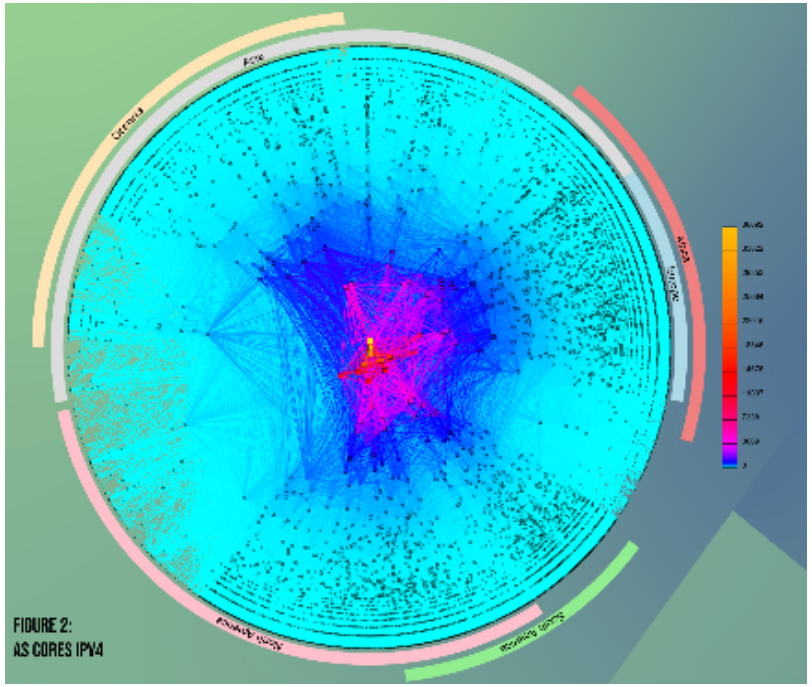
The Hitchhiker's Guide to Traffic Analysis

Jacob Chiang, CTO, Genie Networks

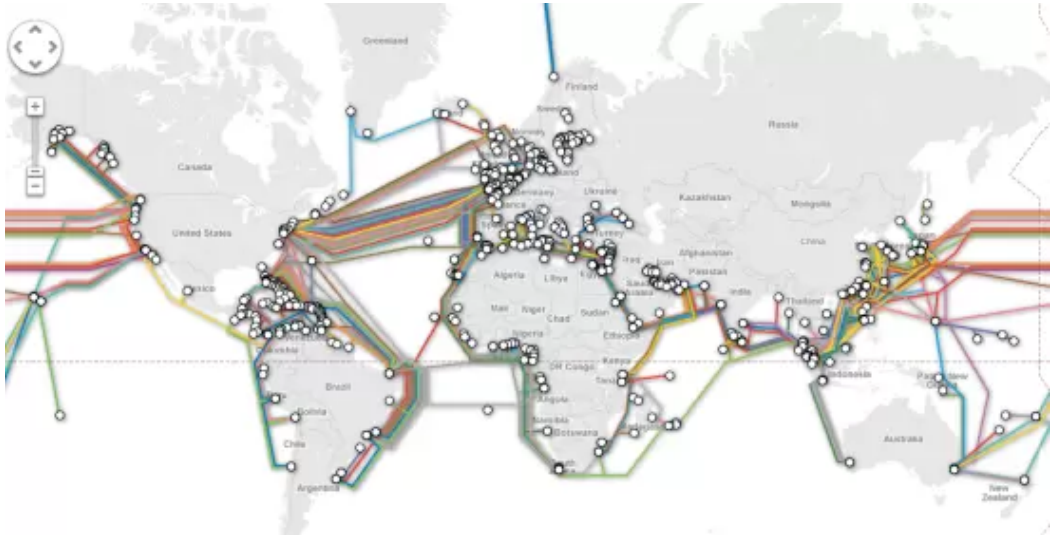
Routing Analytics

The Macro View Of Internet



Quick Facts

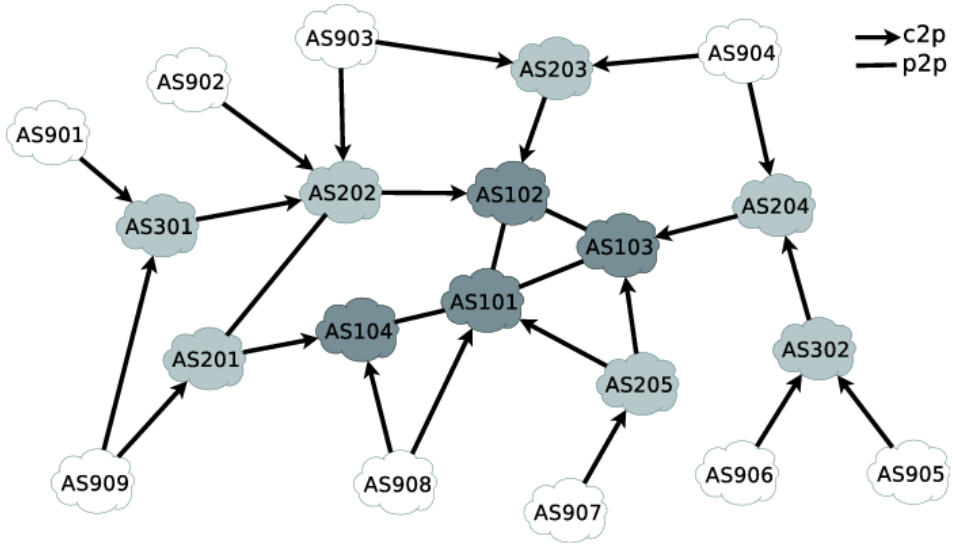
- 71,417 autonomous systems (2021/03)
- 878,585 prefixes (2021/04)



Quick Facts

- ~ 1,200,000 km submarine cables
- ~ 380 submarine cables in use

Internet Topology



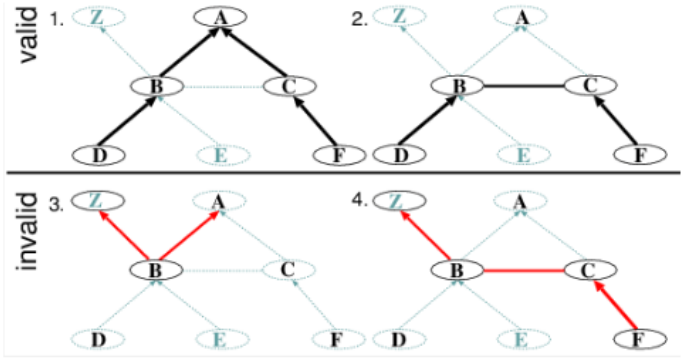
A simplified example of the AS-level Internet topology

Autonomous System

A collection of connected Internet Protocol (IP) routing prefixes under the control of one or more network operators on behalf of a single administrative entity or domain that presents a common, clearly defined routing policy to the Internet.

Valid and Invalid Route

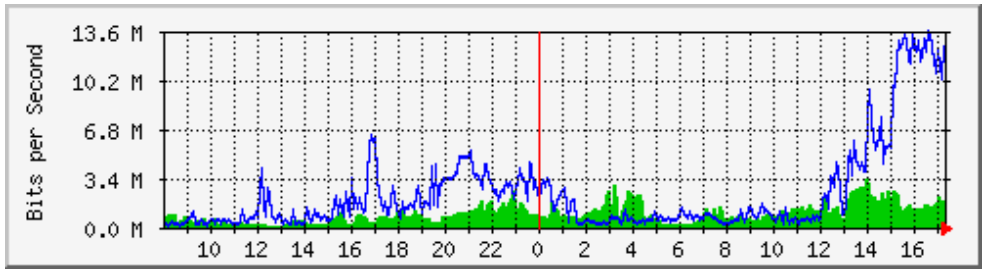
No pay, no transit.

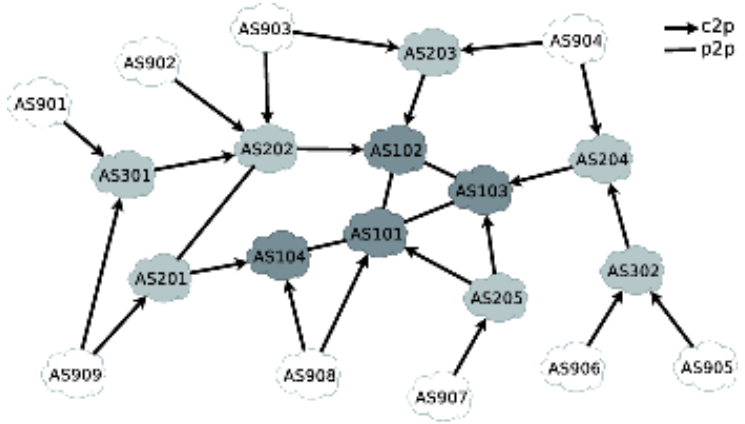


The Demand Of Profiling

A long time ago in a network far, far away

- There's a protocol called SNMP
- There's a tool called MRTG





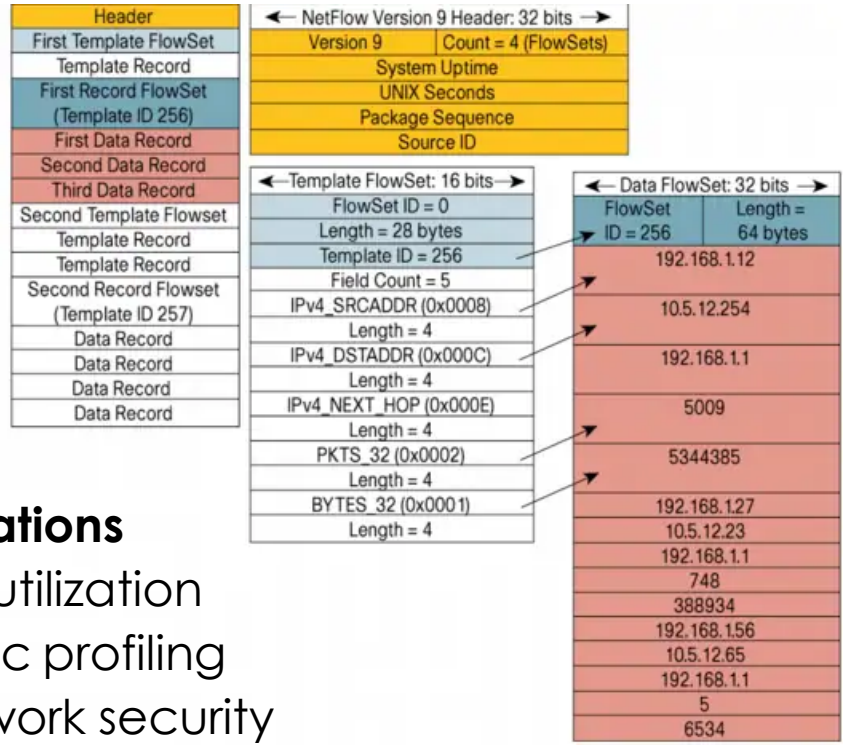
Typical demand of profiling

- The bandwidth from AS203 to AS202

IPFIX - IP Flow Information Export

A smart person said

- Let's survey network traffic on router
- Randomly sample one packet every S packets
- Exports aggregated number of packets and bytes

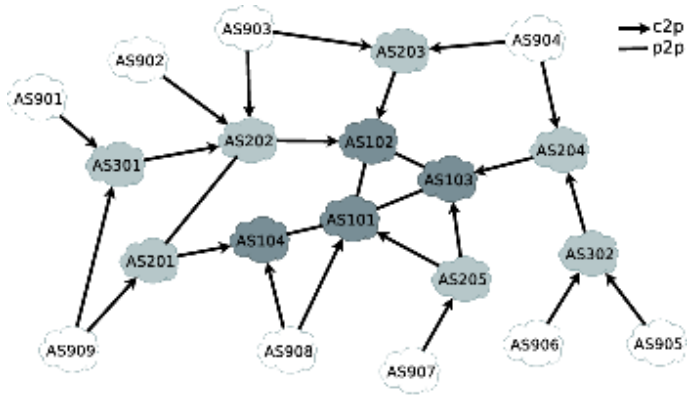


Applications

- Link utilization
- Traffic profiling
- Network security
- Traffic engineering
- Accounting
- QoS monitoring

Challenge I - Confidence Interval

- Each packet sampled by router is a success-failure experiment.
  - Binomial proportion confidence interval - an interval estimate of a success probability p when only the number of experiments n and the number of successes n_s are known.
- The success probability p is estimated as
  - $p = \hat{p} \pm \hat{e}$ where $\hat{e} = z\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$
  - $\hat{p} = \frac{n_S}{n}$ is the proportion of successes
  - $z = 1.645$ with 90% confidence level; 1.960(95%); 2.576(99%)
- Convert the confidence interval to proportion of the metric.
  - $\hat{E} = \frac{\hat{e}}{\hat{p}} = z\sqrt{\frac{1-\hat{p}}{n \times \hat{p}}} = z\sqrt{\frac{1-\hat{p}}{n_s}} \leq \frac{z}{\sqrt{n_s}}$
- $\hat{E}$ is bounded by n_s only



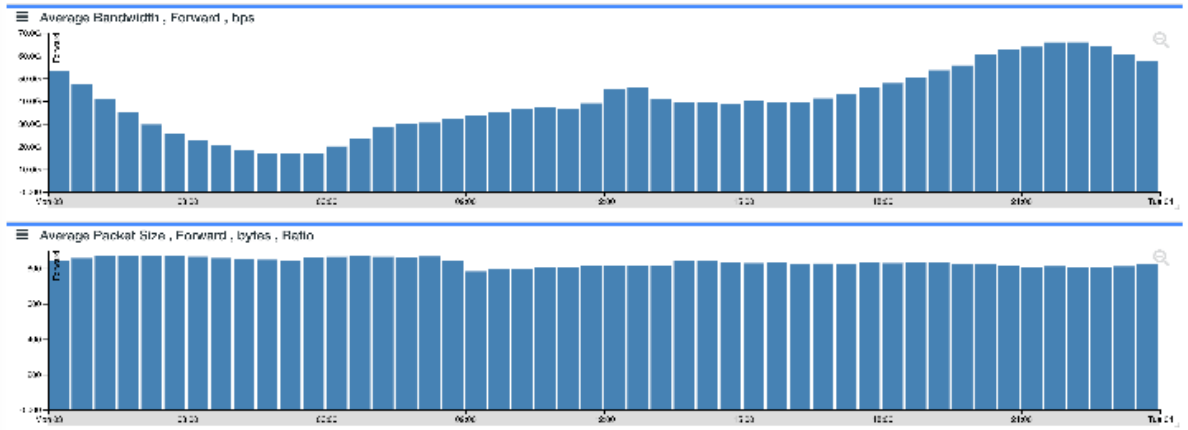
	confidence		
counted packets	99%	95%	90%
100	25.76%	19.60%	16.45%
1000	8.15%	6.20%	5.20%
2000	5.76%	4.38%	3.68%
3000	4.70%	3.58%	3.00%
10000	2.58%	1.96%	1.65%

Confident Granularity

- To have ~10% confidence interval, we need ~1000 sampled packets.
  - Assume the sampling rate is 1000:1, that's 1M packets before sampling
  - Assume the average packet size is ~800B, that's ~800MB traffic volume
  - 1 minute - 800M*8/60 = 106Mbps
  - 5 minutes - 800M*8/300 = 21.3Mbps
- Observation
  - **Time Granularity** and **Metric Granularity** is interchangeable

Challenge II - Volume

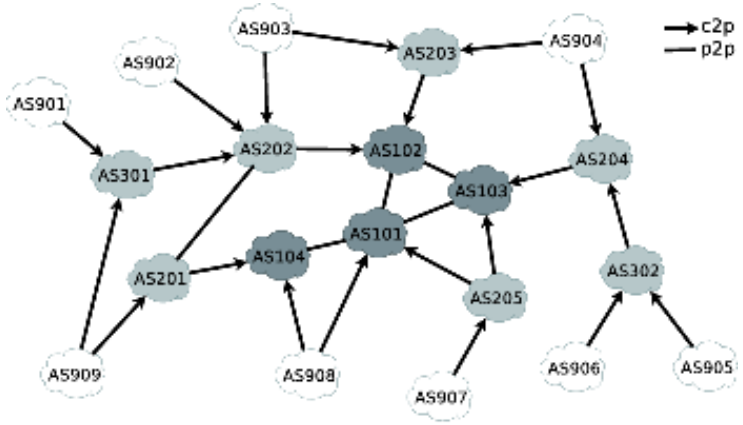
- Real numbers from a real router
  - ~400K mobile subscribers
  - ~40Gbps traffic in average
  - ~800B per packet
- At 1000:1 sampling rate
  - That's (40G/8/800)/1000 = 6250 records/sec
- CHT has 10M subscribers
  - That's 6250*(10M/400K)*86400 = 13.5 billions records/**day**
- Here're are numbers in 2019
  - 7.2 billion invoices/**year**, 2.0 billion mails/**year**.



Solution - Data Binning

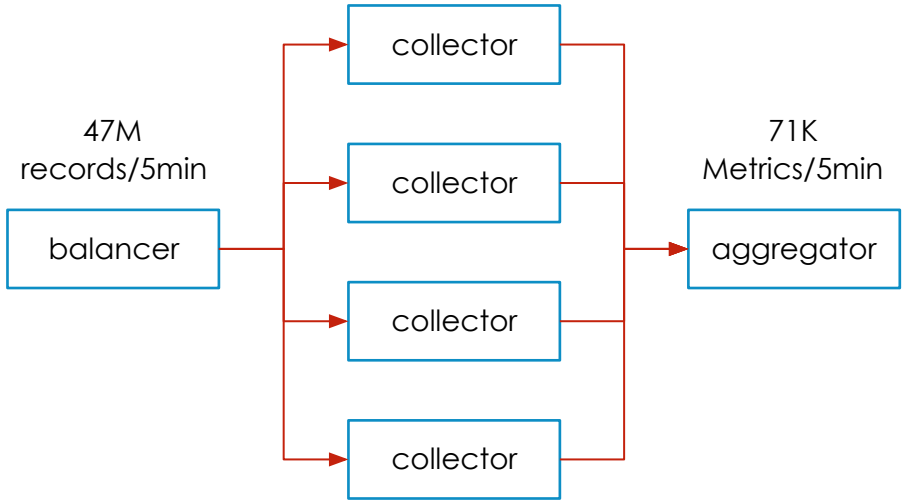
EX: Traffic from AS202 to each ASN

- Input - 47M records/5min
- Output - 71K metrics/5min



Converting records to metrics

- Distribute records to multiple nodes
- Compute metrics in each node
- Aggregate metrics of all nodes



Aggregable Metrics

- SUM, AVERAGE, MAX, MIN, …

Non-Aggregable Metrics

- PERCENTILE, DISTINCTCOUNT, …

Challenge III - Cardinality

Situation

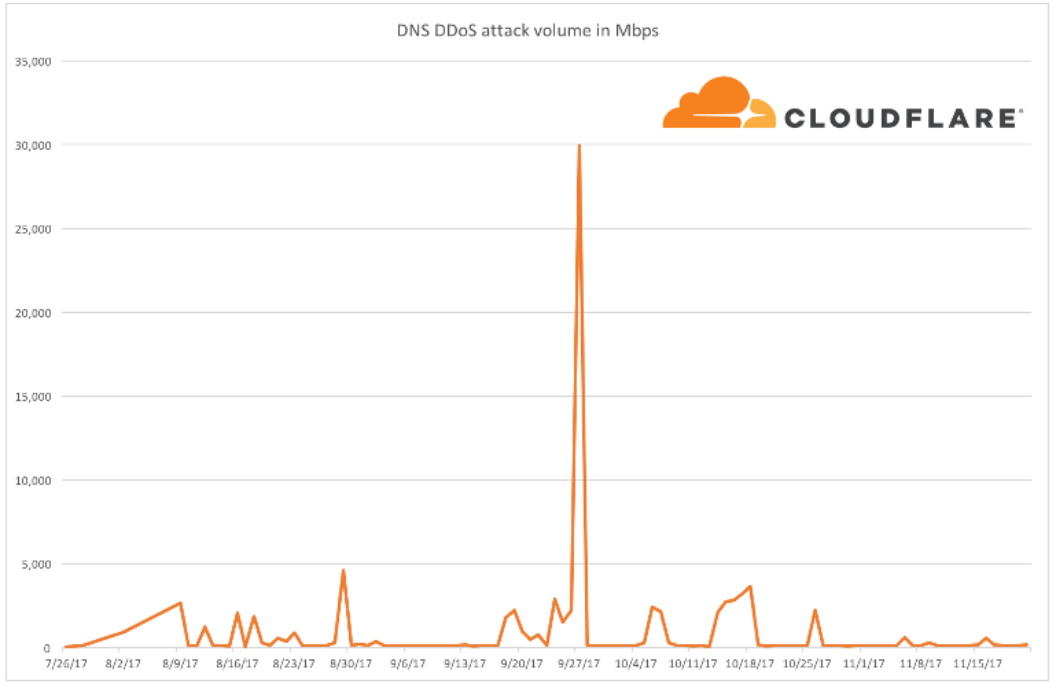
- We have a major DNS amplification attack. The scale is 50Gbps.

Requirement

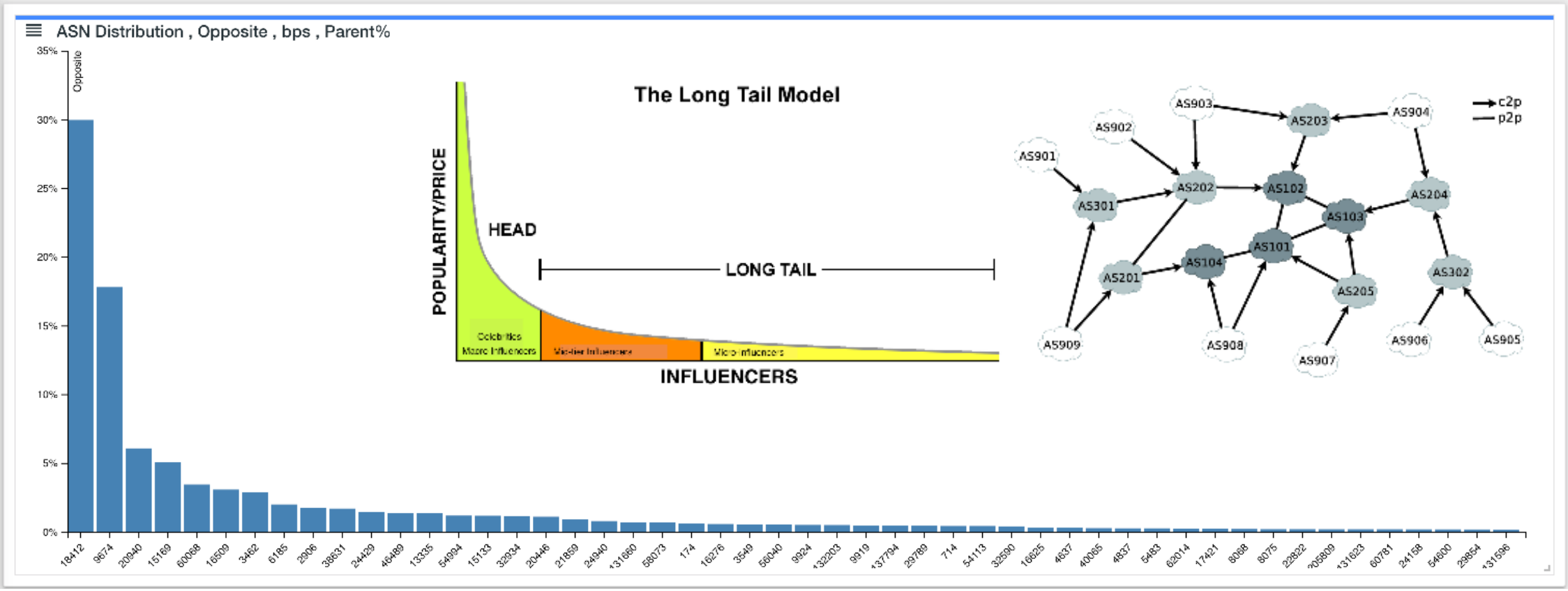
- Find the IP of reflective servers and block them.

Problem

- There are 4 billion IP addresses.



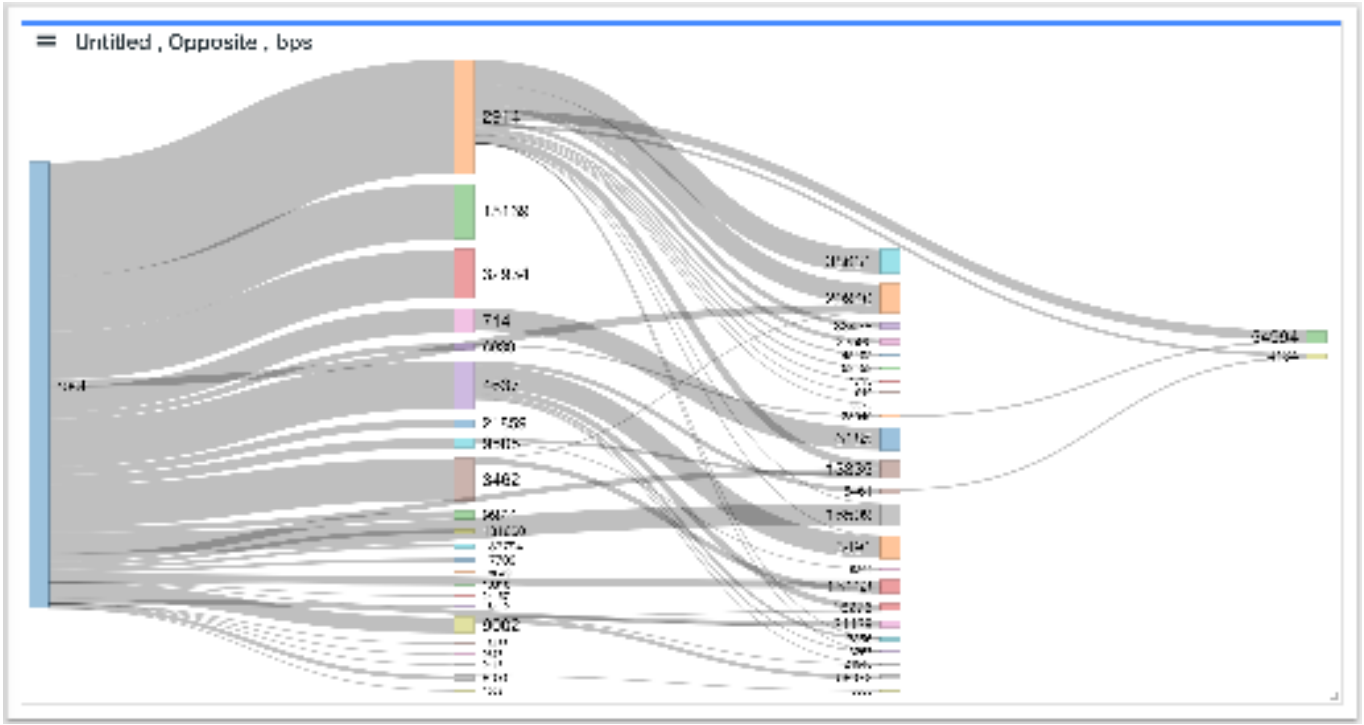
Long Tail Distribution



Real Site Examples

Discussion

- Which ASN to peering with next



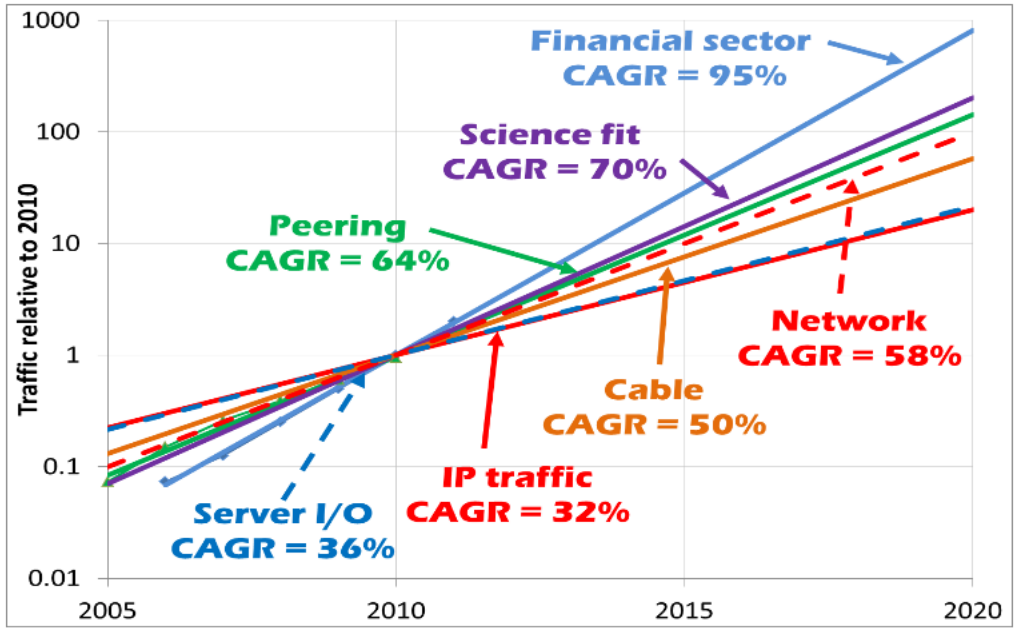
Solution - The Majority Algorithm

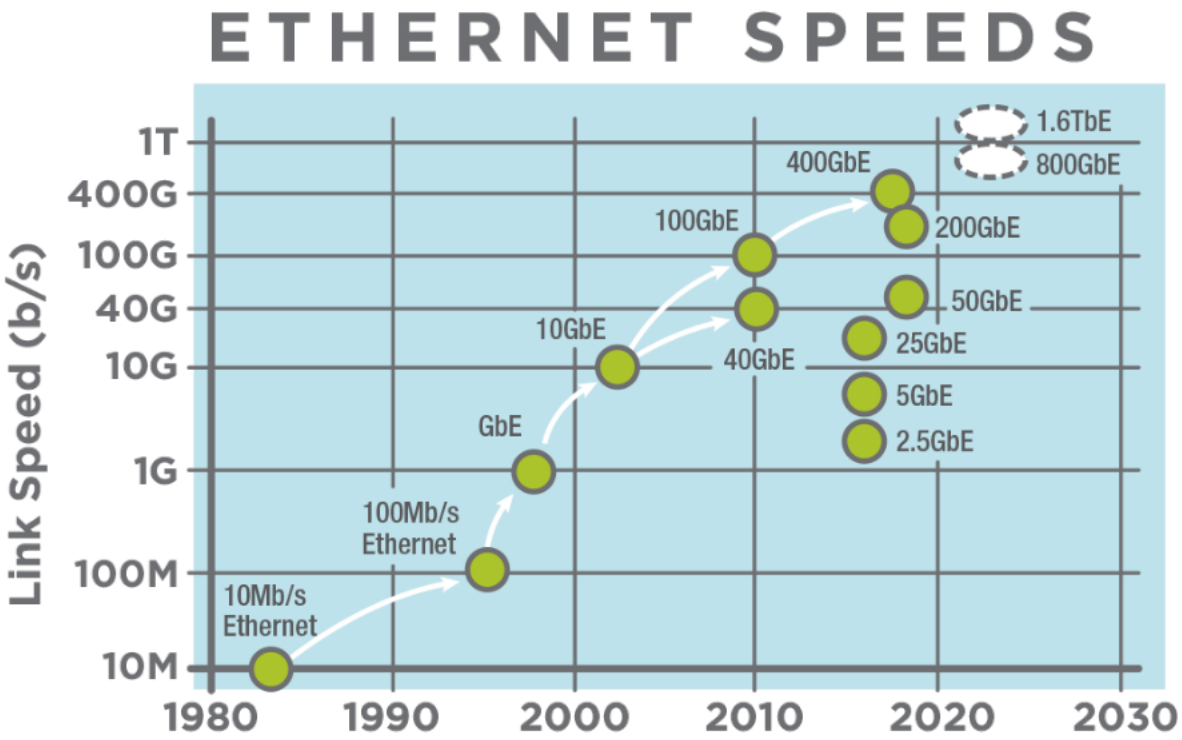
- Boyer–Moore majority algorithm
  - Finding the majority ($> \frac{1}{2}$) of a sequence of elements in linear time and O(1) space
  - Initialize an element m and a counter c with c = 0
  - For each element x of the input sequence:
    - if c = 0, then m = x and c = 1
    - else if m = x, then c = c + 1
    - else c = c − 1
  - Return m
- False positive
  - 2nd pass required to confirm majority

- Finding the majority ($> \frac{1}{N}$) of a sequence
  - Initialize an array of elements $\mathbf{m_{0..N-1}}$ and their counters $\mathbf{c_{0..N-1}}$ and a threshold $\mathbf{s} = 0$
  - For each element x of the input sequence:
    - if $m_i = x$ and $c_i > s$, then $c_i = c_i + 1$
    - else if $c_j \leq s$ for some j, then $m_i = x$ and $c_i = s + 1$
    - else $s = s + 1$
  - Return m
- The most important algorithm of traffic analysis
  - Finding **significant** elements in linear time and O(N) complexity

Enriching Data

The Trend

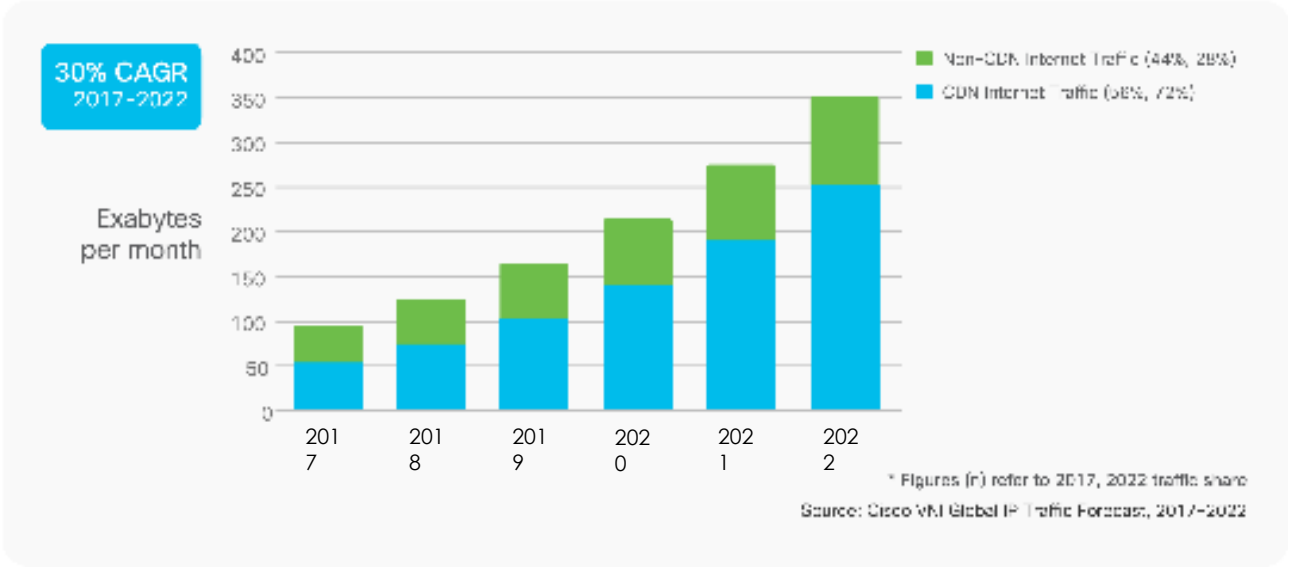




Discussion

- What we see from these two charts

Content DeliverY Network



Discussion

- Why Content Delivery Network

Tracing IP To Domain Name

Site - the logical service

- Site FQDN – download.skype.com
- OTT Provider – Microsoft
- OTT Service – Microsoft Skype

Host - the physical server

- Host FQDN – e4707.dspg.akamaiedge.net
- CDN Provider – Akamai

The challenge of cardinality

- 365M registered domain names
- 9700M hostnames
- 4000M allocated IP addresses

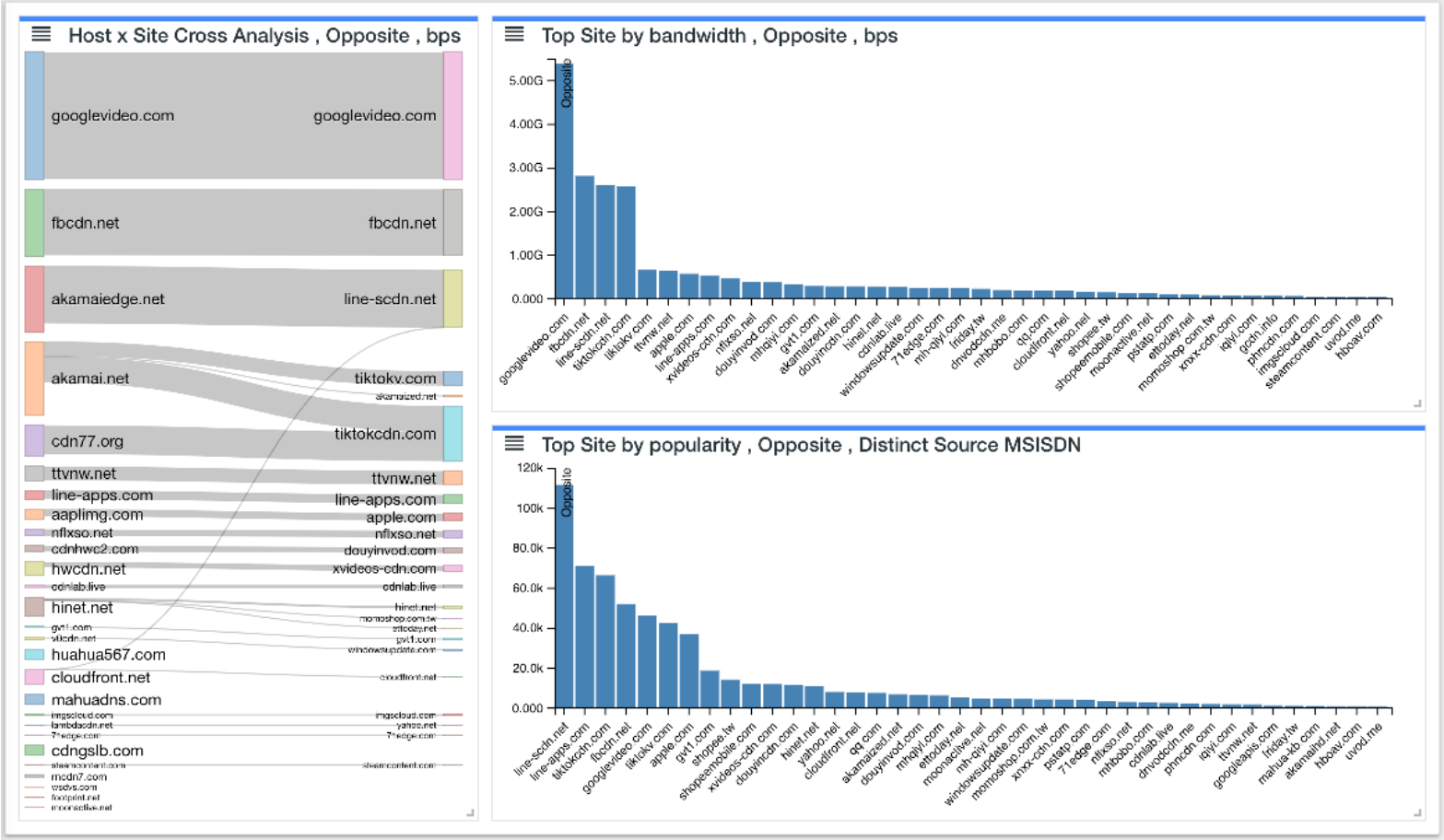
DNS record

```
download.skype.com.              300    IN CNAME  download.skype.com.edgekey.net.
download.skype.com.edgekey.net.  20542  IN CNAME  e4707.dspg.akamaiedge.net.
e4707.dspg.akamaiedge.net.       20     IN A      173.222.180.229
```

flow record

```
flow.addr.src:   173.222.180.229
flow.addr.dst:   10.20.89.215
flow.protocol:   6
flow.port.src:   443
flow.port.dst:   51294
…
_.dns.site_fqdn:      download.skype.com
_.dns_site_name:      skype.com
_.dns.ott_provider:   Microsoft
_.dns.ott_service:    Microsoft Skype
_.dns.host_fqdn:      de4707.dspg.akamaiedge.net
_.dns.host_name:      akamaiedge.net
_.dns.cdn_provider:   Akamai
```

Top Sites And Their Hosts



Quick facts

- 365M registered domains
- 9700M hostnames
- 4000M allocated IP

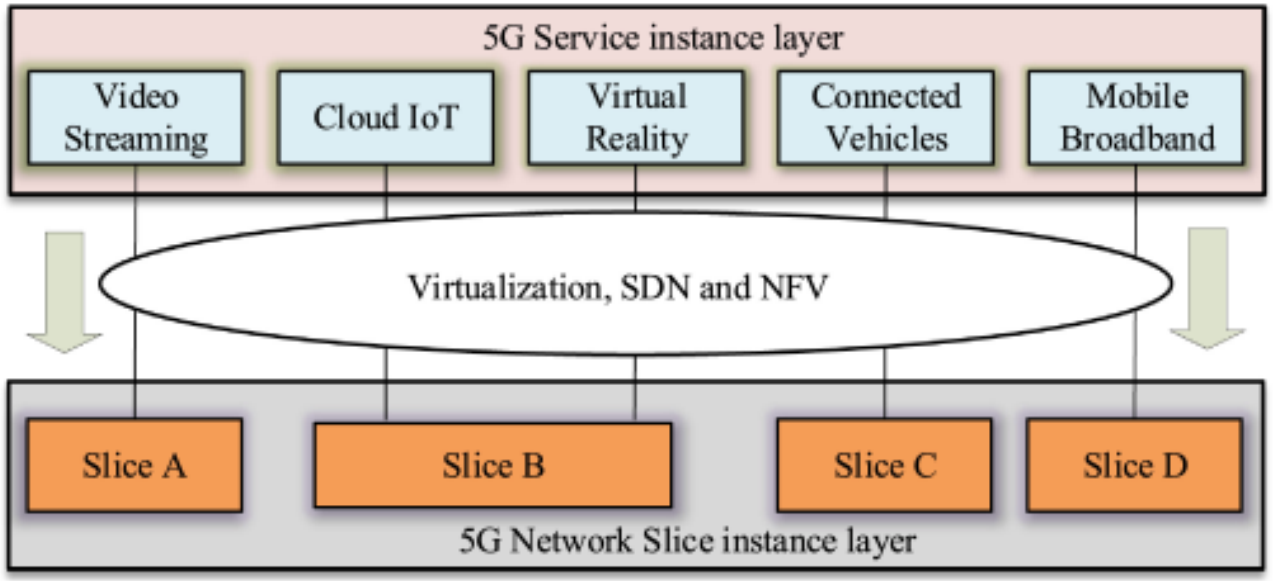
Long tail distribution

- Top 4 sites > 25%
- Top 1000 sites > 98%

Majority algorithm again

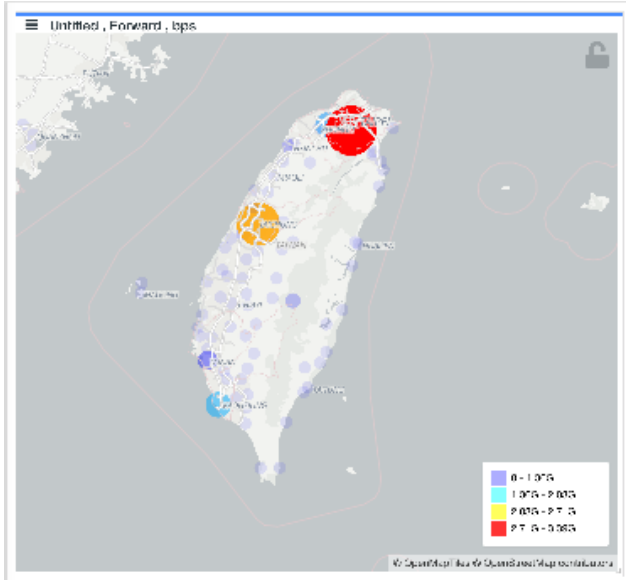
- N = 100,000 is enough

5G Network Slicing



Slicing applications

- Enhanced Mobile Broadband
- Critical Communications
- Enhanced Vehicular to Everything
- Massive Internet of Things



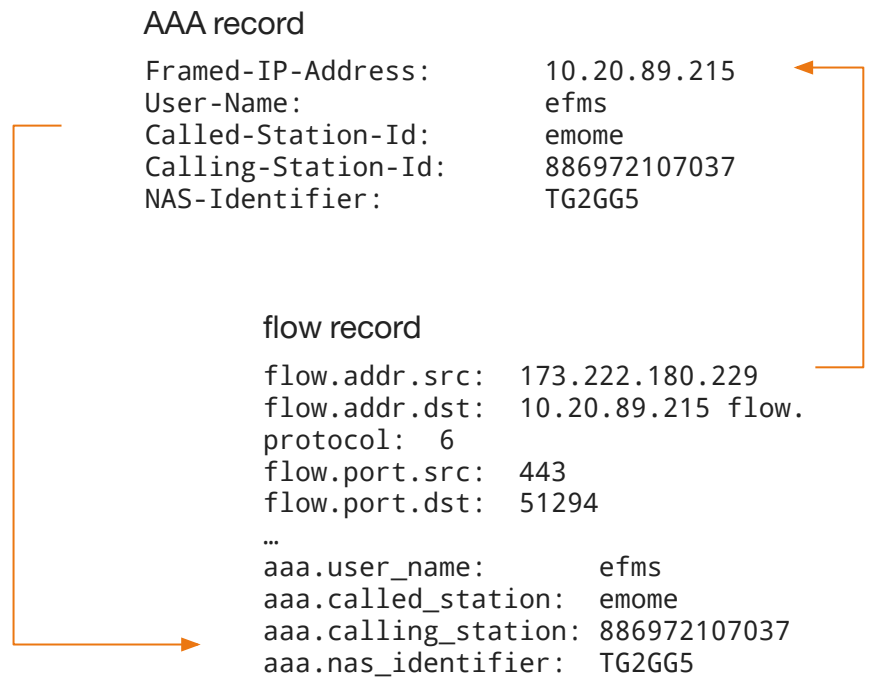
Tracing AAA

Example – Mobile Subscriber

- Framed-IP-Address – 10.20.89.215
- User-Name – efms
- Called-Station-Id – emome
- Calling-Station-Id – 886972107037
- NAS-Identifier – TG2GG5

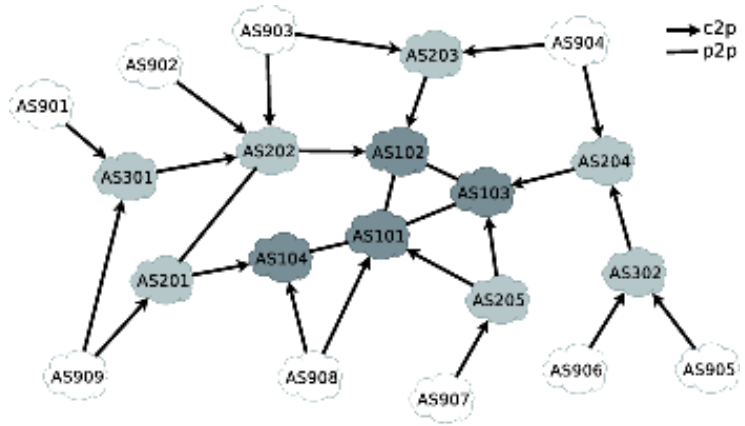
Example – Broadband Subscriber

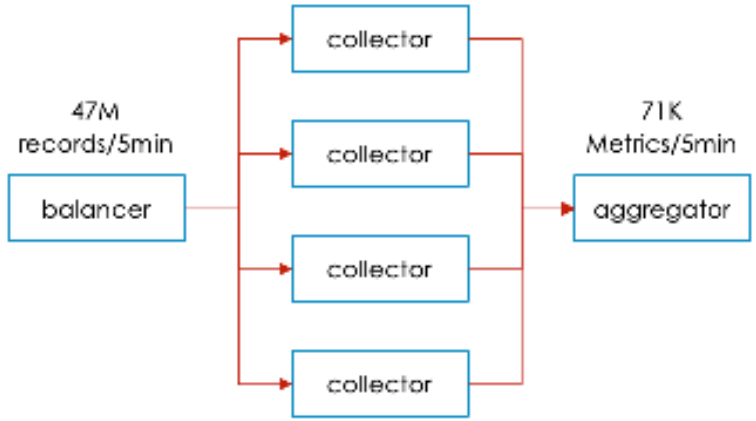
- Framed-IP-Address – 110.210.73.150
- User-Name – 13509820397@dg.cttgd
- Calling-Station-Id – f8:0f:41:24:c6:7d
- NAS-Identifier – GDZH-MS0420211510218 75fb33b025923



Ad-hoc Analytics

Limitation Of Data Bining

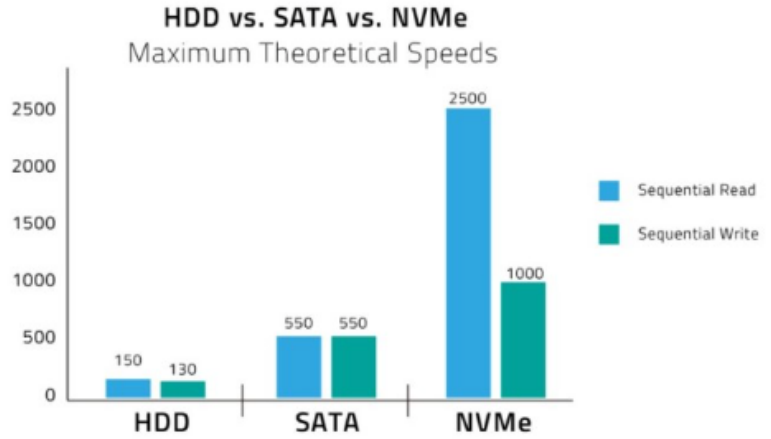


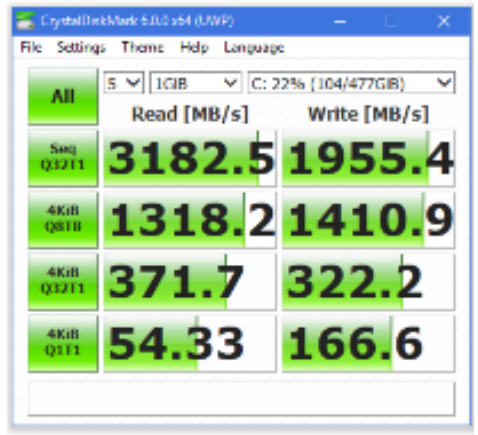


Static Report	Ad-Hoc Analytics
Automated and produced regularly	Produced once
Developed by an analyst	Run by a user
Reports on ongoing activity	Answers a specific question
More formatted with text and tables	More visual
Distributed to larger audience	Shared with smaller audience

The Challenge Of Volume

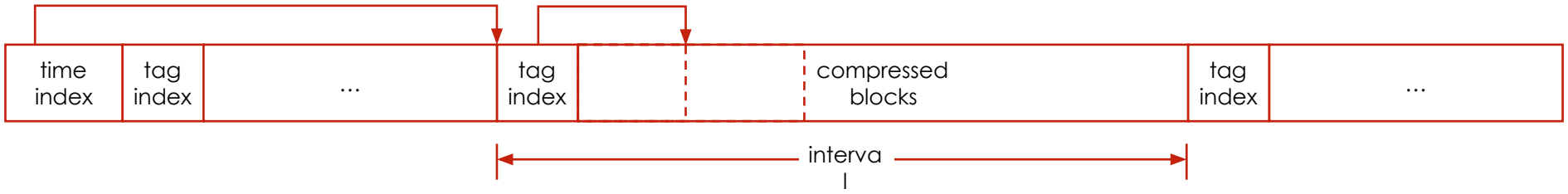
- CHT has 10M subscribers
  - That's 13.5 billions records per day
- Assume record size is 200 bytes
  - That's 2.7TB data per day
- To generate a **daily** report in 5 minutes
  - That's 2.7TB/300=9000MB data per second





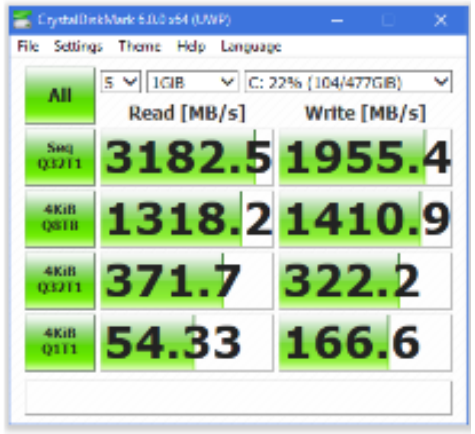
Time Series Database (TSDB)

- **Time Series Data**
  - Timestamp + Tag + Data
- **Time Series Database**
  - Indexed by timestamp and tag only
  - Efficent write - append only
  - Efficent read - sequential
  - Efficent purge - oldes
- **Very suitable for ad-hoc query of network traffic data**



The Challenge Of Timespan

- CHT has 10M subscribers
  - That's 2.7TB data per day
- To generate a **monthly** report in 5 minutes
  - Daily Report - 2.7TB/300s = 9 GB/s
  - Monthly Report - 2.7TB*30/300s = 270 GB/s
  - Annually Report - 2.7TB*365/300s = 3285 GB/s
- We need a smarter solution for long timespan report



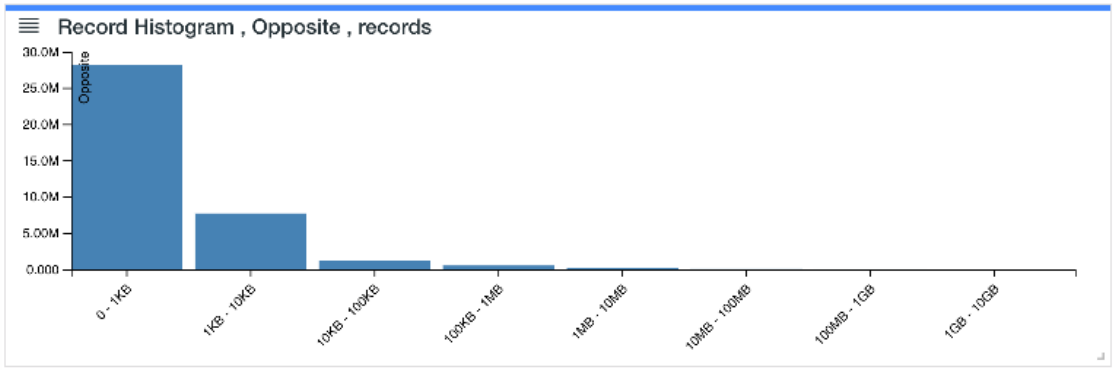
Solution - Resampling Records

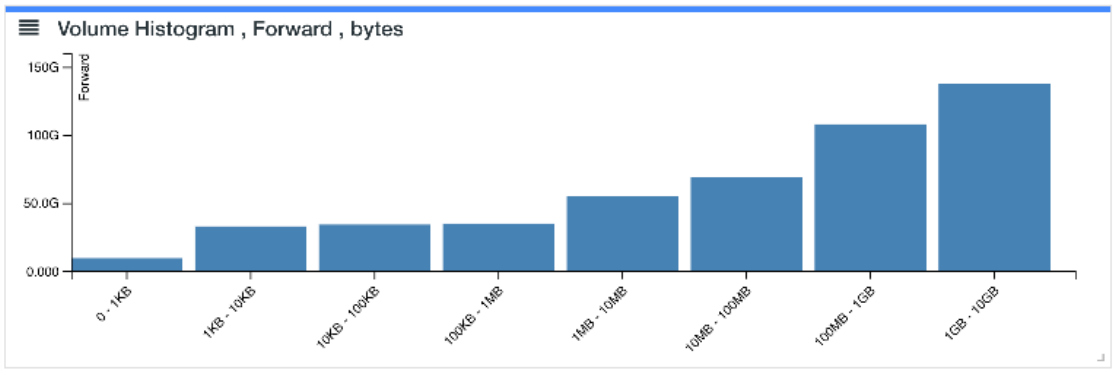
- **Confidence interval formula**
  - $\hat{E} = \frac{\hat{e}}{\hat{p}} = z\sqrt{\frac{1-\hat{p}}{n\times\hat{p}}} = z\sqrt{\frac{1-\hat{p}}{n_s}} \leq \frac{z}{\sqrt{n_s}}$

	confidence		
counted packets	99%	95%	90%
100	25.76%	19.60%	16.45%
1000	8.15%	6.20%	5.20%
2000	5.76%	4.38%	3.68%
3000	4.70%	3.58%	3.00%
10000	2.58%	1.96%	1.65%

- **Naive implementation**
  - Randomly resample one record every P records.
  - $\hat{M} = P\Sigma m_i$, m_i is metric of each resampled record.
- **Discussion**
  - What's wrong with this method

Resampling Records





Untitled

Actions	Filter	records opposite	records.parent%(records.same) opposite	bytes opposite	bytes.parent%(bytes.same) opposite
+	0 - 1KB	28.1M	74.58%	4.53G	0.23%
+	1KB - 10KB	7.68M	20.35%	25.4G	1.26%
+	10KB - 100KB	1.19M	3.15%	39.1G	1.95%
+	100KB - 1MB	507k	1.34%	175G	8.71%
+	1MB - 10MB	186k	0.49%	503G	25.04%
+	10MB - 100MB	27.6k	0.07%	746G	37.18%
+	100MB - 1GB	1.85k	0.00%	434G	21.60%
+	1GB - 10GB	40.0	0.00%	81.0G	4.03%

Records per page: 25 1-8 of 8

Not all flows are equal

- Mice flows dominate record volumes
- Elephant flows dominate traffic volumes

Per-flow resampling probability

- $\hat{M} = \Sigma \frac{m_i}{p_i}$, p_i is resampling probility of record

Summary

Conclusion

- Traffic Analysis is a kind of big data analysis.
  - It is too big in both time and space complexity.
  - Many big data algorithms don't work.
- We can constrain the complexity at certain cost.
  - data bining - query agility ⇔ storage efficiency
  - data sampling - time granlarity ⇔ metric confidence
  - majority algorithm - element visibility ⇔ space complexity
- The hitchhacker's guide to traffic analysis.
  - Sometimes there is no perfect solution, and good is enough.