Classifying DNS Heavy User Traffic by using Hierarchical Aggregate Entropy

2012/3/5

Keisuke Ishibashi, Kazumichi Sato

NTT Service Integration Labs

Motivation

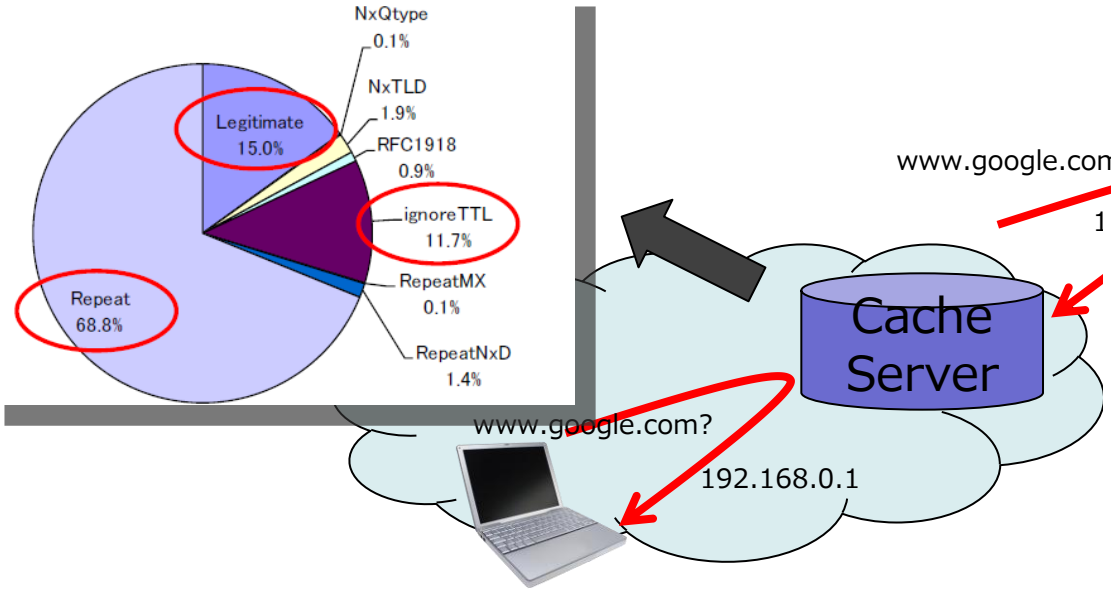
- Network resources are consumed by a small number of heavy users
- Controlling traffic from heavy users is a crucial task for efficient use of network.
  - filtering, rate limiting, charging
- Before controlling the heavy user traffic, we need to understand what type of traffic they send
- If heavy user traffic are mostly anomalous, then filtering such traffic is rather acceptable.

– Anomalous traffic: DDoS attack, spam, illegal file exchange etc.

- Thus, we need to classify heavy user traffic whether normal or abnormal
- In this talk, we focus on heavy users in DNS traffic, one of the most important control traffic in the Internet

Bogus traffic in DNS

- DNS: mainly used for mapping domain name to IP address or vice versa
- Two types of servers: caching server and authoritative server
- Bogus queries are consuming resources of both DNS authoritative servers and caching servers
  - repeated queires for a single name (bug?)
  - scaning quereis for non exiting names (worm?)



Type	Count	Percent
Unused Query Class	36,313	.024
A for A	10,739,857	7.03
Unknown TLD	19,165,840	12.5
Nonprintable in query	2,962,471	1.94
RFC1918 PTR	2,452,806	1.61
Identical Query	38,838,688	25.4
Repeated Query	68,610,091	44.9
Referral Not Cached	6,653,690	4.36
Legitimate	**3,284,569**	**2.15**

TABLE II
QUERY CLASSIFICATION RESULTS (24-HOUR PERIOD ON 4 OCTOBER 2002 AT THE F-ROOT DNS SERVER).

[Wessels] D. Wessels et.al., "Wow, That's a Lot of Packets, " PAM 2003.
[Toyono] T. Toyono et. al., "An analysis of the queries from the view point of caching servers," 2007 DNS-Operations Workshop.

Motivation cont'd

- Most of bogus queries are sent by small number of heavy clients [toyono]
  - Filtering queries sent by those heavy clients is efficient to protect DNS server resources

(Percentage of total queries)

type \ rate	100qps	200qps	300qps	400qps	500qps
Legitimate	0.09%	0.01%	0%	0%	0%
NxQtype	0%	0%	0%	0%	0%
NxTLD	0%	0%	0%	0%	0%
RFC1918	0.80%	0%	0%	0%	0%
ignoreTTL	1.63%	0.05%	0.01%	0%	0%
RepeatMX	0.01%	0%	0%	0%	0%
RepeatNxD	0.64%	0%	0%	0%	0%
Repeat	59.69%	59.69%	59.69%	59.69%	59.69%

- However, not all queries from heavy clients are bogus
  - PTR queries from web servers (analog)
  - Aggregated queries from DNS proxies

Normal heavy user

- DNS prefetch
  - resolves all the domain names of the URLs in a browsed web page before the URL is actually clicked
  - Faster web but burst (unnecessary) queries for a page that contains huge URLs
- Log analyzer
  - Log analyzers in web server send reverse queries (resolve domain names for IP addresses) for addresses in their access logs
  - What organizations access our web servers?

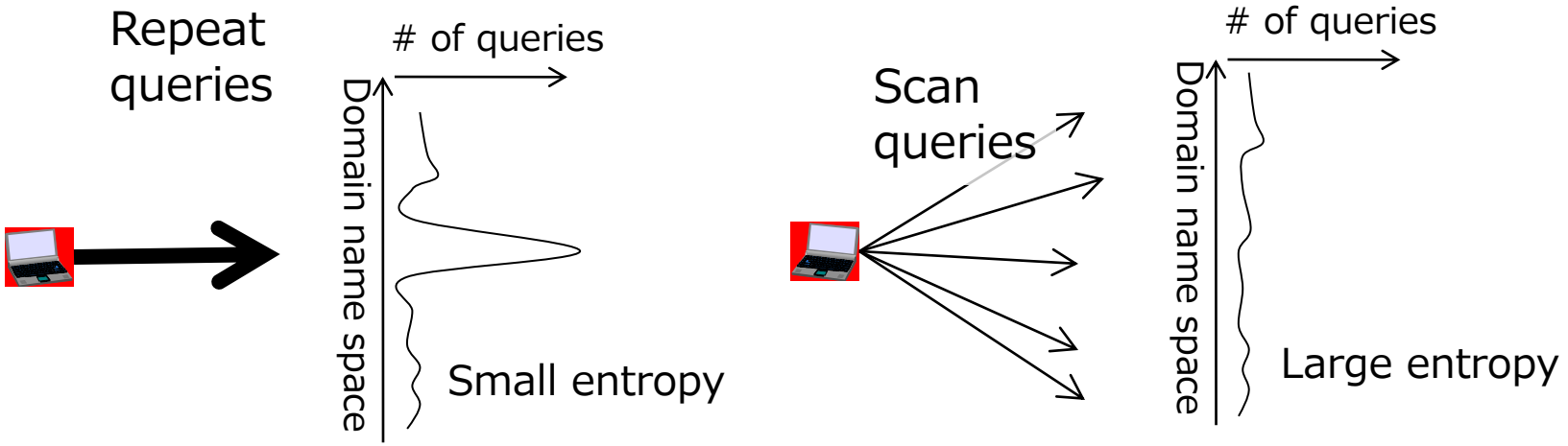
Entropy based classification

- Needs to classify heavy clients into normal users and abnormal users

⇒Classify heavy clients by their query pattern

- How to capture query patterns?

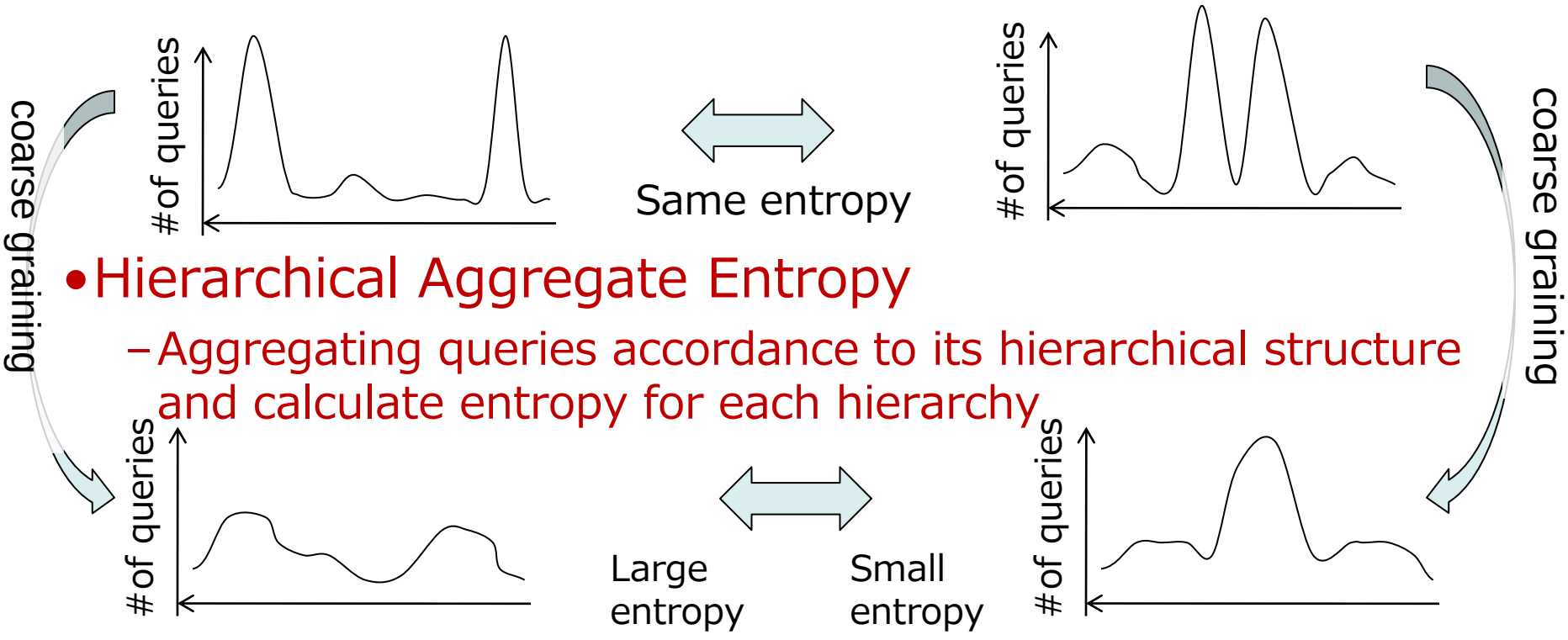
⇒Use of entropy of queries in domain name spaces



- Entropy of legitimate queries: expected to lie between them

Hierarchical Aggregate Entropy(1/2)

- Does not have information on spatial characteristics
  - Independent on where queries concentrate or diverse in domain name spaces
  - Only depends on how queries concentrate or diverse



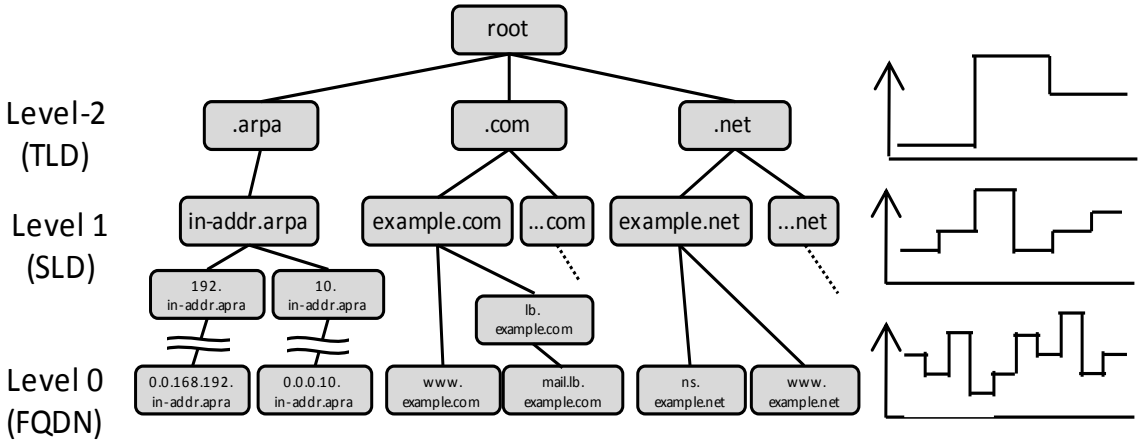
- Hierarchical Aggregate Entropy
  - Aggregating queries accordance to its hierarchical structure and calculate entropy for each hierarchy

Hierarchical Aggregate Entropy(2/2)

- DNS: tree based hierarchical structure
  - Fully qualified domain name (FQDN): www.google.com, www.ntt.co.jp
    ⇒FQDN-level entropy $H(D^{(0)}|D^{(1)})$: deviation in www.example.org level
  - Second level domain (SLD): .google.com
    ⇒ SLD-level entropy $H(D^{(1)}|D^{(2)})$:deviation in example.org level
  - Top level domain (TLD) : .com, .net, .jp…
    ⇒ TLD-level entropy $H(D^{(2)})$ dispersion in queries for com, .net, .jp…

$$H(D^{(0)}) = H(D^{(2)}) + H(D^{(1)}|D^{(2)}) + H(D^{(0)}|D^{(1)})$$

Domain Tree Query Distribution

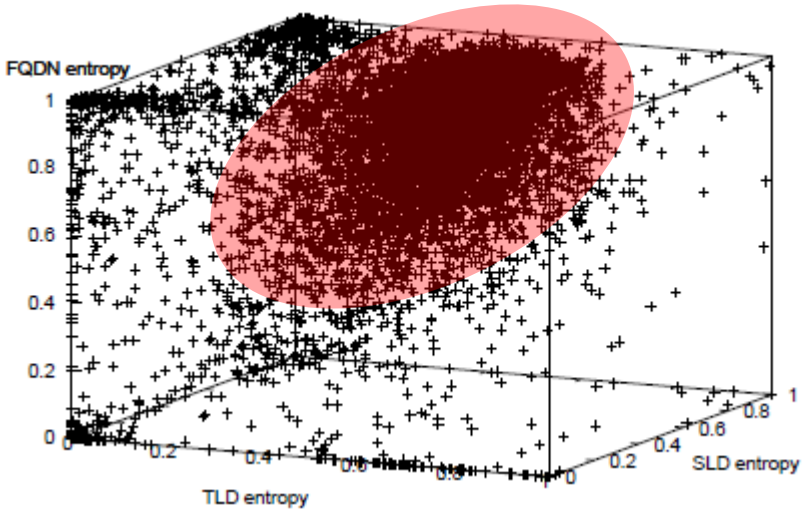


Identify the deviation occurs intra TLD or inter TLD.

Experimental results

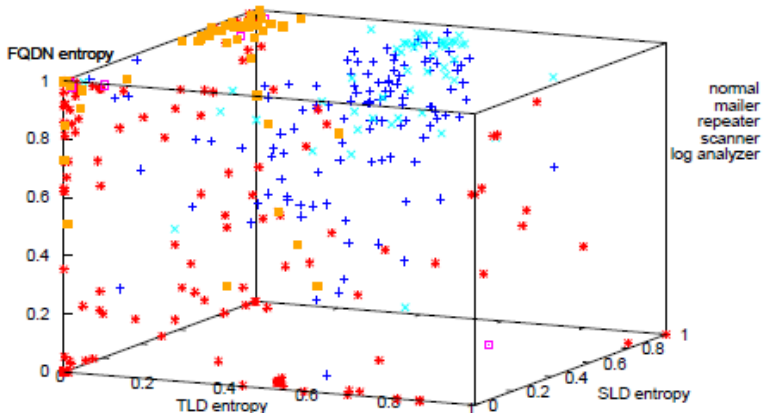
- Calculate entropies of top 10,000 heavy clients for DNS traffic monitored at DNS caching servers
- Entropies from normal clients concentrated in a specific region

⇒Clients whose entropies are out of the region can be expected to be abnormal

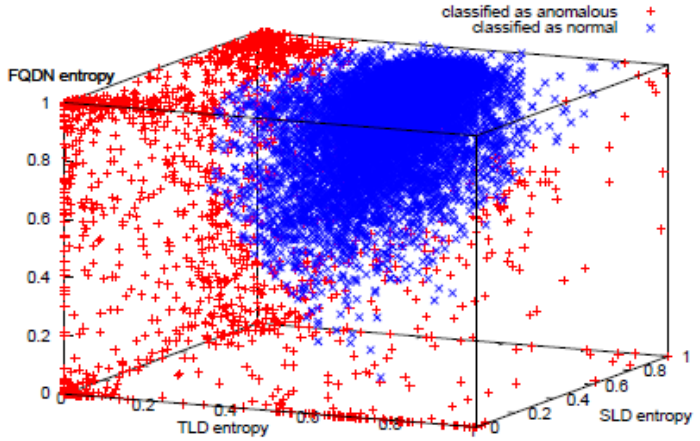


Classification by using SVM

- Extract normal domain by using SVM
  - Training Data: manually labeled data for host sending over 1 query per second
  - SVM(Support Vector Machine): generates boundary between normal region and abnormal region based on the training data



Training Data



Classification Results

Accuracy of classification

- Evaluate the accuracy with 10 cross-fold validation
  - Separate training data into 10 groups.
  - Classify host in a group by using training data of the rest of nine groups and compare the classification results and manual label.
- 10% improvement can be achievement by using hierarchical aggregate entropy

Entropy	Mis-classification ratio (FP+NP)
Hierarchical Aggregate Entropy	8.7%
FQDN Entropy	18.9%
SLD Entropy	23.3%
TLD Entropy	19.8%

Reverse query

- Reverse queries (IP addr -> FQDN) have common SLD and TLD -> entropy is zero.
  - 1.0.168.192.in-addr.arpa.

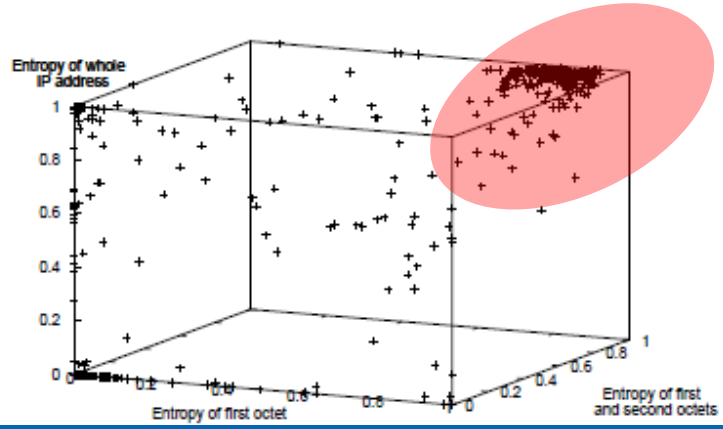
TLD

SLD

- Apply our hierarchical aggregate entropy to IP address part
  - 1.0.168.192.in-addr.arpa.

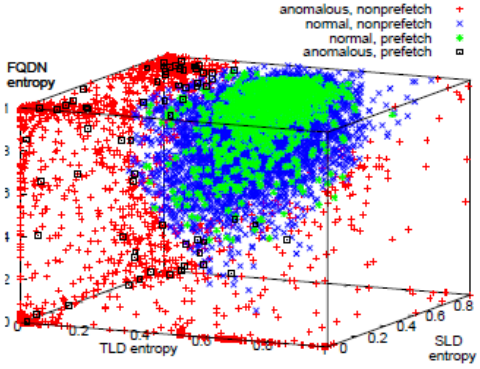
TLD SLD

- Confirm concentration of entropy of normal clients to a domain

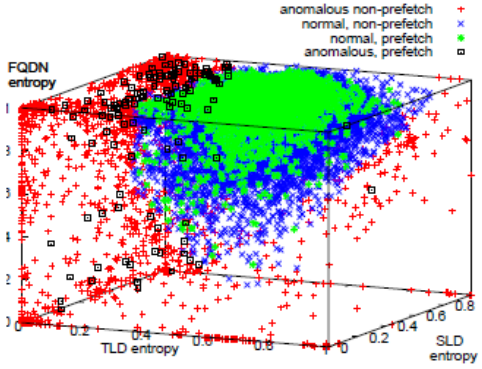


Effect of DNS prefetch

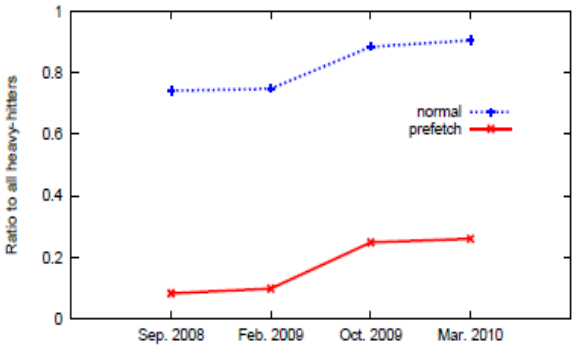
- Extract Firefox users, and compare their entropies before and after Firefox implements DNS prefetch
- After the implementation, ratio of Firefox users among heavy users increases, and that of normal heavy users increases as well.
- Filtering queries from heavy users may impede Internet access of normal users



(a) Before (Feb. 2009)



(b) After (Mar. 2010)



Conclusion

- Propose the use of hierarchical aggregate entropies to classify DNS heavy clients
- Can capture spatial dispersion of queries among domain name spaces
- Entropies from normal clients concentrated in a specific region
- Experimental results show that the proposed method achieve 10 % improvement in classification accuracy