An Automated Classifier Generation System for Application-Level Mobile Traffic Identification

- Thesis Defense -

Yeongrak Choi

Supervisor: Prof. James Won-Ki Hong

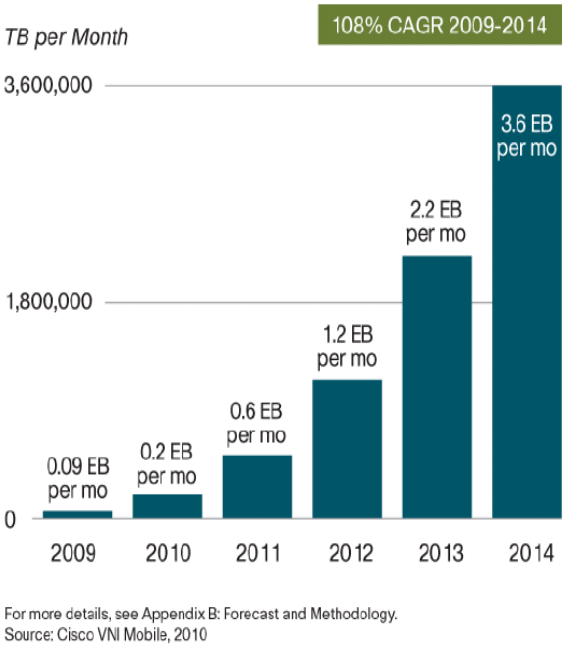
June 20, 2011

Division of IT Convergence Engineering
POSTECH, Korea
dkby@postech.ac.kr

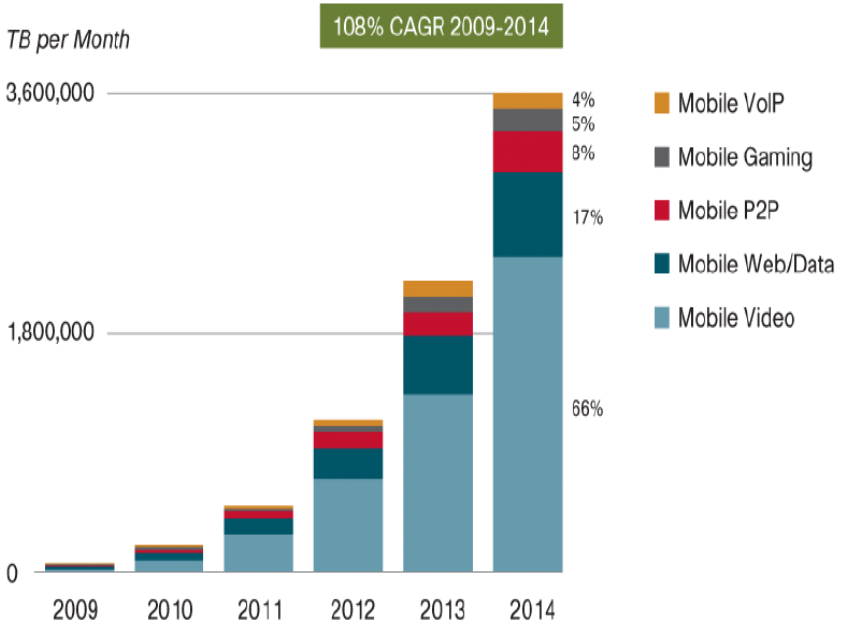
Table of Contents

Introduction – (1)

- **Flood of mobile devices with Internet access**
  - Smart phones: iPhone, Galaxy S, Blackberry phones, …
  - Smart tablets: iPad, Galaxy Tab, …
  - Other mobile devices: iPod Touch, Sony PSP, …
- **Forecast of mobile data traffic**
  - The volume of mobile data traffic is increasing rapidly as more mobile devices are used



For more details, see Appendix B: Forecast and Methodology.
Source: Cisco VNI Mobile, 2010



Introduction – (2)

- **Importance of application traffic classification**
  - Understand network usage
  - Provide high-quality network service
  - Manage selfish-application traffic
- **Difficulties in application traffic classification**
  - Low accuracy of port-based classification
  - High computational complexity for payload inspection
  - Selecting classifier(s) which guarantee high accuracy
  - Exhaustive tasks to extract classifier(s)
- **Characteristics of mobile traffic**
  - Vast majority of HTTP & HTTPS (~80% of total traffic) for port-based observation [Falaki, IMC'10]
  - Most mobile applications run as client in the client-server architecture

Motivations & Problems

- **Application traffic classification in mobile networks**
  - Understanding mobile application traffic to provide high quality mobile network service
  - QoS guarantee for different application class
- **Existing signature-based classification approach and automated signature generation**
  - Reliable classification result
  - Require high computation and memory cost
- **Difficulty in mobile traffic classification**
  - Unclear ground-truth traffic to find classifiers of application traffic
  - Weak processing power and limited memory of mobile devices to collect accurate ground-truth data

Research Goals

- **Define classifiers for mobile traffic identification**
  - Destination IP address and port
  - HTTP-host and user-agent
  - Common payload strings
- **Propose an architecture and algorithms to generate mobile traffic classifiers automatically**
  - Collection of ground-truth data for automated generation
  - Generating classifiers for each application
- **Apply mobile traffic classifiers**
  - Validate generated classifiers
  - Compare results with collected ground-truth data

Related Work – (1)

❖ Analysis of mobile traffic

- **Analyzed hand-held mobile device traffic from residential broadband DSL lines [Maier, PAM'10]**
  - **HTTP used 80-97% of hand-held mobile devices**
  - **Exhaustive work of manually finding application signatures**
- **Measured port application usage from 43 users [Falaki, IMC'10]**
  - **Application Identifitication: port matching**
  - **HTTP & HTTPS are almost 80%**
- **Compared the characteristics of handheld and non-handheld device traffic [Gember, PAM'11]**
  - **Video traffic is over twice (40%) compared to non-handhelds (17%)**
  - **Their identification method depends on the collection of information from all Wi-Fi APs**

Related Work – (2)

❖ Classifiers for existing traffic classification methods

Category	Approach	Classifiers	Cost of Operation	Accuracy
Session-based	Well-known Port Matching [IANA]	Port	Low	Low
	Session Behavior Modeling [Karagiannis, SIGCOMM'05]	{srcIP, dstIP, srcPort, dstPort}	Medium	Medium
Content-based	Signature Matching [Sen, WWW'04]	Payload strings	High	High
	Automated Generation of Signatures [Park, NOMS'08]	Payload strings	High	High
Statistics-based	Group of features [Lim. CoNext'10]	Ports, number of packets, transferred bytes, duration	High	Medium
Proposed method	**Adaptively choose lower-cost Classifiers for automated generation**	**dstPort, dstIP, HTTP-host, HTTP-useragent (or common strings)**	**Medium**	**High**

Definition of Mobile Traffic Classifier

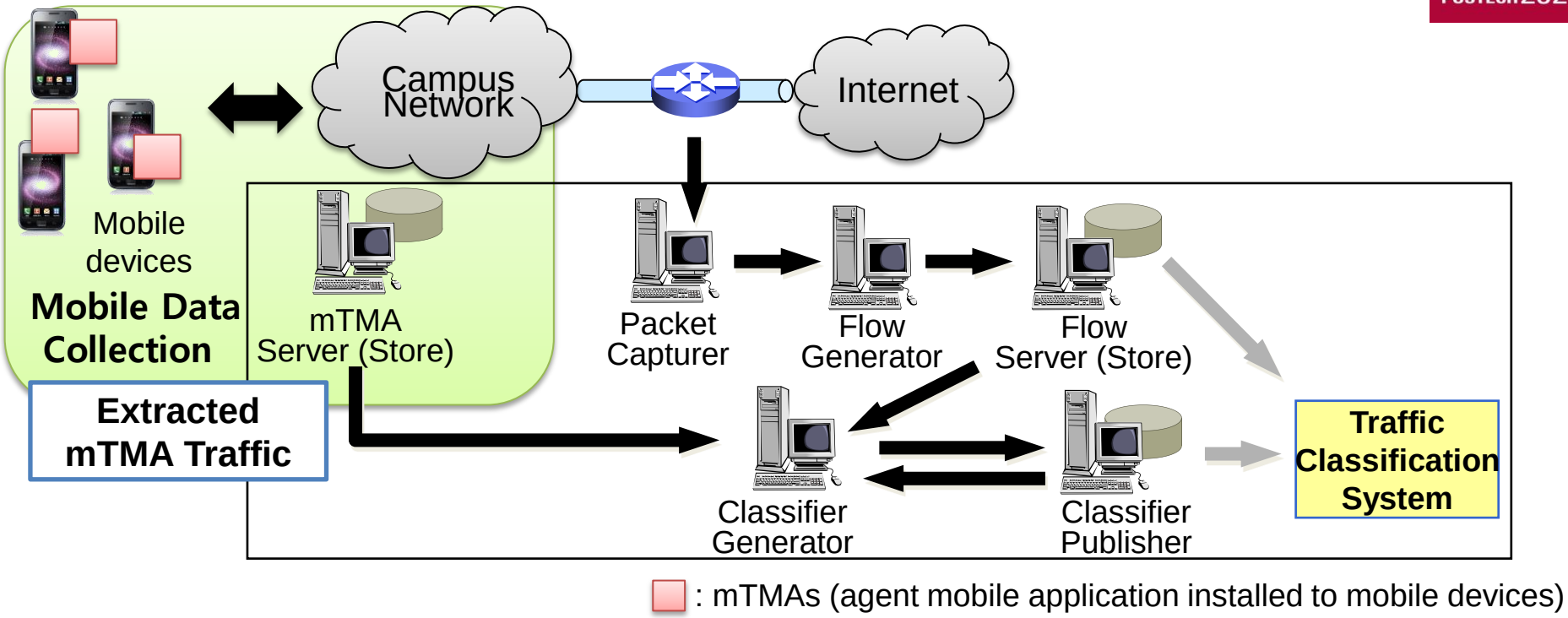
❖ Classifier vs. Signature

- **Classifier**
  - A rule for classifying traffic
  - Example
    - KakaoTalk for tcp any any -> 110.76.140.0/24 80 content: “kakao.com”
- **Signature**
  - Extracted strings (or data) from payload

❖ Definition of mobile traffic classifier

- **Composition of rules for identifying mobile application traffic**
  - Destination IP and port number
    - Client-server architecture
  - ‘Host’ & ‘User-agent’ for HTTP protocol
    - Most HTTP traffic with formatted payload
  - Common payload strings for non-HTTP protocol

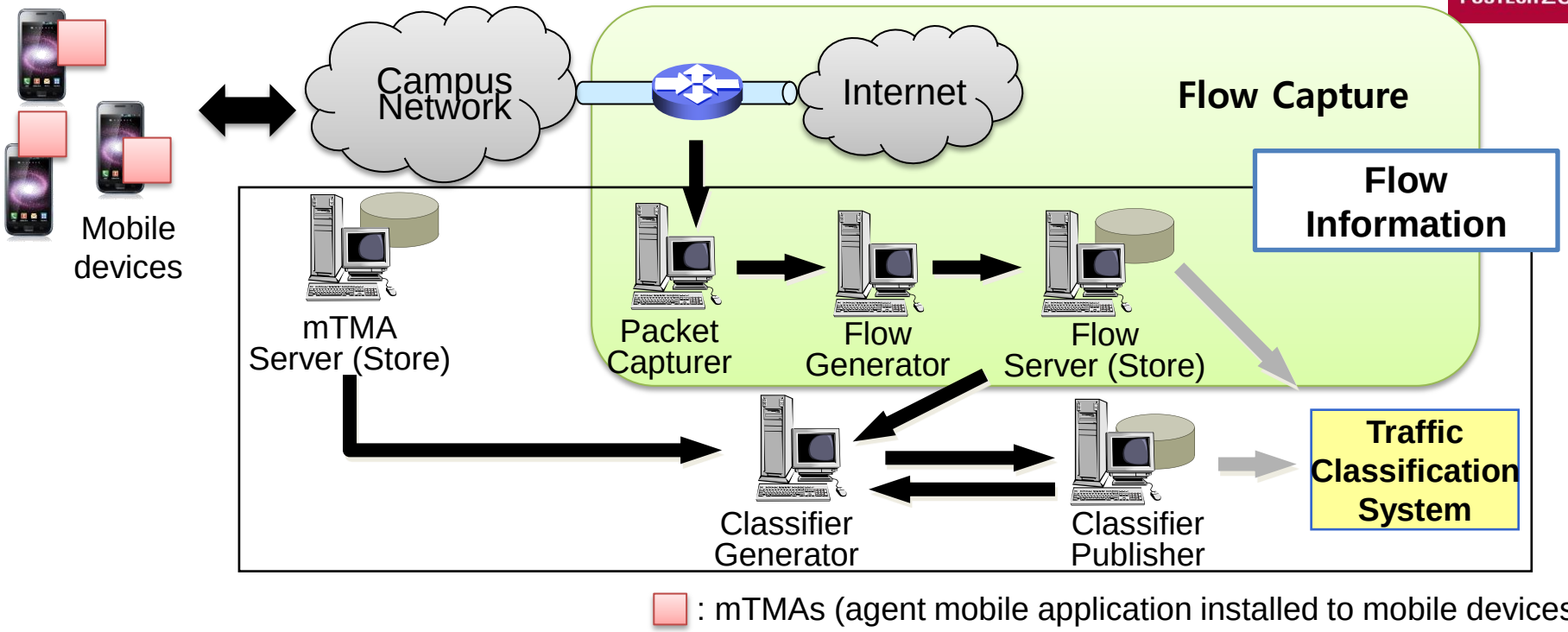
Classifier Generation Architecture – (1)



❖ Mobile Data Collection

- **mTMAs (mobile Traffic Measurement Agents)**
  - **Capturing and matching packets to their application name**
  - **Sending results to mTMA Server Store**
- **mTMA Server (Store)**
  - **Storing all the data from mTMA clients as <flow, part of payload bytes, appname>**

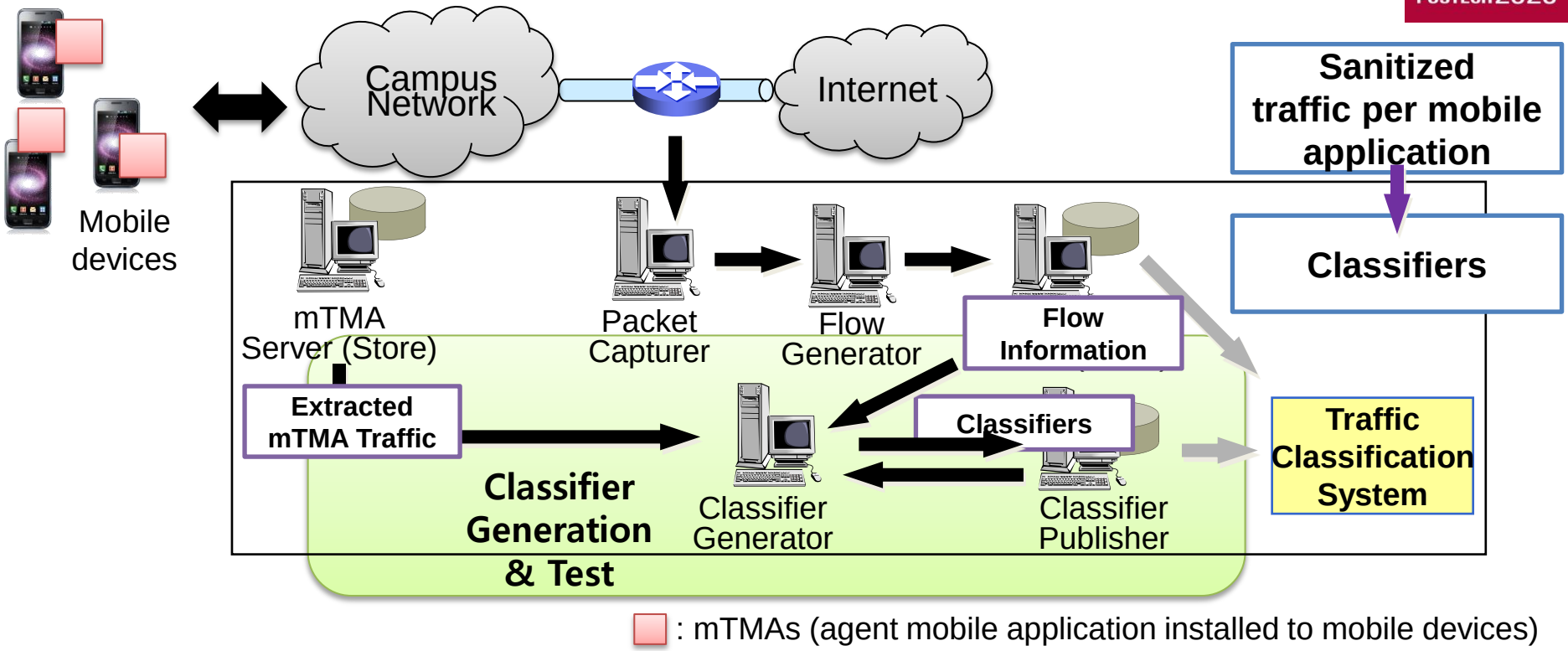
Classifier Generation Architecture – (2)



❖ Flow Capture

- **Packet Capturer**
  - Capturing packets in backbone
- **Flow Generator**
  - Generating flows from captured packets
- **Flow Server (Store)**
  - Storing all flows captured from backbone as <flow, payload>

Classifier Generation Architecture – (3)



❖ Classifier Generation & Test

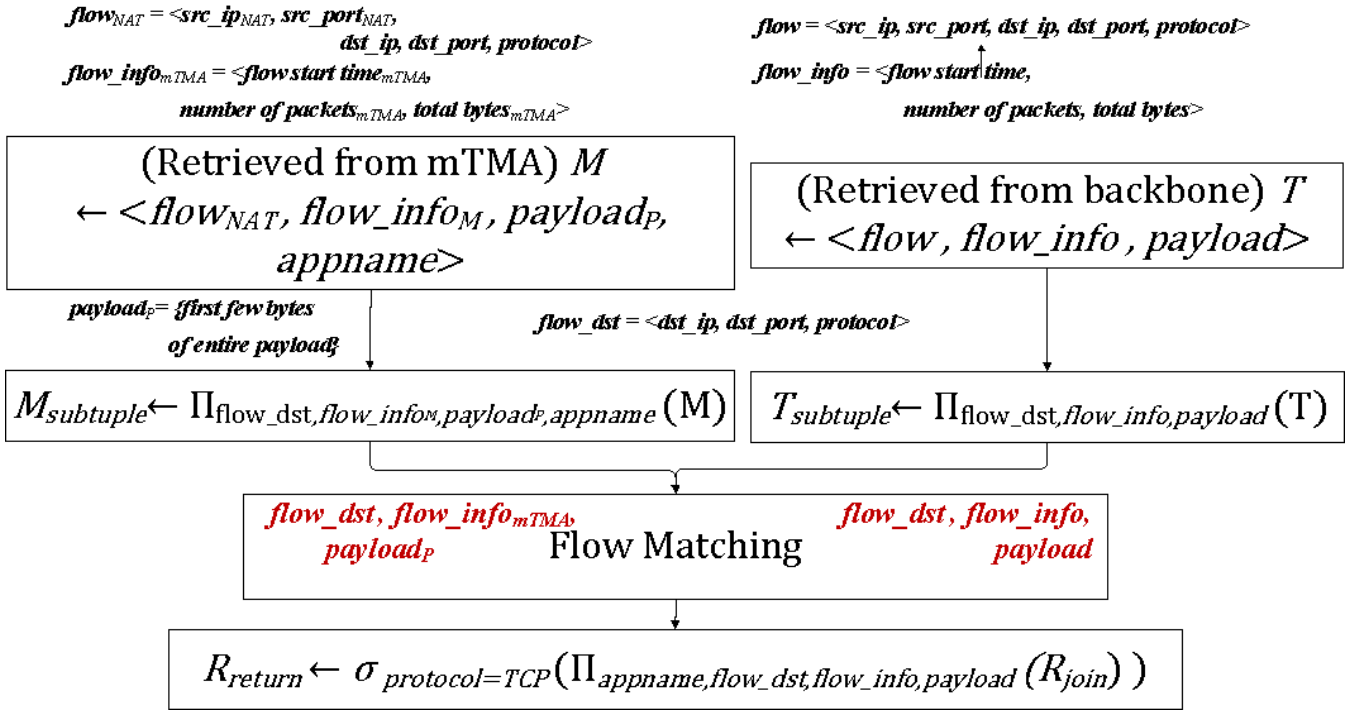
- **Classifier Generator**
  - Matching <flow, payload> in Flow Server with <flow, part of payload bytes, appname> in mTMA Server
  - Automated generation of classifiers for each application using sanitized traffic data per each mobile application
- **Classifier Publisher**
  - Publishing classifiers to a public web page

Ground-truth Data Collection

❖ Difficulty of finding mTMA traffic in backbone trace

- **Flow information from mTMA is not 100% trustable**
  - **Because of limited process power and memory space in mobile devices**
- **Source IP address & port numbers are different (NAT)**

❖ 89% of flows from test mobile devices are matched



Flow Matching Procedure to Collect Ground-truth Traffic Data

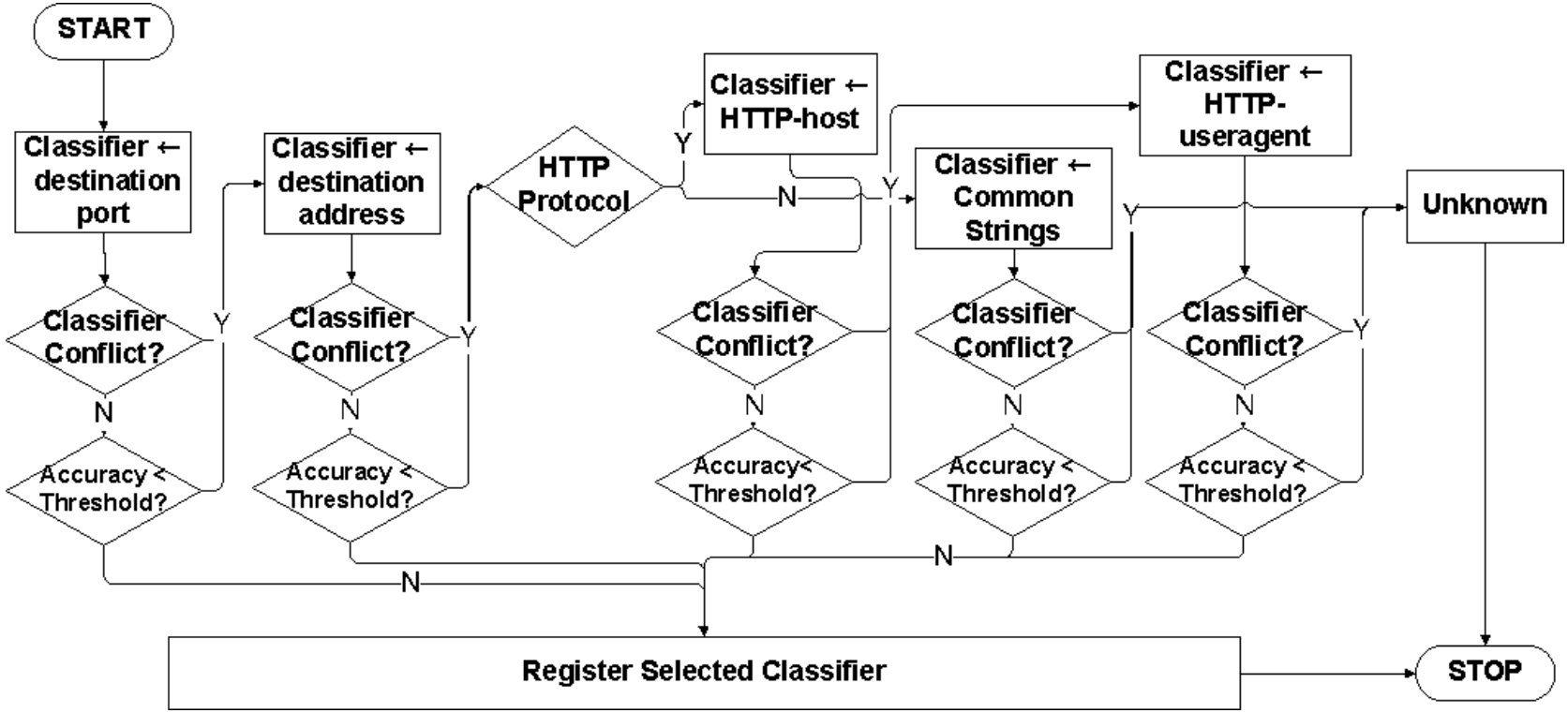
Ratio	89%
# of matched flows	2,396
# of flows from test mobile devices	2,692

NAT: Network Address Translator

Algorithm: Automated Classifier Generation

❖ Automated classifier generation

- Start to observe low-cost classifiers to more-cost classifiers for mobile traffic
- Adaptive decision of classifiers within acceptable accuracy
  - Checking classifier conflict and accuracy with the ground-truth data



Flow Chart of Automated Classifier Decision Algorithm

Examples of Generated Classifiers

❖ Target mobile device

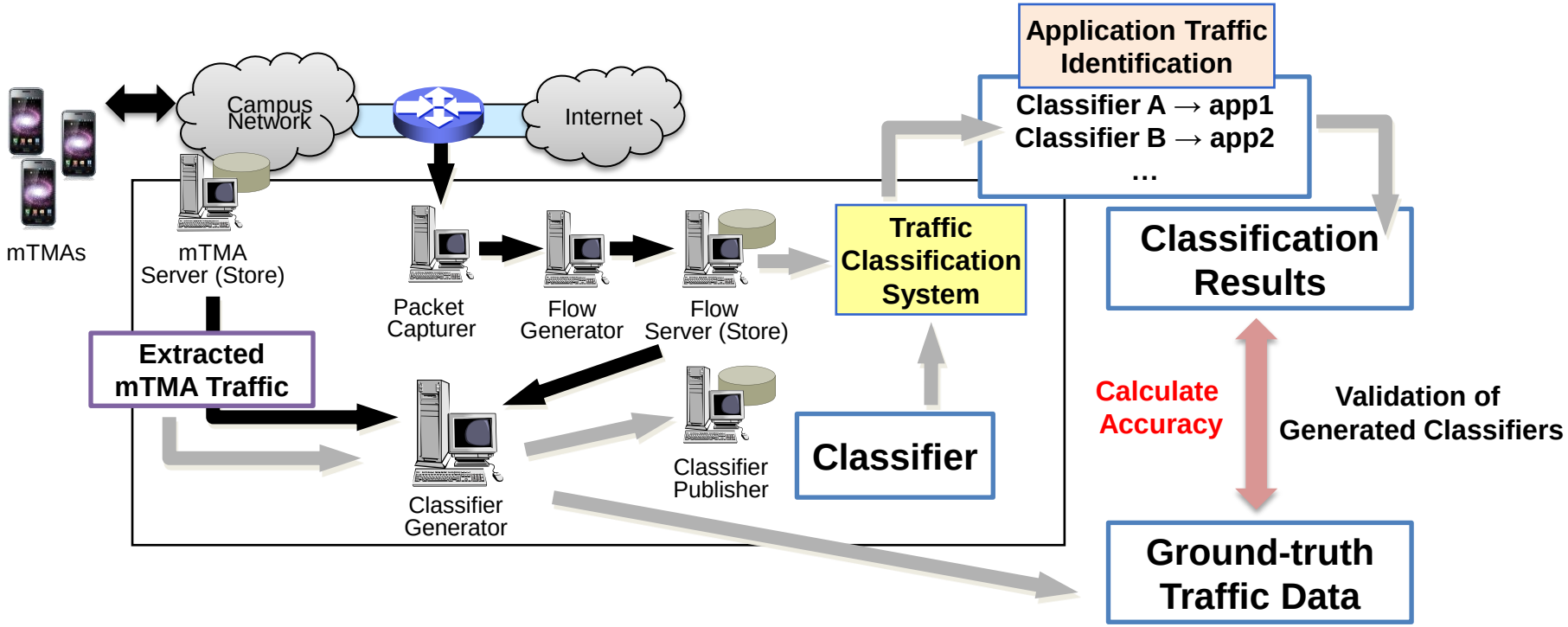
- **Android platform (Gingerbread)**

Application Name	Protocol	Classifier Type	Classifier	Application Process Name
Multimedia player	TCP	HTTP User-Agent	stagefright/1.1 (Linux;Android 2.3.3)	/system/bin/ mediaserver
			stagefright/1.1 (Linux;Android 2.3.4)	
Android Market (File Transfer)	TCP	HTTP User-Agent	AndroidDownloadManager	android.process. Media
Android Web Browser	TCP	HTTP User-Agent	Mozilla/5.0 (Linux; U; Android 2.3.3; ko-kr; SHW-M110S Build/GINGERBREAD) Apple WebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1	com.android.browser
T Store	TCP	Port Number	444, 8205, 9104, 9200, 9401	com.skt.skaf. A000Z00040
KakaoTalk	TCP	Subnet	110.76.140.0/24	com.kakao.talk
			203.246.172.0/24	

Validation of Generated Classifiers

❖ Accuracy of generated classifiers

- Apply classifiers to campus mobile traffic
  - Consider traffic matched to mTMA Server as ground-truth traffic
- Calculate accuracy of generated classifiers
  - Compare classification results with ground-truth traffic



Packet Trace & Target Mobile Applications

❖ Packet trace

- **One hour backbone traffic with total 2,692 flows (mTMA)**

	Date	Start Time	End Time	Total Packets	Total Volume	Ground-truth Packets	Ground-truth Volume
Trace	Jun 12, 2011	21:03:37	22:02:32	66,578,145	54.78GB	706,416	608.57MB

❖ Choice of applications

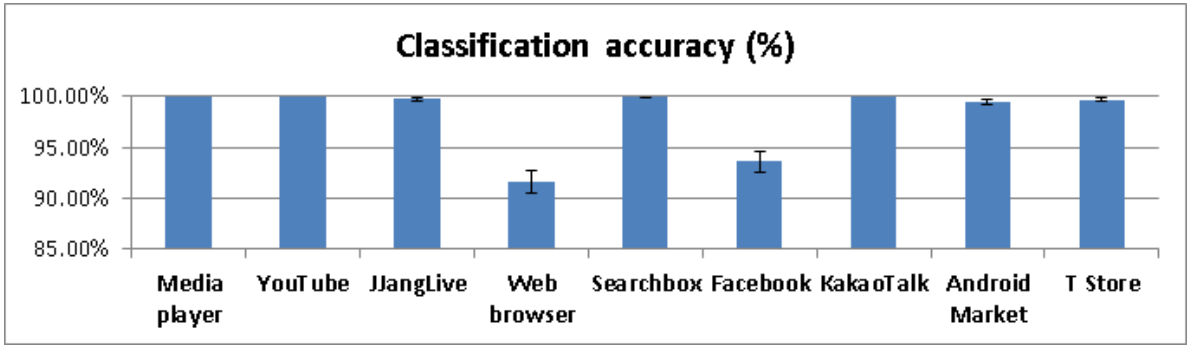
- **Popular Android mobile applications**

Category	Mobile Application	Process name	Displayed Name
Multimedia	Multimedia player	/system/bin/mediaserver	YouTube, Daum TV Pot, JJangLive
	YouTube	com.google.android.youtube	YouTube
	JJangLive	com.uajjang.android	JJangLive
Browser	Web browser	com.android.browser	Internet
	Searchbox	com.google.android.googlequicksearchbox	(attached to other mobile applications)
Messaging	Facebook	com.facebook.katana	Facebook for Android
	KakaoTalk	com.kakao.talk	KakaoTalk
Market	Android Market	android.process.media	Android Market
		com.android.vending	
	T Store	com.skt.skaf.A000Z00040	T Store

Validation of Generated Classifiers – (1)

❖ Accuracy of generated classifiers

- **Estimating the population proportion**
  - **Confidence interval with 95% confidence**



Category	Mobile application name	Accuracy (95% confidence)
Multimedia	Media player	100.00±0.0000%
	YouTube	100.00±0.0000%
	JJangLive	99.71±0.2161%
Browser	Web Browser	91.53±1.1151%
	SearchBox	99.96±0.0818%
Messaging & SNS	Facebook	93.57±0.9820%
	KakaoTalk	100.00±0.0000%
Market	Android Market	99.42±0.3052%
	T Store	99.67±0.2310%

$$Accuracy = \frac{Total\ Traffic - FP\ traffic - FN\ traffic}{Total\ Traffic}$$

Validation of Generated Classifiers – (2)

❖ Analysis of low-accuracy mobile applications

- **JJanglive**
  - Too small sample size
- **Web browser**
  - Other mobile applications use the same classifier
- **Facebook**
  - Encrypted traffic

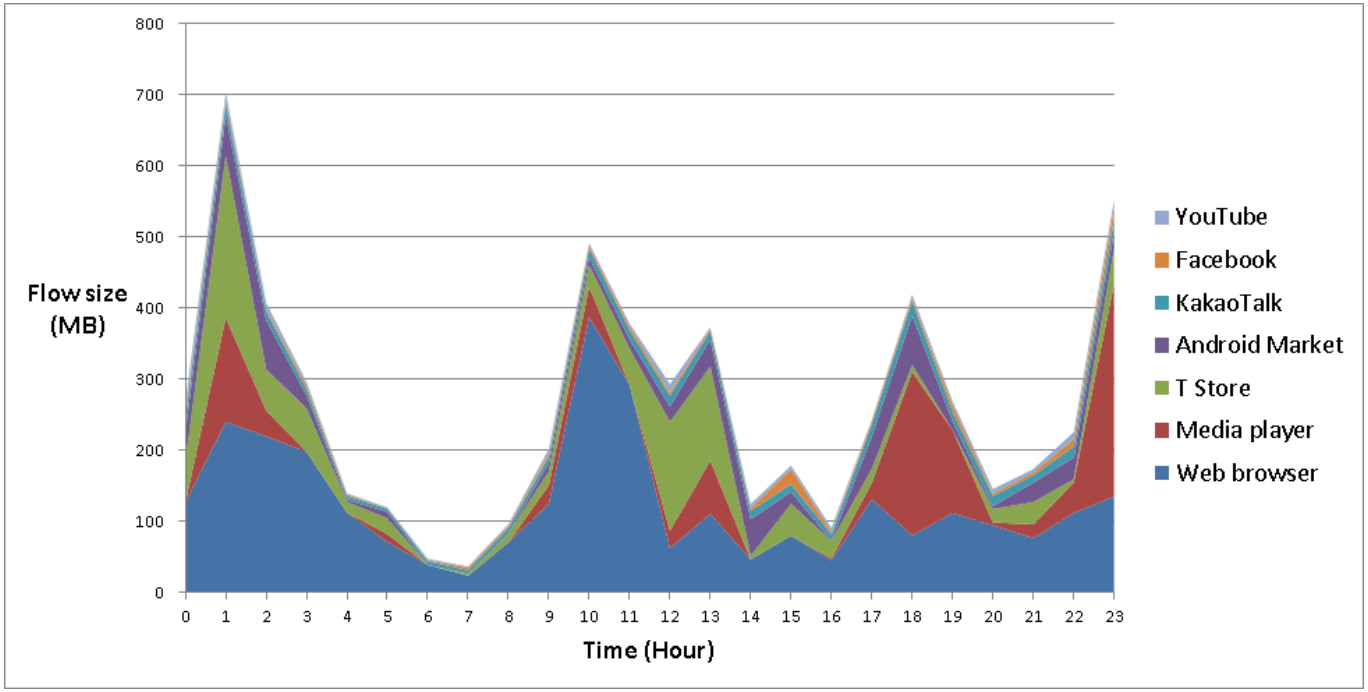
Mobile Application	True Positive (KB)	False Negative (KB)	False Positive (KB)	Precision (%)	Recall (%)
Media player	369,364	0.0	0.0	100.00%	100.00%
YouTube	5,814	0.0	0.0	100.00%	100.00%
JJangLive	52	154.6	6.3	89.18%	25.01%
Web browser	15,822	109.0	2,728.8	85.29%	99.32%
Searchbox	17	0.1	0.0	100.00%	99.57%
Facebook	750	1,130.4	80.1	90.34%	39.88%
KakaoTalk	6,221	0.0	0.0	100.00%	100.00%
Android Market	125,793	88.1	0.0	100.00%	99.93%
T Store	91,932	0.0	490.5	99.47%	100.00%
Total Traffic			623,173 KB		

❖ Packet trace

- One day traffic (April 16th, 2011) at POSTECH dormitory Internet junction with entire packet payloads

❖ Analysis result

- Almost half of traffic volume is from web browsers
- Peak time: 10am~2pm, 6~8pm and 11pm~3am



Proportion of Mobile Application Traffic Volume over 24 Hours

Summary & Contributions

- **Survey on the analysis of mobile traffic and existing application traffic identification classifiers**
- **Proposed an automated classifier generation system and related algorithms**
  - Proposed flow matching algorithm overcomes the difficulty of collecting ground-truth data because of NAT functionality and limited mobile device resources
  - Automatically selecting & checking the accuracy trying low-cost classifiers for each mobile application
- **Validation of proposed system**
  - The validation consists of the ratio of flow matching for collecting ground-truth traffic and the accuracy of generated classifiers

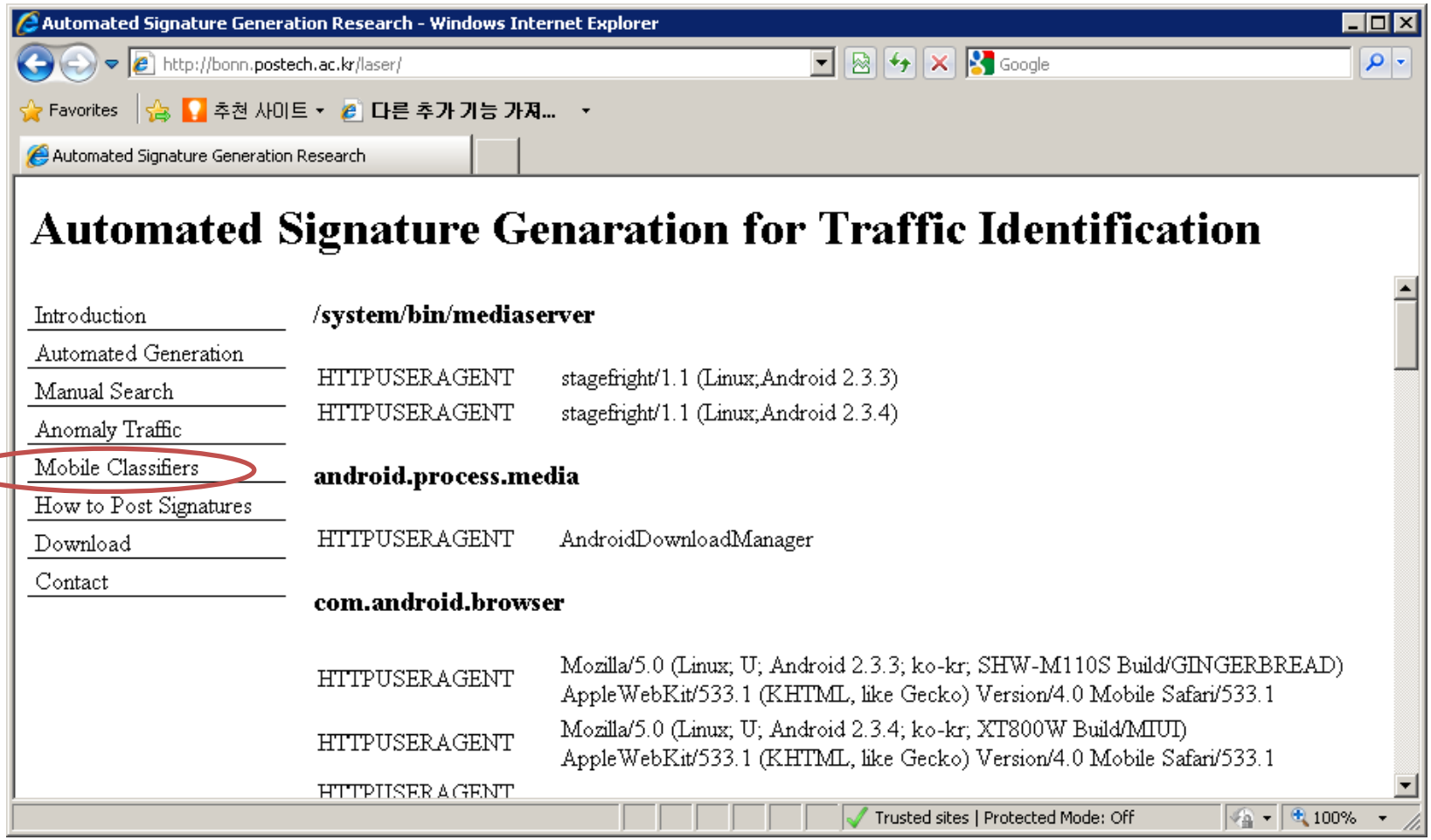
Future Work

- **Increasing the accuracy of generated classifiers**
  - Addressing encrypted mobile application traffic such as Facebook
  - Considering combination of applied classifiers
- **Finding new candidate of classifiers for mobile traffic**
  - Can statistics-based classifiers perform mobile traffic identification with high accuracy?
- **Applying to other mobile platforms**
  - iOS, Blackberry, Windows Phone 7
- **Comparative application-level measurement analysis**
  - Mobile and non-mobile traffic
  - Wi-Fi and 3G network traffic


THANK
YOU
THANK

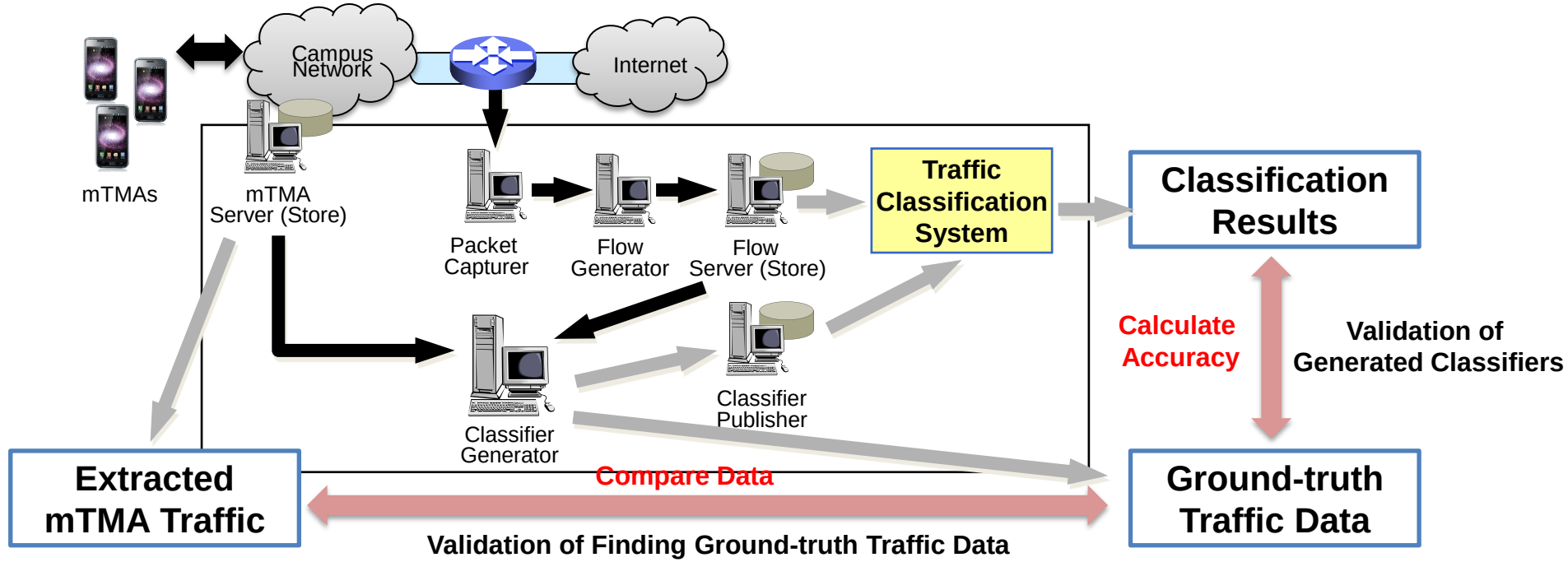
❖ Classifier publisher

- Upload generated classifiers to a public web page
- URL: http://bonn.postech.ac.kr/laser



Validation of Two Proposed Algorithms

- **Matching ratio of finding ground-truth traffic data**
  - Compare the data with extracted mTMA traffic
- **Accuracy of generated classifiers**
  - Apply classifiers to campus mobile traffic
    - Consider traffic matched to mTMA Server as ground-truth traffic
  - Calculate accuracy of generated classifiers
    - Compare classification results with ground-truth traffic



Algorithm: Extracting Ground-truth Data

❖ Approach of proposed algorithm

- **Match with more to less accurate flow information with priority**
- **Apply high-cost payload inspection at the last**

```
procedure Flow_Matching ()
   for each item in M_subtuple
      T ← subnet of T_subtuple that matches
            <dst_ip, dst_port, dst_protocol> for item within time period
      if # of T for exactly matching <pktno, bytes> = 1 then
         Insert <appname,dst_ip,dst_port,flow_info,payload> into R
      end if
      else if # of T for exactly matching <pktno> = 1 then
         Insert <appname,dst_ip,dst_port,flow_info,payload> into R
      end if
      else if # of T for more or less <pktno> than item = 1 then
         Insert <appname,dst_ip,dst_port,flow_info,payload> into R
      end if
      else
         Perform Match payload_P and payload for item and T
         if # of matched results = 1 then
            Insert <appname,dst_ip,dst_port,flow_info,payload> into R
      end if
   end for
   return R
end procedure
```

Proposed Flow Matching Algorithm

Validation of Extracted Ground-truth Data

POSTECH 2020

❖ Overall result

- **Overall ratio is 89%, which is fine although proposed algorithm cannot extract 100% traffic from mobile devices**

❖ Analysis of parameters used in proposed algorithm

- **Exact matching with packet number & flow size results less match ratio**
- **Packet number in a flow is powerful, it is not the same when multimedia mobile applications generated traffic**
- **Payload inspection supports the incompleteness of these two parameters**

Category	Mobile Application	Exact packet number & flow size	Exact packet number	More packet number	Shorter packet number	Payload inspection
Multimedia	Media player	11.29%	37.10%	70.97%	100.00%	100.00%
Web	Web browser	6.48%	72.81%	77.08%	78.04%	81.58%
SNS	Facebook	19.42%	93.04%	95.36%	96.23%	96.81%
	KakaoTalk	7.10%	40.65%	67.10%	72.90%	85.81%