Models for the Simulation of a Name-Based Interdomain Routing Architecture

ANDREW KEATING

KTH Information and
Communication Technology

Degree project in
Communication Systems
Second level, 30.0 HEC
Stockholm, Sweden

Aalto University
School of Science
Degree Programme of Computer Science and Engineering

Andrew Keating

Models for the Simulation of a Name-Based Interdomain Routing Architecture

Master's Thesis
Espoo, June 29, 2012

Home Supervisor: Professor Gerald Q. Maguire Jr.
Royal Institute of Technology, Sweden

Host Supervisor: Professor Tuomas Aura
Aalto University, Finland

Instructor: Kari Visala, M.Sc. (Tech.)
Helsinki Institute for Information Technology, Finland

Aalto University
School of Science
Degree Programme of Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author: Andrew Keating

Title:
Models for the Simulation of a Name-Based Interdomain Routing Architecture

Date:	June 29, 2012	**Pages:**	xiv + 71
Professorship:	Computer Science	**Code:**	T-110
Supervisors:	Professor Gerald Q. Maguire Jr. Professor Tuomas Aura		
Instructor:	Kari Visala, M.Sc. (Tech.)		

Researchers who aim to evaluate proposed modifications to the Internet's architecture face a unique set of challenges. Internet-based measurements provide limited value to such evaluations, as the quantities being measured are easily lost to ambiguity and idiosyncrasy. While simulations offer more control, Internet-like environments are difficult to construct due to the lack of ground truth in critical areas, such as topological structure and traffic patterns.

This thesis develops a network topology and traffic models for a simulation-based evaluation of the PURSUIT rendezvous system, the name-based interdomain routing mechanism of an information-centric future Internet architecture. Although the empirical data used to construct the employed models is imperfect, it is nonetheless useful for identifying invariants which can shed light upon significant architectural characteristics. The contribution of this work is twofold. In addition to being directly applicable to the evaluation of PURSUIT's rendezvous system, the methods used in this thesis may be applied more generally to any studies which aim to simulate Internet-like systems.

Keywords: Information-centric networking, rendezvous routing, AS-level topology, simulation, object popularity, traffic modeling

Language: English

Aalto-universitetet
Högskolan för teknikvetenskaper
Examensprogram för datateknik

SAMMANDRAG AV
DIPLOMARBETET

Utfört av: Andrew Keating

Arbetets namn:
Models for the Simulation of a Name-Based Interdomain Routing Architecture

Datum:	Den 29 Juni 2012	**Sidantal:**	xiv + 71
Professur:	Datavetenskap	**Kod:**	T-110
Supervisors:	Professor Gerald Q. Maguire Jr. Professor Tuomas Aura		
Handledare:	Diplomingenjör Kari Visala		

Forskare som syftar till att utvärdera föreslagna ändringar av Internet arkitektur står inför en unik uppsättning utmaningar. Internet-baserade mätningar ger begränsat värde för sådana utvärderingar, eftersom de kvantiteter som mäts är lätt förlorade mot tvetydighet och egenhet. Även om simuleringar ger mer kontroll är Internet-liknande miljöer svåra att konstruera på grund av bristen på kända principer i kritiska områden, såsom topologiska struktur och trafikmönster.

Denna avhandling utvecklar en nättopologi och trafikmodeller för en simulering baserad utvärdering av PURSUIT mötesplatsen systemet, den namn-baserade interdomän routing mekanismen för en informations-centrerad arkitektur av framtidens Internet. Även om de empiriska data som används för att konstruera modeller är bristfällig, är det ändå användbart för att identifiera invarianter som kan belysa viktiga arkitektoniska egenskaper. Bidraget från detta arbete har två syften. Förutom att vara direkt tillämplig för utvärderingen av PURSUITs rendezvous system, kan de metoder som används i denna avhandling användas mer allmänt för studier som syftar till att simulera Internet-liknande system.

Nyckelord:	Informations-centrerad nätverk, rendezvous routing, AS-nivå topologi, simulering, objekt popularitet, trafik modellering
Språk:	Engelska

Acknowledgements

I would like to acknowledge the exceptional guidance which I received from my home supervisor, Professor Gerald Q. Maguire Jr. Professor Maguire's dedication to his Master's students is truly inspiring, and his consistently insightful comments were invaluable to my thesis.

I am also grateful to my instructor, Kari Visala, who provided me with feedback on numerous occasions.

I would like to thank Andrey Lukyanenko for providing me with mathematical advice.

Finally, I would like to thank my host supervisor, Professor Tuomas Aura, for his helpful comments.

Espoo, June 29, 2012

Andrew Keating

Abbreviations and Acronyms

AId	Algorithmic Identifier
AS	Autonomous System
ASN	Autonomous System Number
BGP	Border Gateway Protocol
CAIDA	Cooperative Association for Internet Data Analysis
CCN	Content-Centric Networking
CDF	Cumulative Distribution Function
CDN	Content Distribution Network
CP	Content Provider
DAG	Directed Acyclic Graph
DDoS	Distributed Denial of Service
DHT	Distributed Hash Table
DNS	Domain Name System
DONA	Data-Oriented Network Architecture
HTTP	Hypertext Transfer Protocol
i3	Internet Indirection Infrastructure
ICMP	Internet Control Message Protocol
ICN	Information-Centric Networking
IP	Internet Protocol
IS-IS	Intermediate System to Intermediate System
ISP	Internet Service Provider
IXP	Internet Exchange Point
OSPF	Open Shortest Path First
P2P	Peer to Peer
PI	Persistent Interest
PMF	Probability Mass Function
PoP	Point of Presence
ReNe	Rendezvous Network
RId	Rendezvous Identifier

RN	Rendezvous Node
ROFL	Routing on Flat Labels
SId	Scope Identifier
SIP	Session Initiation Protocol
TM	Topology Manager
URL	Uniform Resource Locator
VoIP	Voice over IP
VoPSI	Voice over Publish/Subscribe Internetworking
ZM	Zipf-Mandelbrot

Contents

Abbreviations and Acronyms vi

List of Tables xi

List of Figures xiii

1 Introduction 1
- 1.1 Problem Statement . . . 3
- 1.2 Organization . . . 3

2 Background 5
- 2.1 Internet Topology Inference . . . 5
  - 2.1.1 Topology Mapping . . . 6
  - 2.1.2 Traceroute Measurements . . . 7
  - 2.1.3 BGP Measurements . . . 8
- 2.2 Internet Traffic Analysis . . . 10
- 2.3 Rendezvous Routing . . . 12
- 2.4 Information-Centric Applications . . . 14
- 2.5 PURSUIT . . . 16
  - 2.5.1 Tenets . . . 16
  - 2.5.2 Identifiers . . . 17
  - 2.5.3 Rendezvous System . . . 18
  - 2.5.4 Topology Management and Forwarding . . . 20
  - 2.5.5 Prototype and Applications . . . 20

3 Evaluation Methodology 23
- 3.1 Prior Evaluation . . . 24
  - 3.1.1 Areas for Improvement . . . 25
- 3.2 Distributed Rendezvous Simulator . . . 26
- 3.3 Internet Topology Maps . . . 27
  - 3.3.1 Dataset Analysis . . . 28

3.3.2 Routing 30
3.4 Application Traffic Modeling 32
3.4.1 Traffic Volume 33
3.4.2 Object Collections 33
3.4.3 Web Traffic 34
3.4.4 P2P Traffic 36
3.5 Object Popularity Distribution 38
3.5.1 Power Law Distributions 39
3.5.2 Web Objects 39
3.5.3 P2P Objects 40
3.5.4 Distribution Parameters 41
3.5.5 Generating Object Identifiers 45
3.6 Spatial Locality 49
3.7 Workload Generator Design 52
3.7.1 Generating Rendezvous Requests 53

4 Discussion **55**
4.1 Topology 55
4.2 Traffic Models 56
4.3 Reflections 58

5 Conclusions and Future Work **59**
5.1 Future Work 60

Bibliography **61**

List of Tables

3.1 Summary of CAIDA and UCLA datasets 29
3.2 Hybrid UCLA*-IXP topology 30
3.3 BitTorrent content size data gathered by our crawler 37
3.4 Comparison of AS rankings 50
3.5 Workload Generator parameters 52

List of Figures

2.1 Invisible peering link 9
2.2 Scope and rendezvous identifiers 17
2.3 Example of a two-level interconnection overlay 19

3.1 Distributed rendezvous simulator architecture 27
3.2 Example IXP topology 28
3.3 Valid path between ASes 31
3.4 Sample HTTP request 35
3.5 Sample HTTP response 35
3.6 Probability mass function of the Zipf distribution 43
3.7 Log-log plot of the Zipf distribution's probability mass function 43
3.8 Comparison of the Zipf and Zipf-Mandelbrot distributions . . 44
3.9 Percent error of our approximation of the Zipf CDF 47

Chapter 1

Introduction

One of the primary incentives for the research which led to ARPANET, the precursor to today's Internet, was resource sharing [1]. Computing devices were extremely expensive, and remote time sharing promised access to these devices at a fraction of the cost of duplicating them. An end-to-end communication model emerged with host identifiers serving as the central abstraction. Now more than four decades after the inception of ARPANET, Internet traffic is dominated by a different class of applications which deal with the acquisition and dissemination of chunks of information.

Despite the growing popularity of information-centric applications, the primary function of the network has remained the best-effort forwarding of packets between endpoints. Information-centric networking (ICN) addresses the fact that today's networked applications are far more concerned with *what* than *where*, proposing a major functional shift whereby the network's main purpose becomes locating and delivering information [2]. Arguments in favor of ICN cite potential increases in availability, efficiency, and security. Consider the fact that as of 2011, approximately 1.8 trillion gigabytes of information were accessible via the Internet [3]. Additionally, this figure has been observed to more than double every two years. To increase the availability of the Internet's huge volume of information, solutions such as content distribution networks (CDNs) and peer-to-peer (P2P) overlays were developed. ICN would make such technologies obsolete, as the network would provide equivalent services.

Two fundamental design principles drive most modern ICN architectures. First is the concept that every piece of information is assigned a unique identifier. Second is the idea that networking should follow the

publish/subscribe paradigm, thus users *subscribe* to receive information which is *published* by content providers. The naming of each piece of information greatly improves the ability to cache data within the network, ensuring that popular information can retrieved from local sources whenever possible. The use of publish/subscribe operations at the network level simplifies the Internet's service model by eliminating the need to specify endpoint addresses and limits the effectiveness of distributed denial of service (DDoS) attacks, as users only receive content which they have explicitly subscribed to.

The location of named content is a vital function of ICN systems. Several ICN architectures have been proposed [4–8], each having their own content location mechanisms. One such approach is known as rendezvous routing, in which a decentralized network of rendezvous servers routes information requests toward content publishers. Internet-wide rendezvous routing faces clear efficiency and reliability challenges which recent studies have sought to overcome [9–11]. However, demonstrating that a rendezvous routing system is scalable and fault-tolerant enough to be considered as a replacement for traditional Internet routing has proven to be a challenging task.

Global networking systems are inherently difficult to evaluate. Merely studying the characteristics of the current Internet is a delicate task with numerous potential pitfalls. Using the Internet as a measurement platform for new global systems is at best extremely challenging, if not impossible, especially for studies which propose changes to the Internet's core architecture. The ideal evaluation methodology for such systems often involves simulation, which can uncover characteristics that may not be revealed by mathematical models alone. However, this approach also has its own set of challenges.

Fine-grained network simulators such as ns-3 [12] do not scale well with large topologies, limiting their usefulness in Internet-wide simulations. For some studies, utilizing a high-level simulation which models the Internet as a graph of interconnected autonomous systems (ASes) can provide acceptable levels of both detail and scalability. However, despite numerous studies which aim to capture the AS-level topology and global traffic patterns, the rapidly-evolving characteristics of the Internet remain elusive to the research community.

1.1 Problem Statement

This thesis contributes a network topology and application traffic models to a methodology for evaluating the rendezvous routing system of a clean-slate future Internet architecture. The models are intended to be utilized directly by a simulator which implements rendezvous routing on the autonomous system level. The goals of the thesis are:

1. to analyze existing methods for mapping the Internet's topology and produce a dataset which captures the structure of the Internet as closely as possible,
2. to study the traffic characteristics of popular Internet applications and develop methods for generating the rendezvous requests which may be produced by their information-centric equivalents,
3. to model the popularity of objects in each class of generated application traffic, ensuring that they follow empirically observed object popularity distributions, and
4. to introduce realistic spatial locality to the generated rendezvous requests.

1.2 Organization

The organization of the remainder of this thesis is as follows. Chapter 2 presents background information. This includes an introduction to Internet topology inference and an overview of rendezvous routing and information-centric applications, in addition to a survey of Internet traffic analysis studies, and a presentation of the PURSUIT future Internet architecture. Chapter 3 contributes Internet topology and application traffic models to an evaluation methodology for the PURSUIT rendezvous system. This chapter discusses the construction of an AS-level Internet topology dataset and presents methods for generating rendezvous requests based on the behavior of popular Internet applications, capturing crucial characteristics such as object popularity and spatial locality. In Chapter 4, we consider the implications of our contributions to the rendezvous system's evaluation methodology and discuss the shortcomings of our methods. Chapter 5 concludes the thesis and suggests future work.

Chapter 2

Background

This chapter presents fundamental concepts and prior studies from several research areas which are central to this thesis. It provides an introduction to the methods used for inferring the Internet's topological structure and the numerous difficulties faced by researchers in this area. This leads to an overview of Internet traffic analysis, where several studies of Internet traffic patterns are presented. The remainder of the chapter focuses on information-centric networking, specifically rendezvous routing, information-centric applications, and the PURSUIT information-centric future Internet architecture.

2.1 Internet Topology Inference

End-to-end global connectivity via the Internet is enabled by the Border Gateway Protocol (BGP). BGP is an interdomain path vector routing protocol which facilitates the dissemination of network reachability information between anonymous systems. A BGP routing update, sent from one BGP-speaking router to another, contains the path of ASes which can be traversed to reach a set of Internet Protocol (IP) addresses. The "best" route is generally determined by *policies* which represent relationships between ASes, rather than traditional routing metrics such as path length, delay, or throughput.

Let us first consider what an AS is. The original BGP specification in RFC 1163 [13] defines an AS as "a set of routers under a single technical administration, using an interior gateway protocol and common metrics to

route packets within the AS, and using an exterior gateway protocol to route packets to other ASes." The most recent BGP specification in RFC 4271 [14] updates the original definition, noting that single ASes commonly employ several IGPs (and sometimes multiple routing metrics). The updated definition states that an AS "appears to other ASes to have a single coherent interior routing plan, and presents a consistent picture of the destinations that are reachable through it."

Autonomous systems vary in size and function. For example, AS 2914 is operated by the multinational corporation NTT Communications, which has points of presence (PoPs) in North America, Europe, Asia, and Australia, while AS 39857 is operated by the relatively modestly-sized Aalto University Student Union. It is also fairly common for a single organization to operate multiple ASes [15].

All ASes can be classified as either transit or stub ASes. A transit AS is a network which provides packet forwarding service to and from other networks. A stub AS does not forward packets for other ASes and relies on one or more transit providers for Internet access. Most transit ASes are themselves customers of larger transit ASes, forming a hierarchy with global "tier-1" providers at the top. Tier-1 providers sell transit service to regional Internet service providers (ISPs), who then sell transit service to smaller residential and business providers.

Transit ASes offer what is known as the *customer-provider* relationship, where the transit AS is the provider and the AS receiving the transit service is the customer. ASes may also enter a *peer-to-peer* relationship, where they mutually agree to exchange traffic between each other and each others' customers without any payment. For example, all of the tier-1 ISPs are peers in a full-mesh topology which forms the core of the Internet. Peering between autonomous systems often occurs at an Internet exchange point (IXP), a physical infrastructure maintained by a third party where ASes directly interconnect via layer-2 switching to exchange traffic [16]. It is also possible to peer privately by physically connecting two ASes, but IXPs often tend to be more convenient and cost-efficient.

2.1.1 Topology Mapping

In an ideal world, all service providers would gladly share their network configurations for the benefit of the research community. However, many providers are secretive about the business relationships their interconnections

are based upon. As a result, researchers are forced to *infer* the Internet's topology, a task which is difficult to perform and nearly impossible to fully validate.

Mapping the Internet's topology at the router level would require capturing all physical interconnections between routers on the Internet. Given the large number of Internet routers, such a fine-grained topology is not only infeasible to collect, but its sheer size would introduce insurmountable scalability issues to most simulation environments. An alternative to a router-level topology is to map the Internet at the level of PoPs, physically co-located groups of routers which are deployed by ISPs. While some research efforts have attempted to compile PoP-level Internet topology maps [17, 18], these are still widely considered to be works-in-progress.

A domain-level map of the Internet's topology is one level of abstraction higher than a PoP-level map, capturing the links between ASes. Efforts to infer the AS-level topology of the Internet fall into two main categories based on the source of data used: traceroute-based measurements (active) and BGP-based measurements (passive). The former involves mapping AS numbers (ASNs) to IP address ranges and analyzing traceroute probes to determine adjacencies between ASes, while the latter uses BGP routing information collected by route monitors to infer AS interconnections.

BGP route monitors are very useful for gathering information about the Internet's topology. A BGP route monitor collects and organizes BGP routing information, often gathered from Internet backbone links. This routing information is extracted from the *AS_PATH* attribute of BGP UPDATE messages, which contains an ordered list of the ASes on the path to a given IP prefix. RouteViews [19] and RIPE RIS [20] provide publicly available BGP data collected from numerous route monitors, which several studies have used to infer the Internet's topological structure.

2.1.2 Traceroute Measurements

One method of measuring the Internet's AS-level topology is through the use of the *traceroute* tool. Traceroute infers the routing path to an end host by successively sending packets addressed to the host with incremented IP time-to-live parameters, causing each router on the path to return an Internet Control Message Protocol (ICMP) Time Exceeded error. The routing path is derived by collecting the source address from each Time Exceeded packet until the final destination is reached.

Chang, et al. [21] presented a method for inferring the Internet's AS-level topology using traceroute-based measurements. To achieve this, they mapped IP address prefixes to the ASes in which these prefixes reside. This mapping was created from paths captured by BGP route monitors, supplemented with publicly available route origin information (which introduces prefixes that are invisible due to route aggregation). Once the mapping was created, they could determine which AS an IP address resides in via longest prefix matching. They realized that some AS paths produced by this method contained anomalies such as routing loops. One cause of such errors was that the IPv4 standard for routers only requires that the source address used in an error message is assigned to one of the router's physical interfaces [22]. This is problematic because border routers have interfaces residing in multiple ASes. If a border router specifies the source address as one of its outgoing interfaces which lies in another AS, the path will be incorrectly inferred.

In addition to the interface issue mentioned above, traceroute-based measurements have been found to incur a number of other pitfalls. Zhang, et al. [23] and Mao, et al. [24] presented several of these pitfalls, which can be summarized as follows:

1. Aggregation and filtering of routes can cause the list of ASes drawn from BGP UPDATE messages to differ from the actual path taken during data forwarding.
2. Some traceroute hops do not return an ICMP reply.
3. Successive traceroute packets may take different paths.
4. A single IP prefix may be announced by multiple ASes [25].

DIMES [26] and Ark [27] are two additional research efforts which, among other objectives, aimed to map the Internet's topology using traceroute measurements. We do not explore these projects, as their utility is limited by the previously mentioned issues.

2.1.3 BGP Measurements

A more widely accepted method for measuring the AS-level topology is to infer connections between ASes from publicly available BGP routing data. The UCLA Internet Research Lab's AS-level topology [28] 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. ISP route servers and looking glasses allow network users to run a limited set of router commands (e.g., output the contents of the BGP routing table) for the purpose of network troubleshooting. While limited in number, these additional views can uncover links which are not captured by route monitors. Internet Routing Registries are databases of route configurations which some operators voluntarily provide to allow for automated route filtering and to alleviate the troubleshooting of interdomain routing issues.

The Cooperative Association for Internet Data Analysis (CAIDA) AS Relationship dataset [29] is another BGP-derived AS-level topology dataset which augments the AS graph with per-link business relationships. These are computed using heuristics adapted from methods proposed by Gao [30]. Dimitropoulos, et al. [31] presented an inference methodology and validation of the AS relationships and mentioned a third type of AS relationship, *sibling-to-sibling*, but these links are very rare and no such links actually appear in the dataset. Currently, the UCLA AS-level topology dataset is also augmented with business relationships, but these were not introduced until several years after the topology data was first made available.

Both of these AS-level topologies suffer from one major shortcoming: the absence of most peering links. This can be attributed to the *valley-free* routing policy, which mandates that ASes do *not* announce routes containing peer-to-peer links to their providers or any 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. Figure 2.1 illustrates how a peer-to-peer link can be invisible to a route monitor. If a route monitor is present at AS A, the monitor will not capture the peering link between its customer ASes, B and C, because the valley-free policy ensures that this link is not advertised to A.

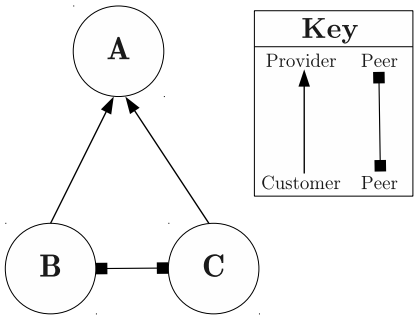

Figure 2.1: Invisible peering link

Oliveira, et al. [32] 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 is eventually able to capture every *customer-provider* link in the Internet's AS-level topology. However, the authors estimated that as many as 90% of *peering* links may be missing from existing AS topology datasets. Dhamdhere, et al. [33] noted that 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.

2.2 Internet Traffic Analysis

Internet traffic studies are notoriously difficult to conduct. Service providers are often hesitant to divulge network configurations and usage statistics, as these are widely considered to be business secrets. Researchers are occasionally granted access to the traffic data of a high-tier provider, but more often they are forced to make clever use of limited publicly available resources.

Chang, et al. [34] devised an inter-AS traffic model by ranking the *utility* of ASes based on three traffic types: web hosting, residential access, and business access. To quantify web hosting utility, they queried Google to retrieve the top 10 uniform resource locators (URLs) for the 10,000 most common search keywords in seven different languages during the years 2003 and 2004. They mapped the web server IP address from each URL to its corresponding AS and ranked the ASes by volume, using a PlanetLab [35] test bed to detect Domain Name System (DNS)-based load balancing. Residential access utility was determined by monitoring several popular P2P file sharing networks and measuring the number of users per AS. Business access utility was calculated by measuring the number of downstream ASes which are reachable from each AS. The authors used the AS utility values to create a gravity model, from which they computed inter-AS traffic matrices. While this research presented a novel use of publicly available information to estimate Internet-wide traffic patterns, the authors noted that their methodology for calculating utility values had some flaws. For example, their metric for web hosting utility excluded embedded content, their business access utility metric failed to consider providers who assign

private AS numbers to their customers, and their residential access utility metric assumed that P2P usage was uniformly distributed across residential networks.

Maier, et al. [36] monitored the network activity of over 20,000 European residential DSL customers in 2008 and 2009, using deep packet inspection to classify traffic types. They found web traffic to be dominant, representing nearly 60% of all traffic, while P2P contributed about 14%. Additionally, the volume of web traffic was observed to be increasing over time, while P2P's traffic volume was decreasing. These results were extremely interesting, as several prior studies had found P2P traffic to be dominant [37–39]. Among web traffic, the authors found that 25% of all bytes carried Flash video, while 14% consisted of RAR archives. They found BitTorrent to be the most prevalent P2P application, while older P2P systems such as Gnutella were almost non-existent.

Labovitz, et al. [40] analyzed global interdomain traffic patterns between July 2007 and July 2009, examining changes in the type and volume of traffic, as well as differences in interconnection relationships between providers. By instrumenting edge routers at 110 large service providers in North America, Europe, Asia, and South America, they were able to observe changing application traffic volumes, identifying a significant rise in web traffic and a roughly equivalent decline in P2P traffic. Another observation of their research was that as of July 2009, over 50% of all interdomain traffic was originated by just 150 ASes.

Ager, et al. [41] used crowdsourced DNS measurements and BGP route monitors to analyze infrastructures for hosting and distributing web content. One product of this study was a location matrix of web content, which captured the continents where web content was originated and subsequently served. The authors noted that 46% of popular content hostnames were served from North America, with a further 20% and 18% being served from Europe and Asia, respectively. Another observation from this data was that 11.6% of content hostnames were served in the same continent from which they originated, confirming that a significant amount of web content was replicated in multiple global regions. While this research contributed valuable insight into global Internet traffic, it unfortunately failed to capture traffic from CDN nodes and data centers located within ISPs' boundaries.

The Sandive Fall 2011 Global Internet Phenomena Report [42] investigated Internet-wide trends using data collected by ISPs in over 85 countries. They found that their data varied significantly between geographic locations. For example, by far the most prevalent Internet application by traffic volume

in North America was Netflix[1], a provider of on-demand Internet video streaming. Netflix accounted for 27% of all North American downstream traffic (32.7% during peak hours) in Fall of 2011, but it had not yet been deployed in any other continents. The authors also reported large variations in traffic volumes during different times of day. These findings highlight the difficulty of accurately modeling Internet traffic patterns, as they differ significantly between geographic locations and times of day, and they are constantly evolving.

2.3 Rendezvous Routing

The ability to efficiently and reliably locate named content is crucial in information-centric networking systems. Rendezvous routing is one approach to content location in which information requests are routed towards content publishers by a decentralized network of rendezvous servers. Rendezvous routing is a major focus of this thesis, specifically the rendezvous routing system of the PURSUIT ICN architecture. We provide a high-level overview of other rendezvous systems in this section. A detailed description of the PURSUIT rendezvous routing system is presented in Section 2.5.3.

In our evaluation of rendezvous routing architectures, we pay close attention to the feasibility of Internet-wide deployment. There is no simple metric which captures deployment feasibility, but we try to critique solutions from the perspective of service providers, who are influential in deciding the fate of new networking technologies. Handley [43] argued that new technologies are only deployed in commercial networks for reasons of greed or fear – that is, to make money or avoid losing money. Modeling the complex tussles [44] introduced by ICN is outside the scope of this thesis, but an interested reader may reference Trossen, et al. [45] for more on this topic. Instead, we will use some common sense and the fact that ISPs are unlikely to invest in new technologies which affect their existing business models *unless* the monetary benefits are extremely clear.

TRIAD [8] is an interdomain content routing system which maps URLs to next hops through the use of *content routers*, IP routers which have been extended to support name-based routing. In TRIAD, URLs are aggregated by their suffixes (e.g., http://domain.org/dir/content.html would be converted to http://dir.domain.org/content.html). This aggregation is

[1] http://netflix.com

crucial to the scalability of TRIAD, as it enables routing to be performed using efficient longest-suffix matching. TRIAD's scalability depends heavily on this aggregation, which requires information to closely follow the hierarchical naming structure of the DNS. TRIAD's advocacy for a global content routing system was very influential in the area of information-centric networking.

The Internet Indirection Infrastructure (*i3*) [9] is a structured overlay network based on the Chord distributed hash table (DHT) [46] where information is sent and received using logical identifiers, thus eliminating the need to specify endpoint addresses in sending and receiving operations. Within the overlay, *i3* servers store subscription records in a distributed fashion such that one server is responsible for any given information identifier. The servers forward packets over IP between other *i3* servers and eventually to their final destinations. While the underlying concept of rendezvous routing proved to be significant, *i3* is poorly suited to interdomain rendezvous routing due to Chord's inability to support domain-specific routing policies.

Routing on Flat Labels (ROFL) [10] is a name-based interdomain routing system which uses flat labels as identifiers in a Canon-based [47] hierarchical DHT overlay network. In ROFL, each AS joins a global ring while additionally maintaining its own intradomain Chord ring. The main advantage of adopting a hierarchical DHT is the ability to enforce AS-level routing policies such as peering, customer-provider, backup, and multi-homing. Although ROFL makes a strong case in support of Internet-wide name-based routing, it is built upon the arguably unrealistic assumption that all participating ASes are willing to perform similar roles in the global ring, with no distinction between small enterprise domains and top-tier service providers.

The Data-Oriented Network Architecture (DONA) [4] is an ICN platform designed upon the publish/subscribe networking paradigm (although publish and subscribe operations are called *register* and *find* in DONA). Flat, self-certified names serve as identifiers, and information lookups are performed by *resolution handlers*, which support interdomain routing using BGP-like policies. Lookups in DONA are first performed locally by the resolution handler of the originating AS. If the local resolution handler cannot resolve the lookup, the request is forwarded upwards in the AS hierarchy. Each AS's resolution handler maintains routing state for all data residing below or equal to it in the AS hierarchy. This places a large burden on tier-1 service providers, who must index and resolve queries for all actively registered data items. The authors estimated the memory and computation overhead of

DONA based on the number of public web pages on the Internet in 2005, concluding that they were well within the capabilities of modern datacenter technology. However, it should be noted that this overhead increases linearly with the number of registered data objects, which may be a cause of concern for tier-1 transit providers.

Content-Centric Networking (CCN) [5] is an information-centric networking system which uses hierarchical names similar to web URLs. In CCN, users request information via *Interest* packets, which are routed to content providers. Content providers respond to requests which they can fulfill by forwarding *Data* packets back to the requester on the reverse path used in the lookup. Unlike the previous rendezvous routing solutions which use DHT-based overlays for content routing, CCN is designed to be incrementally deployed via the general *type label value* capabilities of traditional link-state routing protocols such as Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS). Instead of operating on IP addresses, CCN *content routers* perform longest prefix matching on content names. For interdomain routing, the authors envision that eventually content identifier prefixes will be integrated into BGP. CCN also includes a transport service which guarantees reliability with TCP-like sequence numbers and implements window-based flow control by limiting the rate at which Interest packets are sent.

2.4 Information-Centric Applications

While many popular Internet applications are inherently information-centric, the Internet's communication model forces them to adopt host-centric implementations. It is useful to consider how modern applications and services would operate in proposed ICN architectures. This section introduces several ICN applications, taking into consideration differences in implementation from their host-oriented versions and drawing performance comparisons where possible. PURSUIT's applications are discussed in Section 2.5.5.

One major difference between modern Internet applications and their ICN equivalents is that information-centric applications are completely receiver-driven. Users of ICN applications do not receive any content which they have not explicitly expressed interest in, a guarantee which is facilitated by the network's matching of interest and availability. This is in stark contrast to the behavior of today's Internet applications, where information

can be sent to arbitrary IP addresses without any prior context. It is not particularly difficult to envision how simple applications might operate in ICN architectures. For example, the *ccnputfile* [48] and *ccngetfile* [49] utilities for the CCN platform perform basic file transfer operations. Files are published to content repositories by *ccnputfile*, and users can express interest in named files using *ccngetfile*. Once the networking system has matched a receiver's interest with a publisher's availability, the receiver creates a new Interest packet for each segment of the file.

Jacobsen, et al. [50] noted an air of uncertainty surrounding the topic of real-time applications in information-centric networks. To investigate this subject, they created a prototype implementation of a voice over IP (VoIP) system called VoCCN on their CCN platform. The first problem they encountered was the need to support *service rendezvous*. Before a user can receive a call, a subscription must be created which indicates interest in an incoming call. Their solution to this problem was *on-demand publishing*, in which a request for as of yet non-published content is routed to a potential publisher of this content, who may subsequently publish the desired content. Another challenge was the need to maintain a bi-directional conversation flow, because by default CCN packets do not identify the destination where responses should be sent. They solved this problem with *constructable names*, through which the caller can determine how to formulate a request that will reach the callee without any prior information. It is interesting to note that in traditional VoIP signaling via the Session Initiation Protocol (SIP) [51], the call setup process involves registrar and proxy servers, whereas in the ICN equivalent, the network's added functionality renders them unnecessary.

Tsilopoulos, et al. [52] argued that although it is important for ICN systems to maintain the property that users only receive information which they have explicitly requested, the *one request per packet* model is not an ideal fit for some traffic types. They noted that sending one request per packet in real-time applications such as VoCCN (explained above) wastes uplink bandwidth and excessively burdens routers, proposing an alternative mode of operation known as Persistent Interests (PIs). A PI expresses interest in an arbitrary *stream* of information (e.g., a conversation) by prepending data packet identifiers with a common prefix, known as the *channel name*. Forwarding is then performed based on the channel name, with the PI persisting in routers until the user unsubscribes from the channel or the PI expires.

2.5 PURSUIT

The FP7 PURSUIT project [53] is an ongoing research effort which aims to develop a clean-slate Internet architecture based on the publish/subscribe networking paradigm. In this section, we first discuss the main tenets upon which the PURSUIT architecture is based. Next we present the different types of identifiers used by PURSUIT. We then introduce the three core functions of the PURSUIT architecture: rendezvous, topology management, and forwarding. Finally we discuss the project's prototype software and present two applications which have been developed for PURSUIT.

2.5.1 Tenets

The PURSUIT Architecture Definition deliverable [54] outlines several tenets which are fundamental to the project. The first of these is the need to identify individual pieces of information. This allows for the separation of *what* from *who* in an information exchange. The underlying network is responsible for performing *late binding* of location and information.

The second tenet is the ability to establish context for information items. PURSUIT achieves this through a concept called scoping. A scope organizes a set of information items which exist to fulfill a common purpose. Each information item belongs to at least one scope, and a scope is itself an information item. This property enables the nesting of scopes.

By combining the first two tenets, it is possible to construct complex directed acyclic graphs (DAGs) composed of information. The third tenet is the definition of a service model which supports performing computations directly upon these information graphs. This enables a class of solutions to computational problems which operate over information flows without the need to specify the communicating parties.

The fourth tenet addresses the modularity of information dissemination, which is achieved through the use of three core functions: rendezvous, topology management, and forwarding. Rendezvous refers to matching interest in and availability of information items, topology management determines the delivery path of information which parties have expressed interest in, and forwarding executes the data transfer over this path.

The fifth and sixth tenets aim to achieve modularity across computational problems by defining information dissemination strategies and resolving

conflicts between different strategies. The goal of these tenets is that individual solutions to computational problems can be directly applied to larger problems.

2.5.2 Identifiers

All information items in PURSUIT are assigned a statistically-unique fixed-size Rendezvous Identifier (RId) [55]. A scope identifier (SId) is a special instance of a RId which is used to group related information items. A scope may contain additional sub-scopes, which allows for the creation of information graphs such as the one shown in Figure 2.2. Information items are identified by their full paths starting from the root scope, and multiple identifiers can resolve to the same information item. For example, /$\mathrm{SId_A}$/$\mathrm{SId_C}$/$\mathrm{RId_5}$/ and /$\mathrm{SId_B}$/$\mathrm{SId_C}$/$\mathrm{RId_5}$/ refer to the same piece of information.

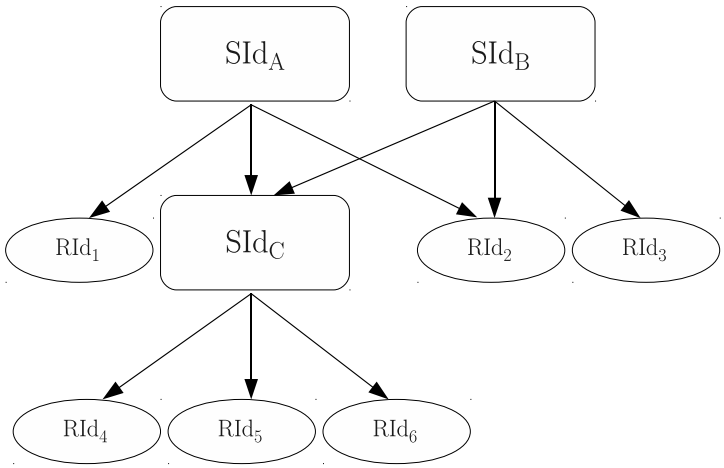

Figure 2.2: Scope and rendezvous identifiers

Another type of identifier, the Algorithmic Identifier (AId), exists for the purpose of application-specific content labeling. For example, sequence numbers for a video streaming application may be implemented with AIds, where the video file is identified by its $< SId, RId >$ pair and each frame is identified with an AId sequence number.

2.5.3 Rendezvous System

PURSUIT's interdomain rendezvous system routes communication requests toward available copies of named information [11]. This process can be viewed as the matching of availability as reported by publishers and interest as reported by subscribers. The design of the rendezvous system follows five guiding principles:

1. The system employs a flat, self-certified namespace.
2. Only top-tier providers need to participate in the core of the rendezvous system, so enterprise domains are not forced to provide transit service.
3. Rendezvous networks consist only of willing service providers.
4. Whenever possible, locality of communication is preserved.
5. Reachability state for rarely-requested objects is not distributed globally.

The second design principle ensures that PURSUIT's rendezvous system maintains compatibility with the business relationships which exist on the Internet today. As such, the rendezvous system supports the customer-provider and peer-to-peer interdomain routing policies.

The most basic component of the system is the *rendezvous node* (RN). A RN is a server which handles rendezvous requests (to publish or subscribe to information). A *rendezvous network* (ReNe) is a hierarchical network of RNs which maintain either customer-provider or peer-to-peer business relationships. The lowest level of a ReNe consists of stub networks, which propagate reachability information for objects in their networks to their peers and providers. This upward propagation is repeated by all RNs in the ReNe, resulting in the top tier service provider receiving the entire set of reachable objects in each of the lower-tier networks.

The top tier rendezvous nodes in each ReNe interconnect to form an *interconnection overlay*. The interconnection overlay is a Canonical Chord DHT [47] in which each overlay node is responsible for a portion of the identifier space for objects. Connections between rendezvous nodes in the overlay are logical, so only willing networks need to participate, and the overlay is able to function without participation from upstream transit providers. Figure 2.3 depicts a sample interconnection overlay, with the shaded regions representing rendezvous networks and the unshaded circles representing levels of the interconnection overlay DHT. The uppermost ring

is the top tier of the interconnection overlay, and the lower ring portrays the portion of the overlay consisting of AS A's customers. In this example, A provides rendezvous service to B, D, and G. Both D and G are top tier providers of their own rendezvous networks, hence they provide rendezvous service to their customers as well.

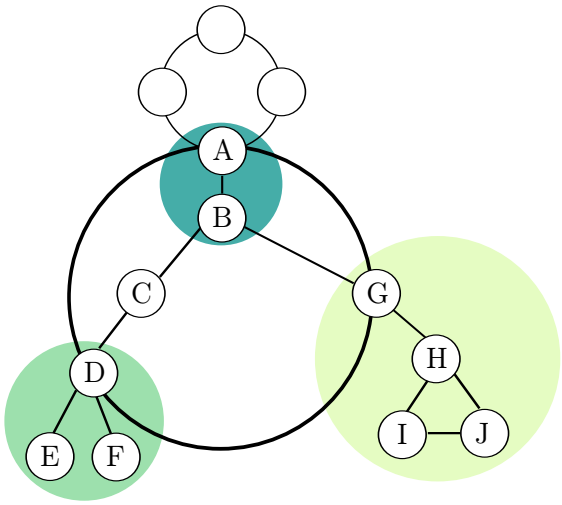

Figure 2.3: Example of a two-level interconnection overlay

An object enters the rendezvous system when the object owner sends a publication request to a local rendezvous node. The local RN routes the request to the node which is responsible for the portion of the namespace in which the object to be published lies. Entries are replicated in multiple overlay nodes for increased fault-tolerance.

Rendezvous subscription requests are first routed within the requester's local domain. If an intradomain copy of the information is not available, then the request is routed through the local ReNe. The request propagates through the interconnection overlay, starting at the bottom and progressing upwards. If at any point during this process the requested object is located, then the rendezvous request is forwarded to the responsible node. The path is incrementally recorded in the request message at each routing domain, and responses are sent back over the reverse of the recorded path.

The scalability of PURSUIT's rendezvous system relies heavily upon the caching of popular objects. The system's specifications do not explicitly define how caching should be performed, leaving this decision to the administrators of individual rendezvous nodes. A likely strategy for caching at rendezvous nodes is to maintain a set of the most frequently-requested information objects, updating the observed object popularity based on

incoming rendezvous requests and caching as many popular objects as the rendezvous node's available memory will allow.

2.5.4 Topology Management and Forwarding

While this thesis deals primarily with PURSUIT's rendezvous system, we provide a brief overview of the topology management and forwarding components, as they are vital for a complete understanding of the architecture. The topology manager (TM) is responsible for forming delivery graphs between publishers and subscribers. Each administrative domain updates its local topology information when nodes join or leave the network. The TM is typically queried when the the rendezvous system matches a publish/subscribe request pair. Depending on the dissemination strategy defined by the object publisher, the TM determines the forwarding path which enables the transfer of information between the publisher and the subscriber(s). For example, the dissemination strategy for a particular publication may require the topology manager to construct forwarding information for a shortest-path multicast tree between the publisher and several subscribers.

The forwarding function is responsible for delivering information along the delivery graph produced by the TM. The forwarding component uses unidirectional *link identifiers* to represent the link connecting two interfaces. A path in PURSUIT is encoded into a Bloom filter [56], a probabilistic bit vector data structure used to efficiently verify set membership [7]. Packets are source routed with the entire path to the receiver(s), represented by a set of link identifiers, included in the packet header. When a router receives a packet, it tests each of its interfaces against the Bloom filter by computing the XOR of each link identifier and the Bloom filter. The router then forwards the packet on any interfaces which are thought to be present in the encoded path. Both the topology management and forwarding components are discussed in more detail in PURSUIT's architectural documentation [54].

2.5.5 Prototype and Applications

Blackadder [57] is the open-source prototype of the PURSUIT ICN architecture. This prototype implements the major networking functions of PURSUIT (i.e., rendezvous, topology management, and forwarding).

Source code and documentation for Blackadder, including several sample applications, are available on the project's Github page[2].

One fairly straightforward application which highlights the mechanics of the PURSUIT architecture is a video streaming application [57]. In this application, video publishers advertise video channels (scopes) under which multiple video information items may be published. When a user subscribes to a publisher's channel, the rendezvous system supplies the publisher with a forwarding identifier to include in all data packets. Sequence numbers are used to sequentially identify video data. Once all viewers have unsubscribed from the channel, the rendezvous system informs the publisher to cease transmission of the video.

Voice over Publish/Subscribe Internetworking (VoPSI) [58] is a SIP-like VoIP application for the PURSUIT ICN architecture. The call setup signaling is receiver-driven, in that the recipient of the call creates a subscription under a desired unique name to indicate willingness to receive a call. The caller creates a publication to the recipient's name to initiate the call. VoPSI utilizes a Skype-like user search service [59] to facilitate the discovery of the $< SId, RId >$ pair used to call a particular user based on, for example, the user's first and last names. Once the call has been established, both parties begin to publish (and subscribe to) information items with increasing sequence numbers.

[2]https://github.com/fp7-pursuit/blackadder

Chapter 3

Evaluation Methodology

Evaluating the architectural components of a clean-slate future Internet design is a challenging task. Although PURSUIT's Internet-based prototype, Blackadder, implements the system's core functions, an evaluation of the prototype's rendezvous system would be severely impacted by the ambiguities and idiosyncrasies introduced by Internet-based measurements. Willinger, et al. [60] offered the following warning to researchers who wish to leverage the Internet as a measurement platform: "A very general but largely ignored fact about Internet-related measurements is that what we can measure in an Internet-like environment is typically not the same as what we really want to measure (or what we think we actually measure)." We heeded this warning, opting to evaluate PURSUIT's rendezvous system in a high-level simulation.

While simulation gives us complete control over what is actually being measured, great care must be taken to construct a simulation environment which accurately resembles the Internet. Floyd, et al. [61] discussed this topic in the aptly-titled article, *Difficulties in Simulating the Internet*. A major obstacle in Internet simulation is constructing an Internet-like topology. The Internet is constantly changing and its topology is extremely difficult to determine. Generating realistic Internet-like traffic is another challenging issue. Although many 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. Floyd, et al. also noted that simulations where each individual traffic source is modeled do not scale well, arguing that large-scale simulations can benefit from utilizing aggregate models. Additionally, they suggested that Internet simulations should be built upon *invariants*,

characteristics which empirical evidence has shown to be true in a wide range of scenarios. These two principles – the use of aggregate models and invariants – guided the design of our simulation environment.

This chapter contributes to an evaluation methodology for PURSUIT's rendezvous system. Section 3.1 analyzes an existing evaluation of the rendezvous system, presenting several aspects of the evaluation which we aim to improve upon. Section 3.2 discusses the architecture of a distributed rendezvous simulator and introduces the components of the simulator which were developed in this thesis. Section 3.3 deals with the problem of constructing an Internet-like topology for the distributed simulator. In Section 3.4, we describe how rendezvous traffic is generated in the simulation environment. Section 3.5 considers the object popularity distributions of popular Internet applications and explains how these probability distributions are used to generate object identifiers for simulation events. In Section 3.6, we introduce spatial locality to the generated rendezvous requests. Finally, in Section 3.7, we discuss the design details of our Workload Generator, which combines the methods presented in Sections 3.3-3.6 to produce rendezvous requests that serve as input to the simulator.

3.1 Prior Evaluation

PURSUIT's rendezvous system was evaluated by Rajahalme, et al. [11] in a study which formed much of the foundation of this thesis. This study measured four properties of the interdomain rendezvous routing system:

1. routing latency,
2. path stretch, which is the ratio of the path taken by routing message to the optimal policy-compliant path,
3. load distribution among rendezvous nodes, and
4. caching efficacy.

The evaluation was performed with a custom simulation environment which used the CAIDA AS Relationship dataset [29] as the network topology. As discussed in Section 2.1, this dataset is known to be missing the majority of peering links. Rajahalme, et al. addressed this deficiency by augmenting the topology with 900% additional peering links. When generating the additional peering links, none were introduced at or above domains containing Route View route monitors, as all these peering links were captured by the monitors.

Additionally, peering was not introduced between transitive customers, and no peering links were created for singly-homed stub ASes.

Rajahalme, et al. constructed rendezvous networks over the Internet's AS-level topology by assuming that all transit providers offer rendezvous service to their customers. The interconnection overlay consisted of a three-level Canon DHT hierarchy which was formed based on the topological distance between nodes. To determine the required number of rendezvous nodes, they took into consideration the expected number of objects to be handled by the system and the memory overhead which would be incurred by each node. The number of globally accessible objects in the system was assumed to be 10^{10}, which is one order of magnitude larger than the number of registered domain names in the DNS. The size of object pointers was assumed to be 64 bytes, with 32 bytes reserved for the rendezvous identifier and an additional 32 bytes for routing and indexing overhead.

Traffic was modeled by classifying each AS into one of three categories: business access, web hosting, and residential access. This model was developed by Chang, et al. [34], as discussed previously in Section 2.2. To model caching, Rajahalme, et al. assumed that the popularity of objects followed a Zipf distribution with a shape parameter of 0.91. This distribution was borrowed from a study which evaluated the popularity of DNS names from university DNS server traces [62]. They used a latency model developed by Zhang, et al. [63], in which the latency between domains was assumed to be 34ms, while intradomain hops were assumed to incur a 2ms latency. The number of intradomain hops for each AS was determined using a model developed by Tangmunarunkit, et al. [64]. Thus, the intradomain hop counts were set to $1 + \lfloor logD \rfloor$ where D is the degree of the domain.

3.1.1 Areas for Improvement

The prior evaluation of PURSUIT's rendezvous system presented a strong argument in support of the feasibility of a global name-based routing system. However, one might argue that the evaluation could have benefited from employing more realistic models. Rajahalme, et al. noted that their study could be improved with a more accurate delay model, by including link failures, and by estimating the computational load incurred by overlay maintenance, request routing, and cache management. We note the following additional shortcomings:

1. Although reasonable rules were employed in the generation of the 900% additional peering links for the CAIDA AS Relationship dataset, it is clear that this methodology is bound to generate many peering links which do not exist on the Internet.
2. A single probability distribution derived from a 2002 study of DNS lookups on a university network was used was used to determine the popularity of all objects.
3. The volume of traffic used in the simulations was never mentioned in the article. Although they noted that each simulation run consisted of 30,000 requests, the frequency of these requests was never specified.

3.2 Distributed Rendezvous Simulator

The main challenge involved in developing solutions to the issues discussed in the previous section is scalability. The prior evaluation simulated events *independently* and simply summed their characteristics to produce results. While this simple approach was computationally inexpensive enough to be executed on a single machine, it was impossible to evaluate how the individual events *interacted*. For example, a link failure might cause temporary localized congestion, but the evaluation was not robust enough to capture this. To meet the demands of a more fine-grained evaluation of the rendezvous system, a *distributed* discrete event-based simulation environment has been designed.

This Python-based distributed rendezvous simulator consists of four main components: the Nameserver, the Worker Nodes, the Coordinator, and the Workload Generator. The Nameserver, which is built upon the Pyro distributed object middleware framework [65], manages the registration of the Worker Nodes. Worker Nodes participating in the simulation register their presence with the Nameserver and await further commands. Prior to the start of a simulation, the Coordinator queries the Nameserver to retrieve a list of all participating Worker Nodes. The Workload Generator creates events and passes them to the Coordinator, which assigns them to individual Worker Nodes. The flow of events from the Workload Generator to the Coordinator and subsequently to the Worker Nodes is pictured in Figure 3.1.

Figure 3.1: Distributed rendezvous simulator architecture

In this thesis, two major components of the distributed rendezvous system are developed. First is the network topology, which all of the simulator's components are dependent upon. Second is the Workload Generator module, which produces timestamped rendezvous requests that serve as input to the Coordinator module.

3.3 Internet Topology Maps

The network topology is a vital component of any networking simulation. Future Internet researchers often strive to demonstrate that their systems are capable of replacing the current Internet's architectural components. To this end, it is highly desirable to use a network topology which resembles the actual structure of the Internet as closely as possible. As discussed in Section 2.1, existing BGP-derived AS-level topology datasets are known to be missing the majority of peering links due to the valley-free routing policy. While the prior evaluation of PURSUIT's rendezvous system introduced additional peering links to an AS-level topology dataset at random, we investigate an alternative approach, analyzing a recent attempt to capture peering links which are missed by BGP route monitors.

Augustin, et al. [66] attempted to identify the AS-level topology's 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 methodology was based around the fact that IXPs typically have a dedicated internal subnet. The addresses of the IXP-facing router interfaces for each AS are within this IXP subnet, which enables the identification of IXPs in traceroute paths. They began by compiling a list of known IXPs and their prefixes from several public IXP databases. To identify IXP member ASes and their peerings, three techniques were used. The first and most reliable technique was to pull mappings from BGP routing tables of route monitors and looking glasses located at IXPs. Since BGP peerings at the IXPs use addresses within the internal IXP subnet, routing

table-derived peerings are guaranteed to be accurate. The second method used traceroute data to determine IXP peerings. Consider the simple IXP topology shown in Figure 3.2. In this example, the labels IXP1, IXP2, and IXP3 are the IP addresses of IXP-facing router interfaces, while A1, B1, C1 and C2 are the addresses of AS-facing interfaces. If the path A1 → IXP3 → C2 is encountered in a traceroute, there is a high probability that ASes A and C have a peering relationship. To reduce false positives introduced by the fact that routers may respond to traceroute probes using any of their interfaces, a *majority-selection* heuristic was applied. This approach simply favors the most frequently-occurring address when multiple ASes are detected following the same IXP-facing address. This heuristic is based upon the fact that routers will *usually* respond to traceroutes using the incoming interface as the source address. The resulting dataset contains over 40,000 high-confidence peering links.

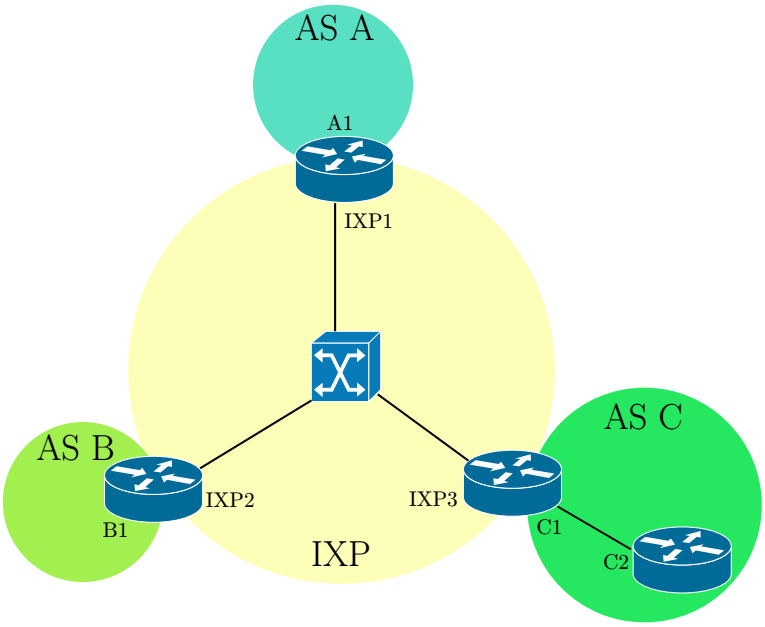

Figure 3.2: Example IXP topology

3.3.1 Dataset Analysis

The two most widely used AS-level Internet topology datasets are the CAIDA AS Relationship dataset (hereinafter CAIDA) and the UCLA Internet Research Laboratory's AS-level topology (hereinafter UCLA). The most recent version of CAIDA was generated on January 16, 2011. We retrieved

the UCLA dataset which was generated on the same date and performed a comparative analysis of these two topologies[1]. Their properties are summarized in Table 3.1. Note that UCLA also contains 289 unclassified links in addition to those listed in the table.

Table 3.1: 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 revealed that only a single AS is absent from UCLA's dataset but present in CAIDA's. Of CAIDA's inter-AS links (not considering relationship annotations), 329 do not appear in UCLA. However, when we also considered the AS relationship, we found that the UCLA and CAIDA datasets disagree about the AS relationships of 34,908 links.

The IXP Mapping Project dataset which we introduced in the previous subsection (hereinafter IXP) contains 53,119 unique peering links, of which 40,076 are high-confidence, 3,801 are medium-confidence, and 9,242 are low-confidence. High-confidence mappings are those which have been observed in both directions (e.g., both AS1→IXP→AS2 and AS2→IXP→AS1), or where both ASes are known to be members of the IXP. Medium confidence links either contain a verified IXP member or have been assigned by the majority selection process, and low confidence links do not have enough data to be verified. We discarded all low and medium-confidence links and performed our analysis using only the 40,076 high-confidence links.

The high-confidence IXP links contain 2,974 unique ASes. Of these ASes, 309 are absent from CAIDA and 200 are absent from UCLA. 14,542 of the IXP peering links also appear in UCLA. Of these links, UCLA considers 820 to be customer-provider links and does not classify 13. While 10,608 of the peering links appear in CAIDA, 9,191 of these are believed to be customer-provider links.

From our analysis of the UCLA, CAIDA, and IXP datasets, we made the following observations:

[1]The UCLA dataset uses the ASDOT notation [67] to represent AS numbers. In ASDOT, AS numbers above 65535 are split into two 16-bit decimal integers separated by a period. The number before the period represents the high-order bits and the number after the period represents the low-order bits. Since CAIDA uses ASPLAIN notation, we converted ASNs in UCLA from ASDOT to ASPLAIN. For example, ASN 4.533 was converted to 262677.

1. UCLA captures more links over more ASes than CAIDA, including a significant number of additional peering links.
2. CAIDA categorizes many links as customer-provider which both UCLA and IXP consider to be peering links.
3. IXP contains many peering links which do not appear in UCLA or CAIDA.

Given these observations, we compiled a hybrid AS-level topology which unites the UCLA and IXP datasets[2]. We utilized a more recent version of the UCLA dataset captured on May 6, 2012 (we refer to this as UCLA*) than the one used in the comparison with CAIDA. 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 3.2.

Table 3.2: 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

3.3.2 Routing

In order to ensure that only valid policy-compliant paths are used in the simulation environment, we selectively export routes between neighboring ASes based on their inferred business relationships. A *valid* path is one where for each transit link, there is a payee who is an immediate neighbor in the path. For example, in the AS structure shown in Figure 3.3, $D \rightarrow C \rightarrow A$ is a valid path because D pays C for transit service and C pays A for transit service. The path $A \rightarrow C \rightarrow B$ is invalid because neither A nor B pay C for transit service, so C should not be expected to forward traffic between them.

[2]We note that a similar application of the IXP dataset was used by Gill, et al. [68]. They utilized a subset of the IXP-derived peering links to create an artificial peering-heavy topology for the purpose of evaluating a Secure BGP deployment strategy.

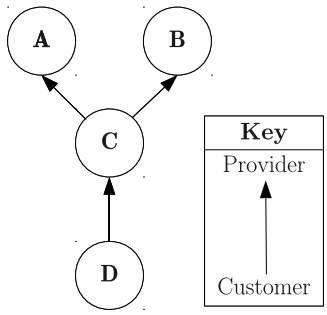

Figure 3.3: Valid path between ASes

We adapted the route export strategy from Gao [30] as follows. For each AS, from the set *neighbors* containing all adjacent ASes, we compute the subsets *customers*, *providers*, and *peers* which contain all neighboring customer, provider, and peer ASes, respectively. All of the AS' routes are then classified into customer, provider, and peer routes based on the first hop. Then we assume that an AS prefers a customer route over a peer route and a peer route over a provider route. If multiple paths of the same type exist, then the shortest path is chosen. If multiple paths of the same type and length exist, then the next-hop with the lower ASN is preferred. The reason for making our earlier assumption is that an AS would ideally prefer to route through a customer (who pays for the transit) or alternatively through a peer (where no party is paid), and if no alternatives exist, through a provider (who must be paid to provide transit). Routes are then exported to neighboring ASes using Algorithm 1, such that customer-provider and peer-to-peer relationships are maintained.

Algorithm 1 Export routes

for each AS $x \in neighbors$ **do**
  if $x \in providers \cup peers$ **then**
    export all customer routes to x
  else if $x \in customers$ **then**
    export all customer, peer and provider routes to x
  end if
end for

3.4 Application Traffic Modeling

Having compiled a suitable Internet-like network topology upon which to perform our evaluation, we now shift our focus to the publish and subscribe requests which serve as input to the rendezvous simulator. As discussed in Section 2.2, the vast majority of the information transferred on the Internet is dominated by two classes of traffic: web and P2P. In order to model the rendezvous requests generated by web and P2P applications, it is first necessary to gain a deep understanding of each of their traffic characteristics. Our model for P2P traffic is based upon BitTorrent, since it was found to be significantly more popular than any other P2P application, as mentioned in Section 2.2.

To our knowledge, no design specifications for ICN implementations of the web or BitTorrent have been published. Therefore, we are forced to make assumptions about the rendezvous traffic which these applications would generate. To keep the number of necessary assumptions to a minimum, we make our ICN application models as simple as possible. Our workload generation methodology involves applying empirically-observed aggregate models to generate rendezvous requests which *might* be produced by the hypothetical information-centric versions of each application. As we are only concerned with evaluating the rendezvous component of the PURSUIT architecture, we do not attempt to model application payloads.

The models developed in this section and Sections 3.5-3.7 determine the functionality of the Workload Generator, a Python module which delivers timestamped rendezvous requests to the Coordinator module of the distributed rendezvous simulator. The generated rendezvous requests have the following format:

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

The $Timestamp$ field 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 rendezvous request is a publish or subscribe request. RId is the 256-bit rendezvous identifier corresponding to the request. The ASN field represents the autonomous system where the request originated. For example, <00002500, Publish, 8a04c201.., 2501> represents a publish request originating from AS2501 which is issued 2.5 seconds after after the start of the simulation.

3.4.1 Traffic Volume

According to the Cisco Visual Networking Index [69], a total of 20,151 petabytes (20,634,624 terabytes) of traffic were transferred on the Internet per month in 2010. This translates to an average throughput of 7.84 terabytes per second. This same report projected monthly usage of 37,603 petabytes per month in 2012 and 80,456 petabytes per month in 2015. IDC [70] provided a slightly more conservative estimate for 2010 of 9,665 petabytes per month, but their report predicted that traffic will increase to an enormous 116,539 petabytes per month in 2015.

Due to the differences in total traffic estimates and the fact that the volume of traffic on the Internet is constantly increasing, the overall throughput is configurable as a parameter in the Workload Generator. This parameter, which is named *Throughput*, is one of the most important parameters in the Workload Generator. The application traffic models which we introduce in Sections 3.4.3 and 3.4.4 are based on the total throughput, thus this parameter significantly influences the number of rendezvous requests created by the Workload Generator per unit time.

The *Throughput* parameter is the first of several Workload Generator parameters which we introduce in this chapter. This particular parameter allows the distributed rendezvous simulator to utilize new Internet traffic studies as they are made available, and it also enables us to evaluate the rendezvous system in a number of different scenarios. A collection of all of the Workload Generator's parameters is presented in Section 3.7, along with a description of each parameter.

3.4.2 Object Collections

Recall from Section 2.5.2 that each individual object in PURSUIT is uniquely identified by a rendezvous identifier, and that a scope identifier is a special instance of a RId which is used to group related objects. Rajahalme, et al. defined an *object collection* as a globally-reachable object which captures the structure of multiple other objects (e.g., a photo album containing multiple photos). They intended that a user should only need to issue a single rendezvous request to receive every object in an object collection and suggested that each collection would be represented by a scope identifier. Thus, a user who creates a rendezvous request for a scope identifier would be able to receive all the items falling under that scope.

Rajahalme, et al. claimed that natural incentives exist for the use of object collections over individual object registrations. They assumed that the registration of a globally-reachable rendezvous identifier would likely cost something, so it would be in the best interest of content providers to aggregate identifiers wherever possible. Additionally, they claimed that object collections could reduce user-perceived latency by reducing the number of rendezvous requests needed to retrieve a collection of objects.

While these assumptions are reasonable, we argue that such object collections are not robust enough to support the behavior of all classes of applications. For example, consider a BitTorrent-like peer-to-peer application which splits large files into numerous chunks and aggregates the chunks under a single scope identifier. In BitTorrent, an interested user retrieves a list of the peers who have the desired chunks and requests different chunks from several peers. An object collection alone would be unable to provide this functionality. One potential solution may be the development of a middleware layer which can coordinate between many peers after a single rendezvous request. However, such a layer has not been defined in any of the PURSUIT project's publications, and we consider the proposal of new architectural concepts to be outside the scope of this thesis. Our goal is strictly to evaluate the architecture as it has been defined in the literature while making as few additional assumptions as possible.

To address this issue, we focus on making our application traffic models as flexible as possible. The characteristics of the rendezvous requests created by the Workload Generator are highly configurable via simulator parameters. In our web and P2P traffic models, we consider both the scenario where a single rendezvous request can acquire multiple objects, as well as the case where each object requires an individual rendezvous request.

3.4.3 Web Traffic

The World Wide Web's foundation lies in the Hypertext Transfer Protocol (HTTP) [71]. HTTP is a stateless request/response application-layer protocol which is used to exchange information between a client and server. A client submits an HTTP request to a server, which performs some task in response to the request (e.g., stores some data or retrieves a document) and sends a response back to the client containing status information about the request in addition to any content which was requested. Figure 3.4 shows a sample HTTP request from a client to a server:

```
GET /documents/sample.html HTTP/1.1
Host: www.example.org
```

Figure 3.4: Sample HTTP request

In this example, the client is requesting a document entitled *sample.html* from the */documents* directory of the server located at *www.example.org*. In order for the client to send this request, first a DNS query must be performed upon *www.example.org* to determine the IP address corresponding to the domain name. Then a TCP connection is opened with the server and the HTTP request is transmitted. Figure 3.5 shows a sample response from the server to the client:

```
HTTP/1.1 200 OK
Date: Sun, 13 May 2012 04:25:12 GMT
Server: Apache/2.2.15 (Red Hat)
Last-Modified: Fri, 11 May 2012 15:00:31 GMT
Content-Length: 402
Content-Type: text/html; charset=utf-8
Connection: close
```

Figure 3.5: Sample HTTP response

This response indicates that the server successfully executed the client's request and that the TCP connection between the client and server will be closed following the delivery of the requested content. The contents of the file specified in the GET request are included after the HTTP header fields and a blank line. In this case, the requested content is a 402-byte HTML document with UTF-8 character encoding.

While this simple example only involves a single DNS query and the transfer of one HTML document, loading the URL of a Web page in a browser generally triggers a number of events. If the requested document is an HTML page, there is a high probability that several additional objects are embedded in the page (e.g., style sheets, Javascript code, bookmark icons, images, videos, or other media types). Individual HTTP GET requests are needed for each embedded object. Additional DNS requests also must be made for any embedded objects which are hosted on other sites.

We refer to Ihm, et al. [72] for a very thorough analysis of modern web traffic. They developed an algorithm which identifies embedded objects in web traffic traces by grouping requests into streams. Using this algorithm, they analyzed the characteristics of web objects in traces from a globally-distributed proxy system with 70,000 daily users spanning 187 countries. Ihm, et al. found that the median size for an entire page, including all of

its embedded objects, was 133 kilobytes. They also found that the majority of web pages contained multiple embedded objects, and the median number of embedded objects per page was 12. We can use these figures to derive a model for the rendezvous traffic generated by web requests.

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. The web throughput is then determined by the *Throughput* and *WebMix* parameters. For example, if *Throughput* is 15GB/sec and *WebMix* is 80%, then web throughput is 12GB/sec. The *WebObjSize* parameter then determines the number of rendezvous requests which are generated per unit time. For example, if the web throughput is 12GB/sec and we set *WebObjSize* to 133KB as observed by Ihm, et al., then 94,608 rendezvous requests for web objects will be generated per second. This scenario assumes that the embedded objects in web pages are all encapsulated within object collections, which would be configured by setting the *WebReqsPerObj* parameter to 1. If we did not assume the presence of object collections, then we could set the *WebReqsPerObj* parameter to 12. In this case, 12 rendezvous requests would be generated for each web page, resulting in 1,135,296 rendezvous requests for web objects per second.

While our model is a very high-level representation of web traffic, its strength lies in the fact that it was derived from the empirical results of a global study. Modeling the behavior of individual web users would not scale to our Internet-wide simulation, thus we opted to use an aggregate model instead. We discuss this shortcoming and its implications in Section 4.2.

3.4.4 P2P Traffic

BitTorrent [73] is a very popular peer-to-peer content distribution system. As mentioned in Section 2.2, BitTorrent is the most prevalent P2P application on the Internet, and thus our P2P traffic model is based on BitTorrent. In BitTorrent, content is divided into *pieces*, which are further split into *blocks*. The publisher creates a .torrent file containing meta-data about the content such as the name and file size, as well as SHA-1 hashes of each piece. The .torrent file additionally contains the address of a *tracker* server, which is responsible for managing the *peers* who are participating in the content distribution. A peer who has all the pieces is known as a *seed*, and those with incomplete content are called *leechers*. A torrent's *swarm* is the set of all of its seeds and leechers. To start retrieving content via BitTorrent, a user

downloads the .torrent file (often from a web server) and contacts the tracker for a list of peers who are sharing the content. Peers exchange information about the blocks they possess and those which they need, determining the rarest pieces in the swarm and requesting them first. Peers decide which other peers to upload to based on their sharing history. This strategy, known as *tit-for-tat*, creates an incentive to share because users who upload more are more likely to experience faster download speeds.

Zhang, et al. [74] presented a very thorough analysis of the BitTorrent ecosystem. They crawled the most popular BitTorrent discovery sites and trackers, collecting statistics about the torrents hosted on them. However, their study was focused upon the characteristics of the peers participating in torrent swarms, rather than the content being distributed by the peers. We are particularly interested in the *size* of BitTorrent content, but Zhang, et al. did not include file sizes in their study. We adapted a subset of their methodology by crawling the website of the most popular BitTorrent discovery site, *The Pirate Bay*[3], and collecting the content size of each torrent on the site.

As of June, 2012, The Pirate Bay is the most popular torrent discovery site on the Internet and the 77th most popular website overall, according to Alexa's web site rankings [75]. We developed a Python program which crawls The Pirate Bay and collects information about each individual torrent file. We ran our crawler over a 16-hour period and collected the content sizes of 1,823,363 torrents. The content size data collected by our crawler is presented in Table 3.3.

Table 3.3: 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

Our results show that there exists great diversity in the sizes of content shared on The Pirate Bay. Given that the mean exceeds the third quartile, we can conclude that a small number of extremely large torrent files have a significant impact on the mean. Thus, the median is a more suitable representation of the data.

The throughput of P2P traffic in the Workload Generator is determined by the *P2PMix* parameter, which serves the same purpose as the *WebMix* parameter in our web traffic model. Returning to our example from the

[3]http://thepiratebay.se

previous subsection, if *Throughput* is 15 GB/sec and *P2PMix* is 20%, then the total P2P throughput is 3 GB/sec. The *P2PObjSize* and *P2PReqsPerObj* parameters determine the number of rendezvous requests which are generated per unit time. Setting *P2PReqsPerObj* to 1 assumes the presence of a middleware layer which can match a single subscriber with multiple publishers (peers). Given a *P2PObjectSize* of 350MB, this scenario would result in the generation of 9 rendezvous requests per second. While we argue that such a middleware layer has not been defined in the current PURSUIT architecture, we believe that this layer could be developed in the future. If we assumed that individual rendezvous requests must be made for each block of a BitTorrent file, with a block size of 64KB, then we would set *P2PReqsPerObj* to 5600, resulting in the generation of 50,400 rendezvous requests per second. 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.

While this simple aggregate model facilitates BitTorrent-like peer-to-peer content distribution, it does not faithfully implement BitTorrent's neighbor or piece selection algorithms. Both of these algorithms require clients to maintain state about and exchange messages with individual hosts. In BitTorrent, the neighbor selection algorithm (*tit-for-tat*) punishes free-riders while rewarding users who share. Without such a mechanism in the ICN equivalent, there is a possibility that some users would simply choose not to share, reducing the availability of the content. Additionally, without the rarest-first piece selection policy, dissemination efficiency and torrent lifetime could suffer. However, the ability to retrieve the nearest copy of the data could yield performance benefits for both users and service providers.

3.5 Object Popularity Distribution

In addition to accurately modeling the behavior of Internet applications, it is important to capture the characteristics of the content they operate over. Object popularity can be challenging to model due to the rapidly-evolving nature of content on the Internet. As such, we construct our object popularity models from the most recent empirically derived results.

In this section, we first introduce the concept of a power law distribution, as this type of distribution has been observed in numerous Internet object popularity studies. Next we analyze empirically observed object popularity distributions from the literature, considering which probability distributions

most accurately represent the popularity of web and P2P objects. Finally, we discuss the generation of rendezvous identifiers for simulation events in our Workload Generator.

3.5.1 Power Law Distributions

A power law distribution occurs when the probability of each value is inversely proportional to the power of that value [76]. Zipf's law and Pareto's law are two common ways of expressing power law distributions. Zipf's law was first used to describe the frequency of words appearing in written text. It states that the frequency of any word is inversely proportional to its frequency ranking. That is, the most frequent word occurs twice as often as the second most frequent word and it occurs three times as often as the third most frequent word, and so on. Zipf's law can be expressed as:

$$y \sim r^{-b}$$

where y is the frequency of the word, r is the word's ranking, and b is a constant close to 1, which is known as the *shape parameter*.

Pareto's law is based upon a study of the distribution of income in society. This study varied slightly from Zipf's in that Pareto was interested in determining how many people earn an income greater than x, as opposed to finding the xth largest income. As such, Pareto's law is expressed by a cumulative distribution function (CDF):

$$P(X > x) \sim x^{-k}$$

where x is a person's income and k is the shape parameter. This distribution shows that there are a few extremely wealthy people and that most people earn a relatively low income.

Plotting a power law distribution on a linear scale yields a heavy-tailed distribution which lies very close to the axis. They are often plotted on logarithmic axes to highlight the upper tail section of the distribution. In a log-log plot, a power law distribution appears as a straight line.

3.5.2 Web Objects

The object popularity characteristics of web traffic have been studied extensively. Much of the research in this area was motivated by the search for

optimal web caching strategies. In an early study of web traffic on a university network, Cunha, et al. [77] observed an object popularity distribution which followed Zipf's law with a shape exponent of 0.986. A subsequent study by Breslau, et al. [78] examined web traffic traces from six networks ranging from a university department to a regional service provider. They discovered Zipf object popularity distributions in each of these traces, with shape exponents ranging from 0.64 to 0.83. Mahanti, et al. [79] performed a similar study of web object popularity using traces from three web proxies, one used exclusively on a university network, one hosted by a regional service provider, and the other being a top-level web cache which serves requests from numerous other proxies. The authors reported Zipf object popularity distributions with shape exponents of 0.84, 0.77, and 0.74 for the university, regional and top-level proxies, respectively.

While web caching was a very popular research area during the late 1990s and early 2000s, interest in this topic has been relatively stagnant as of late. Despite the relative scarcity of recent studies on web object popularity, the few recent studies reported results which are consistent with earlier findings. For example, Callahan, et al. [80] observed a Zipf object popularity distribution which remained relatively constant over three years of traffic traces captured between 2007 and 2009 at a small research institute.

Based on these studies, we can conclude that the Zipf distribution is the most appropriate object popularity distribution for web traffic. We present a thorough overview of the Zipf distribution and its parameters in Section 3.5.4 and discuss our methodology for generating Zipf-distributed random variables in Section 3.5.5.

3.5.3 P2P Objects

Following the rapid growth in popularity of peer-to-peer file sharing applications around the year 2000, many studies sought to understand the characteristics of P2P traffic. In an early study of the once popular Kazaa P2P system, Gummadi, et al. [81] observed that the popularity of objects was not Zipf-distributed. This was particularly true with the most popular objects, which exhibited much lower popularity than Zipf-distributed objects would. They explained this by noting an important distinction regarding the difference in popularity distributions between web and P2P objects. With web traffic, a single user may repeatedly request a popular object, for example a web page which updates its content regularly. However, P2P objects are static, thus there is no reason for a single user to request the same object

more than once, assuming the user stores the object locally after downloading it.

Klemm, et al. [82] observed a similar object popularity distribution in the Gnutella P2P network. The body of this distribution was Zipf-like, but the head was flattened, meaning the most popular objects were *not* Zipf-distributed. They expressed this by merging two Zipf distributions with different shape parameters (one for the head, and another for the body). A subsequent study of the Gnutella network by Hefeeda, et al. [83] presented similar results, but the authors found the Zipf-Mandelbrot (ZM) distribution to be a more elegant representation of their data than two merged Zipf distributions with different shape exponents. The ZM distribution is a generalization of Zipf's law with an additional parameter which determines the flatness of the distribution's head.

While P2P systems such as Kazaa and Gnutella were once popular, today's P2P traffic is dominated by BitTorrent [42]. Dán, et al. [84] performed an 11-month study of the popularity of BitTorrent files by querying trackers daily. They observed a Zipf-like popularity distribution, but with distinct head, trunk, and tail regions. In addition to standard tracker servers, BitTorrent also employs DHTs for decentralized torrent tracking. Currently there are two major BitTorrent DHTs – Mainline and Vuze – both of which are based on Kademlia [85]. Wolchok, et al. [86] collected BitTorrent object popularity statistics by crawling the Vuze DHT using a Sybil attack to create thousands of DHT clients which receive metadata from other DHT peers. By monitoring the peer lists associated with certain torrents over time, they were able to measure the object popularity of torrents. They claimed that the popularity distribution appeared Zipfian without performing any curve fitting. However, we can clearly observe a flattened head in their plots, which suggests that a Zipf-Mandelbrot distribution may be a better fit.

Based on these studies, we conclude that a Zipf-Mandelbrot distribution is the most appropriate object popularity distribution for P2P traffic. We present a thorough overview of the Zipf-Mandelbrot distribution and its parameters in Section 3.5.4 and discuss our methodology for generating Zipf-Mandelbrot-distributed random variables in Section 3.5.5.

3.5.4 Distribution Parameters

In the previous two subsections, we explored several empirical studies of object popularity from the literature. We concluded that the object

popularities of web and P2P traffic are best represented by the Zipf and Zipf-Mandelbrot distributions, respectively. In this subsection, we explore the Zipf and Zipf-Mandelbrot probability distributions and their parameters.

The Zipf distribution is a discrete power law probability distribution. The probability mass function (PMF) of the Zipf distribution is defined as:

$$f(x;\alpha,N) = \frac{1}{x^{\alpha}\sum_{n=1}^{N}\frac{1}{n^{\alpha}}} \tag{3.1}$$

where x is the rank (the most popular item has a rank of 1), N is the number of elements and α is the shape exponent of the distribution. We can rewrite the PMF in a more convenient form by noting that:

$$\sum_{n=1}^{N}\frac{1}{n^{\alpha}} = H_{N,\alpha} \tag{3.2}$$

where $H_{N,\alpha}$ is the Nth generalized harmonic number. Thus, we can represent the PMF of the Zipf distribution as:

$$f(x;\alpha,N) = \frac{1}{x^{\alpha}H_{N,\alpha}} \tag{3.3}$$

The cumulative distribution function of the Zipf distribution is defined as:

$$F(x;\alpha,N) = \frac{H_{x,\alpha}}{H_{N,\alpha}} \tag{3.4}$$

The PMF of the Zipf distribution is shown in Figure 3.6 for shape exponent values $\alpha = 1$, 2, and 3, and $N = 10$. Note that the connecting points do not imply continuous data, and that this distribution is only defined for discrete integer values.

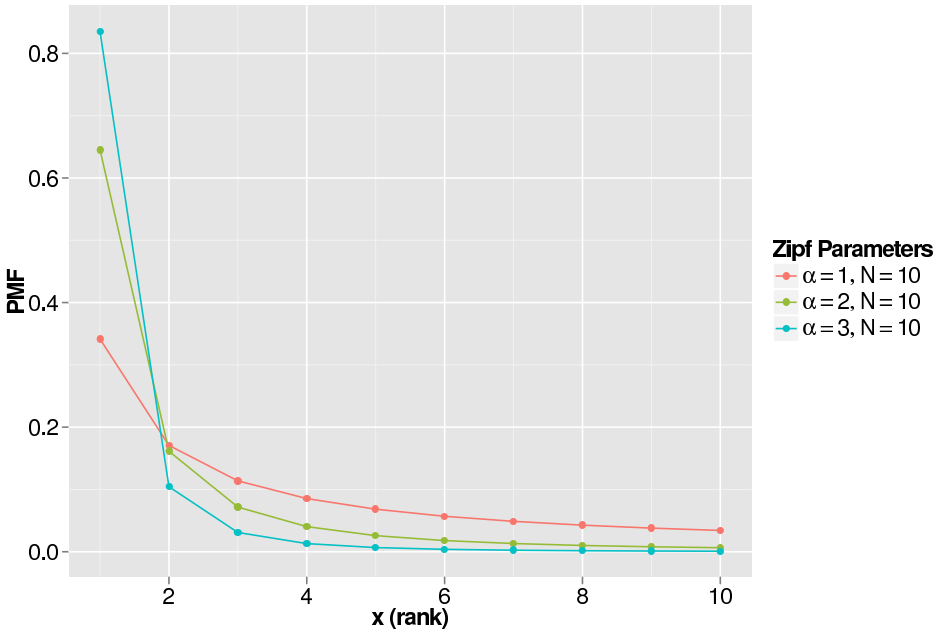

Figure 3.6: Probability mass function of the Zipf distribution

As mentioned in Section 3.5.1, power law distributions appear as straight lines when they are plotted on logarithmic axes. In Figure 3.7, the PMF of the Zipf distribution is shown for the same parameter values ($\alpha = 1$, 2, and 3, and $N = 10$), but as a log-log plot.

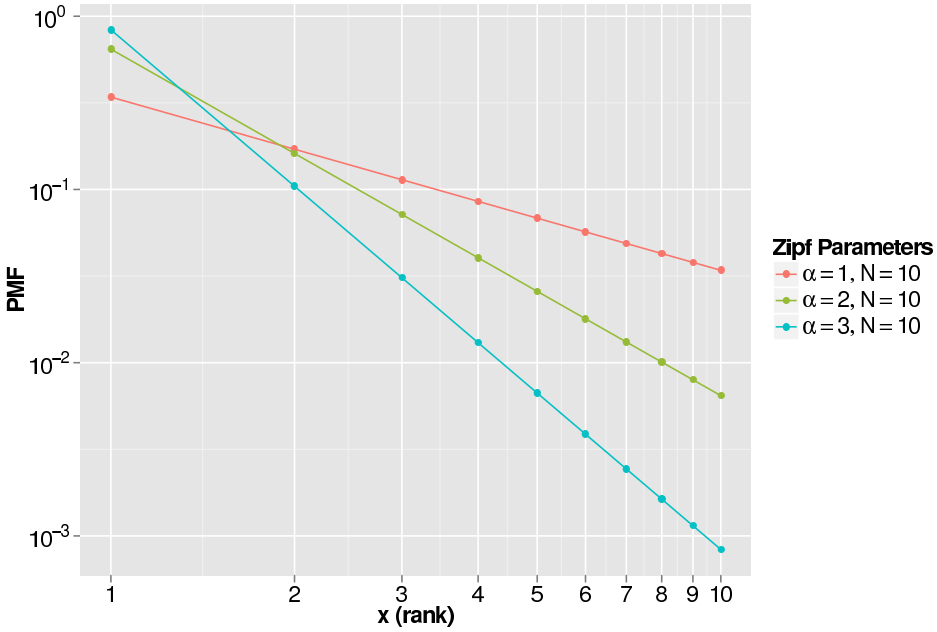

Figure 3.7: Log-log plot of the Zipf distribution's probability mass function

The Zipf-Mandelbrot distribution is a generalization of the Zipf distribution that introduces an additional parameter, q, which we will refer to as the *plateau factor*. It has a PMF of the form:

$$f(x; \alpha, N, q) = \frac{1}{(x+q)^{\alpha} H_{N,q,\alpha}} \tag{3.5}$$

where x is the rank, N is the number of elements, α is the shape exponent and q is the plateau factor. $H_{N,q,\alpha}$ is a further generalization of a harmonic number where:

$$H_{N,q,\alpha} = \sum_{n=1}^{N} \frac{1}{(n+q)^{\alpha}} \tag{3.6}$$

The CDF of the Zipf-Mandelbrot distribution is defined as:

$$F(x; \alpha, N, q) = \frac{H_{x,q,\alpha}}{H_{N,q,\alpha}} \tag{3.7}$$

When $q = 0$, the Zipf and Zipf-Mandelbrot distributions are equivalent. When $q > 0$, the Zipf-Mandelbrot distribution exhibits a flattened head in comparison to a Zipf distribution with the same shape exponent. Figure 3.8 plots the PMF of the Zipf distribution with a shape exponent $\alpha = 3$ and the PMF of the Zipf-Mandelbrot (ZM) distribution with the same shape exponent and three different values of q.

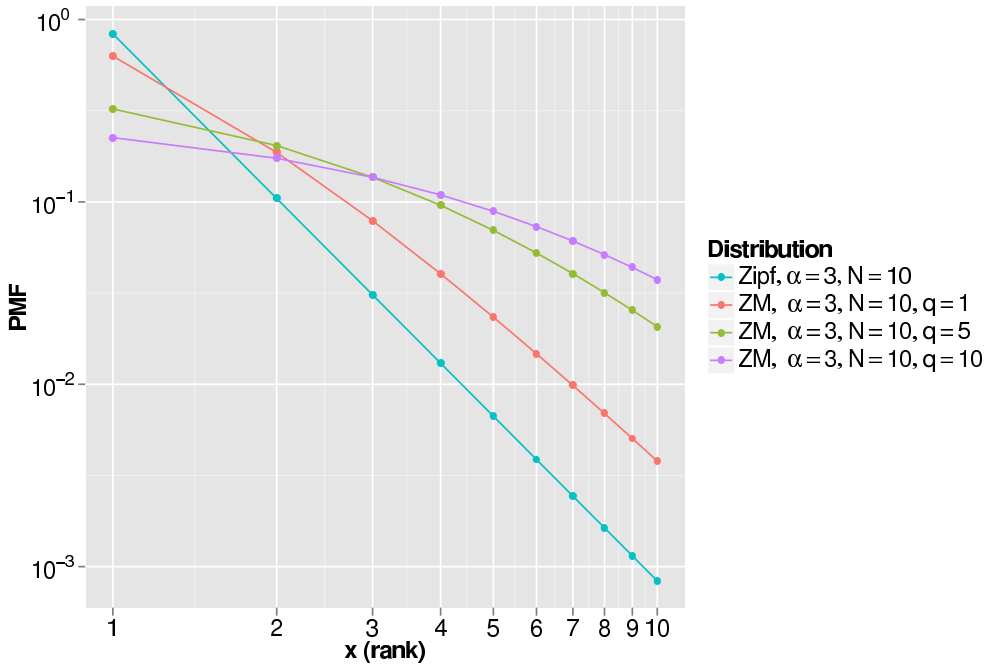

Figure 3.8: Comparison of the Zipf and Zipf-Mandelbrot distributions

3.5.5 Generating Object Identifiers

Although multiple studies have observed Zipf and Zipf-Mandelbrot object popularity distributions in web and P2P traffic, the parameters of these distributions have varied. Thus, we allow the parameters of the object popularity distributions in the Workload Generator to be configurable. This subsection addresses the issue of generating random variables which follow the Zipf and Zipf-Mandelbrot distributions. We use the methodology in this section to produce object identifiers for rendezvous requests in our Workload Generator.

Object popularity is configurable in the Workload Generator via the parameters *WebObjects*, *WebShapeExp*, *P2PObjects*, *P2PShapeExp*, and *P2PPlateau*. The *WebObjects* and *P2PObjects* parameters represent the N parameter of the Zipf and Zipf-Mandelbrot distributions, respectively. In July of 2008, Google reported that their web crawlers had indexed 10^{12} unique URLs [87]. We feel that this figure represents the amount of web objects much more accurately than the number of entries in the DNS, which Rajahalme, et al. used to determine the total number of objects in the prior evaluation of the rendezvous system. Zhang, et al. [74] found that the top twenty torrent discovery sites combined hosted over 1.3 million torrent files. However, when we crawled the Pirate Bay (see Section 3.4.4 for details), we discovered over 1.8 million torrent files. This clearly indicates that the number of BitTorrent objects has increased significantly since 2008, when Zhang, et al. performed their study, and the *P2PObjects* parameter should reflect this fact. The *WebShapeExp* and *P2PShapeExp* represent the α Zipf and ZM parameters. Finally, the *P2PShapeExp* parameter represents the Zipf-Mandelbrot plateau factor, q.

A common technique for generating pseudo-random numbers which follow a particular probability distribution is the *inversion method* [88], which is also known as the *inverse transform method*. It states that if F is the cumulative distribution function and U is a uniform [0,1] random number, then a random variable X can be drawn from the distribution by finding the value such that $F(X) = U$. This can be computed as:

$$X = F^{-1}(U), \tag{3.8}$$

where F^{-1} is the inverse function of F. However, this method is only applicable if F is invertible. The inverse CDFs of the Zipf and Zipf-Mandelbrot distributions do not exist in closed form, so we cannot apply this method directly.

Busari, et al. [89] used a modified form of this approach in their ProWGen workload generation utility. They generated Zipf-distributed random variables as follows:

1. The normalized cumulative distribution function F of the Zipf distribution was computed.
2. For each Zipf-distributed random variable, a uniform [0,1] random number U was generated.
3. A binary search was performed on the CDF for the first position n such that $U \leq F(n)$.

In this approach, n is the Zipf-distributed random variable. Unfortunately, this method is not feasible for large datasets. For example, consider that storing the CDF for 10^{12} objects would require over *3700 gigabytes* of memory, assuming each value is represented as a 4-byte single-precision floating point number. Additionally, the complexity of computing generalized harmonic numbers increases linearly with the size of the input. Thus, if Zipf CDF values were computed on the fly to conserve memory, the complexity of the algorithm for generating *each* random variable would increase drastically from $O(\log\ n)$ to $O(n\ \log\ n)$, which is clearly not efficient enough to be practical.

To address the inability to efficiently compute the CDF of the Zipf distribution for large numbers of objects, we utilize an efficient approximation. We can approximate the CDF of the Zipf distribution by solving the following integral:

$$\int_1^x \frac{1}{z^\alpha}\,\mathrm{d}z = \frac{z^{1-\alpha}}{1-\alpha}\Big|_1^x = \frac{x^{1-\alpha}}{1-\alpha} - \frac{1}{1-\alpha} \tag{3.9}$$

Adjusting for normalization, we define our approximation of the Zipf distribution's CDF[4] as:

$$F(x;\alpha,N) = \frac{\alpha - x^{1-\alpha}}{\alpha - N^{1-\alpha}} \tag{3.10}$$

This approximation allows us to compute Zipf CDF values in constant time, which is a significant improvement over the linear complexity of the actual Zipf CDF. Moreover, since the approximation of the CDF is invertible, we

[4]The approximation is not defined for $\alpha = 1$. In practice, this is not a significant limitation, as we can always use a value for α which is very close to 1.

can draw random variables via the inverse:

$$F^{-1}(x;\alpha,N) = \left((N^{1-\alpha} - \alpha)\left(x - \frac{\alpha}{\alpha - N^{1-\alpha}}\right)\right)^{\frac{1}{1-\alpha}}. \tag{3.11}$$

Although our approximation provides a significant increase in efficiency, this comes at the cost of accuracy. The absolute error between the actual Zipf CDF and our approximation can be computed as:

$$\Delta F(x;\alpha,N) = \left|\frac{H_{x,\alpha}}{H_{N,\alpha}} - \frac{\alpha - x^{1-\alpha}}{\alpha - N^{1-\alpha}}\right|. \tag{3.12}$$

The *percent* error of our approximation is shown in Figure 3.9 for $N = 100$ and $\alpha = 0.8$, 1.2, and 1.5. We note that the percent error is relatively high for the first few elements, but it decreases quickly and approaches 0 as x approaches N.

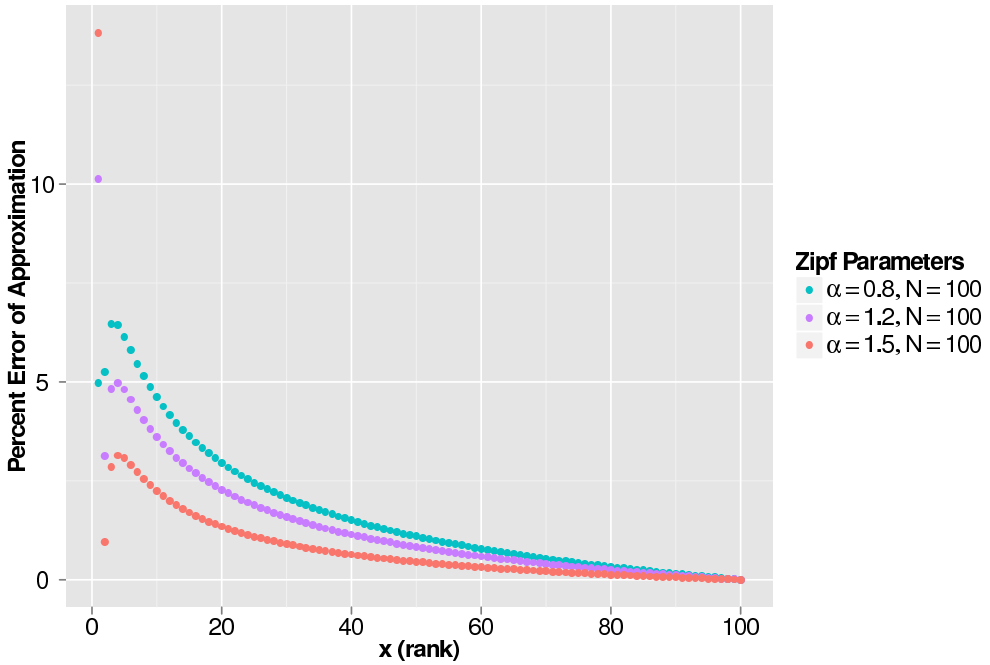

Figure 3.9: Percent error of our approximation of the Zipf CDF

Since the approximation error is greatest for the lowest values of x and it is feasible to compute *some* values of the actual Zipf CDF, we can achieve high accuracy in addition to acceptable efficiency by combining the Zipf CDF and its approximation. We can compute 10^6 values of the Zipf distribution's CDF in a reasonable amount of time (under ten seconds on commodity hardware), and it consumes a reasonable amount of memory (under four megabytes).

Thus, we define a hybrid approach to generating Zipf-distributed random variables based on a pivot value of 10^6. If the number of objects in a simulation run is less than or equal to 10^6, we simply compute the entire Zipf CDF and draw random variables using the binary search method described above. If the number of objects is *greater* than 10^6, then we split the distribution into two parts. The first part consists of the first 10^6 elements of the actual Zipf CDF, and the rest of the distribution uses our approximation of the CDF. We then draw random variables from our hybrid distribution as follows:

1. First, we generate a uniform [0,1] random number U.
2. If $U <=$ the Zipf CDF value at 10^6, then we draw the random variable by performing a binary search on the pre-computed CDF.
3. Otherwise, if $U >$ the Zipf CDF value at 10^6, then we directly apply the inverse transform method on our approximation of the CDF by evaluating Equation 3.11 for $x = U$.

To demonstrate that our approach is not only efficient, but also accurate, we present an example where the Zipf parameters are $\alpha = 1.05$ and $N = 10^7$. Since $N > 10^6$, we apply our hybrid method for generating random variables. The algorithm for pre-computing Zipf CDF values is $O(n)$, but it only needs to be executed once, and as mentioned above, this can be achieved reasonably quickly for 10^6 values. The computational complexity for generating each Zipf-distributed random variable is bounded by the binary search algorithm, which is $O(\log\ n)$, and the best case (where the inverse transform method is applied on our approximation to the CDF) is $O(1)$. The Zipf CDF value at 10^6 is 0.906, meaning that over 90% of the generated random variables will not incur any approximation error. The *greatest* approximation error will occur at $x = 10^6 + 1$, as this is the smallest value for which we use the approximation. The percent error of our approximation to the Zipf CDF at $x = 10^6 + 1$ is 0.36%, which we consider to be a very acceptable amount of error.

We employ a similar method for the approximation of the Zipf-Mandelbrot distribution's CDF by solving the integral:

$$\int_{q+1}^{x} \frac{1}{z^{\alpha}}\,\mathrm{d}z = \frac{z^{1-\alpha}}{1-\alpha}\Big|_{q+1}^{x} = \frac{x^{1-\alpha} - (q+1)^{1-\alpha}}{1-\alpha} \tag{3.13}$$

Adjusting for normalization, we arrive at our approximation of the Zipf-

Mandelbrot distribution's CDF[5]:

$$F(x;\alpha,N,q) = \frac{x^{1-\alpha} - (q+1)^{1-\alpha}}{N^{1-\alpha} - (q+1)^{1-\alpha}}. \tag{3.14}$$

We adapt the same hybrid method for generating Zipf-Mandelbrot random variables as we did for Zipf random variables (described above). Thus, we utilize the inverse transform method for CDF values greater than 10^6 via the inverse of the approximation to the CDF:

$$F^{-1}(x;\alpha,N,q) = \left(xN^{1-\alpha} - (x-1)(q+1)^{1-\alpha}\right)^{\frac{1}{1-\alpha}}. \tag{3.15}$$

Similarly, the approximation error can be computed as:

$$\Delta F(x;\alpha,N,q) = \left| \frac{H_{x,q,\alpha}}{H_{N,q,\alpha}} - \frac{x^{1-\alpha} - (q+1)^{1-\alpha}}{N^{1-\alpha} - (q+1)^{1-\alpha}} \right|. \tag{3.16}$$

To adapt our previous example, consider the case where the Zipf-Mandelbrot parameters are $\alpha = 1.05$, $N = 10^7$, and $q = 2$. The value of the Zipf-Mandelbrot CDF at $x = 10^6$ is 0.893, and the percent error of our approximation to the Zipf-Mandelbrot CDF at $x = 10^6 + 1$ is a very acceptable 0.20%.

3.6 Spatial Locality

By combining the methods developed in Sections 3.4 and 3.5, we can generate rendezvous requests based on empirically-observed web and P2P traffic patterns and object popularities. However, one critical element which remains to be addressed is *spatial locality*. Spatial locality refers to the *location* where a rendezvous request originates. In the context of the distributed rendezvous simulator, locations are autonomous systems in our hybrid UCLA*-IXP AS-level Internet topology.

Introducing realistic spatial locality to our generated traffic is very important for evaluating the requirements of individual rendezvous nodes. It is well known that Internet traffic is unevenly distributed. Recall from Section 2.2 that just 150 ASes originate over half of the traffic on the Internet. Distributing rendezvous requests uniformly among ASes would certainly produce very unrealistic traffic locality. In order to assign spatial locality to

[5]The approximation is valid where $x > q$. This is never a problem in our case, since we only use the approximation for $x > 10^6$, which covers all reasonable values of q.

generated traffic, we must estimate the amount of traffic originated by each AS. One potential approach is to rank the ASes and originate a percentage of traffic proportional to the ranking of each AS.

Several studies have produced autonomous system rankings via a variety of techniques. In CAIDA's AS ranking [90], the rank of an AS is determined by the size of its customer cone (i.e., the number of direct and indirect customers). Labovitz, et al. [40] ranked ASes by observed traffic volumes at 110 ISP vantage points, although four of the top ten ASes were kept anonymous. Ager, et al. [41] ranked *web* content provider (CP) ASes by identifying hosting infrastructures and determining what percentage of popular hostnames are served by each CP.

Table 3.4 compares the AS rankings from CAIDA, Labovitz, et al., and Ager, et al. The CAIDA data was compiled on January 16, 2011. Labovitz, et al. collected their data between July 2007 and July 2009. Ager, et al. began collecting their data in late 2010 and presumably continued the data collection through early 2011.

Table 3.4: Comparison of AS rankings

Rank	CAIDA	Labovitz, et al.	Ager, et al.
1	Level3	Google	Chinanet
2	Hurricane	Anonymous	Google
3	Global Crossing	LimeLight	ThePlanet
4	Metromedia	Akamai	SoftLayer
5	Tinet SpA	Microsoft	China169
6	Sprint	Carpathia	Level3
7	NTT	Anonymous	Rackspace
8	Cogent	LeaseWeb	China Telecom
9	Telia	Anonymous	1&1
10	AT&T	Anonymous	OVH

It is interesting to note that CAIDA's AS rankings consist entirely of large transit providers, while the other two rankings contain a mix of ISPs and CPs. Since we are more interested in where content *originates* than which transit networks it passes through, we believe that CAIDA's AS rankings are less useful for our purposes than the other two rankings. The rankings of Labovitz, et al. feature multiple large web CPs, although all of these except Google are missing from the rankings of Ager, et al., which *exclusively* evaluated web CPs.

While the research community has not reached a consensus regarding the

ranking of autonomous systems, it is clear that a small number of large CPs are responsible for the majority of Internet traffic. Labovitz, et al. [40] found that Google originated over 5% of all Internet traffic in July 2009. Akamai claimed to serve between 15 and 20% of all web traffic in 2010 [91]. Netflix and YouTube combined have been reported to account for over 40% of the downstream peak traffic in the United States [42]. While it is infeasible to obtain accurate traffic volume statistics for *every* CP, we can at least approximate spatial locality for the majority of requests by utilizing estimates for some of the largest CPs.

In addition to capturing the ASes where content is hosted, we need to model the residential networks from which the content is *accessed*. The Internet Systems Consortium [92] attempted to measure the populations of residential networks by crawling the DNS and determining the number of IP addresses which can be resolved via reverse lookup. This technique is flawed, as just because a hostname is assigned to an IP address, it does not necessarily mean that a host exists with that IP address. The same holds in the opposite directions – just because a hostname does *not* exist for an IP address, there is no guarantee that no hosts are assigned the IP address. Chang, et al. [34] ranked residential ASes by collecting IP addresses from P2P swarms and mapping these IP addresses to ASNs. There are some clear issues with this approach as well. First, in order to obtain a diverse sample of users, it is necessary to sample an extremely large number of P2P swarms which relate to many different types of content. Additionally, the accuracy of the rankings is very likely to be affected by the fact that P2P usage is not uniformly distributed among ASes. Moreover, P2P usage is not necessarily a good indicator of web usage. However, while the method of Chang, et al. is clearly not perfect, it is a legitimate, pragmatic approach to the very difficult task of estimating AS access volumes.

The percent of web traffic served by each AS in the Workload Generator is configured via a parameter named *WebProviders*, which is a list of 2-tuples representing ASes and the percent of content that they serve. For example, to simulate the scenario where Google (AS 15169) serves 20% of web traffic and Facebook (AS 32934) serves 10%, we would set *WebProviders* equal to $[(15169, 20.0), (32934, 10.0)]$. The remaining portion of web traffic (in this case, 70%) is uniformly distributed among the other ASes. The spatial locality of web users is modeled by a similar 2-tuple named *WebUsers*.

The spatial locality of BitTorrent content must be considered independently from web content. Although Cuevas, et al. [93] noted that a significant amount of BitTorrent content is *initially* seeded at commercial hosting

providers by profit-driven parties, it is natural for the majority of BitTorrent content to be acquired and subsequently seeded by residential users. While this has the benefit of BitTorrent traffic volumes being roughly symmetric (i.e., the hosts and users are equivalent), it also has the drawback that sizes of residential networks are not as top-heavy or readily available as those of CPs. The spatial locality of P2P providers and P2P users is determined by the *P2PProviders* and *P2PUsers* parameters, respectively. The method presented by Chang, et al. could provide reasonable values for these parameters, given that the set of swarms is sufficiently diverse.

3.7 Workload Generator Design

The Workload Generator is a Python module which combines the models developed in Sections 3.4 - 3.6 to produce rendezvous requests that serve as input to the distributed rendezvous simulator. The traffic volume, object popularity, and spatial locality properties of these requests are configured by a set of input parameters, which are listed in Table 3.5.

Table 3.5: Workload Generator parameters

Parameter	**Description**
Throughput	Overall aggregate throughput (Terabytes per second)
WebMix	Proportion of web traffic
WebObjSize	Size of web objects
WebReqsPerObj	Number of requests per web object
WebObjects	Number of web objects (Zipf parameter N)
WebShapeExp	Web shape exponent (Zipf parameter α)
P2PMix	Proportion of P2P traffic
P2PObjSize	Size of P2P objects
P2PReqsPerObj	Number of requests per P2P object
P2PObjects	Number of P2P objects (Zipf-Mandelbrot parameter N)
P2PShapeExp	P2P shape exponent (Zipf-Mandelbrot parameter α)
P2PPlateau	P2P plateau factor (Zipf-Mandelbrot parameter q)
WebProviders	Per-AS proportion of web traffic served
WebUsers	Per-AS proportion of web traffic accessed
P2PProviders	Per-AS proportion of P2P traffic served
P2PUsers	Per-AS proportion of P2P traffic accessed
P2PShareRatio	Percentage of P2P content which is re-published
P2PShareDelay	Delay between subscription and re-publication

3.7.1 Generating Rendezvous Requests

When a simulation run begins, publish requests are generated for the number of objects specified by the *WebObjects* and *P2PObjects* parameters (all at time 0). The identifiers for web objects are integers starting at 1, where rendezvous identifier 1 represents the most popular web object. P2P objects follow the same identifier scheme, but their values are offset by the RId of the highest (least popular) web object.

With regard to publication, the object popularity and spatial locality parameters are unavoidably coupled. These characteristics are introduced to publications by matching the *WebProviders* and *P2PProviders* with the object popularity distributions for web and P2P traffic. For each AS in *WebProviders* which serves a proportion P of web traffic, we publish the n most popular yet-unpublished objects until the combined object popularity (i.e., the sum of the popularity distribution CDF values for the n objects) reaches P. Consider an example where the top two web providers are Google and Facebook, who serve 10% and 5% of web traffic, respectively, and the four most popular web objects have CDF values of 0.07, 0.05, 0.02, 0.01. In this example, objects 1, 3, and 4 would be published by Google and object 2 would be published by Facebook. Unfortunately, this approach produces imperfect mappings in some scenarios. For example, if the percentage of web traffic served by the highest-ranked AS is *less* than the popularity of the most popular object, then the AS would actually serve more traffic than it was configured to serve. This difference can be computed as:

$$\Delta pop = |P - \sum_{i=1}^{n} C(i)| \tag{3.17}$$

where P is the (configured) proportion of traffic served by the AS, n is the number of objects published by the AS, and C is the set of CDF values for the published objects. We discuss the implications of this in Section 4.2.

The issue of coupling between object popularity and spatial locality does not apply to subscriptions as it does to publications. Subscribe requests for web and P2P objects are produced at rates defined by the traffic volume parameters (i.e., Mix, ObjSize, ReqsPerObj). The object identifier for a subscribe request is produced by our adaptation of the inverse transform method (introduced in Section 3.5.5) using the configured object popularity

distribution parameters. The AS which a subscribe requests originates from is determined by the *WebUsers* and *P2PUsers* parameters for web and P2P traffic, respectively. If multiple requests are made per object, each of these requests originate from the same AS. Subscriptions to P2P objects are followed by corresponding publications of the requested objects with a probability defined by the *P2PShareRatio* parameter, after a delay of *P2PShareDelay* milliseconds.

As previously discussed, it is highly desirable for the distributed rendezvous simulator to be able to approximate the topological structure and traffic characteristics of the current Internet as closely as possible. This can be achieved by selecting parameter values which are derived from the results of empirical studies. We have provided several examples of realistic parameter values in the preceding sections. However, the Workload Generator's flexible parameters also introduce the ability to evaluate the behavior of the rendezvous architecture in hypothetical scenarios. For example, it may be interesting to configure simulations based on future projections, such as IDC's prediction that the observed volume of traffic on the Internet in 2010 will have increased tenfold by 2015.

Chapter 4

Discussion

This thesis presented a network topology and application traffic models for the simulation-based evaluation of the PURSUIT rendezvous architecture. In this chapter, we discuss the implications of our methods and summarize their shortcomings. First, we explain our decision to use an AS-level Internet topology in our evaluation methodology. Next, we consider the implications of using high-level aggregate models to represent Internet traffic. Finally, we reflect on the ethical implications of the research conducted in this thesis.

4.1 Topology

Roughan, et al. [94] claimed that although most published research dealing with the Internet's AS-level topology *seems* scientifically sound, many such studies contain subtle flaws which can lead to specious findings. They warned against abstracting ASes into simple nodes without internal structure and concluded that current Internet topology datasets are *not* sufficiently mature to be used in most meaningful simulation studies. Since the authors of this article include some of the leading experts on Internet topology mapping, we feel obligated to justify our use of an inferred AS-level network topology.

The idea of leveraging a dataset which cannot be validated should be a cause of concern for any scientific researcher. However, this is a reality of Internet topology data which is unlikely to change in the near future. Even as new and improved inference methods are developed, the Internet's topology will still be determined by competing entities, many of whom are unwilling to share their connectivity information. While we cannot fully validate the AS-

level topology, it is possible to perform *partial* validation with the help of partial ground truth. Oliveira, et al. [32] took a pragmatic approach to validating the UCLA topology dataset by comparing the inferred topology data to ground truth from a small number of ASes. This study confirmed that BGP route monitors miss the majority of peering links, a deficiency which we addressed by augmenting the UCLA dataset with additional peering links which were gathered in a study of Internet exchange point members.

It is clear that significant work remains in the area of Internet topology mapping. An accurate PoP-level topology would partially capture the *internal* structure of ASes and could significantly increase the credibility of Internet topology datasets. While recent studies such as iPlane [95] and the Internet Topology Zoo [96] have made some progress in this area, these studies depend on unreliable traceroute data and limited cooperation from service providers. In addition to capturing the internal structure of ASes, PoP-level topology maps can be augmented with accurate *geolocation* data, since unlike an AS, a PoP represents a single physical location. This creates the potential for improved latency and spatial locality models.

We believe that simulations which use inferred AS-level topologies *can* produce meaningful results, but only if the data is treated as an *imperfect approximation* of the Internet's topology and nothing more. It is very important to understand the limitations of inferred topology data and critically analyze all results with these limitations in mind.

4.2 Traffic Models

We attempted to capture the characteristics of Internet traffic through the use of high-level aggregate models. The main advantages of employing such coarse-grained models are that they scale very well, and they are very simple. However, these benefits come at the expense of expressiveness. For example, our model for web traffic does not consider the fact that streaming a video, downloading a file, and browsing images on a social networking site all produce traffic with different characteristics. By aggregating all web traffic into one high-level model, we limited our ability to explore a wide variety of scenarios within the context of web traffic. Since the volume of traffic attributable to video streaming has been observed to be increasing as of late, it may be desirable to simulate the hypothetical situation where video streaming represents 80% of all web traffic. However, our model is too coarse-grained to fully capture such a scenario. In order to explore

this possibility, we would need to divide web traffic into several sub-classes and construct individual models for each sub-class. This would require significantly modifying the base model for web traffic, since it is non-trivial to simply *extract* one sub-class of traffic from an aggregate model.

The high level of abstraction in our application traffic models also imposed limitations on our models of object popularity and spatial locality. Consider the characterization of YouTube traffic locality presented by Brodersen, et al. [97], which reported that for over half of YouTube's videos, 70% of the total views came from a single geographical region. Unfortunately, our web model is not robust enough to capture video traffic independently of other traffic. Additionally, while we acknowledge that there is value in considering the correlation between the locations where requests are originated and served, we did not attempt to model this property in the Workload Generator, as autonomous systems do *not* represent distinct physical locations.

Although our models capture the two most prevalent types of Internet traffic, they clearly do not come close to approximating *all* traffic on the Internet. A more thorough Internet traffic model would consider additional popular applications, such as email, real-time communication (VoIP, video chat), online games, and FTP. However, it would be very difficult to create models for some of these applications which are both realistic and scalable.

Our traffic models produce a constant stream of rendezvous requests based on an estimate of the Internet's total throughput and empirically-observed characteristics of web and P2P applications. However, in reality, Internet traffic is *bursty*, with peak traffic times varying among geographical regions, and for different types of content. Although we did not develop models for traffic patterns such as flash crowds, we note that they could be introduced as additional traffic types in the Workload Generator. It should also be noted that our models do not account for data replication or CDNs. Additionally, the Workload Generator assumes a static collection of existing objects, rather than attempting to capture the characteristics of Internet content creation.

The most valuable lesson learned while developing the models in this thesis was the importance of *simplicity*. We found that attempting to develop highly-detailed traffic models tended to result in over-complication and tight coupling of variables, in addition to increased overhead. Despite employing high-level models, we inadvertantly introduced correlation between spatial locality and object popularity. We will need to carefully analyze the impact which this unintentional correlation has upon caching during the evaluation of the rendezvous system. Earlier versions of our traffic models contained several more unintentional relationships between variables, which

we eliminated by progressing toward higher and higher-level models. Finally, we arrived at a set of coarse-grained, yet very flexible traffic characteristics which are supported by empirical data.

While the models developed in this thesis should not be viewed as complete or detailed representations of the Internet, they do represent a pragmatic approach to approximating the Internet's topology and traffic patterns. Just as we carefully analyzed the error introduced by our efficient approximations to power law distributions in Section 3.5.5, studies which leverage imperfect data should be assessed critically with regard to the limitations of the data. Ultimately, we believe that the goal of a study which attempts to simulate an Internet-like environment should be to identify *invariants* by simulating multiple different scenarios and determining which observed characteristics persist in all cases. We designed our Workload Generator around a set of flexible parameters for this very purpose.

4.3 Reflections

The Internet is thoroughly embedded in many aspects of society. Research which proposes modifications to the Internet's architecture often involves issues such as privacy, censorship, and network neutrality, which have many social and ethical implications. Since our work focused on developing methods for evaluating *previously* proposed architectural changes, these issues did not fall within the scope of this thesis. Overall, we have attempted to conduct this thesis responsibly and in accordance with KTH's policies regarding ethical research. We note that the PURSUIT project should consider developing a rigorous framework for quantifying the social impact of design choices, as advocated by Brown, et al. [98].

To our knowledge, this thesis did not deal directly with any issues which have environmental implications. We note that the distributed rendezvous simulator *could* be extended to evaluate the energy consumption of the PURSUIT rendezvous system.

The data we gathered from The Pirate Bay was collected in a way which preserved the privacy of the website's users. We did not attempt to gather any information which could be used to identify individual users. Our crawler simply collected the overall statistics of each torrent as reported by the torrents' information pages. We did not connect to the torrent trackers or attempt to retrieve information about the peers participating in the swarms.

Chapter 5

Conclusions and Future Work

This thesis developed the network topology and application traffic models for the simulation of a name-based interdomain routing system. The simulation environment was designed to resemble the Internet as closely as possible, in order to demonstrate that this routing system can serve as a viable alternative to traditional Internet routing. The main contributions of this thesis are:

1. a hybrid AS-level Internet topology which captures more peering links than any publicly available dataset,
2. methods for generating the rendezvous requests which may be produced by the information-centric equivalents of popular Internet applications,
3. an approach for efficiently and accurately assigning identifiers to rendezvous requests such that the popularity of objects follows a given probability distribution, and
4. a technique for introducing AS-level spatial locality to generated traffic.

Evaluating proposed modifications to the Internet's core architecture is a challenging task. The Internet's topological structure and traffic patterns are difficult to capture and nearly impossible to validate. Additionally, the size and complexity of the Internet creates a delicate tradeoff between a simulation's scalability and its level of detail. The simulation environment developed in this thesis balances these two properties through the use of *aggregate* models, which were derived from empirically-observed *invariants*, as suggested by Floyd, et al. [61].

The first step in constructing the AS-level topology was to analyze topology inference methods from the literature and compare publicly available

datasets. It was determined that the UCLA Internet Research Lab's AS-level topology dataset [28] captured significantly more *customer-to-provider* links than any other dataset, but that it lacked numerous *peer-to-peer* links, which are invisible to BGP route monitors. The UCLA dataset was augmented with additional peering links collected in a study of Internet exchange point members [66], resulting in a hybrid topology dataset with more than 75,000 customer-to-provider links and 105,000 peer-to-peer links.

The remaining contributions of this thesis, namely models for application traffic, object popularity, and spatial locality, represent a methodology for generating rendezvous requests based on empirically-observed characteristics of Internet traffic. These models were realized in the Workload Generator, a traffic-generating module within a distributed simulator for the PURSUIT rendezvous system. The models are expressed through a set of flexible parameters, thus enabling the simulation of numerous scenarios. In addition to being directly applicable to the evaluation of the PURSUIT rendezvous system, the models and methods presented in this thesis should offer guidance in other studies which aim to simulate Internet-like systems.

5.1 Future Work

Due to the fact that the core of the distributed rendezvous simulator is still under development, it was not possible to evaluate the PURSUIT rendezvous system in this thesis. However, the models developed in this thesis are intended to be used directly in a future simulation-based evaluation of the rendezvous architecture. This future evaluation should leverage the strong points of the prior evaluation performed by Rajahalme, et al. [11] (e.g., rendezvous network formation), while applying the improved network topology and traffic generation models developed in this thesis.

In addition to measuring the performance and scalability, the future evaluation should model possible attacks which aim to disrupt rendezvous service. One such attack could be performed by commanding a large botnet to subscribe to the same RId simultaneously, causing the rendezvous node which is responsible for the RId to become overwhelmed with requests. This attack could be modeled as an additional traffic type in the Workload Generator, and its impact could be measured by monitoring the processing delay at the targeted rendezvous node. We anticipate that this attack will be particularly difficult to detect because the generated rendezvous requests are not easily differentiable from legitimate requests for a very popular object.

Bibliography

[1] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff, "A brief history of the Internet," *SIGCOMM Computer Communication Review*, vol. 39, pp. 22–31, October 2009.

[2] D. Trossen, M. Sarela, and K. Sollins, "Arguments for an information-centric internetworking architecture," *SIGCOMM Computer Communication Review*, vol. 40, pp. 26–33, April 2010.

[3] J. F. Gantz, D. Reinsel, C. Chute, W. Schlichting, J. McArthur, S. Minton, I. Xheneti, A. Toncheva, and A. Manfrediz, "The diverse and exploding digital universe: An updated forecast of worldwide information growth through 2011," *IDC white paper*, March 2008. Available: `http://www.emc.com/collateral/analyst-reports/diverse-exploding-digital-universe.pdf`.

[4] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '07, pp. 181–192, ACM, August 2007.

[5] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, CoNEXT '09, pp. 1–12, ACM, December 2009.

[6] N. Niebert, S. Baucke, I. El-Khayat, M. Johnsson, B. Ohlman, H. Abramowicz, K. Wuenstel, H. Woesner, J. Quittek, and L. Correia, "The way 4WARD to the creation of a future Internet," in *Proceedings of the 19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, PIMRC 2008, pp. 1–5, IEEE, September 2008.

[7] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, and P. Nikander, "LIPSIN: line speed publish/subscribe inter-networking," in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, SIGCOMM '09, pp. 195–206, ACM, August 2009.

[8] M. Gritter and D. R. Cheriton, "An architecture for content routing support in the Internet," in *Proceedings of the 3rd conference on USENIX Symposium on Internet Technologies and Systems - Volume 3*, USITS'01, pp. 37–48, USENIX Association, March 2001.

[9] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," in *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '02, pp. 73–86, ACM, August 2002.

[10] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, and I. Stoica, "ROFL: routing on flat labels," in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '06, pp. 363–374, ACM, September 2006.

[11] J. Rajahalme, M. Särelä, K. Visala, and J. Riihijärvi, "On name-based inter-domain routing," *Computer Networks*, vol. 55, pp. 975–986, March 2011.

[12] "The ns-3 network simulator." `http://www.nsnam.org/`. Accessed May 23, 2012.

[13] K. Lougheed and Y. Rekhter, "RFC 1163 - a border gateway protocol (BGP)," June 1990. Available: `http://www.ietf.org/rfc/rfc1163`.

[14] Y. Rekhter and T. Li, "RFC 4271 - a border gateway protocol (BGP-4)," January 2006. Available: `http://www.ietf.org/rfc/rfc4271`.

[15] X. Cai, J. Heidemann, B. Krishnamurthy, and W. Willinger, "Towards an AS-to-organization map," in *Proceedings of the 10th annual conference on Internet measurement*, IMC '10, pp. 199–205, ACM, November 2010.

[16] B. Chinoy and T. J. Salo, "Internet exchanges: policy-driven evolution," in *Coordinating the Internet*, pp. 325–345, MIT Press, 1997.

[17] D. Feldman, Y. Shavitt, and N. Zilberman, "A structural approach for PoP geo-location," *Computer Networks*, vol. 56, pp. 1029–1040, February 2012.

[18] A. H. Rasti, N. Magharei, R. Rejaie, and W. Willinger, "Eyeball ASes: from geography to connectivity," in *Proceedings of the 10th annual conference on Internet measurement*, IMC '10, pp. 192–198, ACM, November 2010.

[19] D. Meyer, "University of Oregon route views archive project." `http://www.routeviews.org`. Accessed February 13, 2012.

[20] "RIPE routing information service project." `https://www.ripe.net/data-tools/stats/ris/routing-information-service`. Accessed April 25, 2012.

[21] H. Chang, S. Jamin, and W. Willinger, "Inferring AS-level Internet topology from router-level path traces," in *In Proceedings of SPIE ITCom 2001*, August 2001.

[22] F. Baker, "RFC 1812 - requirements for IP version 4 routers," June 1995. Available: `http://www.ietf.org/rfc/rfc1812`.

[23] Y. Zhang, R. Oliveira, Y. Wang, S. Su, B. Zhang, J. Bi, H. Zhang, and L. Zhang, "A framework to quantify the pitfalls of using traceroute in AS-level topology measurement," *IEEE Journal on Selected Areas in Communications*, vol. 29, pp. 1822–1836, October 2011.

[24] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz, "Towards an accurate AS-level traceroute tool," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, pp. 365–378, ACM, August 2003.

[25] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, "An analysis of BGP multiple origin AS (MOAS) conflicts," in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, IMW '01, pp. 31–35, ACM, November 2001.

[26] Y. Shavitt and E. Shir, "DIMES: let the Internet measure itself," *SIGCOMM Computer Communication Review*, vol. 35, pp. 71–74, October 2005.

[27] CAIDA, "Archipelago measurement infrastructure." Available: `http://www.caida.org/projects/ark/`. Accessed May 4, 2012.

[28] B. Zhang, R. Liu, D. Massey, and L. Zhang, "Collecting the Internet AS-level topology," *SIGCOMM Computer Communication Review*, vol. 35, pp. 53–61, January 2005.

[29] CAIDA, "The CAIDA AS relationships dataset." Available: `http://www.caida.org/data/active/as-relationships/`. Accessed February 13, 2012.

[30] L. Gao, "On inferring autonomous system relationships in the Internet," *IEEE/ACM Transacations on Networking*, vol. 9, pp. 733–745, December 2001.

[31] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, k. claffy, and G. Riley, "AS relationships: inference and validation," *SIGCOMM Computer Communication Review*, vol. 37, pp. 29–40, January 2007.

[32] R. V. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang, "In search of the elusive ground truth: the Internet's AS-level connectivity structure," in *Proceedings of the 2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS '08, pp. 217–228, ACM, June 2008.

[33] A. Dhamdhere and C. Dovrolis, "The Internet is flat: modeling the transition from a transit hierarchy to a peering mesh," in *Proceedings of the 6th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '10, ACM, November 2010.

[34] H. Chang, S. Jamin, Z. M. Mao, and W. Willinger, "An empirical approach to modeling inter-AS traffic matrices," in *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, IMC '05, pp. 139–152, USENIX Association, October 2005.

[35] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "PlanetLab: an overlay testbed for broad-coverage services," *SIGCOMM Computer Communication Review*, vol. 33, pp. 3–12, July 2003.

[36] G. Maier, A. Feldmann, V. Paxson, and M. Allman, "On dominant characteristics of residential broadband Internet traffic," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC '09, pp. 90–102, ACM, November 2009.

[37] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and S. Diot, "Packet-level traffic measurements from the Sprint IP backbone," *IEEE Network Magazine*, vol. 17, pp. 6–16, November 2003.

[38] K. Cho, K. Fukuda, H. Esaki, and A. Kato, "The impact and implications of the growth in residential user-to-user traffic," in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '06, pp. 207–218, ACM, September 2006.

[39] H. Schulze and K. Mochalski, "Ipoque: Internet study 2008/2009." Available: `http://www.ipoque.com/sites/default/files/mediafiles/documents/internet-study-2008-2009.pdf`.

[40] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," in *Proceedings of the ACM SIGCOMM 2010 conference*, SIGCOMM '10, pp. 75–86, ACM, August 2010.

[41] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig, "Web content cartography," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement*, IMC '11, pp. 585–600, ACM, November 2011.

[42] Sandvine, "Fall 2011 Global Internet Phenomena Report." Available: `http://www.sandvine.com/downloads/documents/10-26-2011_phenomena/Sandvine%20Global%20Internet%20Phenomena%20Report%20-%20Fall%202011.pdf`.

[43] M. Handley, "Why the internet only just works," *BT Technology Journal*, vol. 24, pp. 119–129, October 2006.

[44] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden, "Tussle in cyberspace: defining tomorrow's Internet," *SIGCOMM Computer Communication Review*, vol. 32, pp. 347–356, August 2002.

[45] D. Trossen and A. Kostopoulos, "Exploring the tussle space for information-centric networking," in *Proceedings of the Telecommunictions Policy Reserch Conference*, TPRC 2011, SSRN, September 2011.

[46] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," in *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '01, pp. 149–160, ACM, August 2001.

[47] P. Ganesan, K. Gummadi, and H. Garcia-Molina, "Canon in G major: Designing DHTs with hierarchical structure," in *In Proceedings of*

the 24th International Conference on Distributed Computing Systems, ICDCS 2004, pp. 263–272, IEEE, March 2004.

[48] D. Smetters, "ccnputfile(1) manual page." `http://www.ccnx.org/releases/latest/doc/manpages/ccnputfile.1.html`. Accessed June 17, 2012.

[49] D. Smetters, "ccngetfile(1) manual page." `http://www.ccnx.org/releases/latest/doc/manpages/ccngetfile.1.html`. Accessed June 17, 2012.

[50] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard, "VoCCN: voice-over content-centric networks," in *Proceedings of the 2009 workshop on Re-architecting the Internet*, ReArch '09, pp. 1–6, ACM, December 2009.

[51] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "RFC 3261 - SIP: Session initiation protocol," June 2002. Available: `http://www.ietf.org/rfc/rfc3261`.

[52] C. Tsilopoulos and G. Xylomenos, "Supporting diverse traffic types in information centric networks," in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, ICN '11, pp. 13–18, ACM, August 2011.

[53] "FP7 PURSUIT project." `http://www.fp7-pursuit.eu`. Accessed February 10, 2012.

[54] D. Trossen, G. Parisis, B. Gajic, J. Riihijarvi, P. Flegkas, P. Sarolahti, P. Jokela, X. Vasilakos, C. Tsilopoulos, S. Arianfar, and M. Reed, "Architecture Definition, Components Descriptions and Requirements. PURSUIT Deliverable 2.3," October 2011. Available: `http://fp7pursuit.ipower.com/PursuitWeb/wp-content/uploads/2011/12/INFSO-ICT-257217_PURSUIT_D2.3_Architecture_Definition_Components_Descriptions_and_Requirements.pdf`.

[55] D. Trossen, G. Parisis, K. Visala, B. Gajic, J. Riihijarvi, P. Flegkas, P. Sarolahti, P. Jokela, X. Vasilakos, C. Tsilopoulos, and S. Arianfar, "Conceptual Architecture: Principles, patterns and sub-components descriptions. PURSUIT Deliverable 2.2," May 2011. Available: `http://fp7pursuit.ipower.com/PursuitWeb/wp-content/uploads/2011/06/INFSO-ICT-257217_PURSUIT_D2.2_Conceptual_Architecture_Principles_patterns_and_sub-components_descriptions.pdf`.

[56] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, pp. 422–426, July 1970.

[57] J. Kjällman, G. Parisis, D. Syrivelis, B. Gajic, M. Reed, C. Tsilopoulos, and C. Stais, "First Lifecycle Prototype Implementation. PURSUIT Deliverable 3.2," September 2011. Available: `http://fp7pursuit.ipower.com/PursuitWeb/wp-content/uploads/2011/09/INFSO-ICT-257217_PURSUIT_D3.2_First_Lifecycle_Prototype_Implementation.pdf`.

[58] C. Stais, D. Diamantis, C. Aretha, and G. Xylomenos, "VoPSI: Voice over a publish-subscribe internetwork," in *Future Network and Mobile Summit 2011*, pp. 1–8, June 2011.

[59] S. A. Baset and H. G. Schulzrinne, "An analysis of the Skype peer-to-peer Internet telephony protocol," in *Proceedings of the 25th IEEE International Conference on Computer Communications*, INFOCOM 2006, pp. 1–11, April 2006.

[60] W. Willinger, D. Alderson, and J. C. Doyle, "Mathematics and the Internet: A source of enormous confusion and great potential," *Notices of the American Mathematical Society*, vol. 56, pp. 586–599, May 2009.

[61] S. Floyd and V. Paxson, "Difficulties in simulating the Internet," *IEEE/ACM Transactions on Networking*, vol. 9, pp. 392–403, August 2001.

[62] J. Jung, E. Sit, H. Balakrishnan, and R. Morris, "DNS performance and the effectiveness of caching," *IEEE/ACM Transactions on Networking*, vol. 10, pp. 589–603, October 2002.

[63] B. Zhang, T. S. E. Ng, A. Nandi, R. Riedi, P. Druschel, and G. Wang, "Measurement based analysis, modeling, and synthesis of the Internet delay space," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, (New York, NY, USA), pp. 85–98, ACM, September 2006.

[64] H. Tangmunarunkit, J. Doyle, R. Govindan, W. Willinger, S. Jamin, and S. Shenker, "Does AS size determine degree in AS topology?," *SIGCOMM Computer Communication Review*, vol. 31, pp. 7–8, October 2001.

[65] I. de Jong, "Pyro - python remote objects." `http://http://packages.python.org/Pyro4/`. Accessed June 17, 2012.

[66] B. Augustin, B. Krishnamurthy, and W. Willinger, "IXPs: mapped?," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*, IMC '09, pp. 336–349, ACM, November 2009.

[67] G. Huston and G. Michaelson, "RFC 5396 - textual representation of autonomous system (AS) numbers," December 2008. Available: `http://www.ietf.org/rfc/rfc5396`.

[68] P. Gill, M. Schapira, and S. Goldberg, "Let the market drive deployment: a strategy for transitioning to bgp security," in *Proceedings of the ACM SIGCOMM 2011 conference*, SIGCOMM '11, pp. 14–25, ACM, August 2011.

[69] Cisco, "Cisco visual networking index: Forecast and methodology, 2010-2015," tech. rep., June 2011. Available: `http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf`.

[70] "Power Users Drive Worldwide Internet Broadband Bandwidth Demand (Press Release). IDC. Mach 14, 2012.." Available: `http://www.idc.com/getdoc.jsp?containerId=prUS23372312`. Accessed May 20, 2012.

[71] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "RFC 2616 - hypertext transfer protocol – HTTP/1.1," June 1999. Available: `http://www.ietf.org/rfc/rfc2616`.

[72] S. Ihm and V. S. Pai, "Towards understanding modern web traffic," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, IMC '11, pp. 295–312, ACM, November 2011.

[73] B. Cohen, "Incentives build robustness in BitTorrent," in *1st Workshop on Economics of Peer-to-Peer Systems*, June 2003.

[74] C. Zhang, P. Dhungel, D. Wu, and K. W. Ross, "Unraveling the BitTorrent ecosystem," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, pp. 1164–1177, July 2011.

[75] "Alexa top 500 global sites." `http://www.alexa.com/topsites`. Accessed June 10, 2012.

[76] M. Newman, "Power laws, Pareto distributions and Zipf's law," *Contemporary Physics*, vol. 46, pp. 323–351, September 2005.

[77] C. R. Cunha, A. Bestavros, and M. E. Crovella, "Characteristics of WWW client-based traces," Tech. Rep. BU-CS-95-010, Computer Science Department, Boston University, July 1995. Available: `http://dcommon.bu.edu/xmlui/bitstream/handle/2144/1571/1995-010-www-client-traces.pdf?sequence=1`.

[78] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: evidence and implications," in *Proceedings of the Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, INFOCOM '99, pp. 126–134, March 1999.

[79] A. Mahanti, C. Williamson, and D. Eager, "Traffic analysis of a web proxy caching hierarchy," *IEEE Network Magazine*, vol. 14, pp. 16–23, May 2000.

[80] T. Callahan, M. Allman, and V. Paxson, "A longitudinal view of HTTP traffic," in *Proceedings of the 11th international conference on Passive and active measurement*, PAM'10, pp. 222–231, Springer-Verlag, April 2010.

[81] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan, "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload," in *Proceedings of the nineteenth ACM symposium on Operating systems principles*, SOSP '03, pp. 314–329, ACM, October 2003.

[82] A. Klemm, C. Lindemann, M. K. Vernon, and O. P. Waldhorst, "Characterizing the query behavior in peer-to-peer file sharing systems," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, IMC '04, pp. 55–67, ACM, October 2004.

[83] M. Hefeeda and O. Saleh, "Traffic modeling and proportional partial caching for peer-to-peer systems," *IEEE/ACM Transactions on Networking*, vol. 16, pp. 1447–1460, December 2008.

[84] G. Dán and N. Carlsson, "Power-law revisited: large scale measurement study of P2P content popularity," in *Proceedings of the 9th international conference on Peer-to-peer systems*, IPTPS'10, USENIX Association, April 2010.

[85] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, pp. 53–65, Springer-Verlag, 2002.

[86] S. Wolchok and J. A. Halderman, "Crawling BitTorrent DHTs for fun and profit," in *Proceedings of the 4th USENIX conference on Offensive technologies*, WOOT'10, pp. 1–8, USENIX Association, August 2010.

[87] J. Alpert and N. Hajaj, "Official Google blog: We knew the web was big...." `http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html`. Accessed May 21, 2012.

[88] L. Devroye, "The inversion method," in *Non-Uniform Random Variate Generation*, pp. 27–39, New York: Springer-Verlag, 1986.

[89] M. Busari and C. Williamson, "ProWGen: a synthetic workload generation tool for simulation evaluation of web proxy caches," *Computer Networks*, vol. 38, pp. 779–794, April 2002.

[90] CAIDA, "AS ranking project." `http://as-rank.caida.org/`. Accessed May 22, 2012.

[91] E. Nygren, R. K. Sitaraman, and J. Sun, "The Akamai network: a platform for high-performance Internet applications," *SIGOPS Operating Systems Review*, vol. 44, pp. 2–19, August 2010.

[92] Internet Systems Consortium, "ISC Internet domain survey." `http://www.isc.org/solutions/survey`. Accessed June 18, 2012.

[93] R. Cuevas, M. Kryczka, A. Cuevas, S. Kaune, C. Guerrero, and R. Rejaie, "Is content publishing in BitTorrent altruistic or profit-driven?," in *Proceedings of the 6th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '10, ACM, 2010.

[94] 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*, vol. 29, pp. 1810–1821, October 2011.

[95] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iPlane: an information plane for distributed services," in *Proceedings of the 7th symposium on Operating systems design and implementation*, OSDI '06, pp. 367–380, USENIX Association, November 2006.

[96] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet topology zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, pp. 1765–1775, October 2011.

[97] A. Brodersen, S. Scellato, and M. Wattenhofer, "YouTube around the world: geographic popularity of videos," in *Proceedings of the 21st International World Wide Web Conference*, WWW '12, ACM, April 2012.

[98] I. Brown, D. D. Clark, and D. Trossen, "Should specific values be embedded in the Internet architecture?," in *Proceedings of the Re-Architecting the Internet Workshop*, ReARCH '10, ACM, November 2010.

TRITA-ICT-EX-2012:122

www.kth.se