate traffic of different types of applications with different requirements have been proposed. One of the architectures is based on the network virtualization [1, 2]. This architecture enables to run multiple virtual networks over the network. By constructing a virtual network for each type of applications, this architecture can accommodate different types of applications. In this architecture, the classification of the flows based on the types of the applications is required at the gateway of the network. Then, the traffic is relayed to the virtual network corresponding to the type of the application.

The traditional classification of the flows uses the port numbers [3, 4]. However, in recent years, a large number of types of the applications, such as YouTube and network game, have become provided through HTTP or HTTPS [5]. All of these applications use 80 or 443 port. As a result, the flow classification using the port numbers is no longer applicable in the Internet.

Therefore, new methods to classify the flow without using the port numbers have been developed [3, 4, 6]. The traffic generated by the applications has characteristics that can be used to identify the types of the applications. Thus, these methods learn the characteristics based on the machine learning techniques.

There are two kinds of the methods based on the machine learning. One of the methods is the supervised method. In this method, the classes are predefined, and the training data set, which include the features of the flows and the labels indicating the corresponding classes, are given as an input. Then, the rules to classify the traffic are learned based on the training data set. The methods based on the supervised machine learning can be applicable only in the case that the labels of the flows are known. However, it is difficult to prepare the training data set including the

suitable labels for the flows if the new applications, which are unknown when the classification system starts, emerge.

On the other hand, the unsupervised method does not require the training data set including the labels. Clustering is one of the representative unsupervised methods. The clustering constructs the clusters so that each cluster includes the data with the similar features. The flow classification methods based on the clustering have been proposed. These methods construct the clusters offline by using the collected traffic traces. Then, they classify the monitored flows online by selecting the cluster which is the closest to the features of the flow.

However, these methods have the following two problems. The first problem is that these methods do not consider sufficiently the case that the new applications emerge. One approach to handling the new applications is to periodically run the clustering method to learn the features of the new applications. However, this approach cannot classify the flows of the new application before running the clustering methods.

Another problem is that these methods assume that the accurate features of the flow are obtained before classifying the flow. However, the accurate features may not be able to be obtained before the flow is completed. One approach to classifying the flows before the flow is completed is to obtain the features when a predefined number of packets are monitored. By setting the required number of packets to the small value, we can classify the flow soon after the flow arrives. However, the features obtained by monitoring a small number of packets may be inaccurate, and some applications may be classified into wrong clusters.

In this thesis, we propose a new hierarchical clustering method for real-time flow classification. Our method constructs the hierarchical cluster, updating the features of the traffic flows; the features are updated every time a packet arrives, and the cluster which a flow belongs to in each layer is determined after the features become accurate enough. If the features of the flow are not accurate enough to determine the cluster the flow belongs to, our method waits another packet to improve the accuracy of the features.

The hierarchical clusters are useful to estimate the characteristics of the flows when the flow is related to an unknown application. In the hierarchical cluster, the flows are clustered into a small number of groups in the upper layer, and the flows belonging to the same group in the upper layers are clustered into several groups in the lower layer. If flows of a new application construct a new group in the lower layer, but they are classified in an existing group in the upper layer, the

characteristic of the new application can be estimated from the characteristic of the existing group.

The rest of this thesis is organized as follows. Section 2 explains the related work. Section 3 introduces our method. Section 4 evaluates our method. Section 5 concludes this thesis.

2 Related Work

2.1 Flow Classification

Roughan et al. proposed a method to classify the traffic through the supervised machine learning [7]. In this method, the newly obtained data is classified as the class of its nearest neighbor in the training data set. Zhang et al. improved the accuracy of the nearest neighbor approach by incorporating the correlation information of the flows when the training data set is small [8].

Moore et al. used another machine learning approach, Naïve Bayes to classify the flows [9]. In this method, the Naïve Bayes classifier is trained by using the hand-classified network data. Then, the new flows are classified by the Naïve Bayes classifier. Nguyen et al. also used the Naïve Bayes to classify the flows [10]. This method uses the statistics of a small number of most recent packets to obtain the features. Then, by applying the Naïve Bayes, this method classify the current traffic flows. Zhang et al. proposed another method based on the Naïve Bayes classifier [11]. In this method, flow correlation information is modeled by bag-of-flow. Then, the features are extracted to represent traffic flows. The traffic is classified by aggregating the output of the Naïve Bayes classifiers using the extracted features.

Li et al. proposed a method to construct the decision tree from the training data set [12]. Then the flows are classified based on the decision tree.

Jin et al. proposed a flow classification method using multiple simple linear binary classifiers [13]. In this method, each classifier can be easily trained. Then, combining the multiple classifiers, we can accurately classify the traffic.

The supervised machine learning methods described above require the training data set. However, as we discussed in Section 1, it is difficult to prepare the training data set including the suitable labels for the traffic if new applications, which are unknown when the classification system starts, emerge.

Another approach of the machine learning is the unsupervised method. Clustering is one of the representative unsupervised methods, and constructs the clusters so that each cluster includes the data with the similar features.

Erman et al. applied the clustering method to the flow classification [14]. They used the K-means method, DBSCAN, and AutoClass, and demonstrated that these clustering algorithms can classify the flow accurately. Bernaille et al. proposed a method to classify the flow online

by using the K-means method [15]. In this method, the clusters are constructed offline by using the training data set. Then, flows are classified online by searching the cluster corresponding to the flow. However, this approach cannot classify the traffic which corresponds to the application, which was not included in the training data set. The clustering method can be used to solve this problem. Zhang et al used the K-means method to detect the unknown flows [16].

Recently, Wang et al. extended the K-means method to improve the accuracy of the flow classification [17]. In this method, the accuracy is improved by considering the information of the flow inferred from the packet headers as the constraint on the clustering.

However, the existing flow classification methods based on the clustering have the following two problems. (1) These methods do not consider sufficiently the case that the new applications emerge. Though the method proposed by Zhang et al [16] can detect the unknown flows, it cannot estimate the characteristic of the new flows. (2) These methods assume that the accurate features of the flow are obtained before classifying the flow. However, the accurate features may not be able to be obtained before the flow is completed.

In this thesis, we propose a clustering method which solves the above problems. Our clustering method can be applicable to the existing flow classification method based on the clustering, because our clustering method can be run by using any features of the flows.

2.2 Clustering

There are many algorithms to construct the clusters.

K-means is one of the most popular clustering algorithms. In the K-means method, the number of clusters k is given as a parameter. Then, the k clusters are constructed so as to minimize the distance from each data point to the center of the cluster the data point belongs to, which is defined by the mean of the data points within the cluster. However, the result of the K-means only indicates the cluster which each data point belongs to. Thus, we cannot understand the similarity of the data points belonging to different clusters.

The hierarchical clustering methods can solve this problem. In the hierarchical clustering methods, the data points are clustered into a small number of groups in the upper layer, and the data points belonging to the same group in the upper layers are clustered into several groups in the lower layer. By constructing the hierarchical groups, we can identify the similarity of the groups.

The ClusTree is one of the hierarchical clustering methods, which allow to update the clusters

online [18]. This method constructs the tree, including two kinds of nodes, inner node and leaf node. The leaf node indicates a fine-grained cluster, and has the pointer to the feature values of the corresponding data points. The inner node corresponds to the coarse-grained cluster, which summarizes the data points included in its children. The inner node has the pointer to the aggregated features of the data points included in its children called as Clustering Feature or Cluster Feature. In the ClusTree, the clusters can be updated by (1) searching the leaf node corresponding to the new data point from the root node, and (2) updating the features on the path from the root node to the leaf node. They propose maintaining CFs by extending index structures from the R-tree family [19–21].

The ClusTree can be updated online, but does not consider the case that the features are inaccurate. Therefore, in this thesis, we propose a new clustering method based on the ClusTree, considering the case that the features are inaccurate.

AntTree [22] is another hierarchical clustering method. AntTree is a method inspired by the behavior of the ants constructing a tree. In this process, each data point acts as an ant; each data point walks around the tree to find the node similar to itself, and connects it to the found node. In the AntTree, the nodes in the constructed tree are corresponding the data points. Therefore, it is difficult to interpret the constructed tree hierarchically; we need to determine the clusters the data points corresponding to the inner nodes belong to when constructing the fine-grained cluster. However, the idea of the AntTree is useful to handle the inaccuracy of the features. In the AntTree, each data point has a function to determine whether the other data points are similar to it. Then, each data point walks around the tree based on the function. Though each ant select its behavior by comparing the similarity with the threshold defined based on the number of nodes the data points visited in the original AntTree, we can consider the inaccuracy of the features by using the function based on the accuracy of the features. Therefore, in this thesis, we introduce the method based on the AntTree to search the cluster each data point belongs to.

3 Hierarchical Clustering Approach for Real-time Flow Classification

3.1 Overview

In this thesis, we develop a new clustering method for the real-time flow classification. We assume our clustering method is used in the traffic classifier, which classifies the flow passing the classifier. The classifier has the definition of flows. When the traffic classifier receives a packet, it identifies the flow the packet belongs to. Then, the traffic classifier updates the features of the flow.

Our method constructs the hierarchical cluster, updating the features of the traffic flows; the features are updated every time a packet arrives, and the cluster which a flow belongs to in each layer is determined after the features become accurate enough. If the features of the flow are not accurate enough to determine the cluster the flow belongs to, our method waits another packet to improve the accuracy of the features.

In this process to determine the clusters, we use an approach based on the AntTree [22], which is a method inspired by the behavior of the ants constructing a tree. In the AntTree, an ant, which corresponds to an item required to be classified, walks around the tree based on the behavior selected by itself. Then, if the similar node is found, the ant connects it to the found node. In our method, each ant uses the information of the inaccuracy of the corresponding features to select its behavior. By doing so, we can construct the clusters, considering the inaccuracy of the features.

In our method, by constructing the hierarchical clusters, we can estimate the characteristic of new clusters; if flows of a new application construct a new cluster in the lower layer, but they are classified in an existing cluster in the upper layer, the characteristic of the new application can be estimated from the characteristic of the existing cluster.

Our method has the following advantages. (1) Our method can identify the clusters a new flow belongs to accurately as soon as possible, because our method searches the clusters, considering the accuracy of the features. (2) The similarity of the flows belonging to the different clusters can be obtained by constructing the hierarchical clusters. (3) Our method can identify the flows of the new applications, because our method updates the clusters online.

3.2 Data Structure

In this thesis, we construct the hierarchical cluster based on ClusTree [18]. Figure 1 shows the data structure of the constructed cluster.

This data structure has two kinds of nodes, *inner node* and *leaf node*. The leaf node indicates a fine-grained cluster, and includes l to L flows. Each leaf node has the pointers to the feature values of the corresponding flows. The inner node corresponds to the coarse-grained cluster, which summarizes the flows included in its children. That is, by constructing the tree of inner nodes, we can hierarchically construct clusters; the inner node near root node corresponds to the coarser-grained cluster, and the node near leaf corresponds to the finer-grained cluster.

The inner node constructs the tree structure by connecting it to m to M children. Each inner node has the entries corresponding to its children. Each entry has the abstracted clustering features and the pointer to the corresponding child. The abstracted clustering features corresponding to the child c have the following values.

- The number of flows included in the abstracted features n_c
- The sum of the features S^{linear}, which is calculated by

$$S_c^{\text{linear}} = \sum_{f \in F_c} S_f$$

  where F_c is the set of flows included in cluster of the node c, and S_f is a vector indicating the clustering features of the flow f.

In our method, there are flows that have not been classified into any clusters. We maintain such flows in a list called *unclassified flows*.

3.3 Process to Cluster Flow

3.3.1 Flow Insertion

If a flow included in the unclassified flows exists and its features are updated, we run the process to find the cluster the updated flow belongs to. We use the process to find the cluster based on the AntTree [22], which is inspired by the behavior of the ants constructing a tree. In this process, an agent corresponding to each flow acts as an ant; each agent walks around the tree to find the node similar to the flow, and inserts it to the found node.

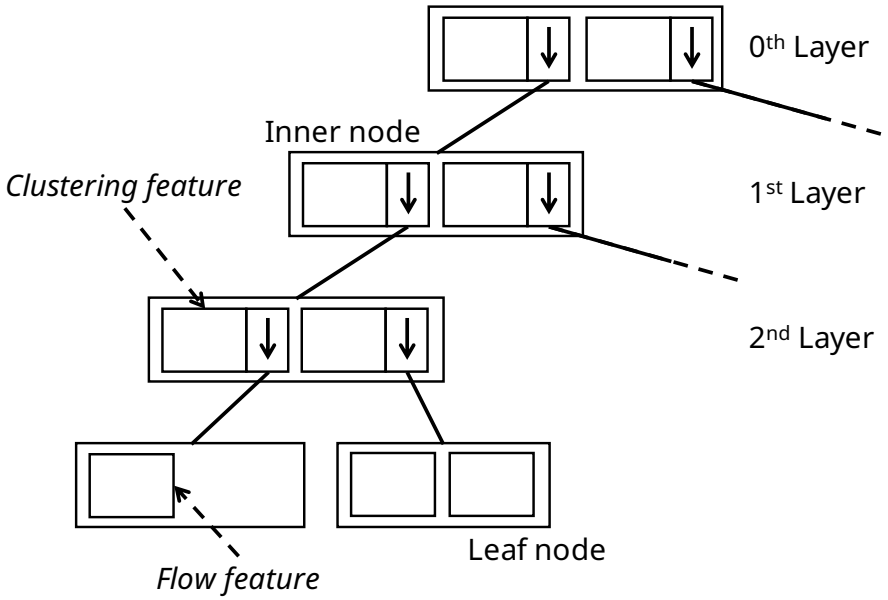


Figure 1: Data Structure

In our method, the agent for the flow f^{new} moves over the tree based on the following two functions.

The first function $F_1(\epsilon_{f^{\mathrm{new}}})$ is a function to check whether the current features are sufficiently accurate. $F_1(\epsilon_{f^{\mathrm{new}}})$ is defined by

$$F_1(\epsilon_{f^{\mathrm{new}}}) = (N - n) \times \epsilon_{f^{\mathrm{new}}} \tag{1}$$

where n is a number of monitored packets of the flow f^{new}, and N is a predefined parameter. $\epsilon_{f^{\mathrm{new}}}$ is defined by

$$\epsilon_{f^{\mathrm{new}}} = |S_{f^{\mathrm{new}}} - S'_{f^{\mathrm{new}}}|$$

where $S'_{f^{\mathrm{new}}}$ is a vector indicating the clustering features before updating the features. $F_1(\epsilon_{f^{\mathrm{new}}})$ indicates the expected difference between the current features and the features after N packets are monitored. Thus, the flow whose $F_1(\epsilon_{f^{\mathrm{new}}})$ is small is regarded as the flow with sufficiently accurate features.

Another function $F_2(d_{c_1^{p_{f^{\mathrm{new}}}}}, d_{c_2^{p_{f^{\mathrm{new}}}}})$ is a function which compares the most similar cluster and the second most similar cluster among the clusters corresponding to the children of the current location of the agent for the flow f^{new}. Hereafter $p_{f^{\mathrm{new}}}$ is the current location of the agent.

$F_2(d_{c_1^{p_{f^{\text{new}}}}}, d_{c_2^{p_{f^{\text{new}}}}})$ is defined by

$$F_2(d_{f^{\text{new}},c_1^{p_{f^{\text{new}}}}}, d_{f^{\text{new}},c_2^{p_{f^{\text{new}}}}}) = \frac{d_{f^{\text{new}},c_1^{p_{f^{\text{new}}}}}}{d_{f^{\text{new}},c_2^{p_{f^{\text{new}}}}}}. \tag{2}$$

where $c_1^{p_{f^{\text{new}}}}$ is the cluster which is the most similar to f^{new}, $c_2^{p_{f^{\text{new}}}}$ is the cluster which is the second most similar to f^{new}. $d_{f^{\text{new}},c}$ is difference between the features of the flow f^{new} and the features of the cluster c, which is defined by

$$d_{f^{\text{new}},c} = \left| S_{f^{\text{new}}} - \frac{S_n^{\text{linear}}}{n_c} \right|$$

By checking $F_2(d_{c_1^{p_{f^{\text{new}}}}}, d_{c_2^{p_{f^{\text{new}}}}})$, we investigate the certainty that the flow f^{new} should be included in the most similar cluster; if $F_2(d_{c_1^{p_{f^{\text{new}}}}}, d_{c_2^{p_{f^{\text{new}}}}})$ is small, the flow f^{new} should be included in the most similar cluster even if the features are inaccurate. But, on the other hand, if $F_2(d_{c_1^{p_{f^{\text{new}}}}}, d_{c_2^{p_{f^{\text{new}}}}})$ is large, we should wait until the features become accurate, because the second similar cluster is also similar to f^{new}.

In our method, the agent for the flow f^{new} moves over the tree by performing the following rule once per one arrival packet.

1. If $F_1(\epsilon_{f^{\text{new}}}) \leq T_1^{f^{\text{new}}} \times A^{h-1}$

   (a) if $p_{f^{\text{new}}}$ has no children, make a leaf node and insert the pointer to the features of f^{new} to the newly added node

   (b) if $p_{f^{\text{new}}}$ has only one child,

      i. if the child is the leaf node, insert the pointer to the features of f^{new} to the child.

      ii. otherwise, go to the child.

   (c) if $p_{f^{\text{new}}}$ has plural children

      i. if $F_2(d_{c_1^{p_{f^{\text{new}}}}}, d_{c_2^{p_{f^{\text{new}}}}}) \leq T_2^{f^{\text{new}}}$,

         A. if $c_1^{p_{f^{\text{new}}}}$ is the leaf node, insert the pointer to the features of f^{new} to $c_1^{p_{f^{\text{new}}}}$.

         B. otherwise, go to $c_1^{p_{f^{\text{new}}}}$.

      ii. if $F_2(d_{c_1^{p_{f^{\text{new}}}}}, d_{c_2^{p_{f^{\text{new}}}}}) > T_2^{f^{\text{new}}}$,

         A. if $p_{f^{\text{new}}}$is the root node, stay and update $T_2^{f^{\text{new}}}$

B. if $p_{f^{\text{new}}}$ is not the root node, go back to the parent node of $p_{f^{\text{new}}}$ and update $T_2^{f^{\text{new}}}$

2. If $F_1(\epsilon_{f^{\text{new}}}) > T_1^{f^{\text{new}}}$,

   (a) if $p_{f^{\text{new}}}$ is the root node, stay and update $T_1^{f^{\text{new}}}$

   (b) if $p_{f^{\text{new}}}$ is not the root node, go back to the parent node of $p_{f^{\text{new}}}$ and update $T_1^{f^{\text{new}}}$

In the above rules, the threshold for $F_1(\epsilon_{f^{\text{new}}})$ is defined by $T_1^{f^{\text{new}}} \times A^{h-1}$ where $T_1^{f^{\text{new}}}$ is a value stored for each flow, A is a parameter, and h is the height of $p_{f^{\text{new}}}$. This threshold becomes strict as $p_{f^{\text{new}}}$ becomes close to the leaf. By doing so, decision of insertion/descent is made carefully at the lower layer and quickly at the upper layer.

In the above rules, $T_1^{f^{\text{new}}}$ and $T_2^{f^{\text{new}}}$ are updated by

$$T_1^{f^{\text{new}}} = T_1^{f^{\text{new}}} \times \alpha \tag{3}$$

and

$$T_2^{f^{\text{new}}} = T_2^{f^{\text{new}}} \times \alpha \tag{4}$$

where α is a parameter. By updating $T_1^{f^{\text{new}}}$ and $T_2^{f^{\text{new}}}$, we relax the threshold so that the flow f^{new} can be included in a cluster.

3.3.2 Node Split

If the number of entries of a node exceeds the threshold, we split the node. In our method, node splitting is done based on pairwise distances between the entries; entries are separated into two groups so as to minimize the sum of the intra-group distances in the same way as ClusTree.

In this thesis, we first apply Ward method [23] to separate entries into two groups. Then one group is removed from the original node and added to newly created node. We also add the node which is the parent of the new node and the original node.

4 Evaluation

4.1 Environment

4.1.1 Traffic Data

In this evaluation, we used the traffic traces monitored at the gateway of our laboratory for 1 hour. We regard the well-known ports as the server ports, and define the flow as the set of packets between the same IP address pair using the same server port. The monitored traffic data includes 4692 flows. In this evaluation, we focus on the flows with a sufficiently large number of packets, because it is important to classify the large flows, considering the case that the flows are handled based on the identified types of the applications. In this evaluation, we focus on the flows which includes more than 50 packets. The number of such flows included in the monitored traffic data is 1268.

4.1.2 Features

In this evaluation, we use the following two features;

- Average size of the downstream packets
- Average size of the upstream packets

where downstream packets indicate the packets from the server, and upstream packets indicate the packets to the server.

In this thesis, we use the above simple features to discuss the constructed clusters, though our method is applicable to the case that more features are used to construct the clusters.

4.2 Compared method

We compare our method with the online clustering method using the features obtained by monitoring a predefined number of packets. In this method, the classifier monitors the downstream and upstream packets. When if either the number of downstream packets or that of upstream packets exceeds the predefined threshold, the classifier set the features of the flow, and run the clustering method. In this method, we use our clustering method with setting ϵ to 0. By this comparison, we

demonstrate that our method constructs the accurate clusters fast, considering the inaccuracy of the features.

4.3 Metric

We use the following metrics.

4.3.1 Accuracy

We evaluate the accuracy of the constructed clusters. We regard the clusters constructed by the offline clustering using the features obtained after the flows are completed as the correct clusters. In this evaluation, we use our method with setting ϵ to 0 as the offline clustering. Then, we compare the clusters constructed by our method with the correct clusters at each layer.

To compare the constructed clusters, we define the following three metrics. The first metric is Precision. Precision for ith cluster is defined by

$$Precision(i) = \frac{|F_i^{\text{correct}} \cap F_i|}{|F_i|} \tag{5}$$

where F_i is the set of flows included in the ith cluster, and F_i^{correct} is the set of flows in the correct ith cluster. Precision for the ith cluster indicates the ratio of the flows that is correctly clustered in the ith cluster.

Similarly, we define Recall for the ith cluster by

$$Recall(i) = \frac{|F_i^{\text{correct}} \cap F_i|}{|F_i^{\text{correct}}|}. \tag{6}$$

Recall for the ith cluster indicates the ratio of the flows in the ith correct cluster that are also included in the ith cluster constructed by the clustering method.

Finally, we define the overall accuracy by

$$\text{Overall Accuracy} = \frac{\sum_i |F_i^{\text{correct}} \cap F_i|}{n} \tag{7}$$

where n is the number of flows. The overall accuracy indicates the ratio of the flows that are correctly clustered.

4.3.2 Time required to find the cluster

In our method, agents for flows find the suitable clusters from the root of the tree. When an agent goes through a node in the tree, the agent determines that the corresponding flow should be included in the cluster corresponding to the node. Therefore, in this thesis, we investigate the number of packets required by the agents to go through the node in each layer. If the agent goes through a node in a layer with a small number of packets, the corresponding flow can be clustered immediately at the layer after the flow starts.

4.4 Results and discussion

In this evaluation, we set the initial value of $T_1^{f^{\mathrm{new}}}$ to 0.04, the initial value of $T_2^{f^{\mathrm{new}}}$ to 0.2, α to 1.1, A to 0.9, N to50 in our method.

Figures 2 through 5 show the clusters constructed by our method, offline clustering using the features obtained after the flows are completed, online clustering using the features obtained when 20 and 40 packets are monitored. In these figures, the horizontal axis is the average size of the downstream packets, and the vertical axis is the average size of the upstream packets. We plot the data point of all flows based on the true features obtained after the flows are completed. The data points are colored according to the constructed cluster.

These figures indicate that all methods construct 3 clusters at the first layer, and 5 clusters at the second layer. Our method and the offline clustering method construct the clusters so that the data points in each cluster are similar. However, especially the clusters constructed by the clustering method using the features obtained when 20 packets are monitored includes data points far from the other data points in the cluster.

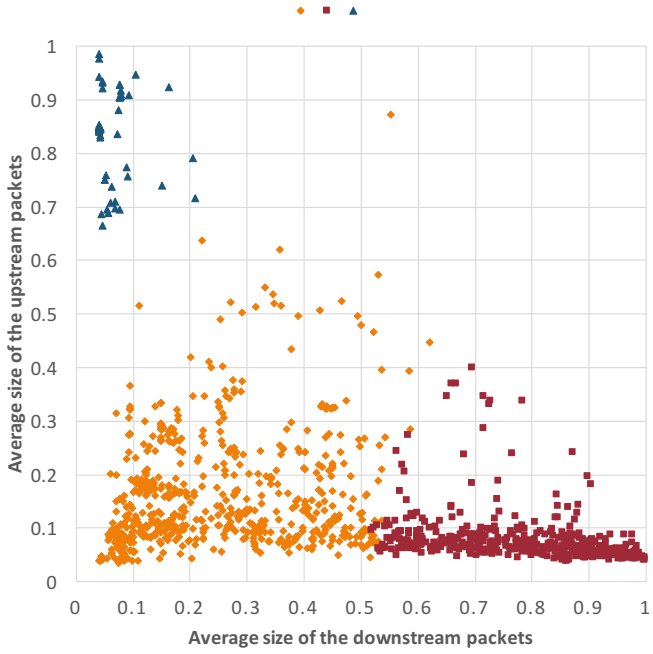
We investigate the accuracy of the constructed clusters. Figures 6 and 7 compares the Precision and the Recall, respectively. In these figures, the Precision and the Recall become low in the lower layer. This is because the difference between the clusters becomes small in the lower layer. As a result, it becomes difficult to construct the accurate clusters in the lower layer. Because the parameters in our method may have an impact on the constructed clusters, our method may construct accurate clusters by setting the parameters properly, which is one of our future work.

From these figures, our method achieves the higher Precision and Recall than the method using the features obtained when 20 or 40 packets are monitored. This is because the features of some

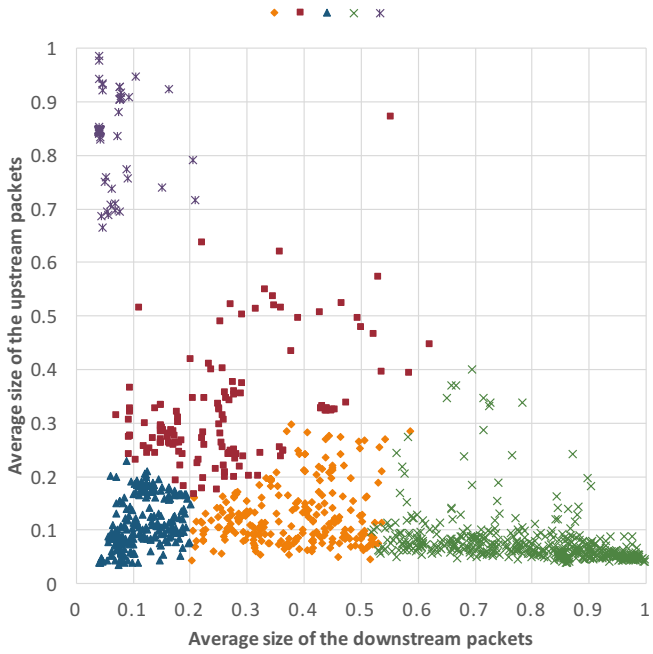
flows obtained when 20 or 40 packets are monitored are different from the true features obtained when the flows are completed. Such flows cannot be grouped into the correct clusters. On the other hand, our method waits until the features of the flow becomes sufficiently accurate. As a result our method achieves the higher Precision and Recall.

The advantages of our method discussed above are also demonsrated by the overall accuracy The overall accuracy is shown in Figure 8. Similar to the Precision and Recall, our method achieves the higher overall accuracy than the method using the features obtained when 20 or 40 packets are monitored.

We also investigate the number of packets required to be monitored to determine the cluster the corresponding flow belongs to. Figure 9 shows the number of packets required to be monitored to determine the cluster in the 1st and 2nd layers. In this figure, the horizontal axis is the number of packets, and the vertical axis is frequency or cumulative distribution. The location of each data point in this figure is determined based on the largest number among the number of upstream packets and that of downstream packets. This figure indicates that 81 % of flows can be grouped into the corresponding cluster in the 1st layer by monitoring 30 packets. Even in the second layer, 99 % of flows can be grouped into the corresponding cluster by monitoring 40 packets. In addition, there are flows whose corresponding cluster is determined only by monitoring 10 packets. This is because our method selects the clusters immediately after the features of the flow becomes sufficiently accurate. Therefore, our method is applicable to the real time flow classification; the flow classification using our method can classify the flows soon after the flow starts, and the classified flows can be handled properly according to the identified types of the flows.

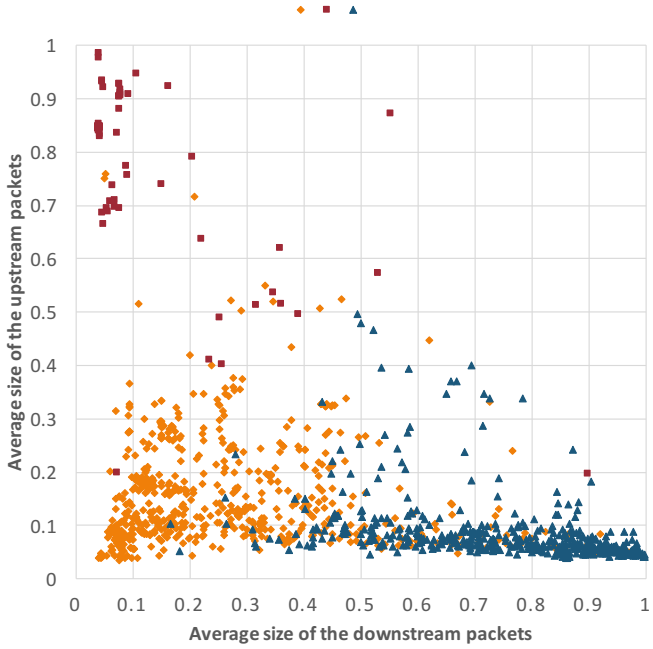


(a) 1st layer

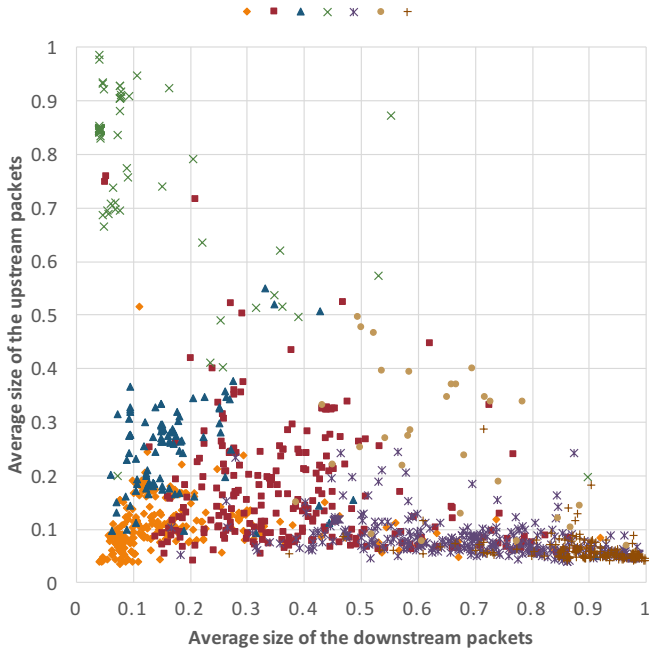


(b) 2nd layer

Figure 2: Constructed clusters: offline

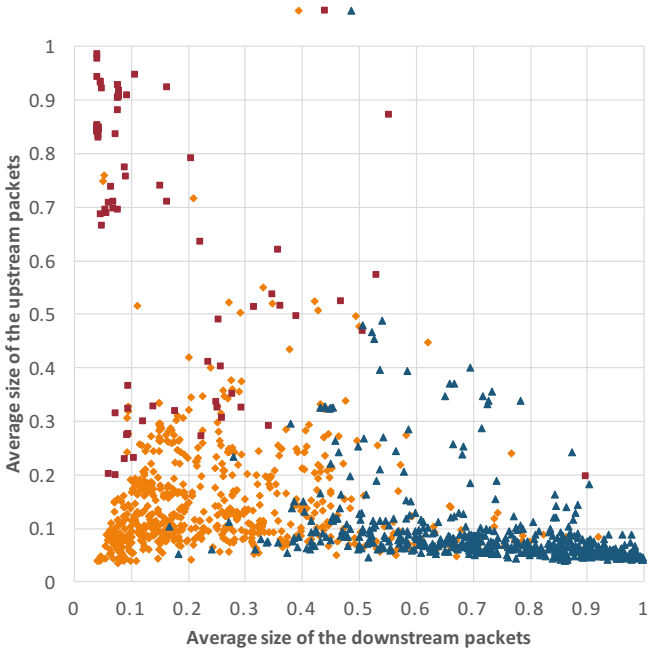


(a) 1st layer

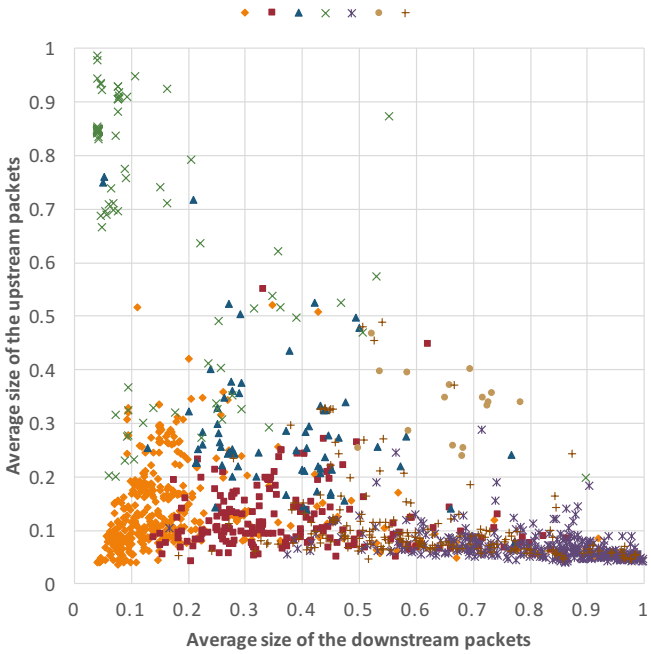


(b) 2nd layer

Figure 3: Constructed clusters: online (proposed)

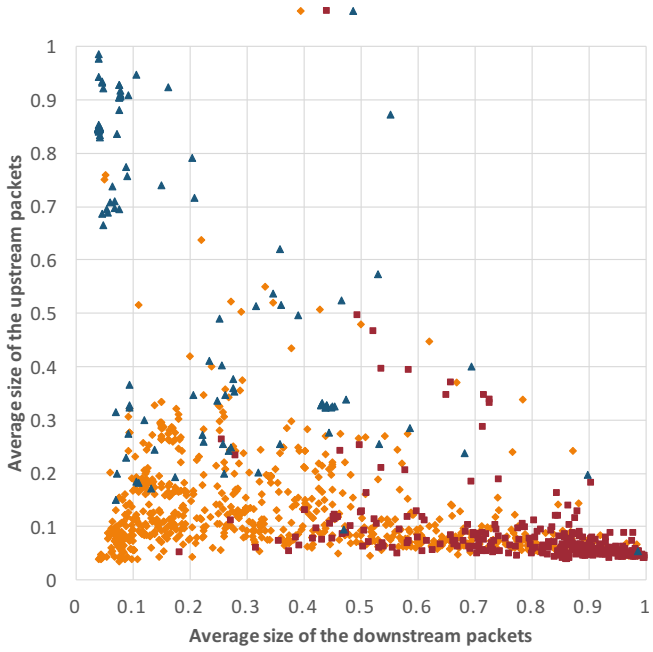


(a) 1st layer

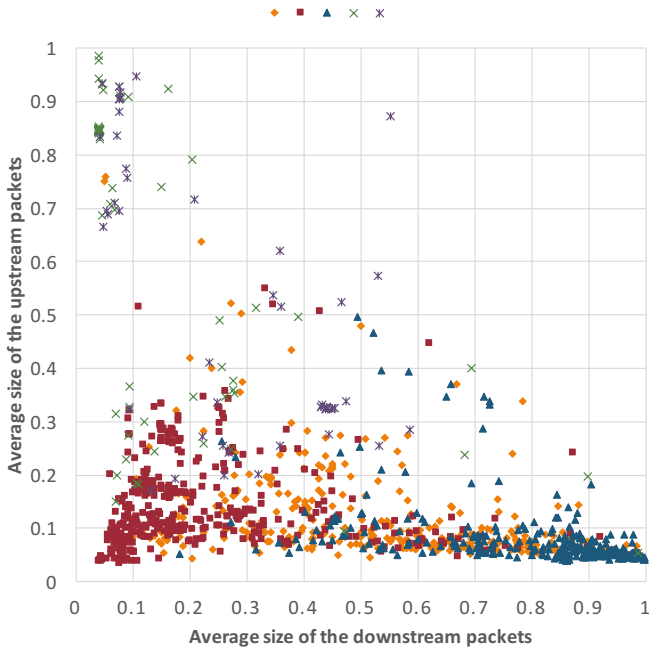


(b) 2nd layer

Figure 4: Constructed clusters: online (using features obtained by monitoring 40 packets)

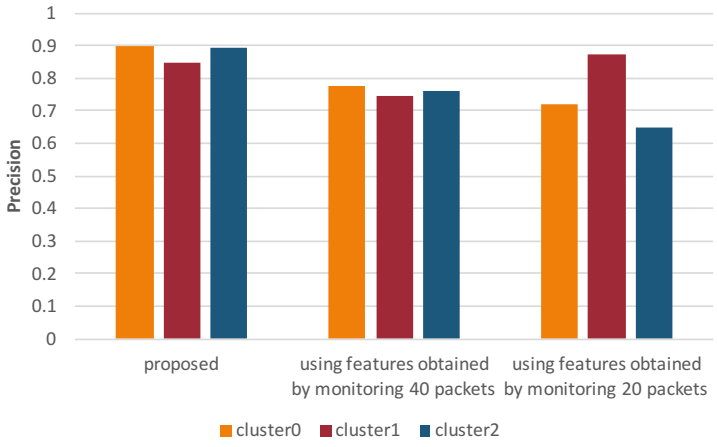


(a) 1st layer

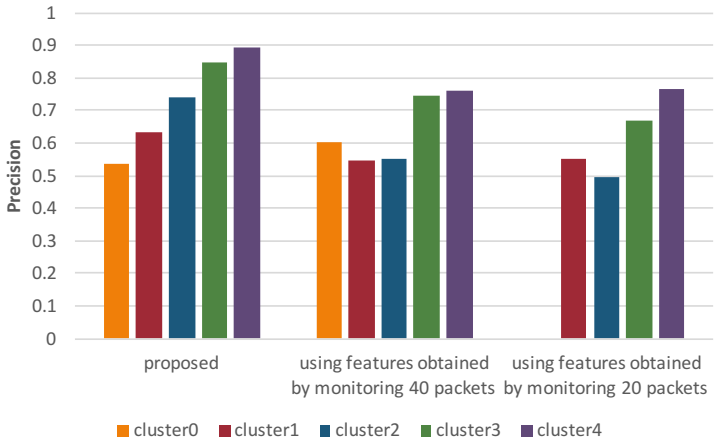


(b) 2nd layer

Figure 5: Constructed clusters: online (using features obtained by monitoring 20 packets)

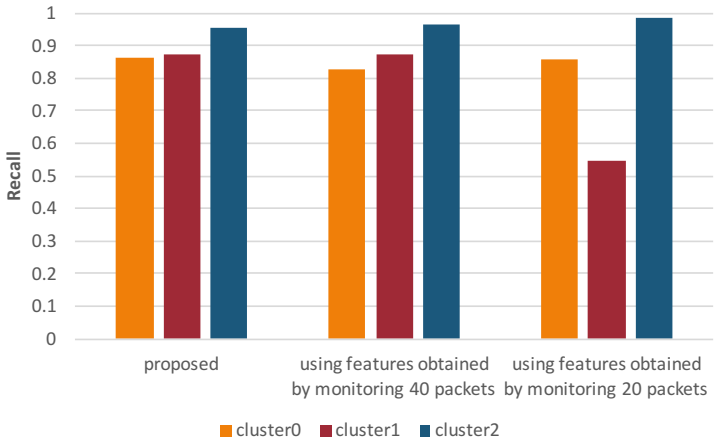


(a) 1st layer

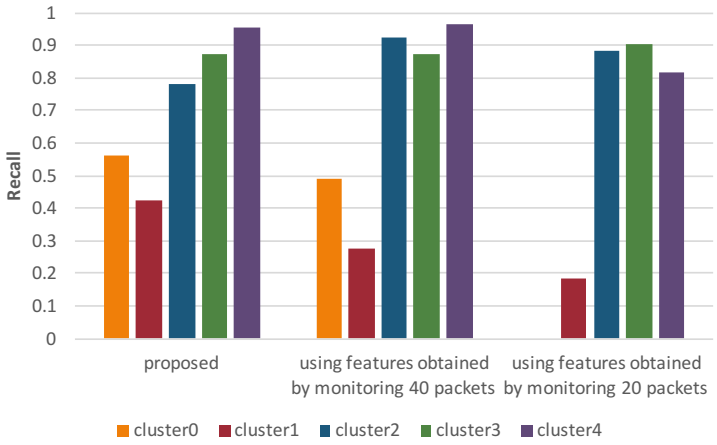


(b) 2nd layer

Figure 6: Precision

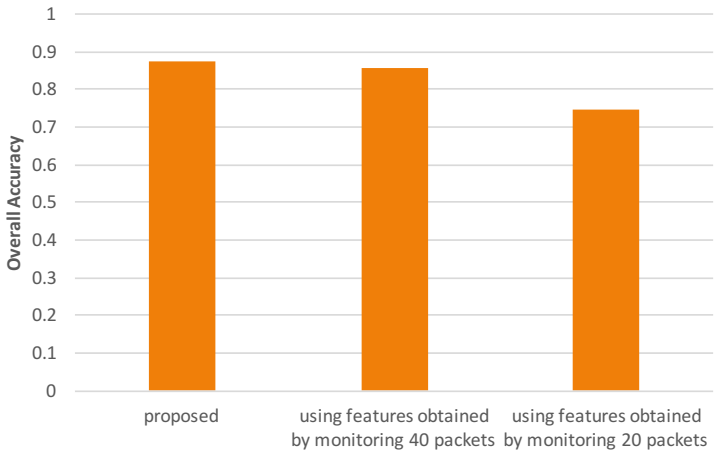


(a) 1st layer

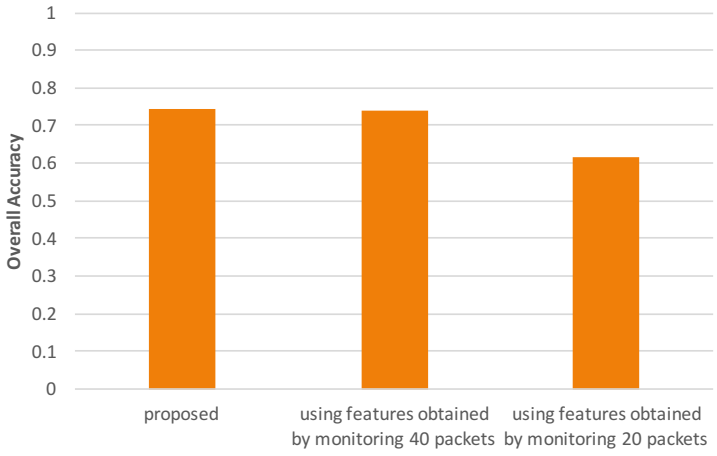


(b) 2nd layer

Figure 7: Recall

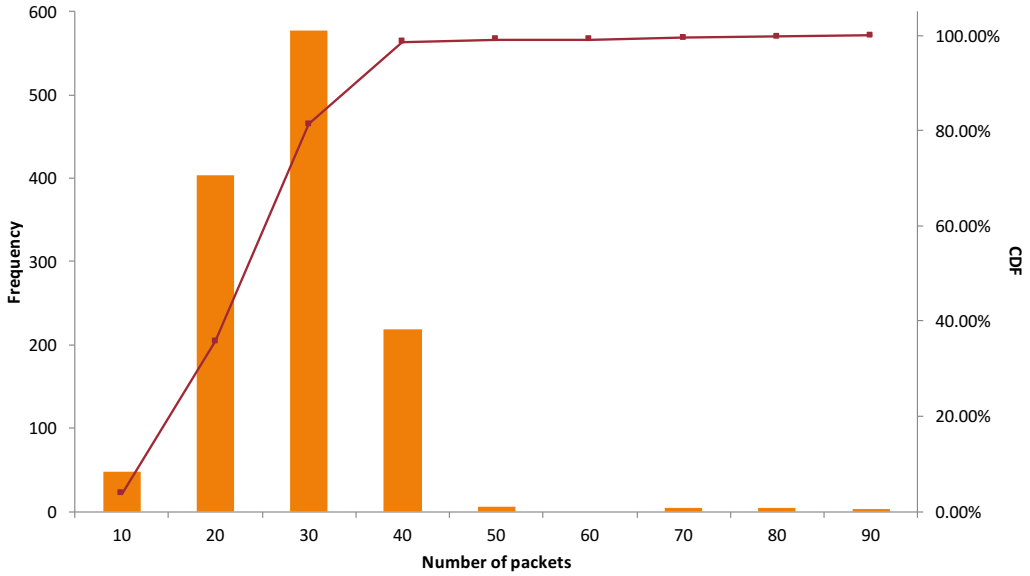


(a) 1st layer

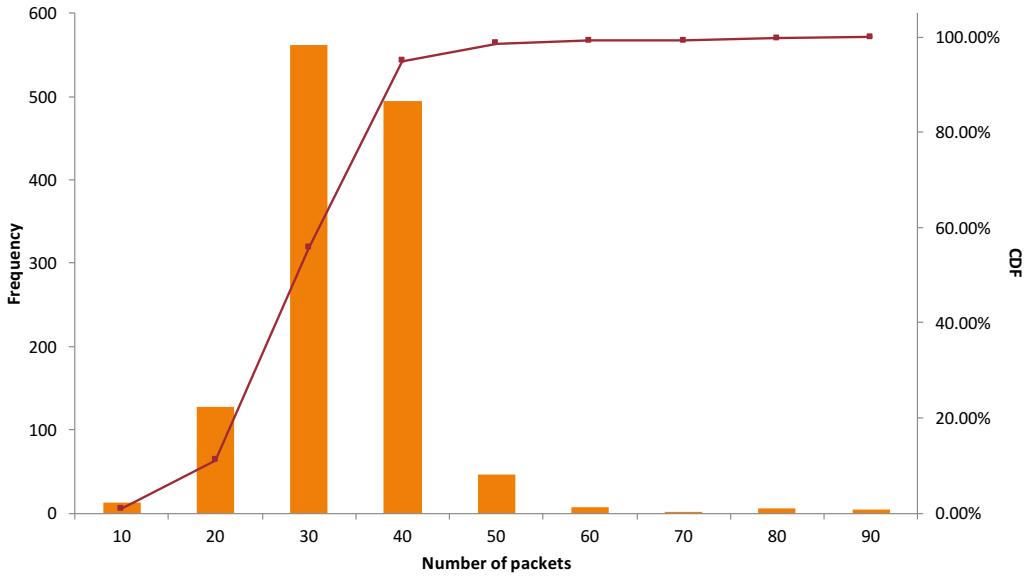


(b) 2nd layer

Figure 8: Overall Accuracy



(a) 1st layer



(b) 2nd layer

Figure 9: Number of packets required to determine the clusters

5 Conclusion and Future Work

In this thesis, we proposed a new hierarchical clustering method for real-time flow classification. Our method constructs the hierarchical cluster, updating the features of the traffic flows; the features are updated every time a packet arrives, and the cluster which a flow belongs to in each layer is determined after the features become accurate enough. If the features of the flow are not accurate enough to determine the cluster the flow belong to, our method waits another packet to improve the accuracy of the features.

The hierarchical clusters are useful to estimate the characteristics of the flow when the flow is related to the unknown application. In the hierarchical cluster, the flows are clustered into a small number of groups in the upper layer, and the flows belonging to the same group in the upper layers are clustered into several groups in the lower layer. If flows of a new application construct a new group in the lower layer, but they are classified in an existing group in the upper layer, the characteristic of the new application can be estimated from the characteristic of the existing group.

We evaluated our clustering method using the actual traffic traces. The results indicated that our method construct the similar cluster as the method that uses the accurate features; our method achieves the accuracy of 88% when 3 clusters are constructed, and achieves the accuracy of 74% when 5 clusters are constructed. In addition, the results demonstrated that our method classifies the flows by monitoring only a small number of packets; 81 % of the flows are classified into 3 clusters before 30 packets arrive, and 99 % of the flows are classified into 5 clusters before 40 packets arrive.

Our future work includes the method to set the parameters in our method, and evaluation of the flow classification method using our clustering method.

Acknowledgments

Foremost, I would like to express my deepest gratitude to Professor Masayuki Murata of Osaka University for his exact guidance, insightful comments and encouragement. Furthermore, I would like to show my sincere appreciation to Assistant Professor Yuichi Ohsita of Osaka University for continuous support, helpful discussions, and meticulous advice. Without their support, I could not achieve all results in this thesis. I am also grateful to the helpful advice from Associate Professor Shin'ichi Arakawa and Assistant Professor Daichi Kominami of Osaka University. Finally, I would like to thank my senior associates including Mr. Tatsuya Otoshi, and my colleagues of Advanced Network Architecture Research Laboratory of Osaka University, for their great encouragement and supports.

References

[1] J. He, R. Zhang-Shen, Y. Li, C.-Y. Lee, J. Rexford, and M. Chiang, "DaVinci: Dynamically Adaptive Virtual Networks for a Customized Internet," in *Proceedings of the 2008 ACM CoNEXT Conference*, CoNEXT '08, (New York, NY, USA), pp. 15:1–15:12, ACM, 2008.

[2] Takashi Miyamura, Yuichi Ohsita, Shin'ichi Arakawa, Yuki Koizumi, Akeo Masuda, Kohei Shiomoto, and Masayuki Murata, "Network Virtualization Server for Adaptive Network Control," in *Proceedings of 20th ITC Specialist Seminar on Network Virtualization - Concept and Performance Aspects*, May 2009.

[3] A. Callado, C. Kamienski, G. Szabo, B. P. Gero, J. Kelner, S. Fernandes, and D. Sadok, "A Survey on Internet Traffic Identification," *Communications Surveys & Tutorials, IEEE*, vol. 11, pp. 37–52, Aug. 2009.

[4] A. Dainotti, A. Pescape, and K. Claffy, "Issues and Future Directions in Traffic Classification," *Network, IEEE*, vol. 26, pp. 35–40, January 2012.

[5] L. Popa, A. Ghodsi, and I. Stoica, "HTTP As the Narrow Waist of the Future Internet," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Hotnets-IX, (New York, NY, USA), pp. 6:1–6:6, ACM, 2010.

[6] T. Nguyen and G. Armitage, "A Survey of Techniques for Internet Traffic Classification using Machine Learning," *Communications Surveys Tutorials, IEEE*, vol. 10, pp. 56–76, Fourth 2008.

[7] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "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*, IMC '04, (New York, NY, USA), pp. 135–148, ACM, 2004.

[8] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, and Y. Guan, "Network Traffic Classification Using Correlation Information," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, pp. 104–117, Jan 2013.

[9] A. W. Moore and D. Zuev, "Internet Traffic Classification Using Bayesian Analysis Techniques," *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, pp. 50–60, June 2005.

[10] T. T. T. Nguyen, G. Armitage, P. Branch, and S. Zander, "Timely and Continuous Machine-learning-based Classification for Interactive IP Traffic," *IEEE/ACM Trans. Netw.*, vol. 20, pp. 1880–1894, Dec. 2012.

[11] J. Zhang, C. Chen, Y. Xiang, W. Zhou, and Y. Xiang, "Internet Traffic Classification by Aggregating Correlated Naive Bayes Predictions," *Information Forensics and Security, IEEE Transactions on*, vol. 8, pp. 5–15, Jan 2013.

[12] W. Li, M. Canini, A. W. Moore, and R. Bolla, "Efficient Application Identification and the Temporal and Spatial Stability of Classification Schema," *Comput. Netw.*, vol. 53, pp. 790–809, Apr. 2009.

[13] Y. Jin, N. Duffield, J. Erman, P. Haffner, S. Sen, and Z.-L. Zhang, "A Modular Machine Learning System for Flow-Level Traffic Classification in Large Networks," *ACM Trans. Knowl. Discov. Data*, vol. 6, pp. 4:1–4:34, Mar. 2012.

[14] J. Erman, M. Arlitt, and A. Mahanti, "Traffic Classification Using Clustering Algorithms," in *Proceedings of the 2006 SIGCOMM workshop on Mining network data*, pp. 281–286, 2006.

[15] L. Bernaille, R. Teixeira, and K. Salamatian, "Early Application Identification," in *Proceedings of the 2006 ACM CoNEXT Conference*, CoNEXT '06, (New York, NY, USA), pp. 6:1–6:12, ACM, 2006.

[16] J. Zhang, C. Chen, Y. Xiang, W. Zhou, and A. Vasilakos, "An Effective Network Traffic Classification Method with Unknown Flow Detection," *Network and Service Management, IEEE Transactions on*, vol. 10, pp. 133–147, June 2013.

[17] Y. Wang, Y. Xiang, J. Zhang, W. Zhou, G. Wei, and L. Yang, "Internet Traffic Classification Using Constrained Clustering," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, pp. 2932–2943, Nov 2014.

[18] P. Kranen, I. Assent, C. Baldauf, and T. Seidl, "The ClusTree: indexing micro-clusters for anytime stream mining," *Knowledge and Information Systems*, vol. 29, no. 2, pp. 249–272, 2011.

[19] A. Guttman, "R-trees: A Dynamic Index Structure for Spatial Searching," *SIGMOD Rec.*, vol. 14, pp. 47–57, June 1984.

[20] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles," *SIGMOD Rec.*, vol. 19, pp. 322–331, May 1990.

[21] T. Seidl, I. Assent, P. Kranen, R. Krieger, and J. Herrmann, "Indexing Density Models for Incremental Learning and Anytime Classification on Data Streams," in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '09, (New York, NY, USA), pp. 311–322, ACM, 2009.

[22] H. Azzag, N. Monmarche, M. Slimane, and G. Venturini, "AntTree: a New Model for Clustering with Artificial Ants," *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, vol. 4, pp. 2642–2647, Dec. 2003.

[23] J. H. Ward, "Hierarchical Grouping to Optimize an Objective Function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.