Models and Tools for the High-Level Simulation of a Name-Based Interdomain Routing Architecture

Kari Visala, Andrew Keating
Helsinki Institute for Information Technology HIIT /
Aalto University School of Science

Rasib Hassan Khan
University of Alabama

17th IEEE Global Internet Symposium, Toronto
April 27, 2014
**Problem: PURSUIT Rendezvous Architecture**

- A hierarchical DHT [Canon] globally interconnecting *rendezvous networks* [DONA]
  - *Scopes* (containing publications) are advertised and previous query results are cached in the DHT nodes
  - Rendezvous networks are assumed to approximately evolve around neighboring stub ASes and Canon hierarchy to follow the structure of the AS graph
- Quantitative evaluation metrics
  - Distribution of latencies and overlay node and link resource usage, scalability, AS path stretch, determination of optimal cache size and number of overlay nodes


Problem: Approaches to Evaluation

- Complete architectures have many interfaces to the external world and require qualitative analysis, comparisons etc.
- Analytical results
  - Either too difficult or require simplifying assumptions in the case of complex, dynamic systems
- Prototyping and testing
  - PlanetLab overlay testbed: network conditions are not fully controllable, topology does not reflect the structure of the whole Internet, and the largest experiments may still not be feasible
  - NetFPGA, The Click Modular Router, OpenFlow..
- Simulation
  - Packet/router-level tools such as ndnSIM on top of ns-3: not scalable to Internet-wide scenarios
  - ⇒ High-level approximate models
HIGH-LEVEL SIMULATION: OUR DESIGN PRINCIPLES

1. Construct models around known invariants, that have been empirically validated under many scenarios [Floyd and Paxson]
   ▶ We also did not use algorithmically generated topologies that could leave out unnoticed features of the Internet

2. Tackle the scale by using aggregate models [Floyd and Paxson]

3. Parametrize the models for the uncertain variables

4. Modularize the different aspects of the simulation

5. Balance the level of detail of the different submodels

6. Use worst-case scenarios to increase confidence (datasets are incomplete etc.)

   ▶ High-level simulation can be thought as a hybrid between analytical results and a detailed simulation
   ▶ Some aspects can be abstracted safely and the difficult parts are simulated
   ▶ Relying on proofs leaves false negatives (too difficult to prove) and simulations allow some false positives (test cases cover the inputs only partially)

The network and traffic models can be simplified by assuming a specific application

- For example, we are only interested in the most important sources of control plane traffic
- Problem 1: The models may not be reused without modifications

PURSUIT is a clean-slate architecture

- Problem 2: The invariants true for the current Internet may not hold anymore
The global topology model should capture the Internet at least at the level of AS business relationships
- Categorized in the datasets into customer-to-provider and peer-to-peer
- Determine routing policies and rendezvous network formation
- PoP-level models are still works-in-progress

AS-level datasets contain mostly the same ASes and links but disagree about 34908 AS relationships
- UCLA [Zhang et al.] dataset combined multiple sources: BGP route monitors, ISP route servers/looking glasses, and Internet routing registries
- CAIDA [CAIDA] is another BGP-derived dataset

90% of the peering links may be missing because of the valley-free routing policies [Oliveira et al.]
- IXP [Augustin et al.] identifies peering links by using a combination of IXP databases, Internet topology datasets, and traceroute-based measurements
- We combined the UCLA and IXP datasets

[CAIDA] The CAIDA AS Relationships Dataset, November 2009
SUMMARY OF THE DATASETS

Table: Summary of CAIDA and UCLA datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Unique ASes</th>
<th>Customer-Provider Links</th>
<th>Peer-to-Peer Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAIDA</td>
<td>36,878</td>
<td>99,962</td>
<td>3,523</td>
</tr>
<tr>
<td>UCLA</td>
<td>38,794</td>
<td>74,542</td>
<td>65,784</td>
</tr>
</tbody>
</table>

Table: Hybrid UCLA*-IXP topology

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Unique ASes</th>
<th>Customer-Provider Links</th>
<th>Peer-to-Peer Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCLA*</td>
<td>42,703</td>
<td>76,083</td>
<td>78,264</td>
</tr>
<tr>
<td>IXP</td>
<td>2,974</td>
<td>0</td>
<td>40,076</td>
</tr>
<tr>
<td>Hybrid</td>
<td>43,018</td>
<td>75,421</td>
<td>105,772</td>
</tr>
</tbody>
</table>
PART OF THE AS RELATIONSHIPS DATASET VISUALIZED
**LATENCIES**

- Underlay latencies (numbers derived from the findings in [Zhang et al.])
  - 34 ms for inter-AS hops
  - 2 ms for intra-domain router hops
  - The number of intra-domain router hops between the nodes in the same AS is
    \[ 1 + \lceil \log D \rceil \]
    , where \( D \) is the degree of the AS. There is a relationship between the degree of the AS and its size [Tangmunarunkit et al.].

---


VALLEY-FREE POLICY ROUTING

- ASes export routes based on the algorithm given below and prefer customer routes to peering and peering to provider routes and secondarily choosing the shortest AS-level path
- ⇒ valley-free routes [Gao]
  - Every path concatenated from 0-n customer-to-provider links followed by 0-1 peering links and ending in 0-n provider-to-customer links

**Algorithm 1 Export routes**

1: for all $a \in AS$, $x \in \text{neighbors}(a)$ do
2: if $x \in \text{providers}(a) \cup \text{peers}(a)$ then
3: export all customer routes of $a$ to $x$
4: else if $x \in \text{customers}(a)$ then
5: export all routes of $a$ to $x$
6: end if
7: end for

AS Utility-Based Traffic Model

- ASes are modelled as points in a three dimensional utility space based on their business model [Chang et al.]
  - Each utility follows a Zipfian distribution with different exponents
  - The rank correlations between different utilities were measured

- The traffic is roughly categorized into the following three utilities:
  - Web hosting $U_{web}$
  - Residential access $U_{ra}$
  - Business access $U_{ba}$
    - $=\text{the cumulative transit provided by the AS (in case of multihoming, the utility is divided equally between all providers)}$

- We assume that the locations of rendezvous networks hosting the scope in a query are distributed to ASes proportional to $U_{web} + \alpha U_{ra}$, where $\alpha$ is a parameter

- Subscriptions originate from ASes proportional to the $U_{ra}$

APPLICATION TYPE-BASED TRAFFIC MODEL

- Application models are based on total *throughput* (parameter)
  - Projected to be 37,000 PB/month (14.6 TB/sec) in 2013 [Cisco]

- Two most popular types of traffic: web and P2P (BitTorrent)
  - *WebMix* and *P2PMix* parameters determine the share of each type of the throughput
  - Web traffic observed to be nearly 60% and P2P contributing about 14% [Maier et al.]

- Labovitz et al. observed that over 50% of interdomain traffic was originated by just 150 ASes [Labovitz et al.]
  - We model this spatial locality of generated traffic by parametrizing the share of each AS for each traffic type
  - Problem: Can conflict with the popularity distribution!


APPLICATION TYPE-BASED TRAFFIC MODEL (2)

- Web traffic parameters
  - WebReqsPerObj determines the number of rendezvous requests per page  
    - Empirical study shows median 12 embedded objects per page [Ihm and Pai]
  - WebObjSize  
    - Median page size of 133KB [Ihm and Pai]
  - Popularity distribution assumed to follow Zipf’s law [Breslau et al.]

- P2P traffic parameters
  - P2PReqsPerObj determines the number of rendezvous requests per unit time per object
  - P2PShareRatio is the percentage of objects republished after P2PShareDelay seconds after they are subscribed
  - Popularity distribution is Zipf-Mandelbrot [Hefeeda and Saleh]
  - We collected information about the content size of torrents by crawling The Pirate Bay:

<table>
<thead>
<tr>
<th></th>
<th>Min.</th>
<th>Max.</th>
<th>Q1</th>
<th>Median</th>
<th>Mean</th>
<th>Q3</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0B</td>
<td>641.40GB</td>
<td>93.33MB</td>
<td>350.47MB</td>
<td>1.05GB</td>
<td>883.39MB</td>
<td>3.60GB</td>
</tr>
</tbody>
</table>


Event Generation

- The traffic generator produces rendezvous request events, that are 4-tuples of type

  \( <\text{Timestamp, RequestType, RID, ASN}> \).

- The number of objects is huge
  - In 2008 Google reported that their web crawlers had indexed \(10^{12}\) unique URLs
  - \(\Rightarrow\) we cannot store per-object state

- Approximate Zipf/Zipf-Mandelbrot laws by using their continuous power law equivalents and use the constant time inverse transform method for generating samples by solving the integral (for Zipf)

  \[
  \int_{1}^{\infty} \frac{1}{z^\alpha} \, dz = \frac{z^{1-\alpha}}{1-\alpha} \bigg|_{1}^{x} = \frac{x^{1-\alpha}}{1-\alpha} - \frac{1}{1-\alpha}
  \]

- Adjusting for normalization, we define our invertible approximation of the Zipf distribution’s CDF as:

  \[
  F(x; \alpha, N) = \frac{\alpha - x^{1-\alpha}}{\alpha - N^{1-\alpha}}
  \]
**Event Generation (2)**

- We can now draw random popularity ranks for requests via the inverse

\[ F^{-1}(y; \alpha, N) = \left( (N^{1-\alpha} - \alpha) \left( y - \frac{\alpha}{\alpha - N^{1-\alpha}} \right) \right)^{\frac{1}{1-\alpha}} \]

where \( y \) is a random number from the uniform distribution in the interval \([0, 1]\) and \( N \) is the total number of objects.

- We precalculated \(10^6\) first values and use binary search to find the exact value.

- The percent error of the approximation is plotted below:
DEPLOYMENT MODEL

- The rendezvous networks were formed by
  1. Extracting a transit hierarchy from the AS topology (in case of multihoming we preferred smaller provider)
  2. Joining ASes in this tree top-down starting from tier-1 domains and offering a rendezvous network service at AS $x$ to its customer $y$ if the number of $y$'s transitive customers is smaller than predefined limit or $y$ and its customers do not host much more content than $x$ and its customers transitively

- The Canon hierarchy formation
  1. Each rendezvous network forms a Chord ring with enough nodes to store the hosted scopes
  2. By traversing the transit tree bottom-up by creating a new layer in the Canon when 5 sub-rings were transitively collected
The Canon overlay routing algorithm is fully simulated

Network failures were not modeled

The main limitation: *linearity* assumption for requests by simulating them independently

- Minimizes the amount of needed memory and allows us to generalize from a small sample size of requests

Each node contains $\beta k$ amount of storage for caching the most recent scope pointers queried via them.

- $k$ is the amount of storage used for storing scopes at the node

An analytical model of the cache performance in steady state

- If each node perfectly caches the $n$ most popular scopes, a scope with a popularity rank $pr$ is found cached at a node $x$ on level $a$ of the Canon hierarchy when

$$pr < \left( \frac{\beta \cdot s \cdot (A/N)}{(A_{x+1,a} - A_{x,a})modA} \right)$$

, where $A_{i,j}$ is the Canon node identifier of $i$th node at level $j$ and $N$ is the total number of nodes, $s$ is the total number of scopes and $A$ is the size of the whole address space.
The simulator environment is written in Python and was also ported into GNU Octave with optimizations for the experiments in [Rajahalme et al.]

Some example outputs of the simulator:

Figure: The graphs on the left show CDFs for the delay caused by the rendezvous phase with different popularity power-law exponents when the number of scopes is fixed to $10^{11}$. On the right, the effect of the node cache size on the rendezvous latency distribution is plotted.

More results in [Rajahalme et al.]

LESSONS LEARNED

- Efficiency and scalability are paramount in large simulations
  - Aggregate algorithms over an AS-level graph
  - Application-specific models can simplify the problem
- Too much detail in the submodels may cause unintentional correlations of variables
- Future work
  - Massive distributed simulation would remove the need for analytical model for caches and linearity assumption, but would probably be very slow
  - Flash crowds etc.
  - PoP-level topology
Thank You! Questions?