Split Validation 방법에서의 트래픽 분류를 위한 최적의 ML 알고리즘과 Feature Set

*정광본, *최미정, **김명섭, *원영준, *홍원기
*포항공과대학교, **고려대학교
*{jkbon, mjchoi, yjwon, jwkhong}@postech.ac.kr, **tmskim@korea.ac.kr

The Best Machine Learning Algorithm and Feature Set for Traffic Classification using Split Validation

*Kwang Bon Jung, *Mi Jung Choi, **Myung-Sup Kim, *Young J. Won, and *James W. Hong
*POSTECH, **Korea University

요 약

현재 Traffic classification의 방법은 payload 분석이나 port를 기반으로 하는 정적인 방법에서 동적으로 변하는 application의 변화에 대처하기 위해 ML 알고리즘을 적용하는 방법으로 변하고 있다. 그러나 현재의 ML 알고리즘을 이용한 traffic classification 연구는 offline 환경에 맞추어 이루어지고 있다. 특히, 현재의 기존 연구들은 testing 방법으로 cross validation을 이용하여 traffic classification을 수행하고 있다. 그러나 실제 네트워크 모니터링 상에서는 training과 testing set이 분리된 split validation 방법으로 트래픽 분류를 수행해야 한다. 본 논문에서는 testing방법으로 cross validation과 split validation을 이용했을 때, traffic classification의 정확도 결과를 비교한다. J48, REPTree, RBFNetwork, Multilayer perceptron, BayesNet, NaiveBayes와 같은 ML 알고리즘과 다양한 feature set을 이용하여 트래픽을 분류한다. 그리고 split validation을 이용한 traffic classification에 적합한 최적의 ML 알고리즘과 feature set을 제시한다.

I. 서론

네트워크에서 제공하는 서비스가 다양해지고 네트워크에 흘러 다니는 트래픽의 종류가 다양해지면서 traffic classification은 네트워크를 효율적으로 그리고 안전하게 관리해야 함에 있어서 더욱 중요해지고 있다. 지금까지 traffic classification은 주로 포트 번호를 기반으로 이루어지고 있으며, payload를 통한 분석도 행해지고 있다. 그러나 최근의 애플리케이션은 dynamic한 포트 번호를 할당하여 패킷을 발생하며, payload를 통한 분석에 있어서도 애플리케이션에서 생성한 패킷의 payload가 암호화되어 전송되는 경우가 많아져 payload 분석을 통한 트래픽 분류를 어렵게 하고 있다. 이러한 추세는 payload나 포트 번호를 이용한 traffic classification의 정확도를 떨어뜨린다. 이러한 문제의 해결책으로 트래픽 특징에서 얻은 트래픽의 통계적 feature들에 Machine Learning (ML) 알고리즘을 적용하여 트래픽을 분류하는 것이 대안으로 제시되고 있다 [4, 6, 7, 11].

기존 ML 알고리즘을 이용한 traffic classification 연구에서는 source IP, destination IP, source port, destination port, protocol의 5가지 정보로 정의할 수 있는 flow를 기반으로 네트워크 트래픽의 feature set을 구성하여 트래픽 분류를 수행하고 있다.

기존 연구에서 제시된 traffic classification 방법은 성능을 평가함에 있어 동일한 시간대에 수집된 하나의 데이터 set 안에서 training set과 testing set을 구성하는 cross validation을 채택하고 있지만, 이는 실제 네트워크를 운영하는 운영자의 입장에서 현실적으로 적용하기 어렵다. 이를 보완하기 위해서 ML 알고리즘을 적용하여 훈련시키는 training set과 성능 평가에 사용되는 testing set을 분리하는 split validation 기법이 적용되어야 한다. 본 논문에서는 기존 연구에서 진행되어 온 cross validation 기반의 traffic 분류 방법에 대한 문제점을 찾아보고 split validation의 필요성을 제시한다. 이러한 필요성을 기반으로 split validation 방법을 적용했을 때 최적의 accuracy 결과를 보이는 ML 알고리즘과 feature set을 실험을 통하여 제시하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문과 관련 있는 traffic classification에 대한 기존 연구에 대해서 살펴보고 기존 연구의 문제에 대해서 정리한다. 3장에서는 본 연구에서 사용한 traffic data set에 대해 설명한다. 4장에서는 본 연구에서 실행한 실험에 대한 내용과 실험 결과에 대해서 설명한다. 마지막으로 5장에서 결론과 향후 연구로 본 논문을 맺는다.

II. 관련 연구

이 장에서는 ML 알고리즘을 적용하여 traffic classification을 수행한 기존 연구를 살펴본다. 또한 기존 연구의 문제점에 대해서 정리한다.

2.1 기존 연구

ML 알고리즘을 이용한 traffic classification 연구는 약 1990년대 말부터 이루어졌으며, 현재에도 많은 연구가 진행 중이다. 표 1을 보면 Protocol, bytes count, connection duration, packet size statistics와 inter-packet arrival statistics를 기본적인 feature set으로 선택하고

본 연구는 2007 년도 두뇌한국 21 사업과 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음(IITA-2007-(C1090-0701-0045))

필요에 의해 새로운 feature를 추가하여 트래픽을 분류하고 있다. ML 알고리즘을 이용한 traffic classification에 관련해서 진행된 많은 연구들이 offline 환경에서 training과 testing 트래픽의 구분 없이 트래픽을 분류하는 것에 초점을 두고 있지만, 본 논문에서는 training과 testing 트래픽이 다른 환경인 practical 네트워크 모니터링 환경에서 네트워크 트래픽을 분류하는 것에 초점을 두고 있다.

	ML 알고리즘	Feature set
Erman et al. [2]	K-Means, DBSCAN, AutoClass	total number of packets, packet size, payload size, number of transferred bytes, inter-packet arrival time
Erman et al. [4]	EM, Naive Bayes classifier	total number of packets, packet size, flow duration, inter-packet arrival time
Nguyen et al. [5]	Naive Bayes	IP address, port number, protocol, inter-packet arrival time, inter-packet length variation, IP packet length
Williams et al. [6]	Naive Bayes, C4.5, Bayesian Network, Naive Bayes Tree	protocol, flow duration, flow volume in bytes and packets, packet length, inter-packet arrival time
Zander et al. [11]	Expectation Maximization (EM)	IP address, port number, inter-packet arrival time, packet length, flow size and duration

표 1. ML 알고리즘을 이용한 Traffic Classification

또한 기존 연구에서는 traffic classification을 하기 위한 데이터를 backbone에서 얻고 있는데 이 데이터가 어떤 애플리케이션에서 발생된 것인지 정확히 알 수 있는 방법이 없어 포트 번호에 따른 트래픽 분류 결과를 제시하고 있다. 즉, 기존 연구에서는 애플리케이션 별로 분류한 결과가 정확하다는 보장이 없다. 또한 기존의 traffic classification 연구에서 많이 사용되지 않은 Neural Network 기반의 ML 알고리즘들로 적용해 봄으로써 기존 연구에서 많이 사용된 ML 알고리즘과의 분류 정확도 결과도 비교할 수 있다.

2.2 기존 연구의 문제

이 장에서는 기존 연구 방법을 실용적인 (practical) 네트워크상에서의 트래픽 분류에 적용할 때 고려해야 할 점을 정리한다. 기존 연구의 문제는 성능평가를 위한 testing 기법과 관련된 문제이다. 기존 연구 [2, 4, 5, 6, 11]를 살펴보면 training과 testing을 위한 기법으로 cross validation 기법을 선택하고 있다. Cross validation 기법을 선택하는 것은 의도하지 않은 문제점을 안고 있다. 예를 들어, 포트 번호와 같은 feature에서 문제점을 찾아볼 수 있는데, 대부분의 애플리케이션은 더 이상 고정적인 포트 번호를 사용하지 않는다. 특히 애플리케이션을 사용하는 client에서 사용하는 포트 번호는 유동성의 정도가 더 심하다. Client 애플리케이션의 포트 번호는 특정한 seed 값에서 1씩 증가하면서 할당되는 특징을 가지고 있다. 특정 시간에 특정한 애플리케이션 'A' 만을 사용한 데스크톱에서 얻은 데이터를 살펴보니 client에서 할당한 포트 번호가 1000번부터 3000번까지 1씩 증가하는 것을 볼 수 있었다. 이 데이터를 cross validation 기법에 대입해 보면, 그 데이터 set 안에서 training을 위한 데이터와 testing을 위한 데이터가 형성되므로, 이렇게 형성된 데이터들을 가지고 분류를 하게 되면 1~3000번에 있는 포트 번호를 가지는 데이터는 'A' 라는 애플리케이션에 의해서 발생된 것이라고 분류되기 쉽고 그 외의 포트 번호를 가지는 데이터는 'A' 라는 애플리케이션에 의해서 발생된 것이라고 분류되기 어렵다. 만약 split validation을 이용하여 분류를 할 때, 앞에서 말한 데이터를 training을 위한 데이터로 사용하고 포트 번호가 6000번부터 9000번까지 형성된 데이터를 testing을 위한 데이터로 사용하여 분류를 수행하게 되면 cross validation을 이용하여 분류를 한 결과만큼 분류의 정확도가 높게 나오지 않는다. 따라서 split validation의 경우는 client의 포트 번호는 적절한 feature set이 될 수 없다. 즉, cross validation을 이용한 분류를 기반으로 하는 기존의 연구 결과가 practical 네트워크 모니터링 환경에서의 네트워크 애플리케이션을 분류하는데 적용하기에 부적절한 면이 있다는 것을 알 수 있다.

III. 트래픽 트레이스 및 Feature Set

이 장에서는 네트워크 트래픽을 수집한 환경과 우리가 ML 알고리즘에 적용하기 위한 feature set의 종류 및 분류 결과의 정확성을 측정하기 위한 방법에 대해서 살펴본다.

3.1 트래픽 트레이스 (Traffic Trace)

본 논문에서는 하나의 데스크톱에서 ethereal [10]을 이용하여서 7가지의 대표 애플리케이션의 data trace를 수집하였다. 이 대표 애플리케이션은 POSTECH의 네트워크 현황을 모니터링 하고 있는 NG-MON [3]을 참조하여 사람들이 많이 사용하고 있다고 판단되는 애플리케이션들 중에 다양한 종류를 선택한 것이다. 7개의 대표 애플리케이션으로 online으로 음악방송을 제공해주는 'alsong', online으로 방송을 제공받거나 제공할 수 있는 'afreeca', Web disk인 'clubbox', ftp를 이용해서 파일을 주고 받을 수 있는 'alftp', Microsoft에서 제공하는 chatting 애플리케이션인 'MSN messenger', 여러 가지 contents를 실시간으로 제공해주는 'Gom', Web browser인 'iexplore'를 선택하였다.

각 애플리케이션마다 training과 testing을 위한 데이터가 각각 필요하기 때문에 packet 데이터도 두 번을 수집했다. 각각 training을 위한 packet은 4시간 정도 모으고, testing을 위한 packet 데이터는 1시간 30분 동안 모았다. Split validation을 위해 두 개의 데이터를 수집한 시간은 서로 다르게 하였다. 표 2는 packet 데이터의 용량과 이렇게 모은 데이터들을 flow 데이터로 가공한 값을 나타낸 것이다.

애플리케이션	Training		Testing	
	Size(MB)	Flow(개수)	Size(MB)	Flow(개수)
MSN	454.07	377	454.07	65
Afreeca	946.97	483	946.97	223
Clubbox	904.244	4490	904.244	835
Gom	168.654	1633	168.654	756
Alftp	1574.53	501	1574.53	234
Iexplore	73.542	1386	73.542	258
Alsong	16.3975	85	16.3975	49

표 2. 각 애플리션별 Data size 및 Flow 개수

3.2 Feature Set 정의

Traffic classification을 위해 수집하는 정보인 feature를 선정하는 기준은 데이터를 aggregation하는 기준에 따라서 달라질 수 있다. 본 논문에서는 flow를 기준으로 데이터를 aggregation하였으며 네트워크 트래픽을 분류하기 위해 가장 많이 사용되는 feature 정보들 [5]은 아래와 같다.

- IP address (source, destination)
- Port number (source, destination)
- Byte counts
- Connection duration
- Packet size statistics (minimum, maximum, mean, standard deviation)
- Inter-packet arrival time statistics (minimum, maximum, mean, standard deviation)

대부분의 기존 연구 [2, 4, 5, 6, 11]에서는 source IP, destination IP, source port number, destination port number 이 네 가지를 feature들을 선택하여 분류를 수행하고 있다. 본 논문에서는 feature set을 형성할 때에, feature중 IP address와 포트 번호를 선택하거나 혹은 선택하지 않은 다양한 set을 형성하여서 최적의 feature set을 구하려 한다. 따라서 다음과 같이 5가지 종류의 feature set을 정의했으며, 이들 중 최적의 feature set을 찾고자 한다.

(1) all: 모든 feature가 선택된 경우
(2) without port: 모든 feature에서 source와 destination의 port number를 제외한 경우
(3) without IP: 모든 feature에서 source와 destination의 IP address를 제외한 경우
(4) without IP&port: 모든 feature에서 source와 destination의 IP address, port number를 제외한 경우
(5) without src IP&src port: 모든 feature에서 source IP address와 port number가 빠진 경우

ML 알고리즘을 통한 분류가 잘 되었는가 평가하기 위해서 가장 일반적인 평가 기준은 overall accuracy이다. Overall accuracy는 전체 데이터를 하나로 놓고 제대로 분류가 된 양이 얼마나 되는지에 대해서 알아보기 위한 평가 기준이다. Overall accuracy는 True Positive (TP)를 사용하여 다음과 같은 식으로 나타낼 수 있다.

$$\text{Overall accuracy} = \frac{\sum TPofEachApplication}{TotalElement} \quad \text{(식 1)}$$

IV. 실험 결과

이 장에서는 5가지 feature set과 다양한 ML 알고리즘 중에서 주어진 traffic trace에 대하여 overall accuracy를 가장 높이는 최적의 feature set과 ML 알고리즘을 찾고자 한다. Cross validation과 split validation의 2가지 testing 기법에 따른 traffic classification 결과를 비교하는 실험과 split validation에서 최적의 알고리즘과 최적의 feature set을 구하는 실험을 수행하였다. 본 논문에서는 ML tool 중 하나인 Weka [1]를 이용하여 실험하였다.

4.1 Testing 기법에 따른 트래픽 분류 분석

이 장에서는 먼저 cross validation과 split validation의 testing 기법에 따른 분류 결과를 비교하는 실험을 수행한다. 4.1장에서 설명한 방법으로 training과 testing을 수행하였다. 여기에서 사용한 ML 알고리즘은 Decision tree인 J48 알고리즘이다. 이 알고리즘을 선택한 이유는 기존 연구 [6, 8, 9]에서 cross validation 방법으로 실험했을 때, J48 알고리즘이 traffic classification에 있어서 뛰어난 성능을 보였기 때문이다. 이 실험에서 사용하는 feature set은 cross validation과 split validation의 분류 정도를 비교하기 위해 대표로 모든 feature (1: all)를 사용했을 때와 전체 feature에서 IP address를 사용하지 않았을 경우 (3: without IP) 2가지를 살펴보았다.

그림 1은 cross validation과 split validation을 testing 방법으로 사용하였을 때의 overall accuracy (식 1)를 보여주고 있다. 먼저 모든 feature로 구성된 feature set을 적용할 때, cross validation의 경우 overall accuracy가 약 95.55%로 측정된 반면에 split validation을 testing method로 사용하면 약 62.64%의 overall accuracy 값을 보인다. IP address를 제외한 feature들로 구성된 feature set을 적용할 때에도 cross validation을 testing method로 사용하면 overall accuracy가 약 95.76%값을 보였고, split validation의 경우는 overall accuracy가 약 63.56%로 측정되었다.

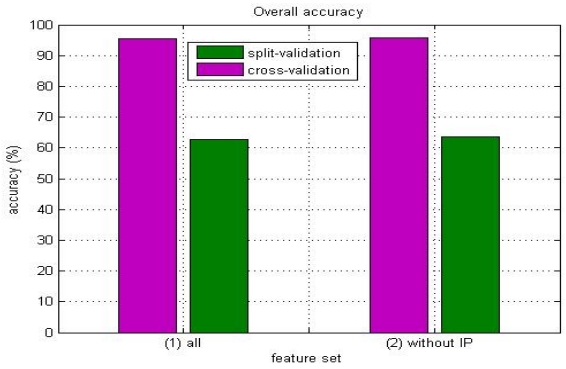

그림 1. Cross & Split Validation의 Overall Accuracy

두 가지 feature set 모두 cross validation을 적용하면 기존 논문과 같이 traffic classification에 있어서 높은 overall accuracy를 가지고 있지만, split validation을 하게 되면 그리 높지 않은 accuracy를 보임을 알 수 있다. 즉, practical 네트워크상의 traffic classification에서는 split validation이 이루어져야 함으로 기존의 cross validation의 traffic classification의 정확도 값을 그대로 받아들이기 어렵다. 또한 split validation 상에서의 최적의 ML 알고리즘과 feature set들 역시 기존의 cross validation 방법과 다를 수 있다. 따라서 본 논문에서는 실험을 통해 split validation에서의 최적의 ML 알고리즘과 feature set을 찾는 것은 의미가 있다.

4.2 Split Validation에서 최적 ML 알고리즘과 Feature Set

이 장에서는 실험을 통해 split validation 환경에서의 최적의 ML 알고리즘과 feature set을 찾아보고자 한다.

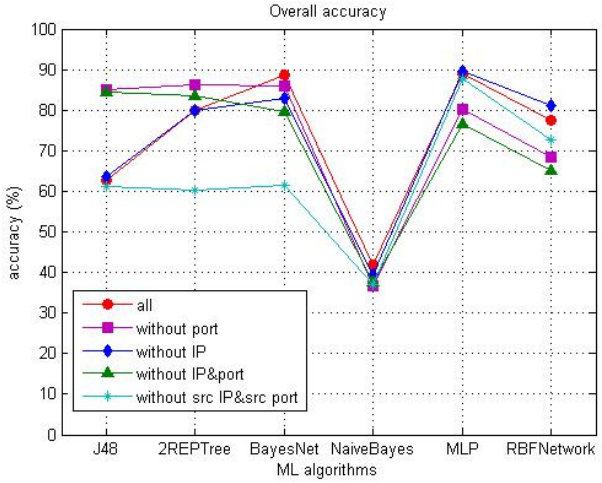

그림 2. Feature set과 ML 알고리즘에 따른 Split Validation을 적용한 Overall Accuracy

그림 2는 6개의 알고리즘과 5개의 다른 feature set을

적용하였을 때의 flow 기반의 분류 결과로써 overall accuracy를 나타내고 있다. Multilayer Perceptron (MLP) 알고리즘에 3.2장의 (3)번 feature set을 적용했을 때가 89.48%로 overall accuracy가 가장 높게 나오는 것을 볼 수 있으며, BayesNet 알고리즘에 (1)번 feature set을 적용했을 때에 88.58%로 두 번째로 높게 나오는 것을 볼 수 있다. 그 다음은 MLP 알고리즘을 이용하여 (5)번 feature set을 사용하면 높은 overall accuracy가 나옴을 볼 수 있다. J48, REPTree 알고리즘을 이용하여 (2)번이나 (4)번 feature set을 사용하였을 때에도 비교적 높은 overall accuracy가 나옴을 알 수 있다. 또한, MLP 알고리즘을 적용하여 (3)번 feature set으로 실험했을 때, traffic classification의 overall accuracy는 89%였다.

4.3 Cross Validation에서 최적 ML 알고리즘과 Feature Set

본 논문에서는 split validation 방법으로 얻은 결과를 기존 연구인 cross validation 방법을 통해 얻은 결과와 비교하기 위해서 5.2장에서 이용했던 알고리즘들과 feature set을 그대로 cross validation 방법을 통해 수행하였다.

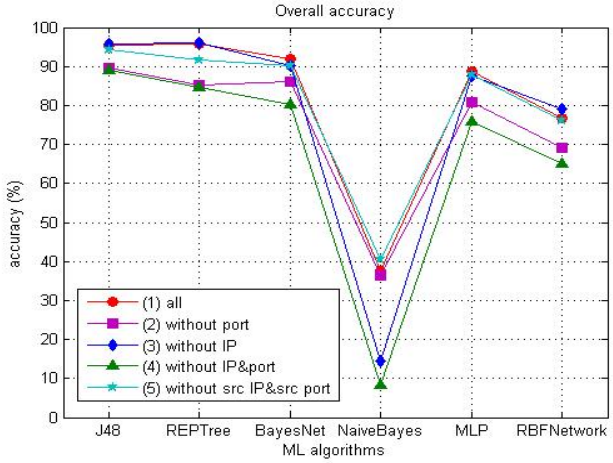

그림 3. Feature set과 ML 알고리즘에 따른 Cross Validation을 적용한 Overall Accuracy

그림 3에서 보듯이 cross validation을 수행했을 때의 최적의 알고리즘은 J48과 REPTree이다. Cross validation을 이용하여 flow를 기준으로 하였을 때에 가장 높은 overall accuracy를 갖는 알고리즘과 feature set은 REPTree와 (3)번 feature set이고 정확도는 96.12%를 보임을 알 수 있다. Split validation을 이용하여 flow를 기준으로 했을 때, 가장 좋은 overall accuracy를 가진 알고리즘과 feature set은 MLP과 IP address를 제외한 (3: without IP) feature set이고 값은 각각 89.48%이다. 즉 cross validation과 split validation에서의 최적의 ML 알고리즘과 feature set이 다름을 알 수 있다.

Split validation에서 MLP 알고리즘을 이용하면, cross validation을 이용해서 얻을 수 있는 overall accuracy 값에 미치지는 않지만, practical 모니터링 환경상에서 적용 가능한 split validation 방법에서 적어도 MLP 알고리즘을 이용하면 cross validation 수준의 충분히 좋은 결과를 얻을 수 있다는 것이다.

V. 결론 및 향후 연구

본 논문에서는 기존의 연구 논문에서 찾아 볼 수 있는 cross validation 기반의 traffic classification의 문제점을 살펴보고 그것을 해결하기 위해 split validation 기반의 traffic classification의 필요성을 제시하였다. Practical 네트워크상의 네트워크 트래픽의 분류를 위해서는 split validation 방법으로 분석해야 한다.

Cross validation과 split validation 방법론의 각 경우에 적합한 최적의 ML 알고리즘과 feature set을 실험을 통하여 제시하였다. Split validation에서의 최적의 알고리즘과 feature set은 Neural Network 계열의 MLP (Multilayer perceptron)가 최적의 overall accuracy 성능을 보였으며, 최적의 feature set은 IP를 제외한 feature set (3: without IP)이 가장 좋은 성능을 보였다.

추후에 이루어져야 할 연구는, 데이터를 수집하는 것이 하나의 데스크톱이 아닌 다수의 데스크톱에서 얻은 데이터를 가지고 classification을 해봐야 할 것이다. 그리고 좀 더 나은 feature set을 선정하기 위해서, packet의 header 정보를 좀 더 가공하여 최적의 feature를 찾는 연구가 이루어져야 한다.

참고문헌

[1] Machine Learning Lab in The University of Waikato, “Weka”, [Online] Available: http://www.cs.waikato.ac.nz/ml.

[2] Jeffrey Erman, Martin Arlitt, Anirban Mahanti, “Traffic Classification Using Clustering Algorithms”, SIGCOMM’06 Workshops, Pisa, Italy, Sep. 2006, pp. 281~286.

[3] Se-Hee Han, Myun-Sup Kim, Hong-Taek Ju and James W. Hong, “The Architecture of NG-MON: A Passive Network Monitoring System”, IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, LNCS 2506, Montreal, Canada, Oct. 2002, pp. 16~27.

[4] Jeffrey Erman, Anirban Mahanti, Martin Arlitt, “ Internet Traffic Identification using Machine Learning ” , IEEE Global Telecommunications Conference, California, USA、Nov.~Dec. 2006, pp. 1~6.

[5] Thuy T. T. Nguyen, Grenville Armitage, “Training on multiple sub-flows to optimize the use of Machine Learning classifiers in real?world IP networks”, IEEE Conference on Local Computer Networks, Tampa, Florida, USA, Nov. 2006, pp. 369~376.

[6] N. Williams, S. Zander, G. Armitage, “A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification”, SIGCOMM Computer Communication Review, Oct. 2006, pp. 7~15.

[7] Andrew W. Moore, Denis Zuev, “Internet Traffic Classification Using Bayesian Analysis Techniques”, SIGMETRICS’05, Banff, Alberta, Canada, Jun. 2005, pp. 50~60.

[8] Junghun Park, Hsiao Rong Tyan, and C. C. Jay Kuo, “Inetnet Traffic Classification For Scalable QoS Provision”, IEEE International Conference on Multimedia and Expo, Jul. 2006, pp. 1221~1224.

[9] Junghun Park, Hsiao Rong Tyan, C.-C. Jay Kuo, “GA-Based Internet Traffic Classification Technique for QoS Provisioning”, International Conference on Intelligent Information Hiding and Multimedia, Pasadena, California, USA, Dec. 2006, pp. 251~254.

[10] Etheral, http://www.ethereal.com.

[11] Sebastian Zander, Thuy Nguyen, Grenville Armitage, “Automated Traffic Classification and Application Identification using Machine Learning”, Proceedings of the IEEE Conference on Local Computer Networks, Sydney, Australia, Nov. 2005, pp. 250-257.