

# 오토마타를 이용한 메신저 트래픽의 기능별 분류에 관한 연구

준회원 이 상 우\*, 박 준 상\*, 윤 성 호\*, 종신회원 김 명 섭\*

## Study on the Functional Classification of IM Application Traffic using Automata

Sang-woo Lee\*, Jun-sang Park\*, Sung-ho Yoon\* *Associate Members*,  
Myung-sup Kim\* *Lifelong Member*

### 요 약

최근 인터넷 사용자의 증가와 다양한 인터넷 기반 응용프로그램의 증가로 네트워크 트래픽이 급증하고 있다. 효율적인 네트워크 트래픽 관리를 위해 기존의 많은 연구들에서 다양한 분류 알고리즘을 제시하였고, 대부분의 제안된 방법들은 응용 단위로 트래픽을 분류하며 분류의 정확성을 높이는데 초점을 두고 있다. 하지만 트래픽 제어의 관점에서 보았을 때 응용단위의 트래픽 분류는 응용의 기능별 제어의 기회를 제공하지 못하고 네트워크 사용자가 각 응용의 어떤 기능을 사용하는지를 파악하지 못하게 되는 문제점을 가지고 있다. 본 논문에서는 이러한 문제점들을 해결 하기 위해 NateOn, MSN, GoogleTalk의 인터넷 메신저 응용에 대하여 기능 오토마타를 설계하고 이를 통해 메신저 응용의 기능별 트래픽 분류를 위한 방법론을 제시한다.

**Key Words** : Traffic classification, Automata, Application function, State transition, Payload signature

### ABSTRACT

The increase of Internet users and services has caused the upsurge of data traffic over the network. Nowadays, variety of Internet applications has emerged which generates complicated and diverse data traffic. For the efficient management of Internet traffic, many traffic classification methods have been proposed. But most of the methods focused on the application-level classification, not the function-level classification or state changes of applications. The functional classification of application traffic makes possible the in-detail understanding of application behavior as well as the fine-grained control of applications traffic. In this paper we proposed automata based functional classification method of IM application traffic. We verified the feasibility of the proposed method with function-level control experiment of IM application traffic.

### I. 서 론

최근 다양한 인터넷 기반 응용프로그램의 등장과 인터넷을 통한 대용량 데이터전송, 사용자수의 증가, 다양한 접속 매체의 등장으로 네트워크 트래픽이

속적으로 증가하고 있다. 증가하는 트래픽을 효율적으로 관리하기 위해 발생하는 트래픽에 대한 다양한 정보가 필요하고 특정한 기준에 따라 트래픽을 분류하는 작업을 해야 한다. 최근 진행되고 있는 트래픽 분류분야의 연구들은 분류의 기준을 응용 단위로 세우

※ 이 논문은 2009년 정부(교육과학기술부)의 재원으로 한국연구재단(2009-0090455)의 지원을 받아 수행된 연구임.

\* 고려대학교 컴퓨터정보학과 {sangwoo\_lee, junsang\_park, sung-ho\_yoon, tmskim}@korea.ac.kr

논문번호 : KICS2011-02-108, 접수일자 : 2011년 2월 7일, 최종논문접수일자 : 2011년 7월 20일

고 그에 따라 다양한 방법론들을 제시하고 있다<sup>5, 6,7,8,9,10,11</sup>. 하지만 기존의 연구들에서 제안하는 분석의 단위인 응용 별 트래픽 분류는 응용의 일부 기능만을 허용하거나 차단하는 세부적인 제어의 기회를 제공하지 못하고, 네트워크 사용자가 각 응용의 어떠한 기능을 사용하는지를 파악하지 못하게 되는 문제점을 가지고 있다. 이러한 문제점들을 해결하기 위해서는 응용단위의 분류와 함께 응용이 제공하는 기능별 트래픽 분류가 필요하게 된다. 응용의 기능별 트래픽 분류라 함은 응용이 제공하는 기능(서비스)별로 응용의 트래픽을 세부적으로 분류함을 말한다. 예를 들어 메시저의 트래픽을 분류 할 때 단순히 해당 메시저의 트래픽으로 분류 하는 것뿐만 아니라 텍스트 채팅, 파일전송, 화상채팅 트래픽 등으로 세부적인 기능별 요소들로 나누는 것을 말한다.

본 논문에서는 기업 네트워크 보안등에서 이슈가 되고 있는 메시저에 대해 초점을 두었다. NateOn<sup>[1]</sup>, MSN<sup>[2]</sup>, GoogleTalk<sup>[3]</sup>의 기능별 분류를 위한 오토마타를 설계하고 이를 통해 메시저 응용의 기능별 트래픽 분류를 위한 방법론을 제시한다. 응용에 대해 오토마타를 설계하면 다양한 분야에 활용이 가능하다. 상태전이에 대한 변화를 추적하면 사용자의 기능사용 패턴을 파악할 수 있을 뿐만 아니라, 상태에 따라 기능별로 트래픽 분류가 가능하며 문제 발생시 해당 기능으로 분류된 패킷만을 살펴보는 것이 가능하므로 사후 처리에 대한 시간을 줄일 수 있다.

본 논문의 구성은 다음과 같다. 서론에 이어 2장에서는 관련연구에 대해 기술하고, 3장에서는 메시저응용의 오토마타에 대해 기술한다. 4장에서는 선정된 응용 별 오토마타에 대해 기술하고, 5장에서는 실험 환경 및 결과에 대해 기술한다. 마지막으로 6장에서는 결론 및 향후 연구에 대해 기술한다.

## II. 관련연구

인스턴트 메시저(Instant Messenger: IM)는 인터넷을 통한 메시저라는 의미에서 인터넷 메시저라고도 한다. 대표적인 IM 응용프로그램에는 NateOn, MSN, GoogleTalk, Skype<sup>[4]</sup> 등이 있으며, 이러한 인터넷 메시저들은 각각의 고유한 프로토콜을 사용하여 동작하게 되는데 각 메시저들의 프로토콜에 대해 분석이 이루어진다면 네트워크내의 메시저 사용자와 각 사용자들이 사용하는 기능에 대한 상세한 분석 및 기능별 트래픽 분류, 제어가 가능하게 된다.

트래픽 분류에 있어서 논문<sup>[5-8]</sup>과 같이 페이로드 기

반 시그니처 트래픽 분류 방법은 응용프로그램의 패킷 페이로드로부터 특정 스트링을 추출하여 이를 이용하는 방법이다. 응용의 트래픽 분류에 있어 패킷의 헤더 정보나 통계 정보를 이용하는 다른 분석 방법들에 비해 상대적으로 높은 분류 정확성과 분석률을 보인다. 하지만 단순히 페이로드 데이터의 매칭을 통해 트래픽을 분류하므로 해당응용의 기능을 사용하는 호스트의 다양한 정보를 추출 할 수 없다는 단점을 가지고 있다.

논문<sup>[9-11]</sup>에서는 기계학습 기법을 이용해 응용의 트래픽 분류하는 방법을 제안하고 있다. 하지만 해당 방법론은 별도의 학습을 위한 데이터 셋이 필요하며 응용이 사용중인 기능에 대한 분석은 제공하고 있지 않다.

논문<sup>[12]</sup>에서는 Sateful Automata메커니즘을 사용하여 응용의 발생 페이로드 시그니처에 따라 트래픽을 분류 하였지만 기능별 제어 관점에서 보았을 때 호스트 별 기능 사용내역 추적이 취약하고 각 응용 고유의 기능과 상태를 의미적으로 잘 나누지 못했다.

따라서 본 논문에서는 NateOn, MSN, GoogleTalk 메시저 응용의 오토마타를 설계하고 이를 통해 기능별 트래픽 분류를 위한 방법론을 제시한다.

## III. 메시저 응용의 오토마타

NateOn, MSN, GoogleTalk 메시저들은 사용자의 환경에 따라 각각 다양한 기능을 제공하지만 메시저의 기본기능인 채팅, 파일전송, 로그인의 기능을 공통적으로 제공한다. 세 메시저에서 제공하는 공통 기능을 기본으로 응용의 기능별 트래픽 분류를 위한 각 메시저의 오토마타를 제안한다.

### 3.1 응용 별 제공 기능 및 오토마타 생성범위

표 1은 각 메시저응용이 제공하는 기능 및 오토마타 생성 범위에 대해 나타낸다. 각 메시저 마다 제공하는 기능을 포함하여 오토마타가 생성되었으며 각 항목은 Login(로그인), Text Chat(채팅 창을 통한 상

표 1. 응용의 제공기능 및 오토마타 생성 범위

기능제공 여부 /분석범위	NateOn (4.0.14.2)	MSN (14.0.8089)	GoogleTalk (kor.1.0.0.105)
Login	O/O	O/O	O/O
Text Chat	O/O	O/O	O/O
File Transfer	O/O	O/O	X/X
Message	O/O	X/X	X/X
Audio, Video	O/O	O/O	X/X

호간 채팅), File Transfer(파일 전송), Message (NateOn 에서만 제공하는 쪽지 기능), Video, Voice (화상, 음성 채팅)을 나타낸다. NateOn의 경우 4.0.14.2버전에 대해 분석이 이루어 졌으며 MSN의 경우 14.0.8089, GoogleTalk의 경우는 한국어 버전 1.0.0.105 버전에 대한 분석이 이루어졌다.

### 3.2 오토마타의 표현 및 정의

메신저의 기본적인 오토마타를 표현하면 그림 1과 같다. 다음 상태로 전이되기 위한 조건은 화살표를 통해 나타내었으며, 타원은 상태, 점선형태의 사각형은 스테드 개념의 병렬적으로 처리되는 메신저의 기능을 나타냈다. 각 상태에 대한 설명은 표 2를 통해 나타내었다.

메신저 서비스 업체는 사용자 구별을 위해 각 사용자 마다 ID, Password를 요구하고 이를 인증하는 서버를 갖추고 있다. 또한 인증이 통과되면 사용자가 메신저를 사용하고 있다는 것을 인지하기 위해 하나의 세션을 유지하게 되는데 이를 유지하기 위한 서버를 갖추고 있다. 그림 1에서 Session IP Received는 사용자의 ID, Password가 올바르게 해당 호스트가 세션

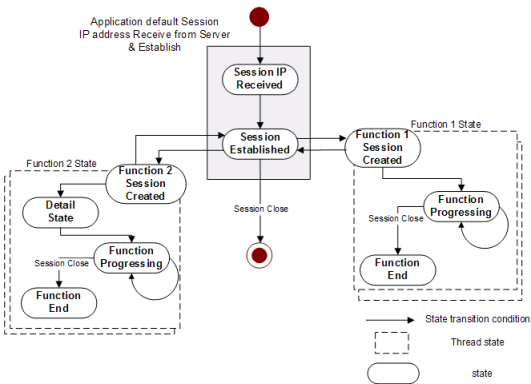


그림 1. 메신저의 기본 오토마타

표 2. 표현된 상태에 대한 설명

State	Description
Start	초기상태
Session IP Received	로그인 서버로부터 접속 유지할 세션 서버의IP를 받아온 상태
Session Established	받아온 세션 서버의IP로 접속을 맺고 유지하는 상태
Function Session Created	새로운 기능의 세션이 생성된 상태
Function Progressing	해당 기능이 진행중인 상태

을 유지할 서버의 IP를 호스트에게 전달 한 것을 의미한다. Session Established는 호스트가 세션유지를 위한 서버로 접속을 하여 세션을 유지하는 상태를 나타낸다.

메신저의 특성상 각 기능은 여러 개의 기능을 동시에 수행 할 수 있다. 예를 들면 동시에 두 개의 채팅 기능을 수행할 수도 있고 화상채팅을 진행하면서 다른 호스트에게 채팅과 파일전송 기능을 동시에 사용할 수 있다. 메신저의 기능을 그림 1에서 Function1, Function2,... 로 표현하였으며 Function1, Function 2,... 의 세션이 생성 되게 되면 각 기능별 스테드 상태가 동시에 수행되게 된다. 즉, 한 사용자가 채팅 창을 두 개를 생성해 동시에 다른 두 호스트와 채팅을 진행 중 이면 두 개의 채팅 상태 오토마타가 서로 다르게 진행 되게 된다.

본 논문에서 사용된 오토마타의 정의는 다음과 같이 3가지 요소로 표현된다. (S, Σ, δ)

#### ■ S: 해당 응용의 상태 집합

NateOn의 경우 그림 2와 같이 23개의 state가 존재하며 초기상태는 start이며 종료상태는 end 로 정의한다. MSN의 경우 그림 3과 같이 14개의 state가 존재하며 마찬가지로 초기상태는 start이며 종료 상태는 end로 정의한다. GoogleTalk의 경우 5개의 state가 존재하며 초기상태는 start, 종료상태는 end로 정의한다.

#### ■ Σ: input data(패킷의 페이로드 데이터)

입력데이터는 타겟 네트워크의 외부 인터넷과 연결된 최상위 라우터에서 패킷을 캡처하고, 캡처된 모든 패킷의 페이로드로 정의한다.

#### ■ δ: transition function(Condition)

전이함수 Condition은 2가지 요소로 이루어져있다. 패킷의 방향성을 나타내는 [+](내부 네트워크 호스트에서 외부 서버로 전송되는 패킷) 또는[-](외부 서버에서 내부 호스트로 전송되는 패킷), 정규표현 형태로 표현된 페이로드 시그니처의 문자집합으로 정의한다.

NateOn의 오토마타에서 어떠한 상태에서 다른 상태로 전이되기 위한 조건은 Condition(1, 2, 3, ... 14)를 통해 표 3을 통해 나타내었으며 각각의 Condition은 정규표현 형태로 표현된 페이로드 시그니처를 나타내며 복수개의 시그니처가 존재하는 Condition은 순차적으로 매칭 되어야 한다는 것을 나타냈다.

정규표현의 표현 형태는 일반적으로 사용하는 정규표현 형식을 따랐으며 다만, E-mail형태의 String은 @

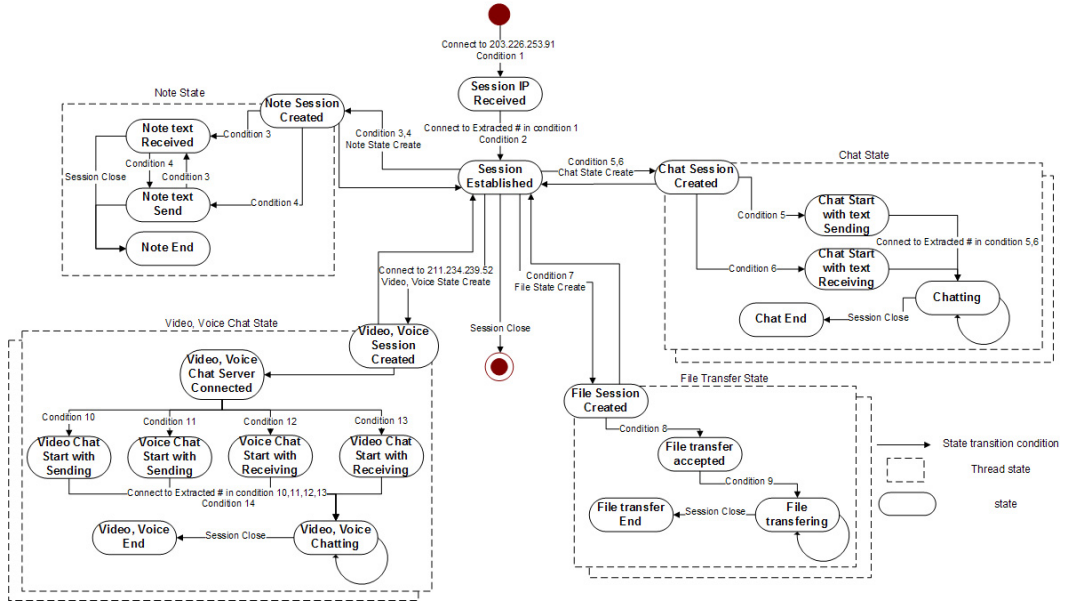


그림 2. NateOn의 오토마타

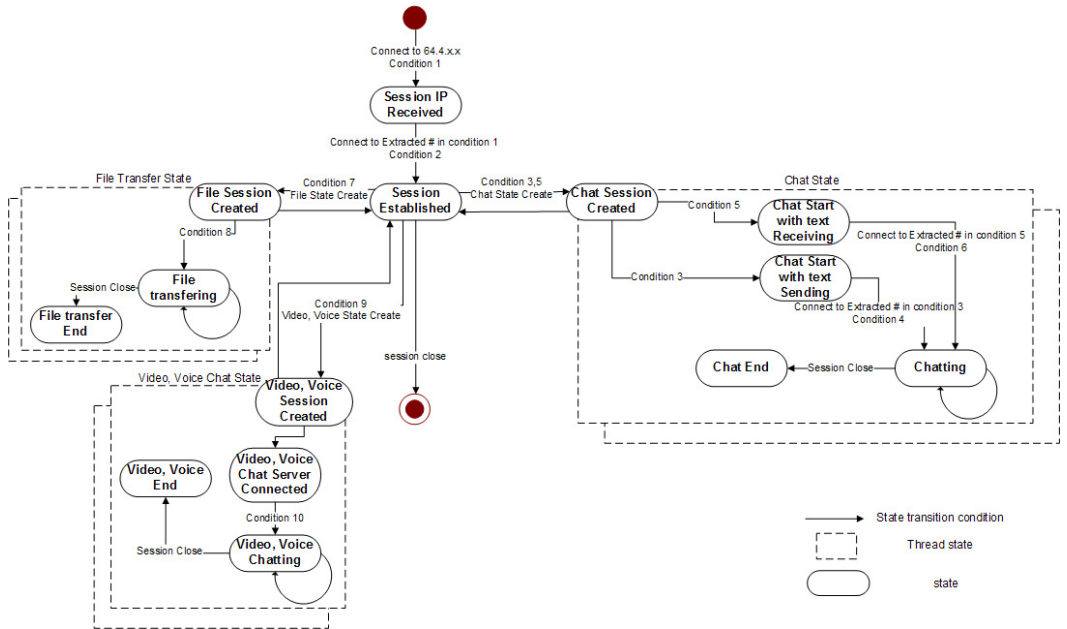


그림 3. MSN의 오토마타

로 나타내었으며, IP형태의 String은 #으로 표현하였다. Action 항목은 시스템에서 추출해야 할 문자열이나 스레드로 동작해야 할 기능의 오토마타를 생성하는 항목을 나타낸다. Connect to # 은 해당 IP로 SYN, SYNACK, ACK 패킷이 발생하여 세션이 생성

되었다는 것을 나타내었다. 또한 Session Close는 해당 세션이 종료되었음을 의미 한다. 본 논문에서는 FIN이나 RST패킷이 하나 이상 발생하면 해당세션이 종료됨 으로 판단하였다.

표 3. NateOn 오토마타를 위한 상태전이 조건 및 시스템이 취해야 할 행위

	State	$\delta(\text{condition})$	Transition	Action
1	Start	[+]^PVER [+]^AUTH [+]^REQS [0-9] DES @ [-]^REQS [0-9] dpc.* #	Session IP Received	Extract #
2	Session IP Received	[+]^NCPT 1 @	Session Established	Extract @
3	Session Established	[-]^CMMSG 0 .* from:@ ref:@	Note text Received	Extract @
4	Session Established	[+]^CMMSG [1-999] .* from:@ ref:@	Note text Send	Extract @
5	Session Established	[+]^CTOC [1-999] @ .* INVT @ #	Chat Session Created	Chat State Create
	Chat Session Created		Chat Start with text Sending	Extract # Extract @
6	Session Established	[-]^CTOC 0 @ .* INVT @ #	Chat Session Created	Chat State Create
	Chat Session Created		Chat Start with text Receiving	Extract # Extract @
7	Session Established	^ATHC [0-9] @	File Session Created	File State Create Extract @
8	File Session Created	^FILE 0 ACCEPT	File transfer accepted	
9	File transfer accepted	^FILE 0 INFO FILENAME ^FILE 1 START	File transferring	
10	Video, Voice Chat Server Connected	[+]INVT .* AVCH REQ 3.* # [-]INVT .* AVCH ACK 0	Video Chat Start with Sending	Extract #
11	Video, Voice Chat Server Connected	[+]INVT .* AVCH REQ 1.* # [-]INVT .* AVCH ACK 0	Voice Chat Start with Sending	Extract #

	State	$\delta(\text{condition})$	Transition	Action
12	Video, Voice Chat Server Connected	[-]INVT .* AVCH REQ 1.* # [+]INVT .* AVCH ACK 0	Voice Chat Start with Receiving	Extract #
13	Video, Voice Chat Server Connected	[-]INVT .* AVCH REQ 3.* # [+]INVT .* AVCH ACK 0	Video Chat Start with Receiving	Extract #
14	Video Chat Start with Sending, Voice Chat Start with Receiving, Video Chat Start with Receiving	UDP Packet 0x1a 0xbc 0x19 0x00	Video, Voice Chatting	

#### IV. 응용 별 오토마타

NateOn 응용 프로그램이 발생시키는 패킷의 페이로드 데이터를 살펴보면 그림 2와 같이 하나의 호스트에 적용 가능한 NateOn의 오토마타를 생성할 수 있다.

Session IP Received는 NateOn의 로그인 인증과정에서 나타나는 상태이며 표현된 Condition1 시그니처(표 3, Condition1)가 순차적이 매칭 된다면 시작 상태에서 Session IP Received 상태로 전이되게 된다. 또한 표 3 에서 Action에 나타나듯이 시스템에서는 패킷의 페이로드에서 IP형태로 된 스트링을 추출하게 된다. 추출된IP는 해당 사용자가 계속적으로 접속을 유지하게 될 서버의 IP를 나타내므로 해당 사용자가 접속을 유지하게 될 서버를 미리 예측 가능하다.

로그인 인증이 통과하게 되면 받아온 서버의 IP로 접속을 시도 하게 되는데 Condition2 시그니처와 매칭이 이루어지면 시스템에서는 사용자의 E-mail 형태로 된 ID를 추출하고 Session Established상태로 전이되게 된다. Session Established 상태로 전이된 후 사용자가 아무런 행동도 취하지 않는다면 호스트는 계속 Session Established 상태에 머무르게 된다.

사용자가 채팅 기능을 사용한다면 Condition 5, 6 과 매칭이 이루어지게 되고 Chat Session Created상태로 전이된다. 시스템은 채팅 기능의 오토마타를 따로 생성하여 해당 호스트에 적용시키고 다시 Session

Established상태로 전이 하게 된다. 다른 기능과 마찬가지로 채팅 기능도 해당 호스트가 다른 여러 호스트들과 동시 다발 적으로 사용 가능 하므로 하나의 채팅 세션이 생성되면 또 다른 하나의 채팅 오토마타가 생성된다.

이러한 방식으로 하나의 호스트에 대해 각각의 기능의 오토마타가 동시 다발 적으로 존재 가능하며 각 오토마타는 서로 다른 상태에 머무를 수 있다.

마찬가지로 MSN 응용 프로그램이 발생시키는 패킷의 페이로드 데이터를 살펴보면 그림3과 같이 호스트에 적용 가능한 MSN의 오토마타를 생성 할 수 있다. MSN은 로그인, 채팅, 파일전송, 화상, 음성 채팅 기능을 제공하며 NateOn과 마찬가지로 Session IP Received 상태를 거쳐 Session Established 상태에 도달한다. Session IP Received의 전이조건인 표4의 Condition1에서 호스트가 다음단계로 접속 할 서버의 IP를 추출한다.

호스트가 Session Established 상태에서 전이 조건을 만족하는 기능을 사용하였을 때 해당하는 기능의 오토마타를 생성하고 다시 Session Established상태로 전이된다.

GoogleTalk의 오토마타도 Session IP Received, Session Established상태를 가지고 있으며 기능별로 상태가 존재하며 NateOn, MSN과 같은 형태를 취하고 있기 때문에 본 논문에서는 나타내지 않았다.

### V. 실험 환경 및 결과

본 절에서는 기능별 트래픽 탐지를 위해 구성한 시스템 환경 및 실험 결과에 대해 기술한다.

실험 환경 구성은 그림 4와 같이 학내 네트워크와 외부인터넷이 연결되어 있는 최상위 라우터 에서 패킷을 미러링 받았으며 패킷을 분석기에 전달하는 형태로 구성하였다. 분석기는 생성된 오토마타를 기반으로 기능별 탐지와 트래픽 분류를 진행하고 결과를 도출한다. 또한 분류된 트래픽을 기반으로 기능별 제어 실험을 진행하기 위해 제어 대상이 되는 호스트를 미리 선정하여 대상 호스트의 기능의 트래픽이 탐지되면 이를 차단하는 실험을 진행 하였다.

표 5는 실험에 사용된 트래픽 트레이스와 실험 기간을 나타낸다. 트레이스는 3000여대의 호스트에서 사용하는 다양한 종류의 응용 프로그램의 트래픽으로 구성된다.

다음 표 6은 기능별 트래픽 분류 결과에 대해 나타낸다. 분석의 대상이 되는 응용인 NateOn, MSN,

표 4. MSN 오토마타를 위한 상태전이 조건 및 시스템이 취해야 할 행위

	State	δ(condition)	Transition	Action
1	Start	[+] <sup>*</sup> VER 1 [+] <sup>*</sup> CVR 2 . <sup>*</sup> USR 3 SSOI @ [-] <sup>*</sup> XFR 3 NS #	Session IP Received	Extract @ Extract #
2	Session IP Received	[-] <sup>*</sup> USR 4 OK	Session Established	
3	Session Established	[-] <sup>*</sup> XFR [0-99] SB #. <sup>*</sup> CKI	Chat Session Created	Chat State Create
	Chat Session Created		Chat Start with text Sending	Extract #
4	Chat Start with text Sending	[-] <sup>*</sup> USR [0-99] OK [+] <sup>*</sup> CAL [0-99] [-] <sup>*</sup> CAL [0-99] RINGING [-] <sup>*</sup> JOI @ [-] <sup>*</sup> JOI @	Chatting	Extract @
5	Session Established	[-] <sup>*</sup> RNG [0-999999999] #. <sup>*</sup> CKI	Chat Session Created	Chat State Create
	Chat Session Created		Chat Start with text Receiving	Extract #
6	Chat Start with text Receiving	[+] <sup>*</sup> ANS [0-9] [-] <sup>*</sup> IRO [0-9] [0-9] [0-9] @ [-] <sup>*</sup> JOI @	Chatting	Extract @
7	Session Established	INVITE MSNMMSG.*To: <msnmmsg: @. ; *From : <msnmmsg: @;	File Session Created	Extract @ File State Create
8	File Session Created	MSNSLP.* 200 OK	File transferring	
9	Session Established	[-]XFR 3 NS #. <sup>*</sup> U D	Video, Voice Session Created	Extract # Video, Voice State Create
10	Video, Voice Chat Server Connected	<sip e="base64" fid="1"	Video, Voice Chatting	

GoogleTalk의 기능별 바이트 발생량, 패킷 수, 호스트 수를 나타낸다. 실험결과 NateOn을 사용하는 호스트가 가장 많았으며 채팅창을 통한 채팅보다 쪽지 기능을 더 많이 사용한다는 것을 알 수 있으며, MSN 사용자

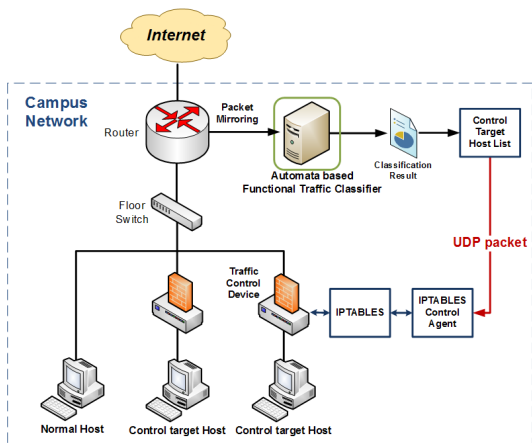


그림 4. 구성환경

표 5. 트래픽 트레이스

Time	# of Packets	Bytes
2011-01-25 12:39 ~ 2011-01-25 20:56	618,016 (K)	487 (GB)

표 6. 기능별 트래픽 분류 결과

		NateOn	MSN	GoogleTalk	Total
Text Chat	Bytes	5MB	598KB	0	6MB
	Packet	46K	5K	0	50K
File transfer	Bytes	53MB	0	-	53MB
	Packet	57K	0	-	57K
Message	Bytes	6MB	-	-	6MB
	Packet	71K	-	-	71K
Audio, Video Chat	Bytes	0	0	-	0
	Packet	0	0	-	0
Host Count		127	12	0	139

들은 주로 채팅 기능만을 사용한다는 것을 알 수 있다.

또한 특정호스트의 NateOn파일전송 기능을 탐지하고 이를 차단하는 실험을 통해 해당 트래픽을 잘 탐지하는 것을 확인 하였다.

## VI. 결론 및 향후 연구

본 논문에서는 NateOn, MSN, GoogleTalk 메신저 응용에 대해 오토마타를 설계하고 이를 통해 메신저 응용의 기능별 트래픽 분류방법론을 제안하였다. 제안된 방법을 사용하여 상태전이에 대한 변화를 추적하

면 사용자의 기능사용 패턴을 파악할 수 있을 뿐만 아니라, 상태에 따라 기능별로 트래픽 분류가 가능하다. 향후 연구로서 응용의 오토마타를 자동 생성하는 시스템과 다양한 응용에 적용하는 방법을 계획 중이다.

## 참고 문헌

- [1] NateOn, <http://nateonweb.nate.com/>.
- [2] MSN, <http://explore.live.com/home>.
- [3] GoogleTalk, [www.google.com/talk/](http://www.google.com/talk/).
- [4] Skype, <http://www.skype.com>.
- [5] Subhabrata Sen, Oliver Spatscheck, Dongmei Wang, "Accurate, scalable in-network identification of p2p traffic using application signatures" *World Wide Web 2004*, May 17-20, 2004, New York, USA.
- [6] L. Deri, S. Suin, "Effective traffic measurement using ntop", *IEEE Communications Magazine* 38, pp.138-143, 2000.
- [7] F. Risso, M. Baldi, O. Morandi, A. Baldini, and P. Monclus, "Lightweight, Payload-Based Traffic Classification An Experimental Evaluation," *IEEE International Conference on Communications*, Beijing, China, May. 19-23, pp.5869-5875, 2008.
- [8] H.Liu, W.Feng, Y.Huang, X.Li, "A peer-to-peer traffic identification method using machine learning", in *International Conference on Networking, Architecture, and Storage*, NAS, July 29-31, pp.155-160, 2007.
- [9] Zhu Li, Ruixi Yuan, and Xiaohong Guan, "Accurate Classification of the Internet Traffic Based on the SVM Method" *Proc. of the IEEE International Conference on Communications*, Glasgow, Scotland, Jun. 24-28, pp.1373-1378, 2006.
- [10] Laurent Bernaille, Renata Teixeira, and Ismael Akodkenou, "Traffic classification on the fly" *Proc. of the ACM SIGCOMM Computer Communication Review*, Vol.36 No.2 April, pp. 23-26, 2006.
- [11] Huang. Nen-Fu, Feng. Yi-Hsuan, "Application Behavior Analysis Stateful Automata Mechanism", *Journal of Computers*, Vol.18, No.4, January, pp.3-14, 2008.



이 상 우 (Sang-woo Lee)

준회원



2010년 2월 고려대학교 컴퓨터  
정보학과 학사 졸업  
2010년 3월~현재 고려대학교  
컴퓨터 정보학과 석사과정  
<관심분야> 네트워크 관리 및  
보안, 트래픽 모니터링 및  
분석

윤 성 호 (Sung-ho Yoon)

준회원



2009년 2월 고려대학교 컴퓨터  
정보학과 학사 졸업  
2011년 2월 고려대학교 컴퓨터  
정보학과 석사 졸업  
2011년 3월~현재 고려대학교  
컴퓨터 정보학과 박사과정  
<관심분야> 네트워크 관리 및 보  
안, 트래픽 모니터링 및 분석

박 준 상 (Jun-sang Park)

준회원



2008년 2월 고려대학교 컴퓨터  
정보학과 학사 졸업  
2010년 8월 고려대학교 컴퓨터  
정보학과 석사 졸업  
2010년 9월~현재 고려대학교  
컴퓨터 정보학과 박사과정  
<관심분야> 네트워크 관리 및 보  
안, 트래픽 모니터링 및 분석

김 명 섭 (Myung-sup Kim)

종신회원



1998년 포항공과대학교 전자계  
산학과 학사 졸업  
1998년~2000년 포항공과대학  
교 컴퓨터공학과 석사 졸업  
2000년~2004년 포항공과대학  
교 컴퓨터공학과 박사 졸업  
2004년~2006년 Post-Doc.,  
Dept. of ECE, Univ. of Toronto, Canada.  
2006년~현 재 고려대학교 컴퓨터정보학과 부교수  
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터  
링 및 분석, 멀티미디어 네트워크