Real-time Classification of Internet Application Traffic using a Hierarchical Multi-class SVM

Jaehak Yu[1], Hansung Lee[1], Younghee Im[2], Myung-Sup Kim[2], and Daihee Park[2]
[1]Electronics and Telecommunications Research Institute, Daejeon, 305-700, Korea
[e-mail: {dbzzang, mohan}@etri.re.kr]
[2]Dept. of Computer and Information Science, Korea University, Yeongi, 339-700, Korea
[e-mail: {yheeim, tmskim, dhpark}@korea.ac.kr]
*Corresponding author: Myung-Sup Kim

*Received June 8, 2010; revised July 20, 2010; accepted July 30, 2010;
published October 30, 2010*

Abstract

In this paper, we propose a hierarchical application traffic classification system as an alternative means to overcome the limitations of the port number and payload based methodologies, which are traditionally considered traffic classification methods. The proposed system is a new classification model that hierarchically combines a binary classifier SVM and Support Vector Data Descriptions (SVDDs). The proposed system selects an optimal attribute subset from the bi-directional traffic flows generated by our traffic analysis system (KU-MON) that enables real-time collection and analysis of campus traffic. The system is composed of three layers: The first layer is a binary classifier SVM that performs rapid classification between P2P and non-P2P traffic. The second layer classifies P2P traffic into file-sharing, messenger and TV, based on three SVDDs. The third layer performs specialized classification of all individual application traffic types. Since the proposed system enables both coarse- and fine-grained classification, it can guarantee efficient resource management, such as a stable network environment, seamless bandwidth guarantee and appropriate QoS. Moreover, even when a new application emerges, it can be easily adapted for incremental updating and scaling. Only additional training for the new part of the application traffic is needed instead of retraining the entire system. The performance of the proposed system is validated via experiments which confirm that its recall and precision measures are satisfactory.

Keywords: Traffic monitoring and analysis, traffic classification, P2P traffic analysis, support vector machine, attribute subset selection

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2009-0090455), the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2007-331-D00387), and the IT R&D program of MKE/KEIT(10035310, Smart Plant Safety Framework based on Reliable-Secure USN).

DOI: 10.3837/tiis.2010.10.009

1. Introduction

With the rapid development of the Internet and the drastic increase in the number of users, a variety of new network services have been developed and commercialized. Especially, according to research conducted by Ipoque [1], Germany (2009), on network services, Peer-to-Peer (P2P) related traffic has comprised over 60% of total Internet traffic over the last several years, and its market importance is expected to increase [1][2]. P2P, which comprises the largest fraction of total traffic on the Internet, not only results in increased network complexity due to the enormous volume of traffic, but also requires huge extra costs such as the cost of upgrading the infrastructure and network repartitioning [2][3]. The network management system viewpoint is concerned with supporting appropriate Quality of Service (QoS) and a safe network environment. Thus, faster and more precise classification of Internet applications including P2P are some of the key recent issues of the network-related community [2][5].

Traditional methodologies for classifying Internet application traffic can be divided into the port number-based method and the payload-based method [2][4]: 1) Port number-based classification analyzes well-known port numbers assigned by IANA. It is rather a simple and practical method, but it is difficult to perform precise traffic classification of an application that allocates dynamic port numbers or uses an existing well-known port number for a different purpose. 2) Payload-based classification is a method to extract the characteristics of the payload and compare it with the packets. However, this method cannot be used if the packets are encrypted. Furthermore, in the present situation where payload access is forbidden due to privacy protection and due to the high cost of access, the use of the payload-based methodology is more difficult and necessitates an alternative.

Recent studies describe many ongoing attempts to apply data mining and machine learning techniques to Internet application traffic classification in order to cope with application changes [4][5][6][7][8][9]. Since machine learning techniques (supervised or unsupervised learning) can extract important patterns from the feature vectors, which are independent of dramatically changing port numbers and encrypted payloads, those techniques have emerged as viable methods to cope with the aforementioned problems of traditional methodologies. Also, the application of a data mining method is appropriate considering that the huge volume of stream data is a characteristic of network traffic. The Support Vector Machine (SVM), which has emerged as a promising tool in the area of intelligent systems, has received attention by researchers on its application to Internet traffic classification [6][7][8][9]. However, employing SVMs in the Internet application traffic classification problem is at a relatively early stage. It involves two methodologies. In the first method [6][7], P2P and non-P2P traffic is simply divided using an SVM (a binary classifier) for the detection of P2P traffic. In the second method [8][9][10], P2P traffic identification and traffic classification involve organically combining various binary classifier SVMs, as with the one-against-one method, then building a multi-class SVM. However, designing a multi-class SVM using binary SVMs has various problems, since it not only requires additional training time but also entails differences in system performance due to the varying quality of the training data, hindering its pactical application [11]. Especially, Internet application traffic classification may have differences in data size according to type. As a result, because of the unbalanced size of the training data, the learning result of one traffic type might be influenced by a different traffic data type. Moreover, it is hard to conclude that the current training data represents the entire class, since new traffic types belonging to one traffic class continue to emerge. Therefore, it is probable that the binary classifier SVM results in misclassification of the new training data by creating a decision boundary that includes an unobserved area. Accordingly, it is preferable to select a decision boundary function with a one-class SVM that independently expresses the corresponding class.

In this paper, we propose a new real-time hierarchical Internet application traffic classification system based on SVM. It can identify P2P traffic and classify it into subsidiary application types in a hierarchical manner. The proposed system hierarchically combines a binary classifier SVM and Support Vector Data Description (SVDDs) that is a representative model of a one-class SVM. First, it selects an optimal attribute subset from the bi-directional traffic flows generated by our traffic analysis system (KU-MON) that enables real-time collection and analysis of campus traffic. Our classification system is composed of three layers: The first layer is the SVM, which is a binary classifier that performs classification between P2P and non-P2P traffic. The second layer classifies P2P traffic into file-sharing, messenger and TV, based on three SVDDs. The third layer performs specialized classification of all individual application traffic types. The Internet application traffic classification system proposed in this paper classifies P2P and non-P2P traffic in real-time, thus, it can safely support a stable network environment. Also, it can guarantee appropriate QoS, since classification of three representative types of P2P traffic is possible. Furthermore, specialized classification of individual applications provides support for seamless bandwidth control and efficient system resource management. Furthermore, when new application traffic emerges, the system can be easily adapted to incrementally update and train only the new part of the corresponding module in the application traffic classification system, instead of retraining the entire system.

This paper is organized as follows. In Section 2, the key considerations for application traffic classification are stated. In Section 3, the proposed hierarchical application traffic classification model based on SVM is presented. In Section 4, the experimental results and performance analysis are described. Finally, in Section 5, the conclusion and future research direction are discussed.

2. Considerations for Internet Application Traffic Classification

The following four points should be considered for developing a real-time Internet traffic classification system using machine learning techniques. First, the classification criterion of Internet traffic must be defined. Three popularly considered classification cariteria are protocol-based, application-based and type-based classification; It must be decided whether Internet traffic is classified by the application-layer protocols, traffic types (e.g., streaming and downloading) or individual applications. Second, the processing unit of the traffic data for traffic classification must be decided. To set the training and test units of the IP network, the processing units can be choosen either by packet, which is the lowest basic unit of the traffic data, or by flow, which is the group of packets. Two types of flow can be considered: uni-directional flow and bi-directional flow. Third, the features extracted from the processing units must be determined. Usually, statistical features extracted from the packet size and capture time distribution as well as the packet header information are considered. Fourth, the time the classification system is applied to the processing units needs to be decided. For real-time classification, it is important to apply the classification system to the processing units as soon as all the required features are determined.

While Internet traffic classification criterion can depend on the analysis purpose, in this paper, Internet traffic is classified by the following principle. From the application viewpoint, all traffic data created by an application will be regulated as that particular application traffic. For example, HTTP traffic generated by a MSN messenger application is regulated as MSN traffic, not as Web traffic. This separation also corresponds with the criterion of traffic control systems such as QoS systems and Intrusion Prevention Systems (IPSs), which means that the classification result can be utilized directly or with minimum changes. One exception is the case that traffic using well-known traditional protocols, such as FTP, SMTP, DNS, etc. (implemented by many different applications) is classified according to protocol, and it is separated from individual application-based classification.

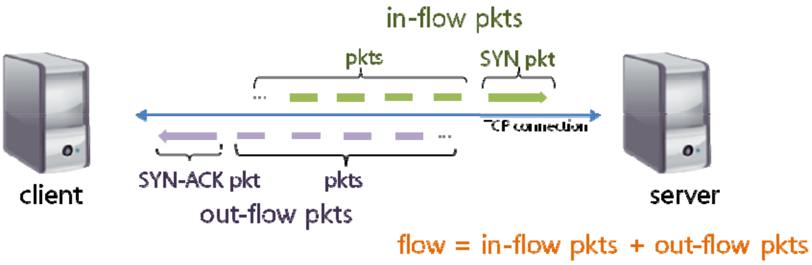

Fig. 1. Traffic flow (in-flow, out-flow) from a TCP connection

For precise analysis of application traffic, the features that will be extracted from the processing units of traffic data must be determined. The flow has been used as the basic unit of traffic classification in many studies [12][13][14][15][16]. In this paper, we also use the flow as the basic processing unit of our traffic classification system. The form of flow that is used in this paper includes most of the flow characteristics with somewhat different definitions in other studies [17][18][19]. It is defined as a collection of bi-directional packets of 5-tuple information (source IP, source port, destination IP, destination port and protocol number). A flow has two different in-flow and out-flow characteristics per connection. In case of TCP, packets moving from the client to the server are defined as in-flow, and packets moving from the server to the client are defined as out-flow. All the bi-directional packets that are created during the entire connection from SYN packet to FIN/RST packets are defined as one flow, as is shown in **Fig. 1**. Further, in case of UDP, since the relationship between client and server cannot be defined, the packets between the point where the first packet is captured and the point where the last packet is captured are defined as one flow.

In this paper, we defined 39 statistical values as a possible features of our bi-directional flow, as shown in **Table 1**. The header information such as IP addresses and port numbers of the 5-tuple information are eliminated from the flow feature set due to the likelihood that most of the collected traffic data can be classified by these special values. In our classification system, we used only a small number features from all of the possible features listed in **Table 1**. These are selected by the Correlation-based Feature Selection (CFS) algorithm [17].

Table 1. Flow feature set for the proposed system

Features		Description
Protocol	proto	TCP or UDP

in-flow	in_duration in_pkts, in_octets in_syn, in_ack, in_rst, in_fin		flow duration(last - first) in in_flow number of packets, bytes in in_flow number of SYN, ACK, RST, FIN packets
out-flow	〃		〃
in-flow	statistics of packet size	in_pkt_min	minimum value of pkt size in in_flow
		in_pkt_max	maximum value of packet size
		in_pkt_avg	Average of packet size in in_flow
		in_pkt_sdev	Standard devation of packet size in in_flow
	statistics of window size	〃	〃
	statistics of jitter	〃	〃
out-flow	statistics of packet size	〃	〃
	statistics of window size	〃	〃
	statistics of jitter	〃	〃

Finally, for real-time determination of the flow identity, we need to decide the time when the classification algorithm is applied to the flow record. In this paper, we apply our algorithm as soon as all of the required features are gathered by our real-time traffic analysis system, named KU-MON.

3. Internet Application Traffic Classification System

In this section, we summarize attribute subset selection, the basic concept of the SVM to classify P2P and non-P2P, and the SVDD that is the basic element of the multi-class SVM. We also introduce the proposed Internet application traffic classification system based on the hierarchical SVM.

3.1 Attribute subset selection

Efficient attribute subset selection for pattern classification is one of the important issues in many studies [20][21][22][23]. Attribute selection is the problem of selecting a subset of attributes from a feature set in order to provide a compact, precise and fast classifier with minimal performance degradation as possible by removing the attributes that are useless, redundant or that are used the least [20][21][22][23]. Reducing the dimensionality of the data reduces the size of the hypothesis space and allows algorithms to operate faster and more effectively.

In this paper, we used the Correlation-based Feature Selection (CFS) that has been verified as the best among the attribute subset selection methods [20]. CFS uses the features' predictive performances and inter-correlations to guide its search for a good feature subset. It can drastically reduce the dimensionality of data sets while maintaining or improving the performance of learning algorithms. At the heart of the CFS algorithm is a heuristic for evaluating the worth or merit of a subset of features. This heuristic takes into account the usefulness of individual features for predicting the class label along with the level of inter-correlation among them. CFS first calculates a matrix of feature-class and feature-feature correlations from the training data and then searches the feature subset space using a best first search. The version of CFS used in this paper includes a heuristic to include locally predictive features and avoid the re-introduction of redundancy [20].

3.2 Binary Classifier SVM

The foundations of Support Vector Machines (SVMs) were developed by Vapnik in 1995 and are gaining popularity due to their many attractive features and promising empirical performance [24][25]. The formulation embodies the Structural Risk Minimization (SRM) principle that has been shown to be superior to the traditional Empirical Risk Minimization (ERM) principle employed by conventional neural networks. SRM minimizes the upper bound of the expected risk, whereas ERM minimizes the error of the training data. This ensures that SVMs have a greater ability to generalize, which is the purpose of statistical learning [24]. In this section, we briefly review some basic works on SVMs for the classification problems [24] that will be used in the proposed system. To explain the principles of SVMs, we first examine the simplest case, a two-class problem, where the classes are linearly separable. In this problem, the goal is to separate the two classes via a function that is induced from the available examples. Consider the example in Fig. 2. Many possible linear classifiers could separate the data, but only one can maximize the margin. This linear classifier is termed the optimal separating hyperplane.

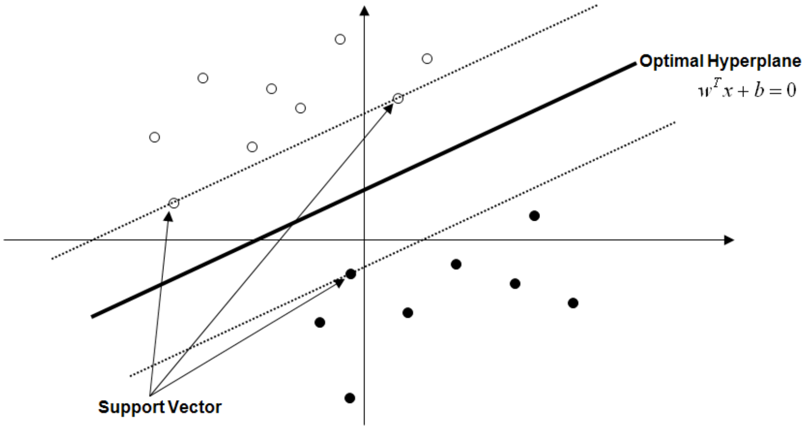

Fig. 2. Optimal separating hyperplane and support vectors

Consider the problem of separating the set of training vectors belonging to two separate classes,

$$D = \{(x^1, y^1), \ldots, (x^l, y^l)\}, \quad x \in R^n, \quad y \in \{-1,1\}, \tag{1}$$

with a hyperplane,

$$\langle w, x \rangle + b = 0. \tag{2}$$

If the set of vectors is separated without error and the distance between the closest vectors to the hyperplane is maximal, then this set is defined as optimally separated by the hyperplane. A separating hyperplane in canonical form must satisfy the following constraints,

$$y^i[< w, x^i > +b] \geq 1, \quad i = 1, \ldots, l. \tag{3}$$

The distance $d(w,b;x)$ of a point x from the hyperplane (w,b) is,

$$d(w,b;x) = \frac{\left|\left\langle w, x^i \right\rangle + b\right|}{\|w\|}. \tag{4}$$

Hence, the hyperplane that optimally separates the data is the one that minimizes the following:

$$\text{minimize} \quad \Phi(w) = \frac{1}{2}\|w\|^2$$

$$\text{subject to} \quad d_i(w^T x_i + b) \geq 1 \quad \text{for} \quad i = 1,\ldots,l. \tag{5}$$

The solution to the optimization problem of equation (5) is given by the saddle point of the Lagrange function:

$$\Phi(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i \left(y^i \left[\left\langle w, x^i + b \right\rangle\right] - 1\right), \tag{6}$$

where α are the Lagrange multipliers. The Lagrangian has to be minimized with respect to w,b and maximized with respect to $\alpha \geq 0$. Classical Lagrangian duality enables the primal problem, given by equation (6), to be transformed to its dual problem, which is easier to solve. The dual problem is given by:

$$\max_{\alpha} W(\alpha) = \max_{\alpha} -\frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \left\langle x_i x_j \right\rangle + \sum_{k=1}^{l} \alpha_k$$

$$s.t. \quad \alpha_i \geq 0, i = 1,\ldots,l \quad \text{and} \quad \sum_{j=1}^{l} \alpha_j y_j = 0. \tag{7}$$

The hard classifier is then given by:

$$f(x) = \operatorname{sgn}\left(\left\langle w^*, x \right\rangle + b\right). \tag{8}$$

The approach described for a linear SVM can be extended to create a nonlinear SVM for classifying linearly inseparable data. There are two main steps. In the first step, we transform the original input data into a higher dimensional space using nonlinear mapping. The second step searches for a linear separating hyperplane in the new space. We are again left with a quadratic optimization problem that can be solved using the linear SVM formulation. The maximal marginal hyperplane found in the new space corresponds to a nonlinear separating hypersurface in the original space [25].

3.3 Multi-class SVM based on SVDD

Recently, the support vector learning method has emerged as a promising tool in the area of intelligent systems. It has shown excellent performance for pattern classification and function approximation, by ensuring the global optimum for a given problem. However, it has the intrinsic structural limitation of the binary classifier. According to a recent study, there are three major types of approaches for multi-class SVM: one-against-all, one-against-one and DAGSVM that involves combining many binary classifiers SVMs [11]. In this paper, instead of adopting one of the previous works, we construct a series of SVDDs operating in parallel at the second and third layer of the proposed system in order to determine the type of Internet traffic applications in detail, as illustrated in **Fig. 4**.

In general, the number of datasets necessary for training varies according to the number of types of applications. Hence, the training results for one class may not be independent of other classes due to the unbalanced size of the training data. In addition, the current training data may not represent entire classes, since new types of applications continue to emerge. Thus, the binary classifier SVM may suffer from misclassification of new application data by creating a decision boundary including an unobserved area. Accordingly, it is preferable to select a decision boundary function using a one-class SVM that independently expresses the corresponding class (one of the most well-known one-class SVMs is the Support Vector Data Description (SVDD)). The multi-class SVM based on SVDD is described as follows [11]:

Given a K -data set of N_k patterns in a d -dimensional input space, $D_k = \{x_i^k \in R^d \mid i = 1,\cdots,N_k\}; k = 1,\cdots,K$, the multi-class SVM based on SVDD is defined as the problem of obtaining a hypersphere that maximizes the number of training datasets while minimizing the radius. It is formalized as the following mathematical optimization problem:

$$\begin{aligned} &\min L_0(R_k^2, a_k, \xi_k) = R_k^2 + C\sum_{i=1}^{N_k}\xi_i^k \\ &\text{s.t. } \left\|x_i^k - a_k\right\|^2 \le R_k^2 + \xi_i^k,\ \ \xi_i^k \ge 0,\ \ \forall i, \end{aligned} \tag{9}$$

where a_k is the center of the sphere that expresses the k -th class, R_k^2 is the square value of the sphere radius, ξ_i^k is the penalty term that denotes the deviation of the i -th training data element x_i^k from a sphere and C is the trade-off constant.

By introducing a Lagrange multiplier for each inequality condition, we can obtain the following Lagrange function:

$$
\begin{aligned}
L(R_k^2, a_k, \xi_k, \alpha_k, \eta_k) = R_k^2 + C\sum_{i=1}^{N_k} \xi_i^k \\
+\sum_{i=1}^{N_k} \alpha_i^k \left[(x_i^k - a_k)^T (x_i^k - a_k) - R_k^2 - \xi_i^k \right] \\
-\sum_{i=1}^{N_k} \eta_i^k \xi_i^k
\end{aligned}
\tag{10}
$$

where $\alpha_i^k \geq 0,\ \eta_i^k \geq 0,\ \forall i.$

Based on the saddle point condition, equation (10) must be minimized with respect to R_k^2, a_k and ξ_i^k, and maximized with respect to α_k and η_k. The optimal solution of (9) should satisfy the following:

$$
\begin{aligned}
&\frac{\partial L}{\partial R_k^2} = 0: \ \sum_{i=1}^{N_k} \alpha_i^k = 1. \\
&\frac{\partial L}{\partial \xi_k^2} = 0: \ C - \alpha_i^k - \eta_i^k = 0 \quad \therefore \alpha_i^k \in [0, C], \ \forall i. \\
&\frac{\partial L}{\partial R_k^2} = 0: \ a_k = \sum_{i=1}^{N_k} \alpha_i^k x_i^k
\end{aligned}
\tag{11}
$$

By substituting equation (11) into the Lagrange function L, we obtain the following dual problem:

$$
\begin{aligned}
&\min \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \alpha_i^k \alpha_j^k < x_i^k, x_j^k > - \sum_{i=1}^{N_k} \alpha_i^k < x_i^k, x_i^k > \\
&\text{s.t.} \ \sum_{i=1}^{N_k} \alpha_i^k = 1, \ \alpha_i^k \in [0, C], \ \forall i.
\end{aligned}
\tag{12}
$$

A sphere can express a more complex decision boundary in feature space, F. We can map an input space into a feature space using the kernel function, K. Therefore, training involves solving the following convex quadratic problem:

$$
\begin{aligned}
&\min \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \alpha_i^k \alpha_j^k k_k(x_i^k, x_j^k) - \sum_{i=1}^{N_k} \alpha_i^k k_k(x_i^k, x_i^k) \\
&\text{s.t.} \ \sum_{i=1}^{N_k} \alpha_i^k = 1, \ \alpha_i^k \in [0, C], \ \forall i.
\end{aligned}
\tag{13}
$$

When the Gaussian function is chosen for the kernel function, it is always the case that $k(x,x)=1$ for each $x \in R^d$. Thus, the above problem can be further simplified as follows:

$$\begin{aligned} &\min \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \alpha_i^k \alpha_j^k k_k(x_i^k, x_j^k) \\ &\text{s.t.} \sum_{i=1}^{N_k} \alpha_i^k = 1, \quad \alpha_i^k \in [0,C], \quad \forall i. \end{aligned} \tag{14}$$

Note that in this case, the decision function of each class can be summarized as follows:

$$f_k(x) = R_k^2 - \left[1 - 2\sum_{i=1}^{N_k} \alpha_i^k k_k(x_i^k, x) + \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \alpha_i^k \alpha_j^k k_k(x_i^k, x_j^k) \right] \geq 0 \tag{15}$$

Since the output $f_k(x)$ of a one-class SVM defined in different feature spaces represents the absolute distance between the corresponding data and the decision boundary, determining the pertaining class by comparing absolute distances in different feature spaces is not recommended. Accordingly, we calculate the relative distance $\hat{f}_k(x) = f_k(x)/R_k$, and decide that the class having the maximum relative distance is the one to which the input data x pertains.

$$\begin{aligned} \text{Class of } x &\equiv \arg\max_{k=1,\cdots,K} \hat{f}_k(x) \\ &\equiv \arg\max_k \left[\left\{ R_k^2 - \left(1 - 2\sum_{i=1}^{N_k} \alpha_i^k k_k(x_i^k, x) + \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \alpha_i^k \alpha_j^k k_k(x_i^k, x_j^k) \right) \right\} / R_k \right] \end{aligned} \tag{16}$$

3.4 Real-time Internet application traffic classification system

The proposed real-time Internet application traffic classification system is composed of four modules: 1) The flow data collection and application traffic classification module of two online process modules, and the attribute subset selection and SVM training module of two offline process modules (See **Fig. 3**.) In the flow data collection module, the basic unit of traffic classification (i.e., the flow) is created from the real-time packet data. 2) The attribute subset selection module selects the optimal flow attribute subset for each hierarchical classification that improves the accuracy and the classification speed of the entire classification system. 3) The SVM training module performs training based on the attribute selection of the flow for each hierarchical classification. 4) In the application traffic classification module, the module that completes training for each hierarchy in the SVM training module is used to perform the incoming flow classification.

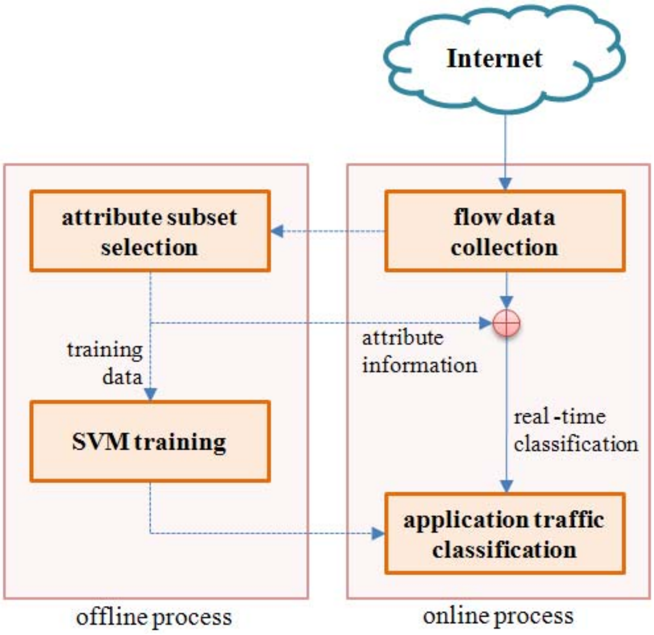

Fig. 3. Overall structure of the Internet application traffic classification system

The hierarchical Internet application traffic classification system schematized in **Fig. 4** is composed of three layers. The first layer is the binary classifier SVM layer that classifies the P2P and non-P2P traffic. The second layer classifies the P2P traffic into three representative forms; file-sharing, messenger and TV. The third layer performs specialized classification of all individual applications. In the experiment, we classify them into 16 individual applications. Feature selection and reduction are performed using the attribute subset selection method for each layer in off-line process. Once it is done, it is no longer needed at on-line process in the proposed real-time Internet application traffic classification system.

The actual traffic classification procedures of the test data are as follows: In the first layer, the traffic flow collected from the network is rapidly classified into P2P and non-P2P traffic using a binary classifier. In the second layer, P2P traffic types are classified into file sharing, messenger and TV using three SVDDs. Therefore, efficient resource management is enabled by managing the bandwidth corresponding to the traffic type that can cause system overload, and by guaranteeing more stable and appropriate QoS. Particularly, classifying by type and managing the P2P traffic causing complicated port avoidance and network congestion enables the system to support the stability of the network environment. In the last layer, specialized classification of all individual traffic types is based on SVDDs; by independently training the SVDDs allocated by traffic, faster and more efficient training and reconfiguration are enabled. Moreover, even when new application traffic is added, the cost of updates and scalability can be reduced by training only the extra part of the SVDD corresponding to the new application traffic, instead of retraining the entire system. As a result, the proposed system can perform flexible network management by controlling the levels of abstraction via generalization and specialization according to the purpose of the network traffic manager, similar to the concept hierarchy of data mining.

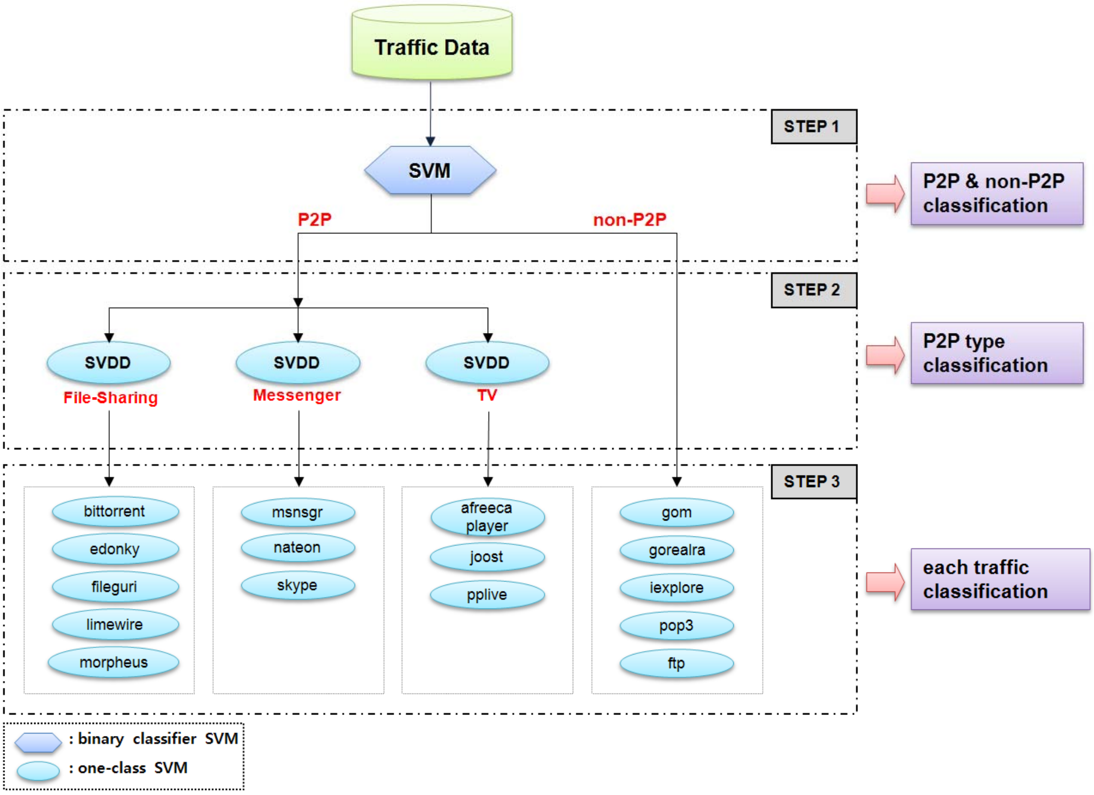

Fig. 4. Architecture of the hierarchical Internet application traffic classification system

4. Experiments

4.1 Data collection and data sets

In our experiment, a real-time classification environment was built using KU-MON. Also a verification network was developed that enabled us to evaluate the classification accuracy of the campus Internet traffic based on KU-MON and TMA.

The KU-MON system that is an extension of the NG-MON system [19] enables real-time data collection and analysis of the Internet traffic in an enterprise network such as a campus network. The KU-MON system is shown in **Fig. 5**. It is composed of three modules; 1) the flow generator module, which captures all the packets from a target network link and creates the corresponding flow information in real-time; 2) the flow store module, which stores the created flow data during some amount of time for on-line cascading analysis and off-line analysis; 3) the traffic analyzer module, which performs various analysis according to the purpose of the network manager. A traffic analyzer can get the flow data either directly from a flow generator for real-time analysis or indirectly from a flow store. The proposed real-time traffic classification module in this paper is one of the KU-MON traffic analyzer modules. The flow generator creates all of the features for a flow, listed in **Table 1,** on-the-fly and delivers the flow record to the proposed classification system as soon as a flow record is completed.

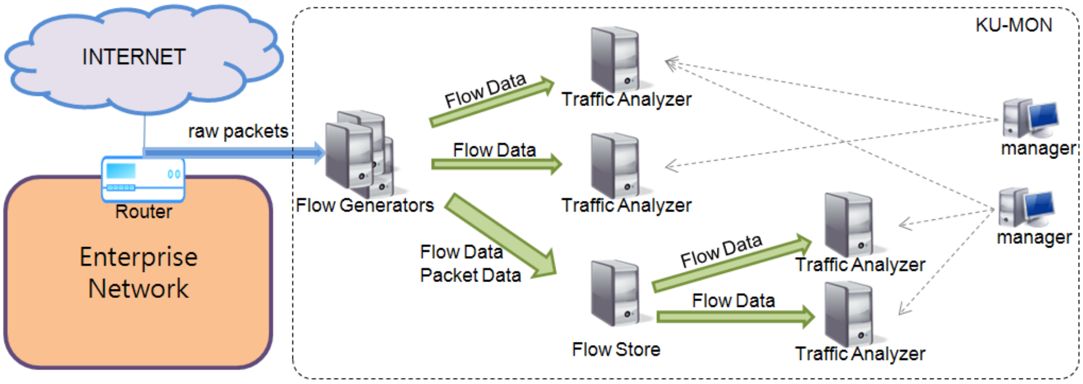

Fig. 5. KU-MON: Real-time traffic collection and analysis system

The flow record created in KU-MON does not include any classification information, such as application traffic type and application name [19][26]. Accordingly, a verification network was built that precisely determines the application traffic type and name by utilizing the socket-process information collected from the end-host's TMA and the flow information from the flow generator [14] (See **Fig. 6**.). TMA is installed at the end-hosts, and it is an agent program that performs the function of collecting TCP/UDP socket information created for network communication and the process information related to that socket in real-time. The socket-process information collected from each end-host is transmitted to the central collection server, the TMS. In the TMS, based on the process-socket information collected from each end-host and the flow information created from the packet data, the application name and type of each flow are determined, and the ground truth data is created. We can exactly verify the classification result of our proposed classification using this ground-truth traffic data.

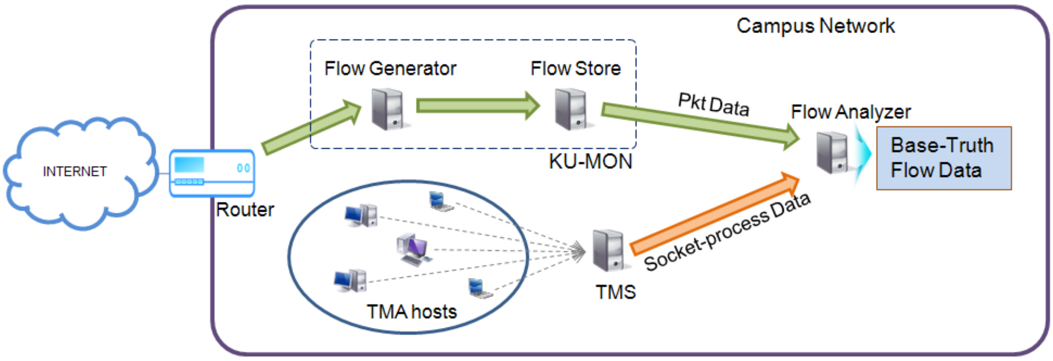

Fig. 6. Method to create the ground truth traffic data for traffic classification

In this experiment, as shown in **Table 2**, we selected 16 applications, including 11 P2P applications that are widely used in South Korea and other countries, and 5 non-P2P applications. In addition, based on KU-MON and TMA, the training and test data were generated from the flow record and socket-process record. For each type of application traffic, 200 flow record sets were collected for the verification test.

Table 2. P2P and non-P2P applications used for the experiment

P2P	File-Sharing	bittorrent, edonky, fileguri, limewire, morpheus
	Messenger	msn, nateon, skype
	TV	afreeca player, joost, pplive
Non-P2P		gom, gorealra, iexplore, pop3, ftp

4.2 Experimental results and analysis

To measure the classification accuracy of the proposed system, the precision and recall were used as the performance measurements [4]. For a given class, the number of correctly classified objects is denoted True Positives (TP). The number of falsely identified objects is denoted False Positives (FP). The number of objects from a class that are falsely labeled as belonging to another class is denoted False Negatives (FN). Precision is the ratio of True Positives to True Positives and False Positives. This determines the number of correctly identified objects. Recall is the ratio of True Positives to True Positives and False Negatives. This determines the number of misclassified objects in a class.

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

4.2.1 Classification of P2P and non-P2P traffic

The first experiment involved rapid classification of P2P and non-P2P traffic. Training was performed by a binary classifier, SVM, with 1,100 P2P traffic flows that were randomly extracted in groups of 100 from each of the 16 applications, and 500 for the non-P2P traffic. The test used a total of 1,600 data sets extracted in groups of 100 from traffic not used in training. In order to select the optimal attribute subset for binary classification of P2P and non-P2P traffic, the Correlation Feature Selection (CFS) of Weka [27] was used. The attribute subsets selected from CFS were {prot, in_fin, in_window_min, in_window_max, in_jitter_avg, out_pkt_max, out_window_min, out_window_avg}. **Table 3** shows the comparison between the result when only the eight attributes selected from CFS were used, and when all 39 attributes were used. The classification precision for the optimal attribute subset was higher than the case with all attributes. This implies that the deleted attributes didn't influence classification, or unnecessary attributes causing misclassification were eliminated. Therefore, by reducing the attributes used in classification, not only can the classification speed be improved, but the classification precision can also be improved.

Table 3. Performance measurement of P2P and non-P2P classification

Evaluation item type	CFS (8 features used)			All features used		
	σ value	recall	precision	σ value	recall	precision
P2P	0.88	98.27	96.35	0.8	96.6	95.8
Non-P2P		91.8	95.6		90.6	92.4

4.2.2 P2P classification by type

In the second experiment, P2P traffic was subdivided into the representative P2P traffic types; file sharing, messenger and TV. For this experiment, five file sharing applications (bittorrent, edonky, fileguri, limewire, morpheus), three messenger applications (msn, nateon, skype) and three TV applications (afreeca player, joost, pplive), for a total of 11 applications of P2P traffic data, were randomly extracted in groups of 100 according to application. The training was performed by three SVDDs (file sharing SVVD, messenger SVDD, TV SVDD), and the test data was from data not used in training. For CFS, the selected optimal attribute subsets were {in_dOctets, in_pkt_min, in_pkt_avg, out_pkt_min, out_pkt_avg, out_pkt_sdev, out_jitter_avg}, and the classification result for only seven attributes and for 39 attributes is shown in **Table 4**. The trade-off constant C was 0.1. The constant of the Gaussian function σ was 0.15 for file sharing, 0.43 for messenger and 0.58 for TV in case of CFS selected features. Even when only seven attributes were used, it was confirmed that the recall and precision measures of the three types of P2P traffic were satisfactory.

Table 4. Performance measurement of P2P traffic classification

Evaluation item type	CFS (7 features used)			All features used		
	σ value	recall	precision	σ value	recall	precision
File sharing	0.15	97.0	95.7	0.12	97.2	97.8
Messenger	0.43	94.3	94.3	0.40	95.3	96.3
TV	0.58	93.7	95.9	0.48	91.3	92.9

4.2.3 Entire traffic classification by individual applications

The third experiment involved specialized classification of all 16 individual applications. 100 data sets were randomly extracted from each application's traffic and trained with each of the SVDDs. Moreover, the test data was from data not used in training. The experimental results are shown in **Table 5**. The optimal attribute subsets selected from CFS were {in_pkt_max, in_pkt_avg, in_window_min, in_window_max, out_dOctets, out_pkt_min, out_pkt_max, out_pkt_avg, out_jitter_max} and the classification results for nine attributes and for all 39 attributes are compared in **Table 5**. Here, the trade-off constant is 0.1 and the constant of Gaussian function σ corresponds to each type of traffic.

Table 5. Performance measurement of the entire application traffic classification

Evaluation application	CFS (9 features used)			All features used		
	σ value	recall	precision	σ value	recall	precision
bittorrent	0.35	98.0	96.1	0.30	94.0	95.9
edonky	0.60	88.0	90.7	0.65	82.0	98.8
fileguri	0.05	100.0	90.1	0.07	100.0	79.4
limewire	0.58	88.0	85.4	0.65	94.0	96.9
morpheus	0.60	71.0	81.6	0.60	64.0	100.0
msn	0.55	85.0	75.9	0.50	84.0	94.4
nateon	0.55	83.0	79.1	0.60	88.0	91.7
skype	0.45	91.0	94.8	0.40	84.0	95.5

afreeca player	0.65	88.0	85.4	0.60	89.0	90.8
joost	0.50	84.0	85.7	0.60	78.0	96.3
pplive	0.50	86.0	89.6	0.50	84.0	93.3
gom	0.30	94.0	91.3	0.25	90.0	90.1
gorealra	0.55	93.0	98.9	0.50	96.0	86.5
iexplore	0.45	84.0	81.6	0.55	89.0	91.8
pop3	0.50	93.0	98.9	0.50	97.0	86.6
ftp	0.35	96.0	100.0	0.3	98.0	85.5

Table 5 shows that the overall accuracy for all 39 features is slightly greater than for the CFS selected features. Both provide recall and precision rates of about 90%, which is accurate enough for classification of Internet traffic into up to 16 application classes. In the case of fileguri and gom, both recall and precision for CFS selected features was greater than for all 39 features. But for most cases, the recall and precision for all features was better than for CFS.

5. Conclusions

In this paper, we proposed a real-time Internet application traffic classification system in order to overcome the problems of the conventional methods that are based on port number and payload. The proposed method hierarchically combines a binary classifier SVM and Support Vector Data Descriptions (SVDDs). It selects an optimal attribute subset from the bi-directional traffic flows generated by our traffic analysis system (KU-MON) that enables real-time collection and analysis of campus traffic. Since the proposed system enables both coarse- and fine-grained classification of Internet application traffic, it guarantees an efficient resource management, which supports a stable network environment, seamless bandwidth and appropriate QoS. Moreover, even when new application traffic is added, it contributes to incremental updating and scalability, thus only new application traffic needs extra training, instead of retraining the entire system. The performance of the proposed system was verified via experiments that confirmed that the recall and precision measures were satisfactory. Our future work will be concerned with useful knowledge discovery and analysis included in the mechanism of Internet application traffic classification, and it will be conducted using a decision tree algorithm and association rule mining for in-depth analysis research. Also we are planning to analyze the importance of the selected attributes by CFS algorithms at each step of our classification hierarchy.

References

[1] H. Schulze and K. Mochalski, “Ipoque Internet Study 2008/2009,” Available from: <http://www.ipoque.com/>.

[2] G. Szabo, I. Szabo, and D. Orincsay, “Accurate traffic classification,” in *Proc. of the IEEE International Symposium on World of Wireless Mobile and Multimedia Networks*, pp.1-8, 2007.

[3] L. Zhou, X. Wang, W. Tu, G. Mutean, and B. Geller, “Distributed scheduling scheme for video streaming over multi-channel multi-radio multi-hop wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol.28, no.3, pp.409-419, 2010.

[4] J. Erman, A. Mahanti, and M. Arlitt, “Internet traffic identification using machine learning,” in *Proc. of the IEEE Conference on Global Telecommunications*, pp.1-6, 2006.

[5] T. Auld, A. Moore, and S. Gull, “Bayesian neural networks for Internet traffic classifications,”

IEEE Transactions on Neural Networks, vol.18, no.1, pp.223-239, 2007.

[6] Y. Liu, R. Wang, H. Huang, Y. Zeng, and H. He, "Applying support vector machine to P2P traffic identification with smooth processing," in *Proc. of the IEEE International Conference on Signal Processing*, vol.3, pp.16-20, 2006.

[7] F. J. Gonzalez-Castano, P. S. Rodriguez-Hernandez, R. P. Martinez-Alvarez, A. Gomez, I. Lopez-Cabido, and J. Villasuso-Barreiro, "Support vector machine detection of peer-to-peer traffic," in *Proc. of the IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, pp.103-108, 2006.

[8] A. Yang, S. Jiang, and H. Deng, "A P2P network traffic classification method using SVM," in *Proc. of the 9th International Conference for Young Computer Scientists*, pp.398-403, 2008.

[9] X. Zhou, "A P2P traffic classification method based on SVM," in *Proc. of the International Symposium Computer Science and Computational Technology*, pp.53-57, 2008.

[10] N. Cascarano, F. Risso, A. Este, F. Gringoli, L. Salgarelli, A. Finamore, and M. Mellia, "Comparing P2PTV traffic classifiers," in *Proc. of the IEEE International Conference on Communications*, pp.1-6, 2010.

[11] H. Lee, J. Song, and D. Park, "Intrusion detection system based on multi-class SVM," *LNAI*, vol.3642, pp.511-519, 2005.

[12] M. Tai, S. Ata, and I. Oka, "Fast, accurate, and lightweight real-time traffic identification method based on flow statistics," *LNCS,* vol.4427, pp.255-259, 2007.

[13] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel traffic classification in the dark," *Proceedings of ACM SIGCOMM*, vol.35, no.4, pp.229-240, 2005.

[14] J. Li, S. Zhang, S. Liu, and Y. Xuan Ye, "Active P2P traffic identification technique," in *Proc. of the IEEE CIS 2007*, pp.37-41, 2007.

[15] G. Zhang, G. Xie, J. Yang, Y. Min, Z. Zhou, and X. Duan, "Accurate online traffic classification with multi-phases identification methodology," in *Proc. of the IEEE International Conference on Consumer Communications and Networking*, pp.141-146, 2008.

[16] G. Munz, H. Dai, L. Braun, and G. Carle, "TCP traffic classification using Markov models," *LNCS,* vol.6003, pp.127-140, 2010.

[17] P. Phaal, S. Panchen, and N. McKee, "InMon corporation's sFlow: A method for monitoring traffic in switched and routed networks," *IETF RFC3176*, 2001.

[18] Cisco Systems, White Papers, "NetFlow services and applications," Available from: <http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm>.

[19] S. Han, M, Kim, H, Ju, and J. W. Hong, "The architecture of NG-MON: A passive network monitoring system," *LNCS,* vol.2506, pp.16-27, 2002.

[20] M. Hall, "Correlation-based feature selection for machine learning," PhD Diss. Department of Computer Science, Waikato University, Hamilton, NZ, 1998.

[21] I. Seok, J. Lee, and B. Moon, "Hybrid genetic algorithms for feature selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.26, no.11, pp.1424-1437, 2006.

[22] F. Fleuret, "Fast binary feature selection with conditional mutual information," *Journal of Machine Learning Research*, vol.5, pp.1531-1555, 2004.

[23] Y. Sun and J. Li, "Iterative RELIEF for feature weighting," in *Proc. of the 23rd International Conference on Machine Learning*, pp.913-920, 2006.

[24] S. Gunn, "The support vector machines for classification and regression," Univ. of Southampton, Technical Report, 1998.

[25] J. Han and M. Kamber, "Data Mining: Concepts and Techniques," Morgan Kaufman, 2nd Ed., 2007.

[26] FileGuri, Available from: <http://www.fileguri.com/>.

[27] Machine Learning Lab in The University of Waikato, Available from: <http://www.cs.waikato.ac.nz/ml>.

Jaehak Yu received his B.S degree in Computer Science from Konkuk University in 2001. He received his M.S and Ph.D degrees in Computer Science from Korea University, Korea, in 2003 and 2010, respectively. He is currently a senior member of Electronics and Telecommunications Research Institute, Korea. His recent research interests include data and information analysis, intelligent network management, RFID/USN, network mining, and home network security.

Hansung Lee received his B.S, M.S, and Ph.D degrees in Computer Science from Korea University, Korea, in 1996, 2002, and 2008, respectively. He is currently a senior member of Electronics and Telecommunications Research Institute, Korea. From July 1996 to July 1999, he worked for DAEWOO Engineering Company.His recent research interests include human recognition, multimedia mining, intelligent database, machine learning, and soft computing.

Younghee Im received her B.S degree in Computer Science from Korea University, Korea, in 1994, and her Ph.D degree in Computer Science from Korea University, Korea, in 2001. She joined Korea University in 2005, where she is currently an Invitational Professor in the Dept. of Computer and Information Science. Her research interests include machine learning, context awareness, and intelligent database.

Myung-Sup Kim received his B.S, M.S, and Ph.D degrees in Computer Science and Engineering from POSTECH, Korea, in 1998, 2000, and 2004, respectively. From September 2004 to August 2006 he was a Postdoctoral Fellow in the Dept. of Electrical and Computer Engineering, University of Toronto, Canada. He has been an Assistant Professor in the Dept. of Computer and Information Science, Korea University, Korea, since September 2006. His research interests include Internet traffic monitoring and analysis, service and network management, and Internet security.

Daihee Park received his B.S degree in Mathematics from Korea University, Korea, in 1982, and his Ph.D degree in Computer Science from Florida State University, USA, in 1992. He joined Korea University in 1993, where he is currently a Professor in the Dept. of Computer and Information Science. His research interests include data mining and intelligent database.