SFMap: Inferring Services over Encrypted Web Flows using Dynamical Domain Name Graphs

TMA 2015

Tatsuya Mori[1], Takeru Inoue[2], Akihiro Shimoda[3],
Kazumichi Sato[3], Keisuke Ishibashi[3], and Shigeki Goto[1]

1 Waseda University
2 NTT Network Innovation Laboratories
3 NTT Network Technology Laboratories

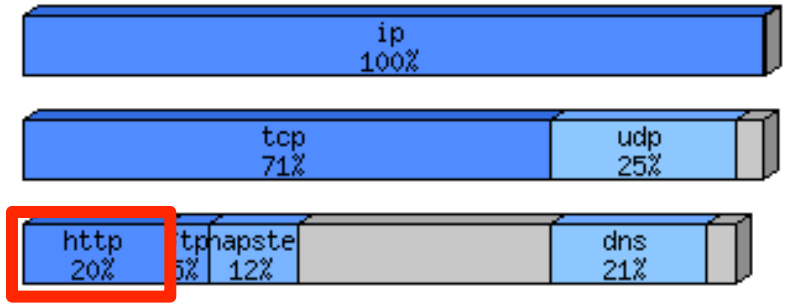
NSL
Networked Systems Laboratory

Background(1) Era of web

Change of Internet traffic

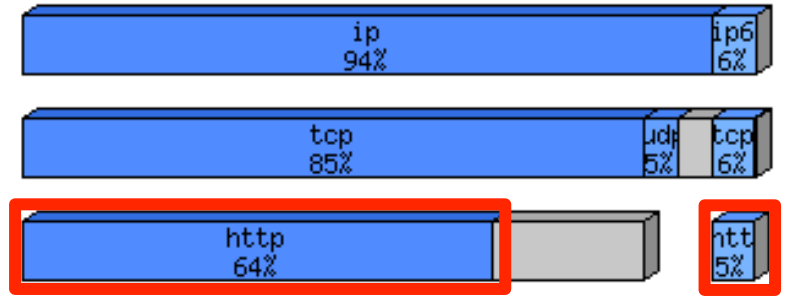
※WIDE Mawi Project http://mawi.wide.ad.jp, samplepoint B, F

2002/12/1



Era of P2P

2012/12/1



Era of Web

Many of primary Internet services have shifted to Web (http): Everything over HTTP

NSL Networked Systems Laboratory

Background(2) Encrypting Web

Deploying HTTPS is not cost any more
Significant portion of web traffic is now encrypted

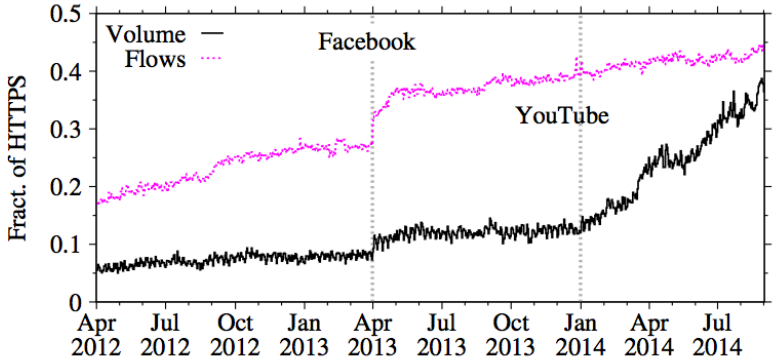


Figure 3: Evolution of HTTPS volume and flow shares ov
2.5 years. Results from Res-ISP dataset. Vertical lines sh
the transition to HTTPS for Facebook and YouTube.

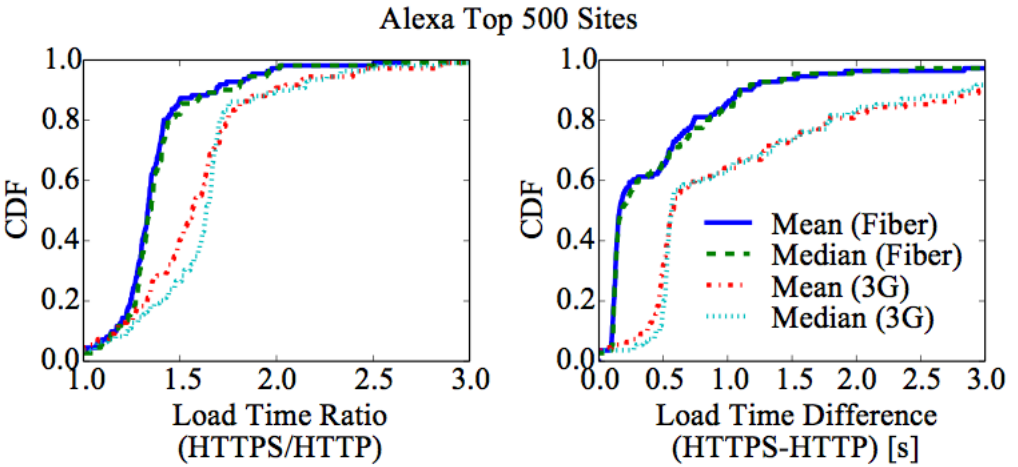
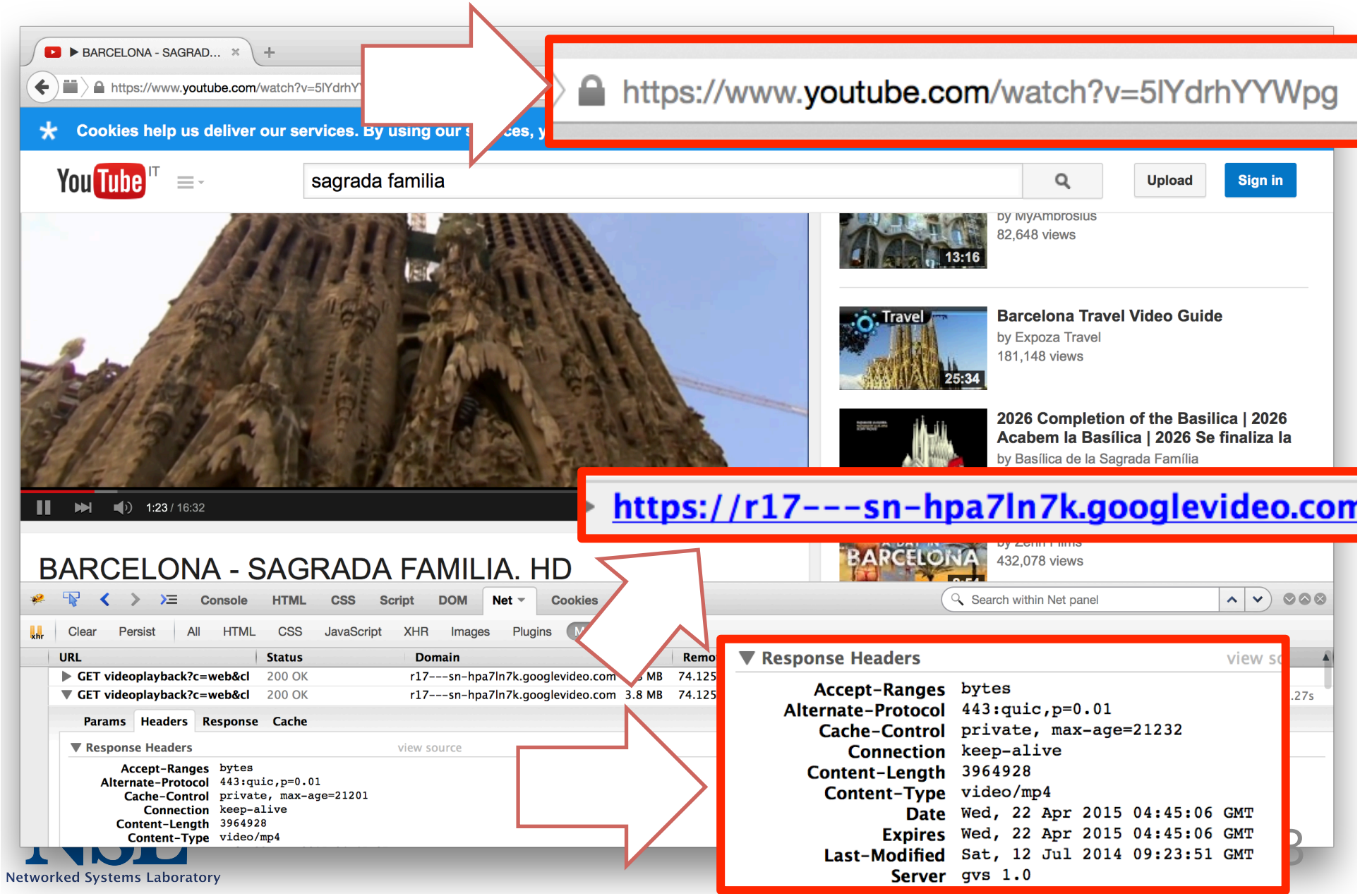


Figure 5: Webpage load time inflation for the Alexa top 500.

D. Naylor et al., **The Cost of the "S"** in HTTP. Proceedings of ACM CoNext, 2014.

NSL
Networked Systems Laboratory

YouTube video over HTTPS!



Netflix started encrypting stream

techdirt

Techdirt | Wireless News | Innovation | Case Studies | Startups | Net Neutrality | Techdirt Deals!

Main | Submit a Story | RSS

PODCAST Techdirt – How The Patent System Can Be Fixed SOUNDCLOUD

<< DailyDirt: More Nature-Inspired Materials | Crowdsourcing The Human Telescope >>

Privacy
by Mike Masnick
Wed, Apr 15th 2015
9:02pm

1

Flattr

Netflix Moving To Encrypted Streams, As Mozilla Moves To Deprecate Unencrypted Web Pages As Insecure

from the *yay-encryption* dept

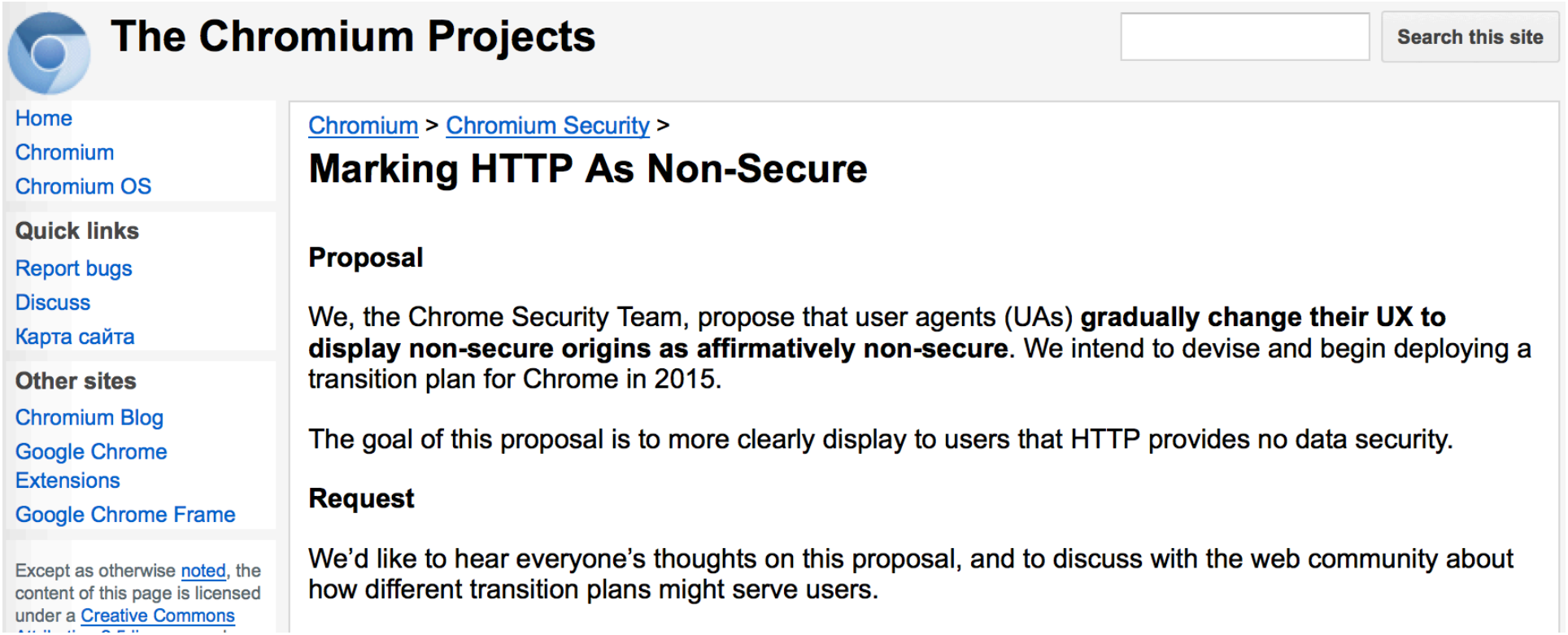
We've been pretty vocal about supporting the encryption of more and more web traffic. It's important for a variety of reasons, not the least of which is your privacy and security. A few months back, we were excited to see the Chrome security team suggest that it should start **marking unencrypted web pages as non-secure**. It appears that Mozilla is now joining in on the fun, proposing **deprecating unencrypted HTTP web pages** to encourage more web developers to go full on in support for encrypted HTTPS:

> *In order to encourage web developers to move from HTTP to HTTPS, I would like to propose establishing a deprecation plan for HTTP without security. Broadly speaking, this plan would entail limiting new features to secure contexts, followed*

Era of WebSocket

NSL
Networked Systems Laboratory

HTTP = non-secure!



The Chromium Projects

Search this site

Home
Chromium
Chromium OS

Quick links
Report bugs
Discuss
Карта сайта

Other sites
Chromium Blog
Google Chrome Extensions
Google Chrome Frame

Except as otherwise noted, the content of this page is licensed under a Creative Commons

Chromium > Chromium Security >

Marking HTTP As Non-Secure

Proposal

We, the Chrome Security Team, propose that user agents (UAs) **gradually change their UX to display non-secure origins as affirmatively non-secure**. We intend to devise and begin deploying a transition plan for Chrome in 2015.

The goal of this proposal is to more clearly display to users that HTTP provides no data security.

Request

We'd like to hear everyone's thoughts on this proposal, and to discuss with the web community about how different transition plans might serve users.

ISPs need to understand traffic mix

- to figure out what to **control** in the presence of congestion.
  - Shaping HTTP flows is too coarse-grained.
  - Shaping flows from a range of IP addresses is also too coarse-grained.
- to know **demand** of end-users
  - What types of services are consuming network resources.

  → Can be used to rethink new architecture or business model

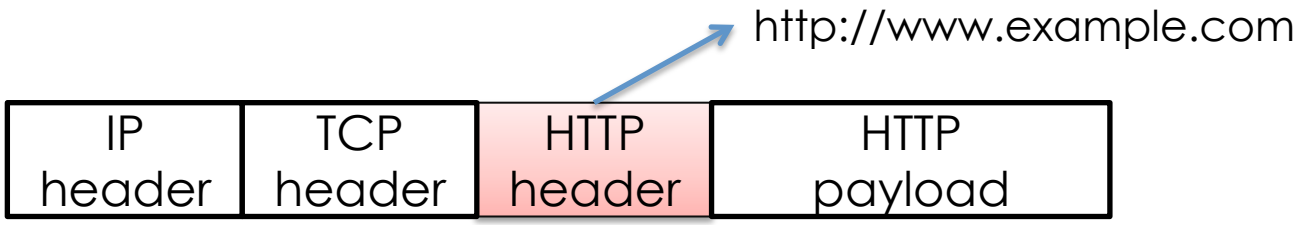
  peering policy, installing cache mechanism,
  WAN optimization, CCN/ICN,

- **Obstacle**： coping with HTTPS

NSL
Networked Systems Laboratory

HTTP vs. HTTPS

- **HTTP**:
  - HTTP header composes of URL information



- **HTTPS**:
  - Entire HTTP protocol including header is encrypted. No URL information is available.

Solution 1: Server IP addresses

- Many of IP addresses can be reverse looked up (PTR record)
- There are many IP addresses that are not configured to have PTR records.
- A single IP address can be associated with many distinct FQDNS (cloud, hosting services, etc.)

```
157.205.136.242 busyu.co.jp
157.205.136.242 edo-ichi.jp
157.205.136.242 gntdns01.alpha-plt.jp
157.205.136.242 wp.tokyo-sports.co.jp
157.205.136.242 www.38shop.jp
157.205.136.242 www.daska.jp
157.205.136.242 www.dnh.co.jp
157.205.136.242 www.edo-ichi.jp
157.205.136.242 www.eme-tokyo.or.jp
157.205.136.242 www.heatwavenet.co.jp
157.205.136.242 www.humax-cinema.co.jp
157.205.136.242 www.j-n.co.jp
157.205.136.242 www.jcsc.or.jp
157.205.136.242 www.jira.or.jp
157.205.136.242 www.kyowa-line.co.jp
157.205.136.242 www.life-bio.or.jp
157.205.136.242 www.needstour.com
157.205.136.242 www.photal.co.jp
157.205.136.242 www.print-value.net
157.205.136.242 www.sayama.com
157.205.136.242 www.tokyo-sports.co.jp
```

Solution 2: SSL/TLS certificates

- Public key certificate of server is exchanged during SSL/TLS handshake stage. The certificate should contain domain name of the server.

- An organization can register a single certificate for many sub-domains, i.e., so called wildcard certificates
  - E.g., *.google.com

Solution 3: SNI extension

- SNI (Server Name Identification) of TLS can be used to obtain FQDN of an HTTPS server.

- Many of client/server implementations have not adopted SNI, yet.
  - In our dataset, roughly half of HTTPS clients did not use the SNI extension.

Solution 4: Decrypting HTTPS

- Anti-virus software or firewall products have mechanisms to intercept HTTPS traffic

- They use self-signed certificates to work as a transparent HTTP(S) proxy.
  - Same as the MTIM (Man-in-the-middle) attack
  - Needs for installation of certificates for each OS/ application

- [cf] IETF Explicit Trusted Proxy in HTTP/2.0 (I-D expired)

Goal

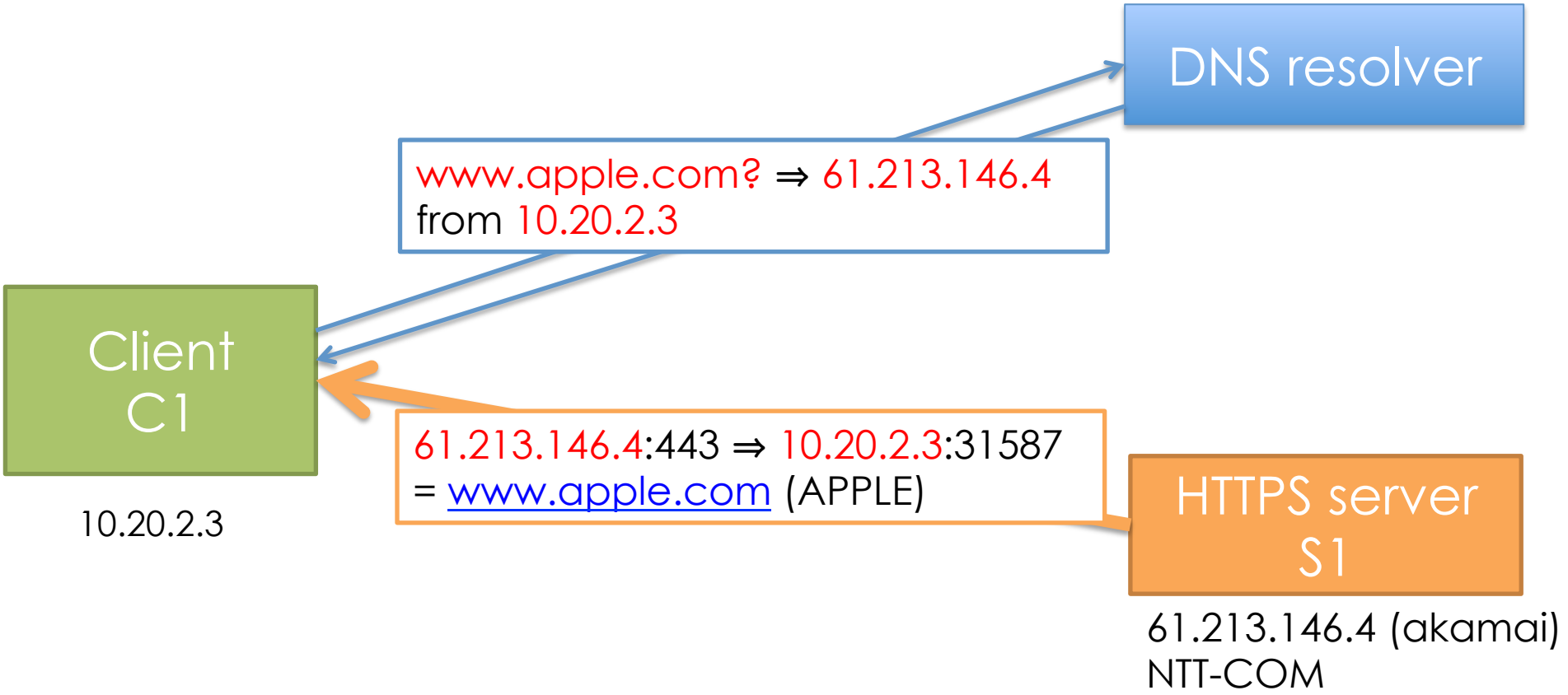
- Estimate server hostnames of HTTPS traffic
  - Server hostnames can be used as a good hint to estimate the **services** provided by the server
  - E.g., www.apple..com, itunes.apple.com , …
- Establish better performance than the existing solution (DN-Hunter)

Idea

- Leverage DNS name resolutions that precedes HTTPS transactions
  - Labeling data plane using control plane
  - This is not a simple task as we will describe soon.
  - [cf] state-of-the-art = **DN-Hunter** (IMC 2012)

- Use statistical inference when measurement is incomplete
  - DNS resolutions can be missed due to some reasons

Illustration of DNS approach



Three practical challenges:

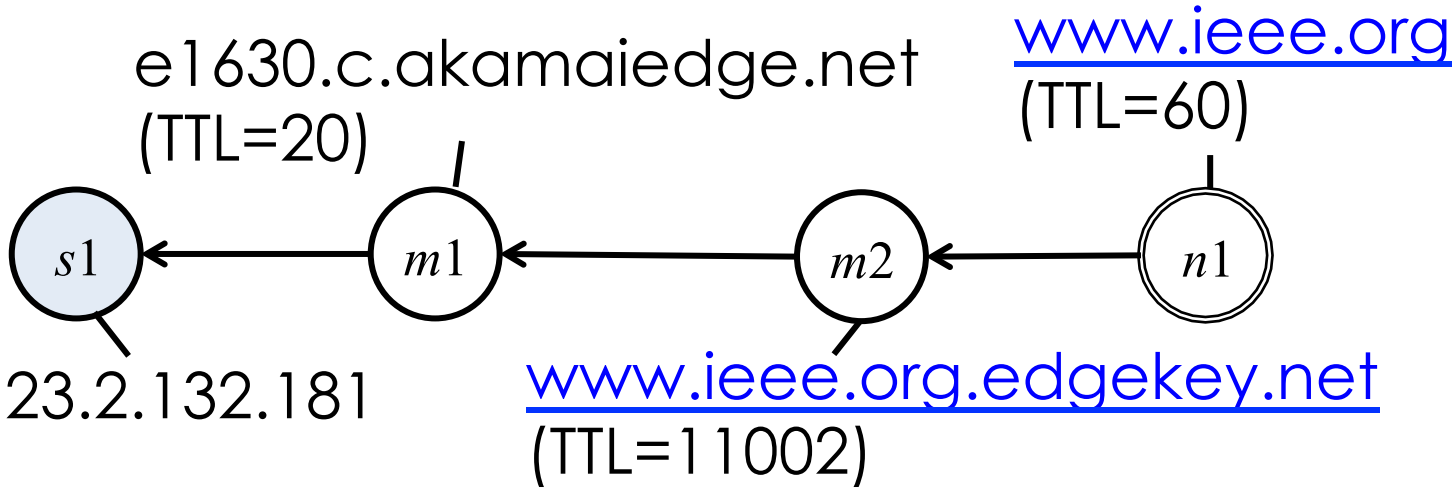
1) CNAME tricks

2) Incomplete measurements

3) Dynamicity, diversity, and ambiguity

1) CNAME tricks

- Modern CDN providers heavily make use of CNAME tricks to optimize content distribution
- We need to keep track of not only client/server IP addresses/hostnames, but also intermediate CNAMEs



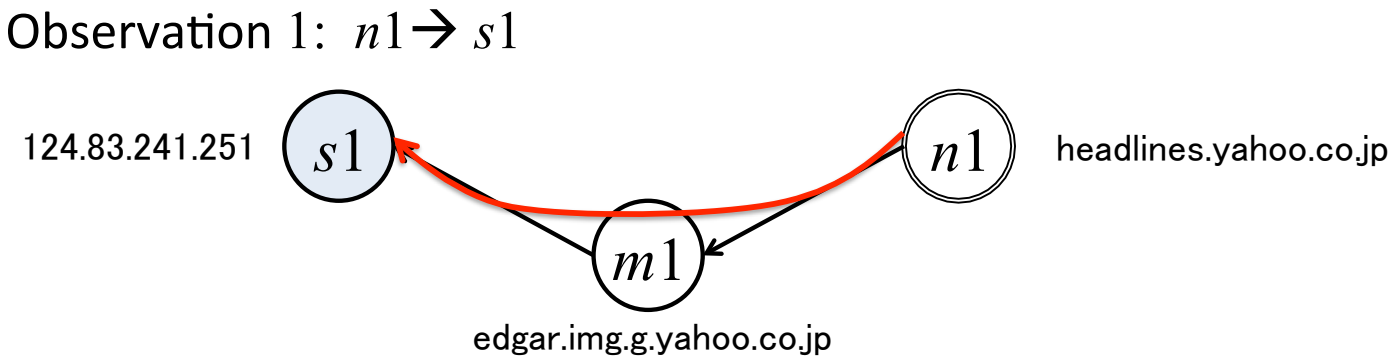
2) incomplete measurements

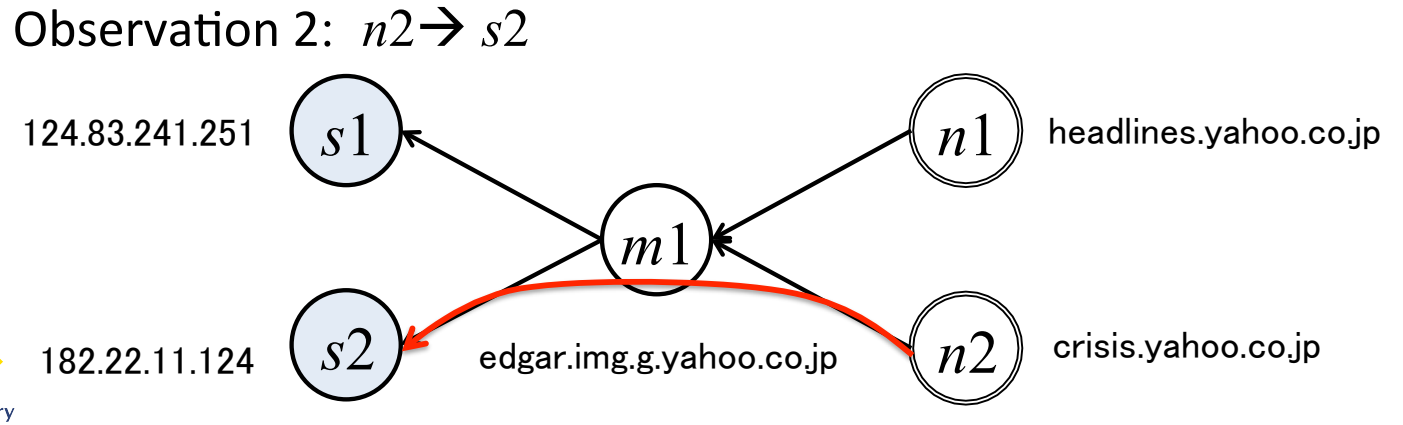
- Various DNS caching mechanisms in the wild
  - Browser/apps
  - OS
  - Home routers w/DNS resolver
  - DNS resolvers (Organization/ISP/Open)

- From the viewpoint of ISPs, DNS queries originated from end-users can be missed due to the intermediate caching mechanisms

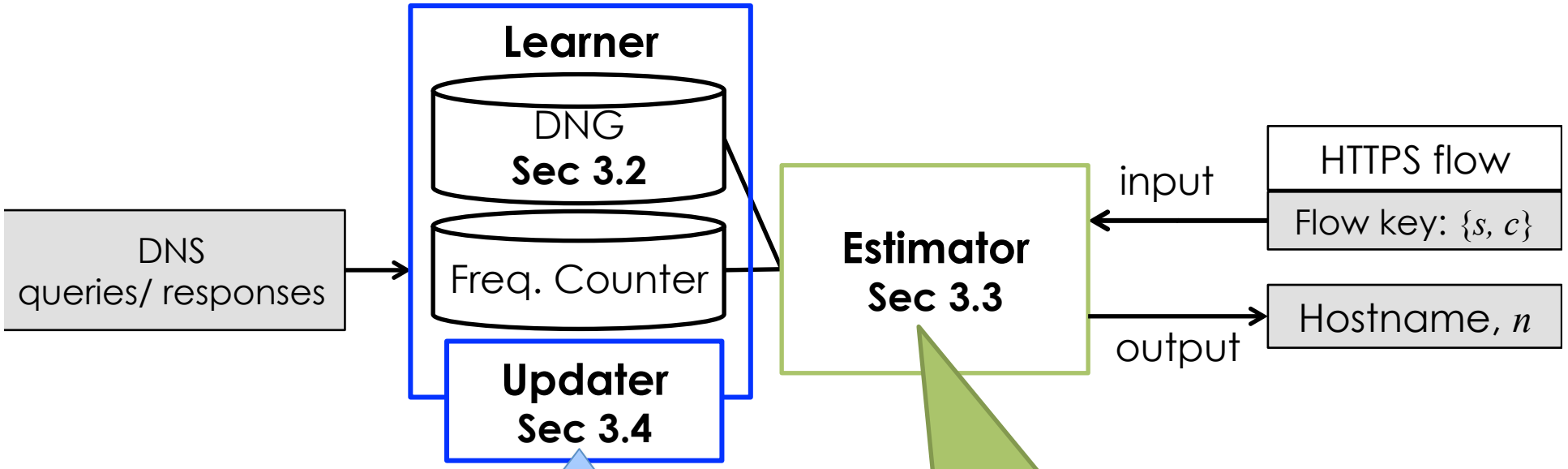
3) Dynamicity, diversity, and ambiguity

- A pathological/popular example





SFMap (Service-Flow Map)



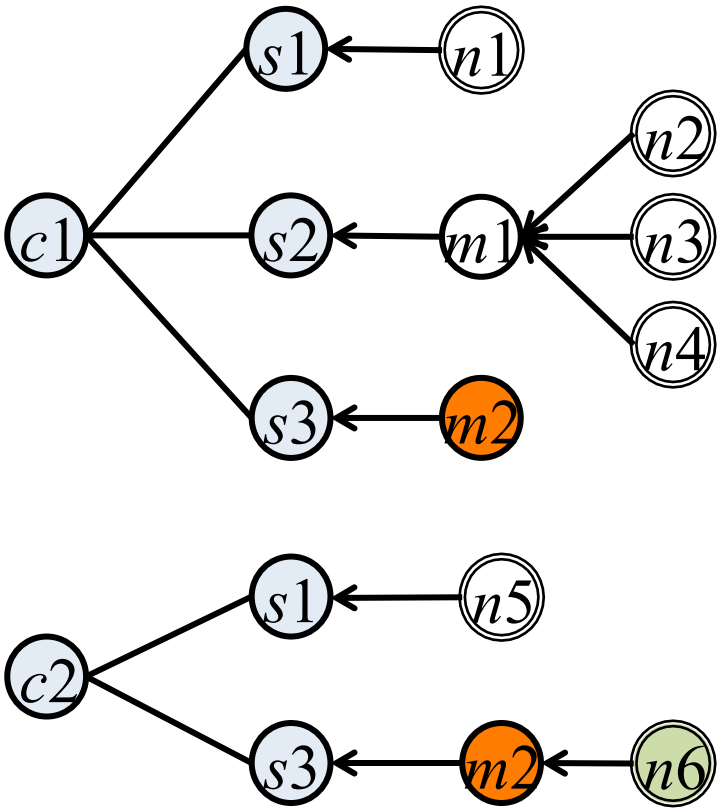
Learning with monitored queries:
Building DNG (Domain Name Graph)

Hostname estimation using DNG
Use maximum likelihood estimation when necessary

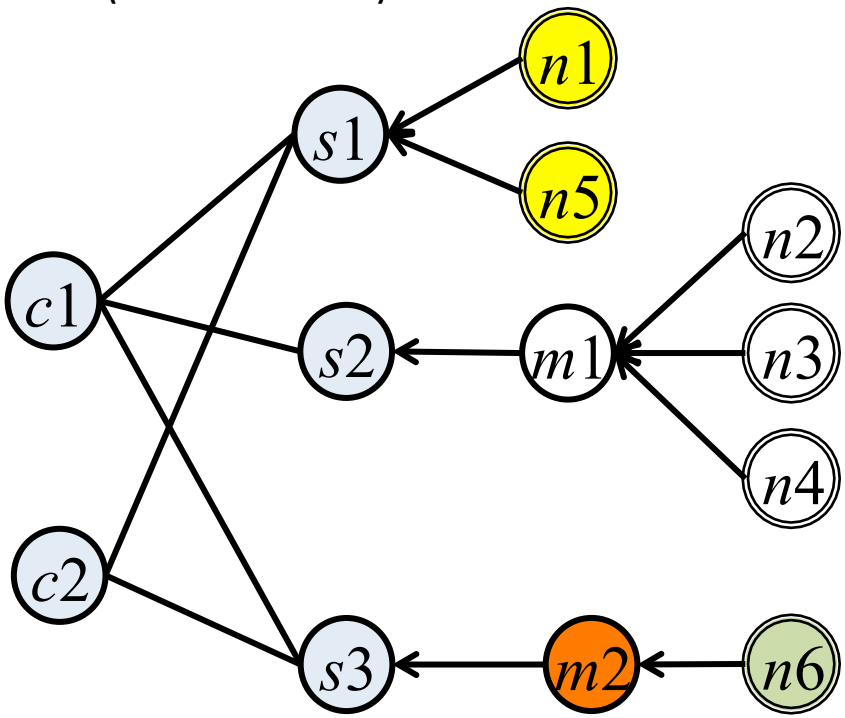
Networked Systems Laboratory

Illustration of DNG

Per-client graphs
(local DNG)



A global graph
(union DNG)

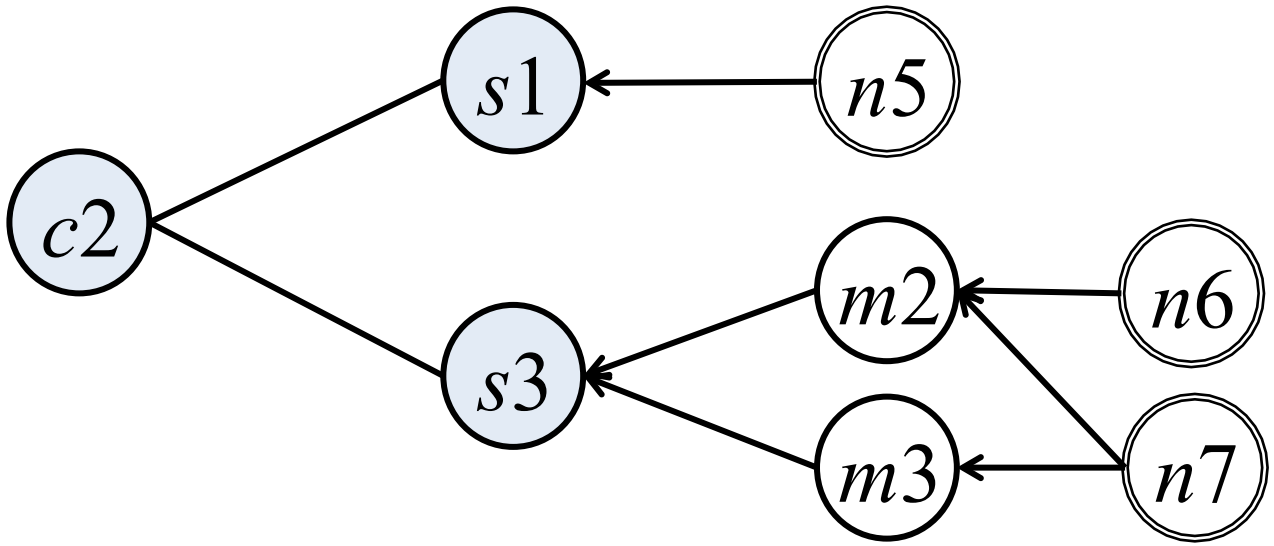


NSL
Networked Systems Laboratory

Overview of hostname estimation algorithm (1)

- **Get** client/server IP addresses (C,S) from an HTTPS flow
- **Search** a set of hostnames **N** corresponding to (C,S) on DNG
  - Enumerate edge nodes **N** that have paths reachable from C to S on DNG
  - Also consider TTL expiration
- If $|\mathbf{N}| = 1$, it is the estimated hostname

An example



(c2, s1) → estimation = n5
(c2, s3) → candidates = n6, n7

Overview of hostname estimation algorithm (2)

- If there are multiple candidates, sort them in descending order, according to the likelihood probabilities
  - Uncertain events → use frequencies
    - Second, third candidates can be informative
  - Note: The statistical inference can be extended to Bayes estimation that uses P(n) (a priori probability)

Updating DNG

- States/Statistics of DNG is updated online when a DNS query is observed

Algorithm 1: Updater

Input: $c, n^\star, A, M$ // DNS response

1 **for** $(u, v) \in A \cup M$ **do**

2 $\quad E_c = E_c \cup \{(u, v)\}$ // to add edge

3 $\quad$ update expire time of edge (u, v)

4 $N' = \{n' \in V_c : (*, n') \notin E_c,\ n' \underset{G_c}{\rightarrow} n^\star\}$ // leaf vertices reachable to $n^\star$

5 **for** $n' \in N'$ **do**

6 $\quad$ **for** $(*, s) \in A$ **do**

7 $\quad\quad F_c(n', s) = F_c(n', s) + \frac{1}{|N'| \cdot |A|}$ // to increment frequency

8 **return** G_c, F_c

Dataset

- LAB:
  - A small LAN used by research group
- PROD:
  - Middle-scale production network

	learning time	# of clients	# of DNS responses	estimating time	# of servers	# of HTTP requests	# of hostnames
LAB	0 ~ 12 h	10	5,226	10 ~ 12 h	1,705	542	1,135
PROD	0 ~ 12 h	4,250	86,854	10 ~ 12 h	10,785	55,091	10,534

Scales of DNGs (12 hours long)

	Local DNG w/o TTL expiration		Union DNG w/o TTL expiration	
	mean # of nodes	mean # of edges	total # of nodes	total # of edges
LAB	460	755	2,849	5,979
PROD	56	80	25,403	172,974

Estimation accuracy (1)

Exact match

	LE	LE-NTE	UE	UE-NTE	DN-Hunter
LAB	54.98%	68.08%	71.59%	92.25%	67.90%
PROD	79.90%	88.29%	90.88%	90.88%	85.40%

UNION DNG without TTL expiration

Public suffix match

	LE	LE-NTE	UE	UE-NTE	DN-Hunter
LAB	57.20%	70.30%	73.80%	94.46%	73.43%
PROD	83.20%	92.12%	94.52%	94.98%	89.98%

Estimation accuracy (2)

Accuracies of top-3 estimations (UE-NTE)

	Exact matching			Public suffix		
	Hit in 1	Hit in 2	Hit in 3	Hit in 1	Hit in 2	Hit in 3
LAB	92.25	97.23	98.16	94.46	98.16	98.16
PROD	90.88	95.77	96.71	94.98	97.01	97.43

The top-3 ranked hostnames were similar in many cases; e.g,

pagead2. googlesyndication.com
pubads.g.doubleclick.net,
googleads.g.doubleclick. net

Discussion

- **Sources of inevitable misclassification**
  - DNS implementations that ignore TTL expiration
    - It keeps holding old information
  - mobility
    - DNS could be resolved in different vantage point
  - Hardcoded IP addresses
    - Some gaming apps did have such mechanism

Discussion (cont.)

- **Scalability**
  - Did not matter for our datasets
  - Size of DNG depends on the number of client IP addresses
    - Some aging mechanism should be incorporated for much large-scale DNGs (future work)
- **URL=hostname + path.**
  - How can we deal with path?
  - Need for a standard mechanism to explicitly expose path like SNI?

Summary

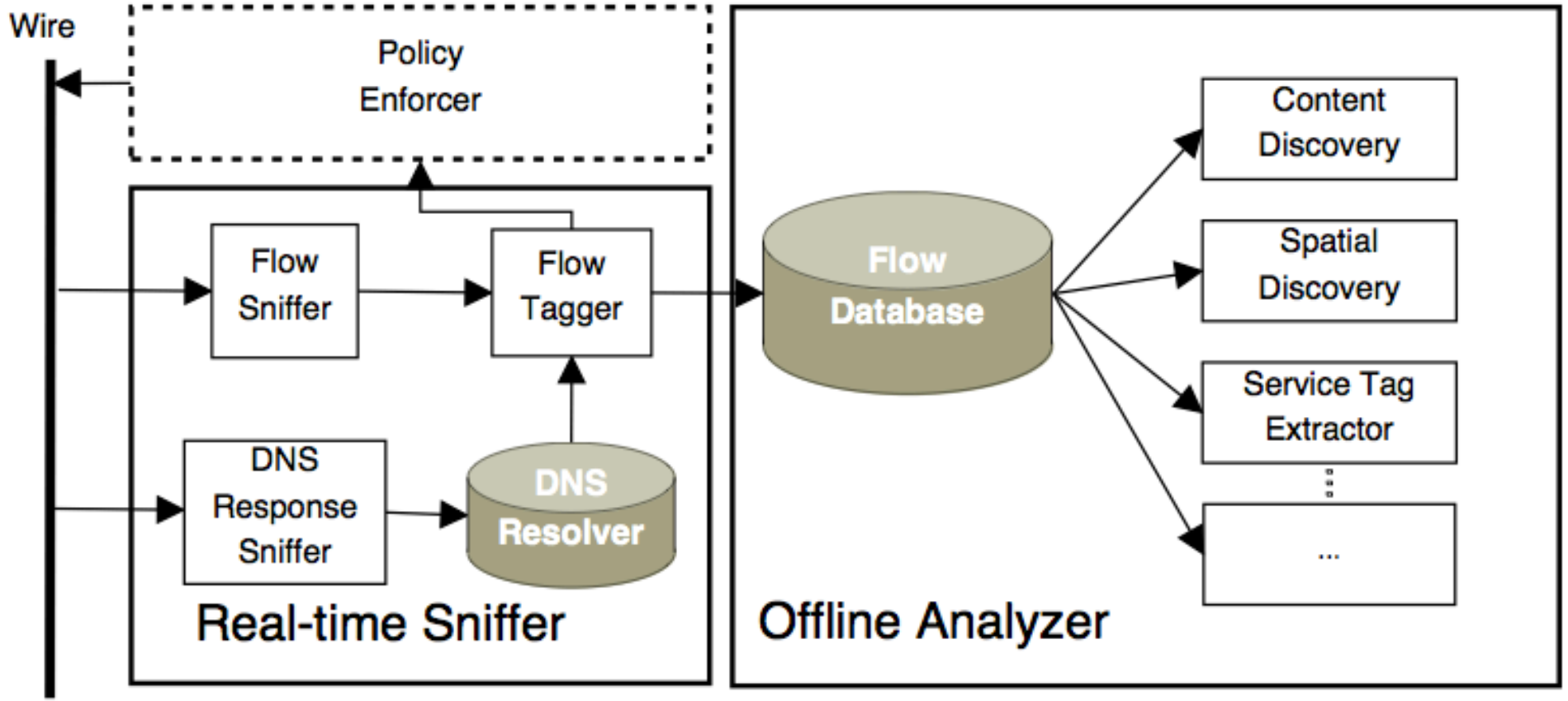
- **SFMap** framework estimates hostnames (~services) of HTTPS traffic using past DNS queries
- **Key ideas**：use of **DNG** and statistical inference
- SFMap achieved better accuracies than the state-of-the-art work (DN-Hunter)
  - Exact match: **90-92%** accuracies
  - Public suffix match: **94-95%** accuracies
  - Top-3 hit: **97-98%** accuracies

Acknowledgements

- This work was supported by JSPS KAKENHI Grant Number 25880020.

Existing research: DN Hunter

- Bermudez et al., "DNS to the Rescue: Discerning Content and Services in a Tangled Web", ACM IMC 2012



Comparison with DN-Hunter

	Distributed monitoring	Statistical estimation
DN Hunter	△	X
SFMap	○	○

Distributed monitoring