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
Email: kari.visala@hiit.fi, ake@issuu.com

Rasib Hassan Khan
University of Alabama
Birmingham AL, USA
Email: rasib@cis.uab.edu

***Abstract*—The deployment and operation of global network architectures can exhibit complex, dynamic behavior and the comprehensive validation of their properties, without actually building and running the systems, can only be achieved with the help of simulations. Packet-level models are not feasible in the Internet scale, but we are still interested in the phenomena that emerge when the systems are run in their intended environment.**

We argue for the high-level simulation methodology and introduce a simulation environment based on aggregate models built on state-of-the-art datasets available while respecting invariants observed in measurements. The models developed are aimed at studying a clean slate name-based interdomain routing architecture and provide an abundance of parameters for sensitivity analysis and a modular design with a balanced level of detail in different aspects of the model. In addition to introducing several reusable models for traffic, topology, and deployment, we report our experiences in using the high-level simulation approach and potential pitfalls related to it.

I. Introduction

Clean-slate Internet architectures, such as the recent ICN proposals [20], [9], [30], aim at potentially global deployment. Before any of these designs becomes popular, it is important to gain a good understanding of how they will behave if widely used in the Internet. For example, we would like to know how the amount of routing state scales in realistic scenarios, is the system resilient to distributed denial-of-service attacks, or the amount of resources needed to avoid congestion. As these architectures are based on a different principles than the current Internet, most of the previous experience may not be directly applicable anymore.

The Internet has evolved as a result of interactions between multiple stakeholders and technologies and we cannot always build its architecture around strong assumptions designed top-down to support reasoning about its properties. The dynamic aspects of the behavior of such complex systems are difficult or impossible to fully capture with theoretical artefacts, which necessitates at least partial black box approach to the evaluation. Experimentation with prototypes is typically too expensive to do in large scale. Using testbeds like PlanetLab [7] can produce valuable data, but network and node conditions are not fully controllable, the used topology does not reflect the structure of the whole Internet, and the largest experiments may still not be feasible.

This leaves us with the *simulation* and the detailed packet/router-level tools, such as ndnSIM [2] built on the ns-3 discrete-event simulator core, can be highly useful for small and medium-scale networks. We are, however, interested in what happens in Internet-wide scenarios and the enormous scale requires us to develop high-level models of network topology and traffic while abstracting the operation of the nodes. It might be possible to use a full-sized model of the evaluated system by scaling only the time dimension and running the simulation slower than real-time, but in our case, the total amount of state stored in the nodes, for example in the form of caches, makes this kind of approach infeasible. Therefore, we can only carefully approximate details in the simulation and concentrate on the selected aspects.

Even though algorithmically generated models might work for our purposes, the goal here is to use measured data as the basis of creating an Internet-like environment. This way we are not as likely to accidentally leave out unnoticed features of the Internet. We heed the warning by Floyd and Paxson [13] about the difficulties of simulating the Internet and try to

1) construct our models around known invariants, that have been empirically shown to be true in many scenarios,
2) parametrize the models for the uncertain variables,
3) tackle the scale by using aggregate models,
4) modularize different aspects of the simulation, and
5) balance the level of detail of the different submodels.

We acknowledge the difficulties in validating the models of the Internet based on incomplete datasets, but note that testing and experiments in general can only produce a negative definitive answer to a generic property. By adjusting the simulation parameters to reflect the worst-case scenario, we can increase our confidence in the results.

A. Application-driven high-level model

Our simulation design consists of four components, which are the *traffic model*, the *network topology model*, and the *application* and its *deployment/evolution model*. Their dependencies are shown in Fig. 1. We start by fixing the application abstraction, which models the network architecture or protocol, whose properties we are interested in. As discussed in [25], the service provided by the application affects the incentives of the network operators to deploy the system. Therefore, before the actual system can be run, we possibly need to first simulate the evolution of the deployment of the application in the network. Modularity of the different

978-1-4799-3088-3/14/$31.00 ©2014 IEEE

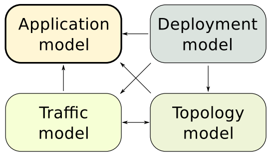

Fig. 1: Simulation modules and their dependencies

aspects of the simulation, such as the traffic generation and the topology, is desirable for reducing complexity and easy experimentation with different submodels, but as we will see in Section III, hidden couplings are easily introduced.

The application can also affect the incentives behind internetworking and routing. This is less and less restricted by the physical connectivity as the Internet is evolving from a tiered hierarchy of customer-provider relationships to a dense mesh of peering links [10]. One of the main difficulties of the validation of clean-slate network architectures stems from the fact that a switch of the communication paradigm may lead to profound changes in the traffic patterns and topology. For example, the exact operation of the ICN equivalents of web and P2P applications is unknown.

For scalability reasons, we have to limit the types of traffic modeled to the protocols relevant to the application in question. Currently, we are able to scale our simulation environment to handle *control plane* scenarios such as the global rendezvous architecture of PURSUIT [26], which we cover in section V as a case study of using our approach.

In this paper, we construct an AS-level topology and delay model of the Internet by studying and combining the existing datasets in section II and then develop two methods for the generation of aggregate application traffic following empirically observed object popularity distributions and introducing spatial locality to the generated requests in section III. We then explain how we implemented a high-level simulation environment incorporating policy routing in Python in section IV. Finally, we give a conclusion in section VI.

II. Topology Model

The global topology model should capture the Internet at least at the level of AS business relationships, because these are needed to determine the inter-domain routing policies and stakeholders in the deployment of new services on the most basic level. On the other hand, Roughan, et al. [28] warned against abstracting ASes into simple nodes without internal structure. Probably the best trade-off between practicality and details would be a graph that captured ASes on the level of *point of presence* (PoP) with their geographical location. However, while some research efforts have attempted to compile PoP-level Internet topologies [12], [27], we consider these to be works-in-progress.

The UCLA Internet Research Lab's AS-level topology [31] (hereinafter UCLA) combines adjacency information from numerous data sources to produce a graph of the interconnections between autonomous systems on the Internet. The topology is constructed with data from BGP route monitors (RouteViews and RIPE RIS), ISP route servers/looking glasses, and Internet Routing Registries. The Cooperative Association for Internet Data Analysis AS Relationship dataset [1] (hereinafter CAIDA), partly visualized in Fig. 2, is another BGP-derived AS-level topology dataset which augments the AS graph with per-link business relationships categorized as either *customer-to-provider* or *peer-to-peer*[1]. These are computed using heuristics adapted from methods proposed by Gao [15]. The UCLA AS-level topology dataset was also later annotated with the business relationships.

TABLE I: Summary of CAIDA and UCLA datasets

Dataset	Unique ASes	Customer-Provider Links	Peer-to-Peer Links
CAIDA	36,878	99,962	3,523
UCLA	38,794	74,542	65,784

Comparing the CAIDA and UCLA datasets, summarized in Table I, revealed that only a single AS is absent from UCLA's dataset but present in CAIDA's. Of CAIDA's inter-AS links, 329 do not appear in UCLA. However, after taking the AS relationship type into consideration, we found that the two datasets disagree about the AS relationships of 34,908 links. UCLA also contains 289 unclassified links in addition to those listed in the table.

Both of the aforementioned topologies suffer from one major shortcoming: the absence of most peering links. This can be attributed to the *valley-free* routing policy stemming from operator incentives: ASes do not announce routes containing peer-to-peer links to their providers or other peers. As such, a peering link between two ASes will only be captured in BGP-derived topologies if a route monitor is installed in either one of the ASes or one of their downstream customers. Fig. 3 illustrates how a peer-to-peer link can be invisible to a route monitor present at AS *A*.

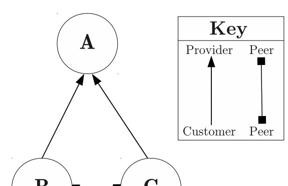

Fig. 3: Invisible peering link

Oliveira, et al. [24] investigated the accuracy of BGP-derived AS graphs, comparing them with complete connectivity data from a small number of ASes. They discovered that over time, any route monitor located at a tier-1 ISP

[1][11] presents an inference methodology and validation of the AS relationships and mentions a third type of AS relationship, *sibling-to-sibling*, but these links are very rare.

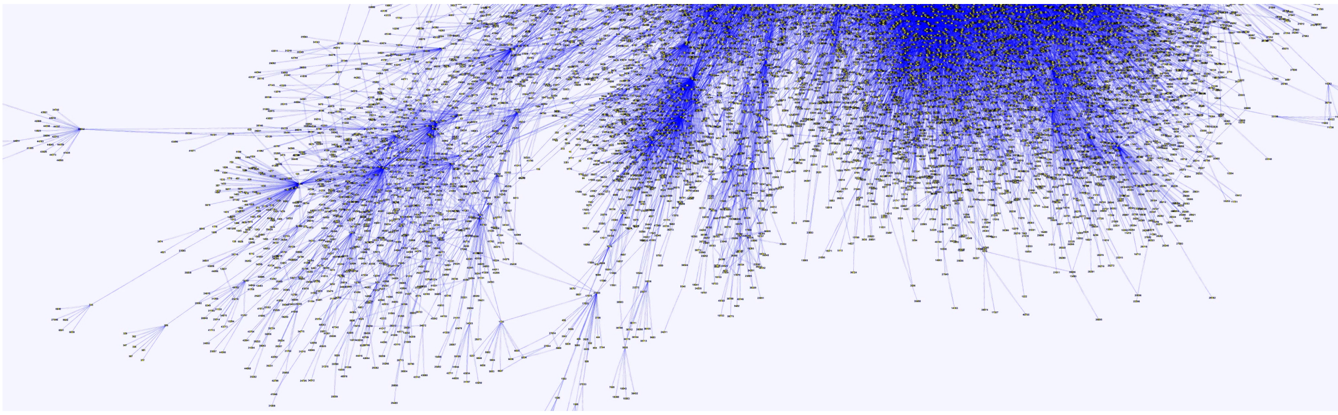

Fig. 2: Part of the CAIDA AS relationships dataset visualized

is eventually able to capture every customer-to-provider link in the Internet's AS-level topology. However, the authors estimated that as many as 90% of peer-to-peer links may be missing from existing AS topology datasets. Dhamdhere, et al. [10] noted that *Internet exchange points* (IXPs) have experienced significant growth as of late predicting that the Internet is evolving from a tiered hierarchy of customer-provider links to a dense mesh of peering links. As more operators adopt direct peering relationships, the importance of capturing peering links in the AS-level topology increases.

In [26] this deficiency was addressed by generating 900% additional peer-to-peer links using simple rules. The shortcoming of this method is that it can generate links that do not exist in the Internet. Augustin, et al. [3] attempted to identify the missing peering links by mapping the members of Internet exchange points through a combination of IXP databases, Internet topology datasets, and traceroute-based measurements. Their resulting dataset (hereinafter IXP) contained 40,076 high-confidence peering links.

Given these observations, we compiled a hybrid AS-level topology which unites the UCLA*[2] and IXP datasets using a similar approach used in [17]. The hybrid UCLA*-IXP dataset contains all classified UCLA* links (552 unclassified links discarded), in addition to all high-confidence IXP links. In the cases where links existed in both datasets but the AS relationships differ, we preferred the IXP categorization over UCLA*. A summary of the two datasets and their resulting union is presented in Table II. More details of our analysis of the AS-level topology datasets can be found in [21].

TABLE II: Hybrid UCLA*-IXP topology

Dataset	Unique ASes	Customer-Provider Links	Peer-to-Peer Links
UCLA*	42,703	76,083	78,264
IXP	2,974	0	40,076
Hybrid	**43,018**	**75,421**	**105,772**

[2]We used a more recent version of the UCLA dataset captured on May 6, 2012

A. Delay Model

In [26], we used the following delay values to model routing latency: 34ms for inter-AS node to node hops and 2ms for intra-domain router hops [32]. The number of intra-domain router hops used between two nodes residing in the same AS was $1 + \lfloor \log D \rfloor$ where D is the degree of the AS. This is based on findings in [29], where a strong correlation between the number of routers in an AS and the degree of the AS is found (May 2001 coefficient of correlation was 0.959) and the assumption that the routing topology is efficiently designed.

III. Traffic Model

Although several Internet traffic traces have been made publicly available, it is not advisable to blindly generate simulation traffic based on these traces, since much of the Internet's traffic uses adaptive congestion control, resulting in packet traces which are specific to the network conditions at the time of the capture [13]. We need to make assumptions about the applications generating the traffic as their implementations depend on the service provided by the measured clean-slate network architecture.

A. AS Utility-based Model

In [26] we used the three dimensional utility model for ASes introduced in [6] as the basis of aggregate sources for *publish* and *subscribe* events. The three utilities are assigned to each AS with the algorithm given in [6] by first calculating the *business access* U_{ba} utility from the topology data by summing the cumulative transit provided by each AS[3]. After the U_{ba} is known, the two other utilities, *residential access* U_{ra} and *web hosting* U_{web}, are randomly generated in a way that their values respect the Zipfian distributions with correct exponents and the pairwise rank correlation between utilities of ASes coincide with measurement data.

We then assumed that subscriptions originate randomly from ASes with a probability proportional to the U_{ra} of the AS.

[3]In case of multihoming, the utility generated by the client is divided equally between all of its providers.

The target *scopes*[4] of subscriptions are distributed to ASes proportional to $U_{\text{web}} + \alpha U_{\text{ra}}$, where α is a parameter. The popularity of scopes, which is needed in the model to simulate caching, is assumed to follow a Zipfian distribution in line with several studies in content delivery networks and other such services [5], [16].

B. Application Type-based Model

In our second traffic model we capture three characteristics: 1. application models, 2. object popularity, and 3. spatial locality. This model is built around the total *Throughput* parameter, which includes all traffic transferred on the Internet (7.84 terabytes per second in 2010 according to [8]). We chose only two applications, Web and P2P, to represent all traffic based on the findings in [23]. In [23] web traffic was found to be dominant, representing nearly 60% of all traffic, while P2P contributed about 14%.

Our model for web traffic is defined by three workload generator parameters, *WebMix* and *WebObjSize* and *WebReqsPerObj*. The *WebMix* parameter defines the percentage of the total traffic volume which is represented by web traffic. *WebObjSize* relates the number of requests to the traffic volume invariant and *WebReqsPerObj* can be used to determine the number of control plane operations such as DNS queries in the Internet or rendezvous requests in the PURSUIT architecture. An empirical study shows that the median of embedded objects per page is 12 and the median page size of 133kB [19]. The web object popularity is assumed to have a distribution following Zipf's law [4].

Our P2P traffic model is based on BitTorrent, which is the most prevalent P2P application currently. We developed a crawler, that collected information about the content size of torrents shared in the most popular BitTorrent discovery site, The Pirate Bay. The results are shown in Table III.

The throughput of P2P traffic in the Workload Generator is determined by the *P2PMix* parameter. The *P2PObjSize* and *P2PReqsPerObj* parameters determine the number of control plane requests which are generated per unit time. The percentage of P2P objects which are re-published after being subscribed to is determined by the *P2PShareRatio* parameter. The re-publication occurs after a delay of *P2PShareDelay* seconds. For the P2P object popularity we used Zipf-Mandelbrot distribution [18]. The list of all parameters and a detailed description of our second traffic model can be found in [21].

Labovitz, et al. [22] observed that as of July 2009, over 50% of all interdomain traffic was originated by just 150 ASes. In order to model spatial locality of generated traffic, our model allows to configure the share of each AS of the total traffic for each traffic type [21]. However, this created an unexpected coupling with the popularity distribution for publications as it may not be possible to arrange the most popular objects to ASes in such a way that the shares of ASes are respected.

[4]each containing a set of publications

C. Generation of Events

We implemented a traffic generator based on our second traffic model. It produces PURSUIT rendezvous request events, which are 4-tuples of type

$$< Timestamp, RequestType, RId, ASN > .$$

The $Timestamp$ represents the amount of time in milliseconds after the start of the simulation that a rendezvous request is made. The $RequestType$ field determines whether the event is a publish or subscribe request. RId is the 256-bit rendezvous identifier and the ASN field represents the autonomous system where the request originated.

The generated pseudorandom requests need to follow the Zipf and Zipf-Mandelbrot laws for object popularity. Because the number of objects is huge[5], we cannot store per-object state. In order to use the *inverse transform method* for generating samples, we approximated the CDF of the Zipf distribution by solving the integral

$$\int_1^x \frac{1}{z^\alpha}\,\mathrm{d}z = \frac{z^{1-\alpha}}{1-\alpha}\bigg|_1^x = \frac{x^{1-\alpha}}{1-\alpha} - \frac{1}{1-\alpha} \tag{1}$$

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

$$F(x;\alpha,N) = \frac{\alpha - x^{1-\alpha}}{\alpha - N^{1-\alpha}} \tag{2}$$

This approximation allows us to compute Zipf CDF values in constant time. Moreover, since the approximation of the CDF is invertible, we can draw random popularity ranks for requests via the inverse

$$F^{-1}(y;\alpha,N) = \left((N^{1-\alpha} - \alpha)\left(y - \frac{\alpha}{\alpha - N^{1-\alpha}}\right)\right)^{\frac{1}{1-\alpha}} \tag{3}$$

, where y is a random number from the uniform distribution in the interval $[0, 1]$ and N is the total number of objects. Although our approximation provides a significant increase in efficiency, this comes at the cost of accuracy. The percent error between the actual Zipf CDF and our approximation is plotted in Fig. 4. Since the approximation error is greatest for the most popular objects, we precalculated 10^6 first values of the Zipf CDF and use binary search to find the exact value in case the popularity rank falls under this pivot value. We used similar method for the Zipf-Mandelbrot distribution [21].

IV. Simulator

The simulator environment is writen in Python and was also ported into GNU Octave with slight modifications for the experiments in [26], where we first simulated the deployment of the rendezvous nodes in ASes and then ran batches of 30000 rendezvous requests. Without the policy-compliant path computation optimizations implemented in the Octave version, the simulator could handle approximately 1000 requests/minute in a modern laptop.

[5]In 2008 Google reported that their web crawlers had indexed 10^{12} unique URLs http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html

TABLE III: BitTorrent content size data gathered by our crawler

Min.	Max.	Q1	Median	Mean	Q3	Std. Dev.
0B	641.40GB	93.33MB	350.47MB	1.05GB	883.39MB	3.60GB

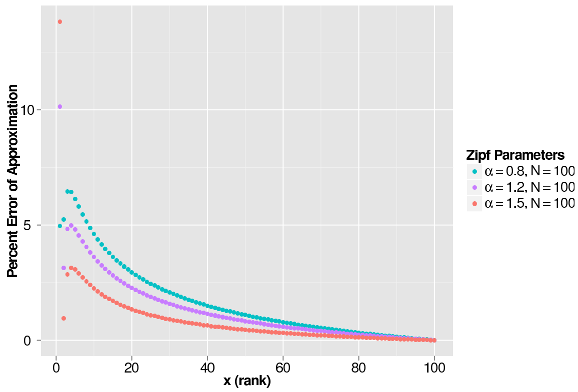

Fig. 4: Approximation error of the Zipf CDF

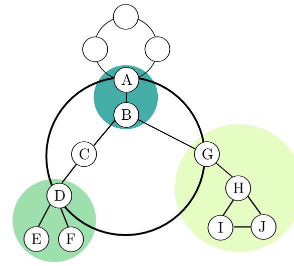

Fig. 5: Rendezvous nodes residing in ASes A, D, and G belong to the topmost level in the Canon hierarchy.

The Canon DHT [14] overlay routing algorithm is fully simulated except that the success of cache operations is based on the high-level cache model described in Section V and network failures are not modeled. The main limitation of the system is that it uses a *linear* model for requests by simulating them independently without any interaction between requests. The whole operation of the rendezvous system is thus modeled as a superposition of the requests. This both minimizes the amount of needed memory and allows us to generalize from a small sample size of requests.

Algorithm 1 Export routes

for all $a \in AS, x \in neighbors(a)$ **do**
  if $x \in providers(a) \cup peers(a)$ **then**
    export all customer routes of a to x
  else if $x \in customers(a)$ **then**
    export all routes of a to x
  end if
end for

The simulator implements a policy-based inter-domain routing algorithm. We assume that ASes export routes based on the Algorithm 1 and prefer customer routes to peering and peering to provider routes and secondarily choosing the shortest AS-level path.

V. An Example: Global Rendezvous

In [26] our tool with the CAIDA topology and the AS utility-based traffic model was used to evaluate the inter-domain rendezvous system of the PURSUIT ICN architecture. The rendezvous system consists of Canon hiearchical DHT [14] overlay interconnect joining *rendezvous networks* spanning multiple ASes as shown in Fig. 5.

The rendezvous networks were formed by 1. extracting the transit hierarchy from the AS topology and 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. We formed the Canon hiearchy by first assuming that each rendezvous network forms a Chord ring with enough nodes to store the amount of scopes hosted in the rendezvous network. Then we traversed the transit tree bottom-up by creating a new layer in the Canon hierarchy at ASes that have transitively at least 5 sub-rings.

We determined the number of rendezvous nodes only based on the amount of content they store. If the capacity of the overlay is not enough for the number of requests, the nodes can easily be replaced by multiple parallel nodes.

In addition to storing scope pointers, each node contains βk amount of storage for caching the most recent scope pointers queried via them. Here k is the amount of storage used for storing scopes at the node and the parameter β describes the relative amount of memory used for publishing and subscribing. An analytical model of the cache performance in steady state was used to estimate the relevant hit probabilities. Assuming that 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

$$pr < \left(\frac{\beta \cdot s \cdot (A/N)}{(A_{x+1,a} - A_{x,a}) mod A}\right) \tag{4}$$

where $A_{i,j}$ is the Canon node identifier of ith 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. This assumes that hashed identifiers are evenly distributed.

As a result of this work, the following simulation results were obtained: the number and locations of the nodes needed, the latency distributions of the rendezvous operations, the consumption of network resources measured as the inter-domain *stretch* distribution of the paths used by the requests, caching effiency, and the relative node load distribution. In Fig. 6, we show two example graphs produced by the simulator.

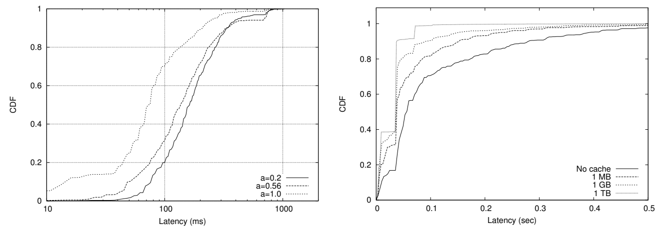

Fig. 6: The graphs on the left show CDFs for the delay caused by the rendezvous phase with different popularity power-law exponents when the number of scopes is fixed to 10^{11}. On the right, the effect of the node cache size on the rendezvous latency distribution is plotted.

VI. Conclusions and Future Work

Efficiency and scalability are paramount in large simulations. In order to be able to just sample the PURSUIT control plane traffic in a single node, we had to use aggregate algorithms over an AS-level graph and analytical model for the per-node caches. Simulating the Internet is hard and it is important to understand the limitations of the approximate models based on imperfect datasets and analyze results critically. Also too much detail can lead to overly complex models with unintentional correlation of variables. However, when the abstraction in the models is justified analytically, high-level simulation can become a marriage of theory and experimentation yielding new knowledge.

We have only touched the problem of high-level simulation of future Internet architectures. We aim to produce evidence towards the validity of the combined model in the future by simulating an existing system such as DNS and comparing the results to measurement data. The topology model can be made independently more fine-grained to take into account PoP-level structure and geographical information in latency calculations when better dataset will be available. The traffic model should eventually incorporate dependencies between communication events to capture essential phenomena such as flash crowds.

References

[1] The CAIDA AS Relationships Dataset, november 2009.
[2] A. Afanasyev, I. Moiseenko, and L. Zhang. ndnSIM: NDN simulator for NS-3. Technical Report NDN-0005, October 2012.
[3] B. Augustin, B. Krishnamurthy, and W. Willinger. IXPs: Mapped? In *Internet Measurement Conference (IMC)*, pages 336–349. ACM, 2009.
[4] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *INFOCOM'99*, volume 1, pages 126–134. IEEE, 1999.
[5] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. In *ACM SIGCOMM IMC'07*, Oct. 2007.
[6] H. Chang, S. Jamin, M. Mao, and W. Willinger. An Empirical Approach to Modeling Inter-AS Traffic Matrices. In *ACM SIGCOMM IMC'05. Proceedings*, pages 139–152, 2005.
[7] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. PlanetLab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*, 33(3):3–12, July 2003.
[8] Cisco. Cisco Visual Networking Index: Forecast and Methodology, 2010-2015. Technical report, Cisco, June 2011.
[9] C. Dannewitz, J. Golić, B. Ohlman, and A. Bengt. Secure Naming for a Network of Information. In *13th IEEE Global Internet Symposium 2010*, 2010.
[10] A. Dhamdhere and C. Dovrolis. The internet is Flat: Modeling the Transition from a Transit Hierarchy to a Peering Mesh. In *ACM CoNEXT*, page 21. ACM, 2010.
[11] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, k. claffy, and G. Riley. AS Relationships: Inference and Validation. *SIGCOMM Computer Communication Review*, 37(1):29–40, January 2007.
[12] D. Feldman, Y. Shavitt, and N. Zilberman. A Structural Approach for PoP Geo-Location. *Computer Networks*, 56(3):1029–1040, February 2012.
[13] S. Floyd and V. Paxson. Difficulties in Simulating the Internet. *IEEE/ACM Transactions on Networking (TON)*, 9(4):392–403, August 2001.
[14] P. Ganesan, K. Gummadi, and H. Garcia-Molina. Canon in G Major: Designing DHTs with Hierarchical Structure. In *ICDCS'04*, pages 263–272. IEEE Computer Society, 2004.
[15] L. Gao. On Inferring Autonomous System Relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9(6):733–745, Dec. 2001.
[16] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. YouTube Traffic Characterization: A View From the Edge. In *ACM SIGCOMM IMC '07*, pages 15–28, New York, NY, USA, Oct. 2007. ACM.
[17] P. Gill, M. Schapira, and S. Goldberg. Let the Market Drive Deployment: a Strategy for Transitioning to BGP Security. In *ACM SIGCOMM'11*, SIGCOMM '11, pages 14–25, New York, NY, USA, August 2011. ACM.
[18] M. Hefeeda and O. Saleh. Traffic Modeling and Proportional Partial Caching for Peer-to-Peer Systems. *IEEE/ACM Transactions on Networking*, 16(6):1447–1460, Dec. 2008.
[19] S. Ihm and V. S. Pai. Towards Understanding Modern Web Traffic. In *Internet Measurement Conference (IMC)*, pages 295–312. ACM, 2011.
[20] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard. Networking Named Content. In *ACM CoNEXT 2009*, 2009.
[21] A. Keating. Models for the Simulation of a Name-Based Interdomain Routing Architecture. Master's thesis, Aalto University, 2012.
[22] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. Internet Inter-Domain Traffic. In *SIGCOMM'10*, 2010.
[23] G. Maier, A. Feldmann, V. Paxson, and M. Allman. On Dominant Characteristics of Residential Broadband Internet Traffic. In *Internet Measurement Conference (IMC)*, pages 90–102. ACM, 2009.
[24] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang. In Search of the Elusive Ground Truth: The Internet's AS-level Connectivity Structure. *SIGMETRICS Perf. Eval. Rev.*, 36(1):217–228, 2008.
[25] J. Rajahalme, M. Särelä, P. Nikander, and S. Tarkoma. Incentive-Compatible Caching and Peering in Data-Oriented Networks. In *ReArch'08*. ACM, 2008.
[26] J. Rajahalme, M. Särelä, K. Visala, and J. Riihijärvi. On name-based inter-domain routing. *Computer Networks Journal: Special Issue on Architectures and Protocols for the Future Internet*, 55:975–986, 2011.
[27] A. H. Rasti, N. Magharei, R. Rejaie, and W. Willinger. Eyeball ASes: From Geography to Connectivity. In *IMC'10*, 2010.
[28] M. Roughan, W. Willinger, O. Maennel, D. Perouli, and R. Bush. 10 Lessons from 10 Years of Measuring and Modeling the Internet's Autonomous Systems. *IEEE Journal on Selected Areas in Communications*, 29(9):1810–1821, October 2011.
[29] H. Tangmunarunkit, J. Doyle, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Does AS Size Determine Degree in AS Topology? *SIGCOMM Comput. Commun. Rev.*, 31(5):7–8, 2001.
[30] D. Trossen, G. Parisis, K. Visala, B. Gajic, J. Riihijärvi, P. Flegkas, P. Sarolahti, P. Jokela, X. Vasilakos, C. Tsilopoulos, and S. Arianfar. PURSUIT Deliverable D2.2: Conceptual Architecture: Principles, patterns and sub-components descriptions. Technical report, PURSUIT, 2011.
[31] B. Zhang, R. Liu, D. Massey, and L. Zhang. Collecting the Internet AS-level Topology. *ACM SIGCOMM Computer Communication Review*, 35(1):53–61, 2005.
[32] B. Zhang, T. E. Ng, A. Nandi, R. Riedi, P. Druschel, and G. Wang. Measurement-Based Analysis, Modeling, and Synthesis of the Internet Delay Space. In *ACM SIGCOMM IMC'06*, pages 85–98. ACM, May 2006.