GATHERING INFORMATION ABOUT NETWORK INFRASTRUCTURE FROM DNS NAMES AND ITS APPLICATIONS

by

ABHIJIT ALUR

A THESIS

Presented to the Department of Computer and Information Science
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Master of Science

December 2014

THESIS APPROVAL PAGE

Student: Abhijit Alur

Title: Gathering Information about Network Infrastructure from DNS Names and Its Applications

This thesis has been accepted and approved in partial fulfillment of the requirements for the Master of Science degree in the Department of Computer and Information Science by:

Prof. Reza Rejaie	Chair
Prof. Jun Li	Member
and	
J. Andrew Berglund	Dean of the Graduate School

Original approval signatures are on file with the University of Oregon Graduate School.

Degree awarded December 2014

THESIS ABSTRACT

Abhijit Alur

Master of Science

Department of Computer and Information Science

December 2014

Title: Gathering Information about Network Infrastructure from DNS Names and Its Applications

DNS (Domain Name System) names contain a wide variety of information, such as geographic location, speed of the interface, type of interface, etc. However, extracting this information is challenging since this information does not have a consistent format across different ISPs (internet service providers) or even a particular ISP.

We present a new tool, GINIE, which extracts useful information and some common dictionary words from a DNS name. We use three ISPs and a CAIDA (Center for Applied Internet Data Analysis) dataset to demonstrate these capabilities.

Information extracted with GINIE provides valuable insight about the infrastructure of the three ISPs and shows the availability and type of information in a collection of DNS names from many ISPs that exist in a typical dataset. The embedded information from DNS names can be used (with some additional active measurements) to infer the geo-aware topology of an ISP.

CURRICULUM VITAE

NAME OF AUTHOR: Abhijit Alur

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene, OR
B.V.B College of Engineering and Technology, Hubli, Karnataka, India

DEGREES AWARDED:

Master of Science, Computer and Information Science, 2014, University of Oregon
Bachelor of Engineering, Computer Science, 2008, BVB College of Engineering

AREAS OF SPECIAL INTEREST:

Network Measurement, Data Mining

PROFESSIONAL EXPERIENCE:

Systems Engineer, Tata Consultancy Services, 3.9 Years

ACKNOWLEDGEMENTS

I thank Professor and advisor prof. Reza Rejaie for his assistance in the preparation of this manuscript. Special thanks are due to PhD student Reza Motamedi, whose guidance and familiarity with the concepts of network measurement has been helpful throughout this undertaking. I was able to use some of his databases of network measurement data and add build it further. Specifically, the databases of information from RouteViews and Team-Cymru that he had collected was really helpful in my research.

TABLE OF CONTENTS

Chapter Page

Chapter Page

LIST OF FIGURES

Figure Page

Figure Page

LIST OF TABLES

Table Page

Table Page

CHAPTER I

INTRODUCTION

It is important to know the information about Internet infrastructure. It helps in mapping out the topology. Information such as geographical location, the speed, types of interfaces etc gives an idea of network and helps build a complete picture of the Internet. Tier-1 ISPs connect major part of the internet. Finding out their router locations, types of routers : physical and logical, the types of interfaces, speed of interfaces etc is one way of understanding their topological structure. Typical approaches to reconstruct the router level ISPs include traceroute probes, multicast advertisements such as mrinfo[8] as used in MERLIN [19], IP options probing , manual analysis of existing dns records etc.

The active probing techniques mentioned above have some problems. The routers might block some of those requests. Even when the probe is successful, they might not always reveal a lot of information. traceroute might use the delay value to infer information which might not be very satisfactory. The process is slow and we may have to run the process multiple times to possibly get more information or to validate the prior results. Whereas using DNS names for information gathering is both fast and easy. Some of the information such as speed and types of interfaces etc which are very hard to obtain are easily available.

The objective of our work is to come up with a parser for DNS names and extract information such as interface type, router function, geographical location, different companies subleasing from a bigger ISP, dictionary words(which might mean something specific about the configuration) etc. We run our parser on 10 different ISPs and discuss the results of 3 of them in detail in this paper. We also run our parser on publicly available CAIDA dataset of DNS names and compare them against 3 ISPs to see structure across multiple ASes and how they fare against our 3 chosen ISPs. We use IFFINDER and XNET to find out aliases of IP addresses and we find the router-level and city-level topologies of individual companies such as verizon-gni and level3.net.

There are many advantages of using reverse DNS names to map the topological structure find and other useful data. But there are inherent challenges [26]. There is no standard on how different ISPs name their routers. Each one follows their own rules. Many times they are misconfigured. This might lead to erroneous inferences. Sometimes even within an ISP, a set of routers follow one naming conventions and another set follow some other conventions. Many

ISPs don't update the names regularly as they add/move/replace their routers. In such cases our measurement approach doesn't work. But we know about some ISPs which follow good naming conventions and update the dns names of servers as that would help them in monitoring their resources. Typically large ISPs tend to follow this culture.

We find that Level 3 communications, Verizon, Cogent are some of the most connected Internet service providers with large networks in both Europe and North America, and they follow some naming conventions. Some of the reverse DNS names which we found out are given here as an example:

se-4-0.hsa1.Baltimore1.Level3.net

fa1-0-0.burlma2-cr1.bbnplanet.net

are, respectively, the reverse DNS records for the level3 owned routers (they have same ASN).

These records include four pieces of information. First, a router location (e.g., Baltimore and burlma which might mean burlingham massachusetts) we have uncovered around a 50000+ interface names that end with 'level3.net' and almost all of them have full city names. Second, the router code within a location (e.g., hsa1 and cr1) Third, the type of interface, which we infer based on Cisco naming conventions (e.g., te for 10 Gbps Ethernet, and fa for 100 Mbps Ethernet). And fourth, the interfaces position within the router (e.g., 1-0 and 4-0, which are, respectively, the first ports on their line cards). Cr1 also hints that its a core router.

To capitalize on this information we first generated all possible IP addresses in a list of subnets that belong to the 10 ISPs listed in the next section. We then request for their reverse dns names.

We have several interesting results which we are releasing to the research community. These results include:

- DNS name Parser named GINIE
- Information about the interface types used by different ISPs.
- Speeds of the interfaces.
- Location of interfaces of ISPs.
- Router level topology of smaller companies which form the larger AS.

- City level topology of smaller companies which form the larger AS.

CHAPTER II

RELATED WORK

Extracting information from DNS names has been done before. A few notable projects are UNDNS [13] which is described in detail in the paper Measuring ISP Topologies with Rocketfuel [21] and PathAudit [9] described in detail in the paper What's in a name.

UNDNS It uses regular expressions to match against the DNS names and parse the information from them. This approach needs a man-in-the-middle approach. Someone has to come up with the rules by looking at the pattern of DNS names and write a regular expression for it. However there are problems with this approach. The DNS names might change over time. A slight change in the DNS name will render the regular expression written for an ISP useless. Also, there are wide variations in the DNS name formats used by system administrators. To write rules which don't lead to erroneous results, the regular expressions have to be very specific in some cases. This defeats the purpose of writing one regular expression for a group of DNS names. This also leads to a need to write lots of regular expressions and a lot of manual inspection. This makes the process very slow and unrealistic for large ISPs and widespread use.

PathAudit PathAudit [9] is another project described in detail in [14]. It uses a dictionary of information such as city names, interface types etc to check for information in the DNS names. They use clustering algorithms to group names into clusters based on the "tags". These tags are: router function, dots ("."), dashes("-"), alphanumeric names [A-Za-z][A-Za-z]+[0-9], interface speed, IP address in dnsname, and router type (cisco,juniper etc). However, since they use clustering algorithms a situation a smaller subset of a part of a name might be matched first and wrongly infer the tags. For example, a name such as "Fibernet" might lead to a city tag "bern".

GINIE Approach We have come up with a parser called GINIE (Gathering Information from Network InfrastructurE). We too use dictionaries of information of cities, interface types etc. But we use the separators in the DNS names such as dots(".") and dashes("-") to split the names and check against our dictionaries. Based on observation of DNS names we were able to come up with a logical flow which most of the DNS names follow. For example, among all information, interface information always comes first in name (if it is present) before other types of information. Based on observation of city names we have also seen that the DNS names follow CLLI [3] name format for city information.

CHAPTER III

METHODOLOGY

Most network interfaces are assigned with DNS names by their ISPs for ease of management. These domain names usually have some structure to them which depends upon the hardware, the functionality of the router etc. Such a typical name is of the form a7-0.lsanca1-ar53.bbnplanet.net. The DNS names are usually separated by "." (DOT). We already know some parts of this name such as the right-most part "net" is a Top Level Domain (TLD). We encounter other TLDs such as .com, .us etc. "bbnplanet" maybe the company subleasing the address space from Level3 communications because the IP address is addressed in the BGP advertisements to be a part of Level3 ASN(3356). The name to the left of the TLD are usually managed by the individual organizations. Our aim is identify patterns in these names and retrieve as much data from these names as possible.

The data from BGP announcements are captured in projects like RouteViews [20]. The BGP announcements are in the form of prefix to ASN mapping. Team-Cymru also has a large database of IP address to ASN mapping too. We gather information about ISPs and their prefixes from these two sources. We issue reverse DNS queries for a sample number of IP addresses for some ISPs and select a list of about 10 ISPs whose DNS names seem to have good structure (e.g a7-0.lsanca1-ar53.bbnplanet.net.). Usually the large ISPs have a good naming structure. The further steps are broadly listed below:

- Generate all IP addresses in prefixes of the selected ISPs.
- Select a list of public DNS servers from public-dns.tk [10] website.
- Issue reverse DNS queries for all the IPs and store the DNS name and error messages that we get (if any).
- Repair any IP addresses for which we encountered errors by re-sending the reverse DNS queries.
- Build dictionaries of city names, city codes, interface types and their codes, states in United States of America from various sources.

– Use the dictionaries to parse the DNS names that we resolved.

– Validate the cities that we parsed from the names with IP-geo location database such as IP2Location [7]

Generating All IP Addresses in Prefixes

The example prefixes for Level3 ISP are in listed in the table 1. below.

TABLE 1. Prefixes of Level3

Prefix	**Size**
4.0.0.0/10	4,194,304
8.0.0.0/10	4,194,304
62.67.0.0/16	65,536
62.140.0.0/19	8,192
63.208.0.0/13	524,288
64.30.32.0/19	8,192
64.152.0.0/13	524,288
64.200.0.0/16	65,536
65.88.0.0/14	262,144
66.170.136.0/22	1024
67.96.0.0/14	262,144
166.90.0.0/16	65,536
195.16.160.0/19	8,192
195.50.64.0/18	16,384
198.17.30.0/24	256

This table shows some of the prefixes we use that belong to Level3 communications Autonomous System Number (ASN). The size of the prefixes shows all possible IP addresses in that prefix. We generate all possible IP addreses for each prefix for issuing reverse DNS queries for them.

Problems in This Approach There are certain issues with this approach.

– Since we acquire these prefixes from Routeviews [20] and Team-cymru[23] which internally gather these information from Router BGP updates, The size and values of these prefixes might vary from time to time.There may be new prefixes associated with the ISPs as well.

– Some of the IP addresses might not be allocated. Since the only method we use is reverse DNS queries, we cannot be sure whether these IP addresses were allocated. Once approach is to send ping requests to the routers. But this approach isn't fool-proof either since many of the routers block ping requests.

– The ISP might rename the routers from time to time. This might change our view of the ISP and its configuration. They might also move the servers from one location to other. But we assume that these changes are quite slow and happen for only a small section of the routers.

In spite of these drawbacks, analyzing the DNS-names is potentially very useful because they have a lot of information in them and many times the information that we gather using this approach might not be acquired by any other means of active or passive network measurement. Moreover, our methodology can be repeated to find a more up-to-date view of the ISPs.

Selecting The Public DNS Servers

There are many public DNS servers online. Public-dns.tk [10] does a good job of listing all the public DNS servers and their statuses. We only use IPV4 DNS servers because we consider only the IPV4 addresses for the ISPs. There are around 3000 such DNS servers. Some of these servers might not be very efficient. So it is essential for us to weed out the servers which are either slow or have high error rate. We pass the list of servers with 12000 sample (one cycle of our algorithm) queries and check the responses and the behavior of the servers. We remove all the servers with more than 6% error rate. Around 273 addresses have losses more than 6% and they are shown in red to show that they are not used further in our analysis. We also see that most of the DNS servers lie within the 6% error percentage rate. We have such a strict check because we have millions of addresses to resolve the DNS names for and the servers with higher error rate than 10% would eventually have a much higher error rate because of our persistent checks. Below is the figure which shows the server timeout percentage on x axis and the number of servers having them. We delete all these DNS servers from our list. 28 of the DNS servers have 100% timeouts. So in all, 272 servers are deleted from our list.

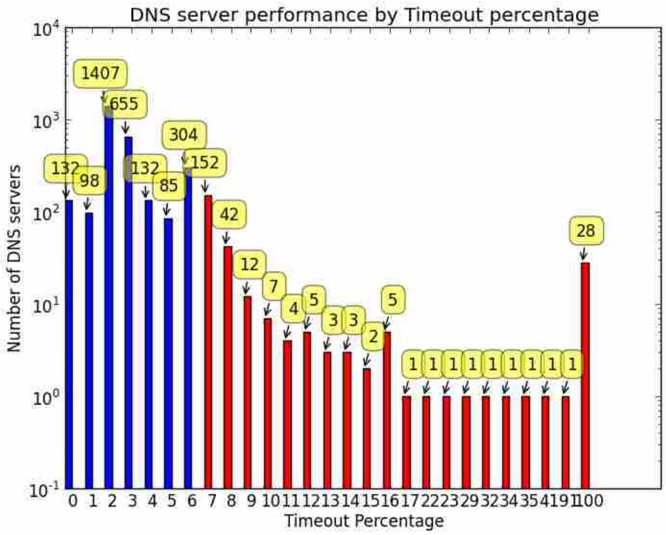

FIGURE 1. DNS Servers Whose Timeout Percentage is More Than 6%

Issuing Reverse DNS Queries

We use `dig` tool to perform reverse DNS queries. Other DNS lookup tools exist such as `nslookup` and `host`. `nslookup` is deprecated. `host` is much more succinct form of `dig`. But we use `dig` as it gives us a lot more information such as

Answer Section: Contains the type of reverse DNS request (in our case it PTR) and the DNS names if it exists.

Question Section: Contains information about the request.

Authority Section: Names of the authoritative DNS servers.

Additional Section: If we query for an MX record, the answer section will show the dns names of the mail servers and the additional section would show the IP address of those name servers (If they are present).

EDNS option: If the DNS server is EDNS enabled, the query is converted into and EDNS dns query and sent to the server. The server then recursively relays the query to the authoritative DNS server for the domain requested in the query. The authoritative DNS server then looks at the EDNS query (The EDNS query contains the prefix of the client which initially made the request) and provides a response which might contain an IP address that is nearer to the client. `host` doesn't have this option.

Statistics: It also shows the statistics of the query such as the time it took for the query to be resolved and the message size received.

which might be useful for analysis later. The following are the responses generated by dig tool and their meanings from RFC 1035 [4]

Setting The Timeout Value in `dig`

`dig` tool has an option for setting the timeout for each query in seconds. To decide an optimal timeout setting for our queries, we ran our script against 300,000 IP addresses of Level3 (Each of the DNS servers would be queried approximately 300 times. This is also the number of requests after which the script is programmed to wait for around 10 minutes.) with values of timeouts ranging from 1 second to 6 seconds. When a DNS server is bogged down by requests, it tends to take a longer (more than the timeout specified in the query) time to respond and hence we are expected to get a higher number of TIMEOUT responses. The distribution of TIMEOUT responses received for each of these set of experiments is shown in 2. and in figure 2..

TABLE 2. Distribution of Number of TIMEOUT or SERVFAIL Responses For Different Timeouts Specified in The Queries

Timeout value specified	**Total No.of IP Addresses**	**No. of TIMEOUT responses**	**Percentage**
1 Seconds	500,127	73,813	14.75%
2 Seconds	500,117	68,850	13.76 %
3 Seconds	500,117	69,439	13.88 %
4 Seconds	500,117	67,590	13.51 %
5 Seconds	489,739	77,739	15.87 %
6 Seconds	480,089	63,765	13.28 %
7 Seconds	480,089	65,507	13.64 %
8 Seconds	480,089	67,614	14.08 %
9 Seconds	480,089	66,541	13.86 %
10 Seconds	480,089	68,569	14.28 %

Methodology of Running Reverse DNS Queries

We produce 100,000 possible addresses from our list of IP address prefixes. We divide them into 4 parts. We pass these 4 lists of IP addresses to 4 new processes. These processes spawn 500 threads each and each of 25,000 addresses are subdivided into 500 parts so that

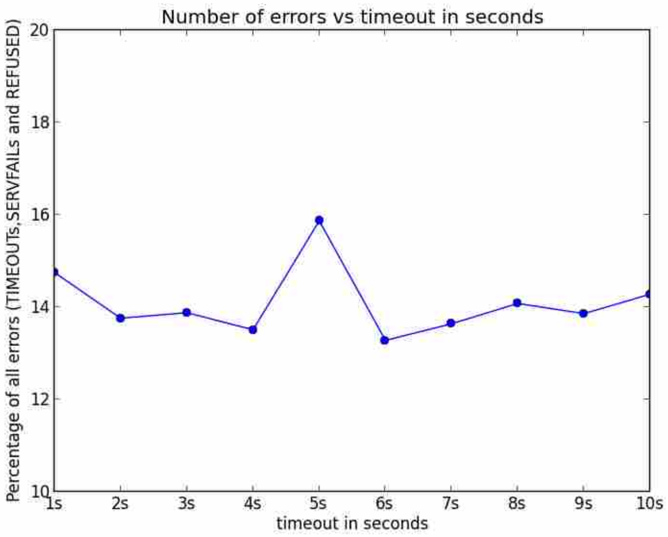

FIGURE 2. Number of TIMEOUTs Observed vs Timeout Value

each thread is responsible for 25 IP addresses. Once all threads of the 4 processes complete, we generate 100,000 more IP addresses. This process constitutes one loop or a cycle. We repeat this process until all the prefixes and IP addresses are exhausted. The program is able to resolve 160-170 IP addresses in a second. After every 300,000 addresses are resolved, we induce a 10 minute sleep for the program so that the DNS servers don't blacklist us. For every reverse DNS query (for each thread), a different DNS server is chosen randomly. On an average, each DNS server is queried 2-3 times a minute.

Correcting The IP Addresses That Had Errors

Some of the reverse DNS requests result in errors as mentioned above namely TIMEOUT and SERVFAIL. We issue reverse DNS requests for these IP addresses again. But to minimize the errors, we reduce the speed with which we query by inducing threads to sleep for random time. We also, choose only those servers which have very high success rate. We list around 600 of such DNS servers which have TIMEOUT errors of less than 1%.

Creating Dictionaries of Interface Names, Router Function, Cities, etc.

The DNS names are composed of a wide variety of information. Broadly the DNS names have the following sections of information (if they are present). [14]

Interface - Which tells about the type of the interface, possibly Its speed, make, interface location (in terms of numbers), model etc.

Router Type - The router type contains information about the function of the router. For example, border router, core router etc.

Location - This contains the information about the location of the router.

The very first thing that needs to be done to analyze the names is to build a database of all these codes so that when we encounter these codes in the DNS names, we can deduce the information present in them. The CLLI codes [3] (which are not to be shared without the permission of Telcodata.us [12]) are stored in city_clli table. The airport codes are stored in city_decode table.

Decoding Interfaces As mentioned before the interface naming techniques by many ISPs follow the naming standards of the company that their routers are made of. Cisco and Juniper are the major vendors of routers to the ISPs. Upon searching the Cisco and Juniper interface naming conventions there are interesting details about the router interface naming procedures. Please find the Cisco and Juniper Interface types mentioned in tables below 3.. Juniper has a much elaborate explanation of the interface naming procedure whereas Cisco just mentions the interface types. In Juniper routers, the physical part of an interface name identifies the physical device, which corresponds to a single physical network connector. This part of the interface name has the format mentioned in the table(Only part of the table is shown here). The full table can be found in [17]). Table shows the Huawei router interface naming guidelines. 4. shows the interface naming guidelines for Cisco routers. Both Cisco and Huawei networks don't explicitly tell how the interfaces might be named. They give guidelines for them. And based on them I have come up with dictionaries for them. For example, F might "FE/GE interface" of any one of the Cisco, Juniper of Huawei. L could be "Simplified Interface" of Huawei network etc.

TABLE 3. Juniper Interface Naming

Code	Description
ae	Aggregated Ethernet interface. This is a virtual aggregated link and has a different naming format from most PICs; for more information— see Aggregated Ethernet Interfaces Overview.
as	Aggregated SONET/SDH interface. This is a virtual aggregated link and has a different naming format from most PICs; for more information— see Configuring Aggregated SONET/SDH Interfaces.
at	ATM1 or ATM2 intelligent queuing (IQ) interface or a virtual ATM interface on a circuit emulation (CE) interface.
bcm	Gigabit Ethernet internal interface.
br	Integrated Services Digital Network (ISDN) interface (configured on a 1-port or 4-port ISDN Basic Rate Interface (BRI) card). This interface has a different naming format from most PICs: br-pim/0/port. The second number is always 0. For more information— see Configuring ISDN Physical Interface Properties.
cau4	Channelized AU-4 IQ interface (configured on the Channelized STM1 IQ or IQE PIC or Channelized OC12 IQ and IQE PICs). ce1 Channelized E1 IQ interface (configured on the Channelized E1 IQ PIC or Channelized STM1 IQ or IQE PIC).
ci	Container interface.
coc1	Channelized OC1 IQ interface (configured on the Channelized OC12 IQ and IQE or Channelized OC3 IQ and IQE PICs). coc3 Channelized OC3 IQ interface (configured on the Channelized OC3 IQ and IQE PICs).
coc12	Channelized OC12 IQ interface (configured on the Channelized OC12 IQ and IQE PICs).
coc48	Channelized OC48 interface (configured on the Channelized OC48 and Channelized OC48 IQE PICs).
cp	Collector interface (configured on the Monitoring Services II PIC).
cstm1	Channelized STM1 IQ interface (configured on the Channelized STM1 IQ or IQE PIC).
cstm4	Channelized STM4 IQ interface (configured on the Channelized OC12 IQ and IQE PICs).
cstm16	Channelized STM16 IQ interface (configured on the Channelized OC48/STM16 and Channelized OC48/STM16 IQE PICs).
ct1	Channelized T1 IQ interface (configured on the Channelized DS3 IQ and IQE PICs— Channelized OC3 IQ and IQE PICs— Channelized OC12 IQ and IQE PICs— or Channelized T1 IQ PIC).
ct3	Channelized T3 IQ interface (configured on the Channelized DS3 IQ and IQE PICs— Channelized OC3 IQ and IQE PICs— or Channelized OC12 IQ and IQE PICs).
demux	Interface that supports logical IP interfaces that use the IP source or destination address to demultiplex received packets. Only one demux interface (demux0) exists per chassis. All demux logical interfaces must be associated with an underlying logical interface.

TABLE 4. Cisco Interface Naming

Type	**Description**
Null	Null interface.
Analysis-module	A Fast Ethernet interface that connects to the internal interface on the
Network	Analysis Module (NAM).
Async	Port line used as an asynchronous interface.
ATM	ATM interface.
BRI	ISDN BRI interface. This interface configuration propagates to each B channel. B channels cannot be configured individually.
BVI	Bridge-group virtual interface. BVI interfaces are used to route traffic at Layer 3 to the interfaces in a bridge group.
Content-engine	Content engine (CE) network module interface.
Dialer	Dialer interface.
Ethernet	Ethernet IEEE 802.3 interface.
Fast Ethernet	100-Mbps Ethernet interface.
FDDI	Fiber Distributed Data Interface.
Gigabit Ethernet	1000-Mbps Ethernet interface.
Group-Async	Master asynchronous interface. This interface type creates a single asynchronous interfaces to which other interfaces are associated. This one-to-many configuration enables you to configure all associated member interfaces by configuring the master interface.
HSSI	High-Speed Serial Interface.
Loopback	A logical interface that emulates an interface that is always up. For example, having a loopback interface on the router prevents a loss of adjacency with neighboring OSPF routers if the physical interfaces on the router go down. The name of a loopback interface must end with a number ranging from 0-2147483647.
Multilink	Multilink interface. A logical interface used for multilink PPP (MLP).
Port channel	Port channel interface. This interface type enables you to bundle multiple point-to-point Fast Ethernet links into one logical link. It provides bidirectional bandwidth of up to 800 Mbps.
POS	Packet OC-3 interface on the Packet-over-SONET (POS) interface processor.
PRI	ISDN PRI interface. Includes 23/30 B-channels and one D-channel.
Serial	Serial interface.
Switch	Switch interface.
Ten Gigabit Ethernet	10000-Mbps Ethernet interface.
Token Ring	Token Ring interface.
Tunnel	Tunnel interface.
VG-AnyLAN	100VG-AnyLAN port adapter.
VLAN	Virtual LAN subinterface.
Virtual Template	Virtual template interface. When a user dials in, a predefined configuration template is used to configure a virtual access interface; when the user is done, the virtual access interface goes down and the resources are freed for other dial-in uses.

TABLE 5. Huawei Interfaces And Their Meanings.

Field	Meaning	Description
A	Product name	AR: application and access routers
B	Hardware platform type. The value can be 1 or 2.	1: four LAN interfaces
		2: eight LAN interfaces
C	Combines with B to indicate different router series using the same hardware platform. The following router series are available:	15: 4*FE LAN interface series
		16: 4*GE LAN interface series
		20: 8*FE LAN interface series
D	Type of major or fixed uplink interfaces on the router	1: FE or GE
		6: ADSL-B/J
		7: ADSL-A/M
		8: G.SHDSL
		9: VDSL over POTS
E	Other interface types supported by the router. This field is optional.	E: enhanced major uplink interface (dual-uplink or two-wire/four-wire DSL enhanced)
		F: uplink GE combo interface

Continued on next page

Table 5. – continued from previous page

Time (s)	Triple chosen	Other feasible triples
		G: uplink wireless interface (GPRS, 3G, or LTE)
		V: voice interface
		W: Wi-Fi access interface
F	Extended information about the router. This field is optional.	HSPA+7: WCDMA HSPA+7 3G standard
		C: CDMA2000 3G standard
	NOTE:	D: DC model
	This field starts with and specifies supplementary interface descriptions or other possible configurations.	P: PoE supported
		L: FDD-LTE, a European standard
A	Product name	AR: application and access routers
B	Hardware platform series code	Currently, three router series are available: 1, 2 and 3. A larger value indicates higher performance.
C	Hardware platform type	2: modular router

Continued on next page

Table 5. – continued from previous page

Time (s)	Triple chosen	Other feasible triples
D	Maximum number of slots supported by the router	AR1200 series: D indicates the maximum number of SIC slots supported.
		AR2200/3200 series: D indicates the maximum number of XISC slots supported.
		NOTE: D can be 0, indicating the cost-effective router model with fixed uplink interfaces or reduced number of slots. E represents the number of fixed uplink interfaces and or reduced number of slots.
E	Fixed uplink interfaces on the router	1: FE/GE
		2: E1/SA
		4: four SIC slots

Continued on next page

Table 5. – continued from previous page

Time (s)	Triple chosen	Other feasible triples
		NOTE: If E is 0, the device has no fixed uplink interface.
F	Other interface types supported by the router. This field is optional.	F: FE LAN interface
		L: simplified interface
		V: fixed voice interface
		W: fixed Wi-Fi access interface
G	Extended information about the router. This field is optional.	A: AC model (AC is the default configuration, and this field can be omitted in AC models.)
		D: DC model
	NOTE:	48FE: 48 fixed 100M switching ports
	This field starts with and specifies supplementary interface descriptions or other possible configurations.	

We stored these descriptions of interface names and their types in our database. Once the interface types of Cisco and Interface naming conventions of Juniper as discovered, its fairly easy to make a fairly accurate guess of the type of interface present in the DNS names. Every name can be checked against these values. And we are a step closer to the process of coming up with a technique to automatically interpret the DNS names without human intervention of writing rules. (The current procedure requires writing rules or regular expressions that explains the classes of DNS names and their meanings.)

Decoding Router Function

Some of the DNS names have coded information about the function the router performs such as border, gateway etc. The codes for these routers are usually br,gw etc. We have stored such router information in a table for use later while parsing the DNS names. An example for such a DNS name is 3e-company.edge2.chicago2.level3.net. This DNS name shows that it is an edge router. Most of the information required to decode the router function is derived from the regular expressions mentioned in [14]. Some other router function is based on observation such as observing that some of the names have 'core', 'gateway', 'border' etc in them.

Decoding Cities

City or region information is abundantly available in DNS names. It is present in 4 forms. In the first case, the city names are fully spelled out. For e.g, `8-2-9.ear1.amsterdam1.level3.net.`. We download the database of world city names from geonames.org. [5]. The database which includes all the cities and their information is too large (9,115,154 cities). The ISPs are not likely to host their routers in cities where the population is less than 5,000 (this conclusion is based on our observation and the probability). So, we use only the cities which have a population of 5,000 or more. The size of this reduced database is 57,021. This also increases the speed of our parser. In the second case, they are present in the form of 3-letter airport codes. For example, `212-162-17-225.edge3.dus1-ge-500`. Here `dus` is an airport code for Dusseldorf, Germany - International airport. It indicates that the router is situated somewhere near the airport, in the same city. We store all the world's airport codes in the database for future use from airportcodes.org [1]. There are 3,833 airport codes. In the third case, 4 letter city names with 2 letter state names are used. Upon some research, the 4 letter city names and the 2 letter state names are mostly the CLLI names used in North American Telecommunication industries. CLLI stands for Common Language Location Identifier code

[3]. These codes are currently owned by telcodata (telcordia telecommunications database) [12]. There are 22,223 CLLI codes. The 4th form is in the form of 2-letter state codes of US states. For e.g, `141-51-97-67-cust-ny.nuvisions.net.` This name states that the router is present somewhere in New York.

Parsing DNS Names

Once we get all the DNS names, we run the parser through two passes. Once we split the names by both ”.” and ”-”. In the the second pass we split by only ”.”. we parse each DNS name to extract the embedded info.

1. we extract each part of the names that are separated by a ”.” and ”-”

2. the right two most part should be com and ISP-name (or something else for leased addresses) we group names based on the two right most parts

3. till the [half of size of array of name segments] +1 of the size of the array of names (actually, and check them against Cisco's, Junipers and Huawei's convention for interface naming. To avoid conflicts with interface codes and location codes, we assume that interface name takes precedence if it exists in the first half of the name. This is because interface names are always at the beginning of dns names (if numbers are present, we ignore those making interface names as the first entities present in a name).

4. checking against 3 letter airport codes. This a standard code taken from airportcodes.org [1]. We have 3613 airport codes.

5. Checking against CLLI-codes. These are maintained by Telcodata and these are proprietary location codes. [12]. We have 22223 CLLI codes of cities. These are codes used in telecommunication. The codes are like DLLSTX which is the code for Dellas, Texas and STTLWA, STTMWA and STTNWA all stand for Seattle Washington. E.g in the DNS name ”evrtwa1-ar2-4-62-114-149.cv.dsl.gtei.net.” evrtwa stands for Everett Washington.

6. Checking against world city names where the population is greater than 5000 obtained from geonames.org [5]. There are 57021 such city names.

7. Checking against 2-letter state codes in United States obtained from Wikipedia [11]

8. We also repeat the above process by splitting only by "." in the second pass. If this results in a higher success in parsing the data, we use the results from this pass and ignore the previous pass.

For example, consider the name s11-0-3-0.london2-cr2.bbnplanet.net. We first split this name by "." and "-". 's11' is of type 's' interface which means it is a serial interface following Cisco's serial interface naming convention. 0,3,0 are not interpreted. They are ignored. london2 is stripped off of numbers and checked against the city. 'cr' defines the router function saying it is a 'Core Router'. bbnplanet.net is the company to which the address space of level3 communications is leased to.

There are certain rules that we follow while parsing names.

1. If a segment(split by either "." or by ".-" depending on which pass it is in) only has numbers, we ignore that segment.

2. We strip all numbers in a code before comparing them to hashmap of codes we have.

3. If a code is followed directly by an English character(without a separator in between), that code won't be found by our method. 99% of the names have separators between logical codes inside a name. For example an airport code SFO sandwiched between other letters of English characters such as airportSFO etc. By observation, we almost never find codes not separated by separators.

CHAPTER IV

CHARACTERISTICS OF INDIVIDUAL ISPS

Selection of ISPs

The ASes used in our study are given in the table 6. below. The sample is selected by resolving the reverse DNS names of a small sample of the addresses in those ASes and checking if they have a well defined naming structure.

TABLE 6. ISPs and Their Details

ASN	ISP Name	Address Space Size
174	COGENT Cogent/PSI	19,984,128
701	UUNET - MCI Communications Services Inc. d/b/a Verizon Business	37,264,384
702	AS702 Verizon Business EMEA - Commercial IP service provider in Europe	6,960,128
703	UUNET - MCI Communications Services Inc. d/b/a Verizon Business	877,056
1239	AS1239 SprintLink Global Network	11,355,200
3356	LEVEL3 Level 3 Communications	10,933,760
5650	FRONTIER-FRTR - Frontier Communications of America Inc.	5,498,368
7018	ATT-INTERNET4 - AT&T Services Inc.	64,134,401
7922	COMCAST-7922 - Comcast Cable Communications Inc.	69,029,376
22394	CELLCO - Cellco Partnership DBA Verizon Wireless	17,186,816
25899	LSNET - LS Networks	186,112
7385	INTEGRATELECOM - Integra Telecom Inc.	1,801,728

Table 7. shows status messages for Level3, Verizon and Cogent in the first run. (Since we observe a lot of TIMEOUT and SERVFAIL errors in the first run, we run the erroneous results in the second run). Verizon seems to have a very high percentage of DNS names and a very low rate of error followed by Level3 and Cogent. In the first run, we focus on the speed of our reverse DNS name resolver to complete large IP address space. In the second run, we select the DNS servers which have error rate of less than 1%(about 600 such DNS servers). And we run our reverse DNS name resolver again with a much slower speed by using lesser threads and inducing wait. Also, we use google DNS server (8.8.8.8) whenever we encounter a SERVFAIL or TIMEOUT as the last check before storing the result as SERVFAIL or TIMEOUT. In the repair

run(second run), 64,333 new DNS names are found in Verizon. 62,298 new DNS names are found in Cogent. 15,739 new DNS names are found in Level3. Table 8. shows the table with different status distributions for Level3, Verizon and Cogent.

TABLE 7. ISPs with Status Distribution Before Repair

Status Message	**Level3**		**Verizon**		**Cogent**	
NOERROR	1,465,536	13.40 %	2,721,661	86.55 %	1,055,825	5.28 %
NXDOMAIN	7,582,902	69.35 %	235,390	7.48 %	14,439,522	72.33 %
REFUSED	516,450	4.74 %	48,023	1.53 %	1,055,110	5.28 %
SERVFAIL	952,011	8.74 %	29,626	0.94 %	773,962	3.87 %
TIMEOUT	1,592,910	14.63 %	109,973	3.49 %	2,638,001	13.21 %

TABLE 8. ISPs with Status Distribution After Repair

Status Message	**Level3**		**Verizon**		**Cogent**	
NOERROR	1,477,998	13.51 %	2,786,216	88.6 %	1,108,234	5.55 %
NXDOMAIN	7,745,486	70.84 %	269,254	8.56 %	17,627,692	88.30 %
REFUSED	16,282	0.14 %	48,520	1.54 %	79,754	0.39 %
SERVFAIL	1,691,460	15.47 %	40,666	1.29 %	1,145,655	5.73 %
TIMEOUT	2,047	0.01 %	17	0.0005 %	1,085	0.0054 %

Table 9. shows the count of all the domains in each ISP. For example, gsa.gov found in Level3 ASN etc.

TABLE 9. Count of All domains in Level3, Verizon and Cogent

ISP	**Number of companies**
Level3	14,403
Verizon	3,759
Cogent	29,908

As we mentioned the number of different domains present in the table 9.. The figures 3., 4. and 5. shows the plot of the different domains and their sizes. The green plot shows the maximum size of the inferred prefix and the blue line shows the number of DNS names we found in that prefix (In other words, it shows the utilization of that prefix). The domains are on the x-axis and are serially indexed. The size of the domains is represented on the y-axis. The y-axis is a log scale to fit all sizes to scale. There are around 30,000 different prefix lists we could find in level3. There are around 3,700 prefix lists that we found for Verizon and around 35,000 prefix lists for Cogent.

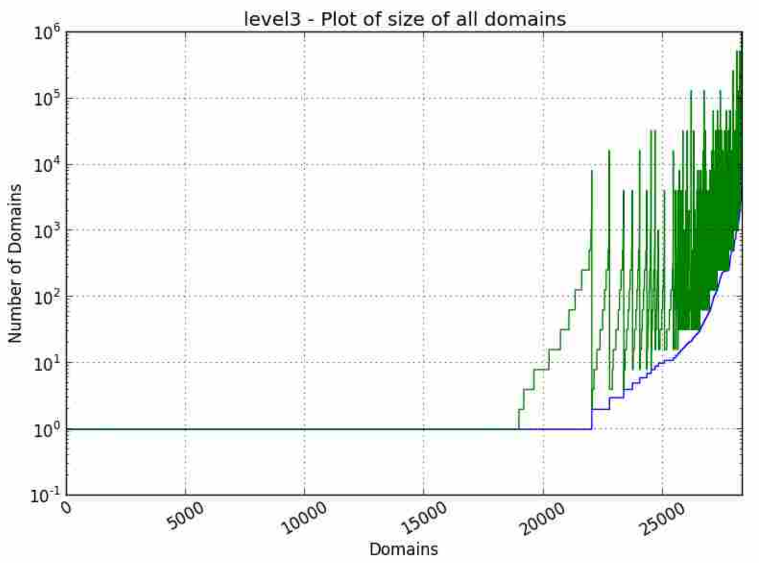

FIGURE 3. Level3 Inferred Prefixes and Their Size Distribution

Level3 Communications

Domain Distribution

Level3 communications is a major ISP. Its a tier 1 ISP. Some of the statistics uncovered in this ISP is given below.The number of distinct domains found are 23,941.

The subnets of the address space taken up by each of those domains is depicted in the table 10. below. The complete results will be shared with the research community. This classification is important because usually the same domains usually follow the same naming conventions and it will be helpful in writing the rules.

DNS Name Count

The total number of addresses which are resolved to DNS names are 1,427,358 out of 10,933,760 IP addresses. This is only about 13% of the address space. The distribution of IP addresses which don't resolve into DNS names are grouped into prefix and subnet length format in null_subnet_level3 table. The table 10. gives a picture of the distribution for the different companies/domains which have IP addresses that belong to Level3 address space. 'gsa.gov' has 3,970 entries with different subnets and the table shows the number of IP addresses in that subnet. The complete results are stored in the database for every such company. The prefix

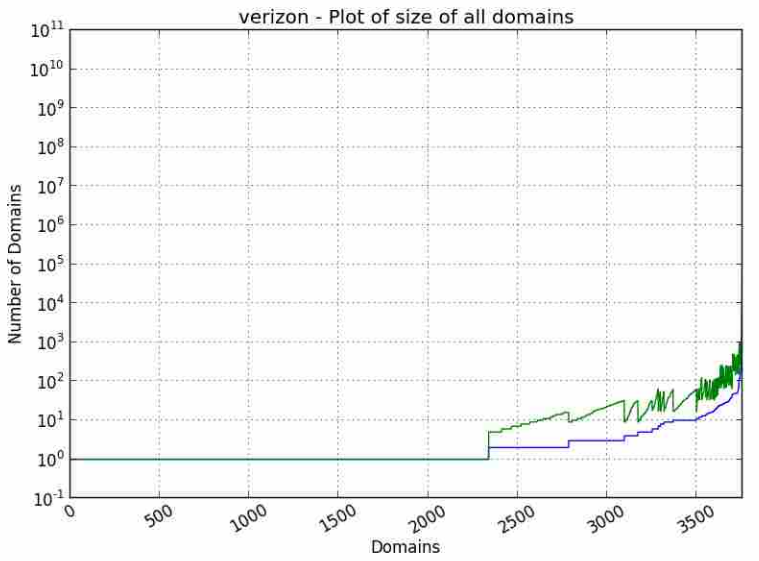

FIGURE 4. Verizon Inferred Prefixes and Their Size Distribution

shows the prefix in which the DNS name containing the domain is found. Count of addrs shows the number of DNS names/IP addresses that had names with that domain.

TABLE 10. Level3-Domain-Subnet Coverage

Domain	**Prefix**	**Count of Addrs**
Level3.net	63.211.96.0/19	3,972
Level3.net	64.154.64.0/19	3,766
Level3.net	63.208.231.192/26	57
gsa.gov	205.130.224.0/19	3,970
buffalo.edu	8.35.160.0/20	3,959
Level3.net	63.214.128.0/19	3,950
Level3.net	209.246.0.0/15	3,918
fibrant.com	8.25.224.0/19	3,868

Table 11. shows the top 10 domains in Level3 and their size and percentage of names with that domain. This table just shows the number of different companies/domains in the descending order of their size. Large portion of the names are Level3.net domain but a significant fraction of the IP address space is used by other companies. Figure 6. shows the distribution of different domains/companies and their count. All the domains of count size 1 are ignored for clarity.

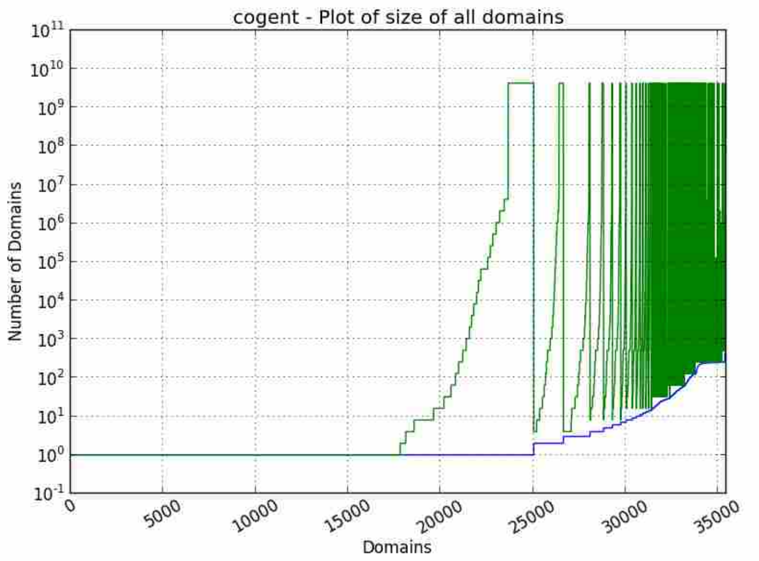

FIGURE 5. Cogent Inferred Prefixes and Their Size Distribution

TABLE 11. Top 10 Domains in Level3 and Their Distribution

Domain	**Count**	**Percentage**
all domains	1,447,628	100 %
Level3.net	1036156	71.57 %
gsa.gov	9,920	0.68 %
buffalo.edu	4,635	0.32 %
fibrant.com	3,868	0.26 %
bbnplanet.net	13792	0.95 %
gtei.net	11,171	0.77 %

Table 12. shows Level3's DNS names and the distribution of the components in the names such as the interface, router, city names, state codes and others. It also shows the dictionary words present in the names which aren't categorized as any of the prior categories mentioned. The number of checks shown in the last column shows the number of times a type of information in a name was checked against a component type and the number of times each of them is found is shown in the second column. There are 1,447,628 IP addresses which have DNS names. In some of the DNS names, there are multiple dictionary words found. Hence, to show the success in finding the dictionary words in the DNS names it is needed to show the number of parts there are in all the DNS names when we split them by "." and "-". Table 22. shows the same.

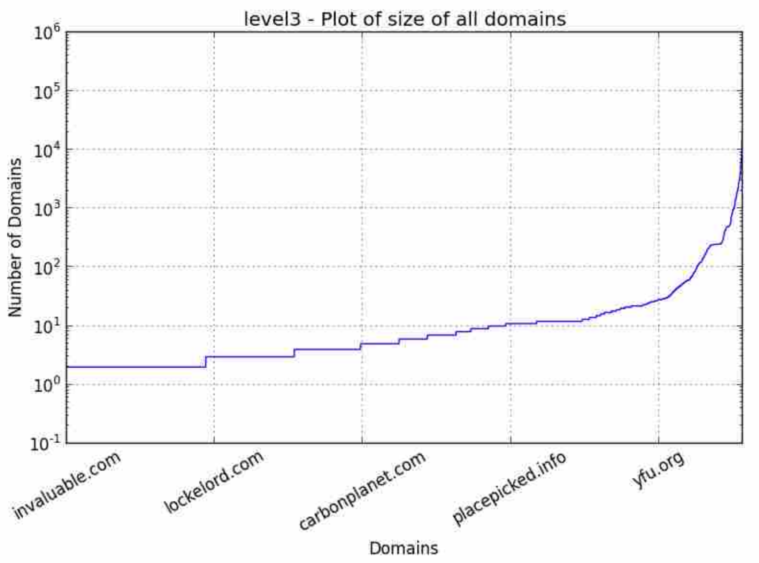

FIGURE 6. Level3 - Domains and Their Size Distribution

TABLE 12. Level3 - Parsed DNS Names

Information gathered	Count	Percentage	Number of Checks
Total no.of DNS Names	1,447,628	-	-
Interface	105,996	7.3 %	2,215,277
Router Function	33,303	2.3%	2,801,113
City Names	78,796	5.44 %	2,783,273
City CLLI	34,084	2.35 %	2,771,077
Airport Codes	53,506	3.69 %	2,772,813
State Codes	51,557	3.56 %	2,770,403

TABLE 13. Level3 - Parsed DNS Names (others)

type of information	Count	Percentage
Number of Segments	2,788,122	-
Dictionary Words	1,117,297	40.07 %
Others	356,485	12.78 %

Fig 7. and 8. shows the pictorial representation of table 12. and 13.. The others bar shows the number of name segments that couldn't be classified as either of interface, router, city categories. Among the "others", there are dictionary words which could tell something more about the dns names. The "others" section contains unusually large number of parts. And a large number of parts in others are dictionary words. It points to a situation where there is some

kind of pattern but it is not consistent. Each of the names have to be studied carefully and we have to study the others section for Level3 more closely to analyze it further.

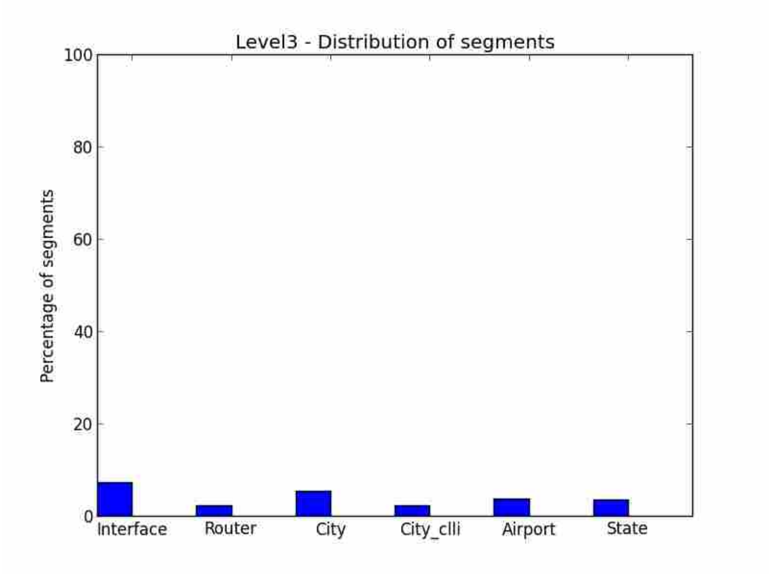

FIGURE 7. Level3 - Names Distribution

Fig. 9. shows Level3's region information clearer along with interface and router function categories. Fair number of names have interfaces and location information. Many names have fully spelled out city names too.

Table 14. shows Level3's dictionary words, their number of occurrences and an example DNS name. Unusually large number of DNS names have the word unknown in them. It just shows that the configuration hasn't been properly done for them. Fair number of routers are host, static, mail servers etc.

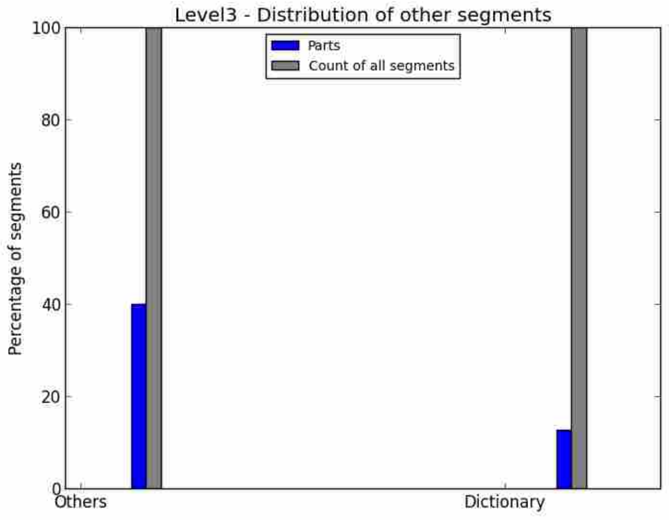

FIGURE 8. Level3 - Names Distribution(Others)

TABLE 14. Level3 - Most Occurring Dictionary Words

ASN	Word	Number of Occurrences	DNS name
3356	unknown	1,002,616	unknown.level3.net.
3356	host	17115	host-23.kletos.net.
3356	static	11247	static.34k.dscga.com.
3356	dynamic	9781	bj-dynamic-245.sys.gtei.net.
3356	bc	6140	8-6-93-255-bc.redplaid.com.
3356	mail	6059	mail.clearwaterhousingauth.org.
3356	wireless	4076	db-wireless.car1.minneapolis1.level3.net.
3356	voice	3618	voice-retri.edge6.dallas1.level3.net.
3356	domain	3306	waident-exch2.domain.waident.com.
3356	customer	2885	customer-co.edge1.minneapolis1.level3.net.
3356	unassigned	2426	unassigned-183.e.active.com.
3356	reverse	2235	reverse.vetronix.com.
3356	unused	1770	8-23-128-124-unused.phx.unsi.net.
3356	dial	1718	dial-800-ll.car1.dallas1.level3.net.
3356	deploy	1259	a8-17-144-105.deploy.akamaitechnologies.com.

Table 15. and fig 10. shows Level3's CDF of information parsed. The x-axis shows the number of items of information parsed. Since this is a cdf, the x-axis shows bins. The first bin is the number of DNS names that have no items of information(x axis from 0 to 1) . The second bin (x axis from 1 to 2) shows the number of DNS names that have 0 or 1 parts of information and so on. Since the ”others” section had a high number of name parts, its clear that we couldn't

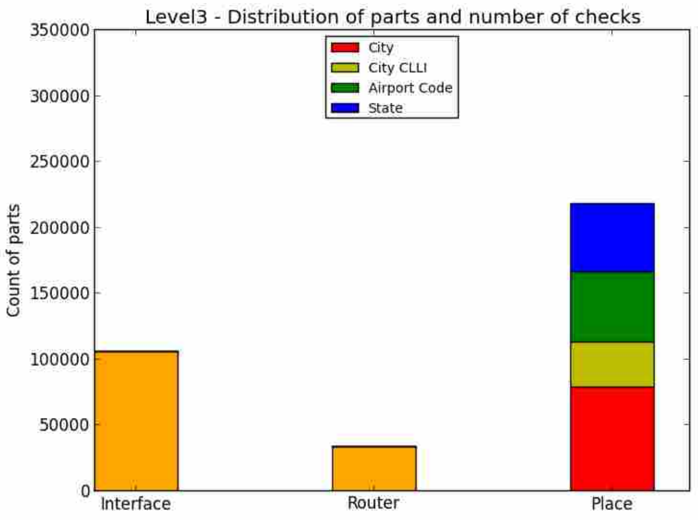

FIGURE 9. Level3 - Parts Distribution

infer any of interface, router or location information for 86% of the names and that can be seen at the bin 1 (x axis 0 to 1).

TABLE 15. Level3 - CDF of Information Parsed

Type	**Count**	**Percentage**
No inference	1,250,287	86.36 %
At least one	96,207	6.64 %
At least two	45,917	3.17 %
At least three	51,672	3.56 %
At least four	3,540	0.24 %
Five and above	5	0.0003 %

Verizon

Table 16. shows the top 10 domains in Verizon and their size and percentage of names with that domain. Figure 11. all the domains whose size is greater than 1. It shows them in an increasing order of their sizes. Some of the domains clearly have a large size. These tend to be big companies.

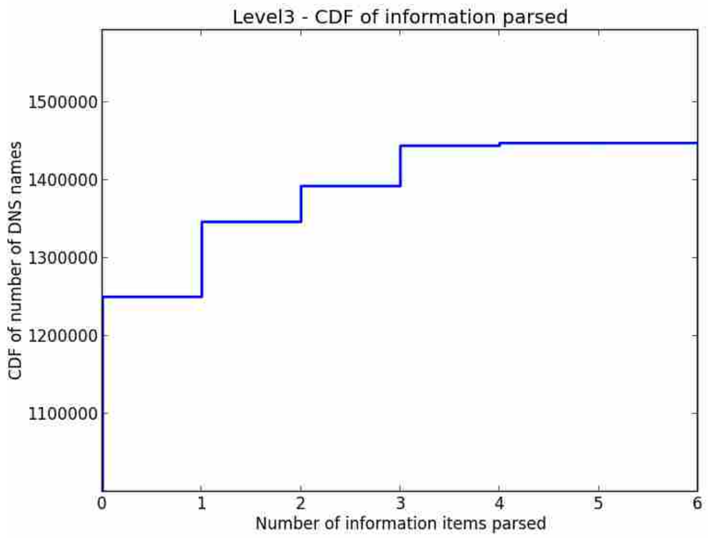

FIGURE 10. Level3 - CDF of Extracted Information

TABLE 16. Top 10 Domains in Verizon and Their Distribution

Domain	**Count**	**Percentage**
all domains	3,144,673	100 %
verizon.net	2,601,667	82.73 %
verizon-gni.net	44,312	1.41 %
ALTER.NET	37,059	1.18 %
ba-dsg.net	1,474	0.04 %
nisgroup.com	490	0.01 %
airg.com	475	0.01 %
bellatlantic.net	347	0.01 %
algorithmics.com	245	0.007 %
dwoskin.com	240	0.007 %

Table 17. shows Verizon's DNS names and the distribution of the components in the names such as the interface, router, city names, state codes and others. It also shows the dictionary words present in the names which aren't categorized as any of the prior categories mentioned. The number of checks shown in the last column shows the number of times a segment of a name was checked against a component type and the number of times each of them is found is shown in the second column.

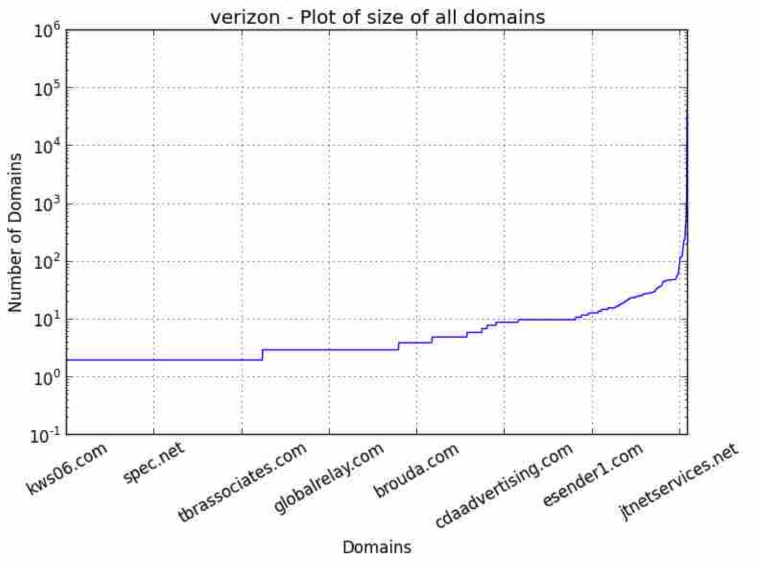

FIGURE 11. Verizon - Domains and Their Size Distribution

TABLE 17. Verizon - Parsed DNS Names

Information gathered	**Count**	**Percentage**	**Number of Checks**
Total no.of DNS Names	2,703,582	- %	-
Interface	102,507	7.18 %	10,855,520
Router Function	31,784	2.22%	18,814,333
City Names	84,773	5.44 %	18,794,999
City CLLI	33,243	2.32 %	18,794,865
Airport Codes	70,706	4.95 %	18,801,068
State Codes	95,068	6.66 %	18,794,865

TABLE 18. Verizon - Parsed DNS Names(Others)

Type of Info	**Count**	**Percentage**
Number of Segments	18,814,155	-
Others	2,148,405	11.41 %
Dictionary Words	3,324,352	17.66 %

Fig 12. shows the pictorial representation of table 17.. Figure 13. shows the number of name segments that couldn't be classified as either of interface, router, city categories. There are dictionary words which could tell something more about the dns names upon further analysis. Some of the names have more than one dictionary names. Hence, to calculate the percentage of dictionary words and percentage of "others", I had to calculate the number of parts in all that we check against in our list of DNS names. This is represented as the grey bar.

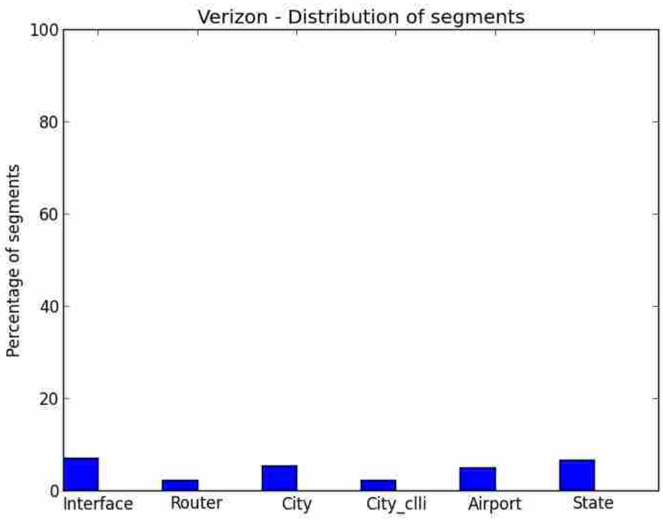

FIGURE 12. Verizon - Names Distribution

Fig 14. shows the parts of the names with region information and its distribution clearly. A very high percentage of the names are in the others section. And most of them are dictionary words. This means that the naming of DNS names has structure but it doesn't tell much about the interface or city. It is possible that it speaks about the router function but the description of router function varies from ISP to ISP. Since there is no consistency and no format for naming, we can't classify them. A detailed analysis of others section along with other types of measurement could help understand these names.

Table 19. shows Verizon's most occurring dictionary words, their number of occurrences and an example name.

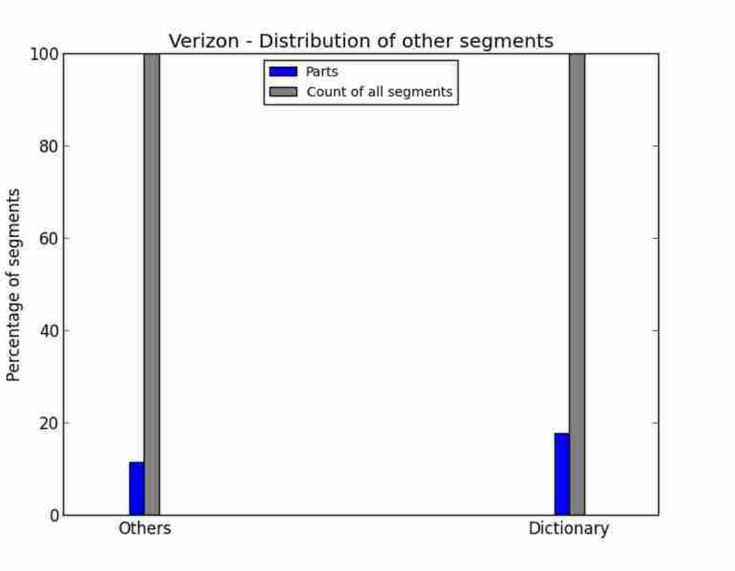

FIGURE 13. Verizon - Names Distribution(Others)

TABLE 19. Verizon - Most Occurring Dictionary Words

ASN	Word	Number of occurrences	DNS name
702	pool	2,321,007	pool-71-165-110-143.lsanca.fios.verizon.net.
702	east	547,152	pool-71-174-0-142.bstnma.east.verizon.net.
702	static	280,704	static-71-165-70-129.lsanca.dsl-w.verizon.net.
702	customer	13,312	customer.bpsoft.com.
702	client	2,365	client-141-156-58-9.ba-dsg.net.
702	internet	1,845	internet-gw.customer.alter.net.
702	bb	1,737	so-7-3-0-0.lax01-bb-rtr1.verizon-gni.net.
702	mail	1,642	mail.abtinc.com.
702	broadcast	386	broadcast.alter.net.
702	reed	181	smtp17.reed-ian-swx.com.
702	digital	161	smtp29.digital.reinforcedplastics.com.
702	charming	135	charming-gw.customer.alter.net.
702	response	119	email1.response.sdgroup.eu.com.

Table 20. and figure 15. shows Verizon's CDF of Information parsed. The x-axis shows the number of items of information parsed. Since this is a cdf, the x-axis shows bins. The first bin is the number of DNS names that have no items of information(x axis from 0 to 1) . The second bin (x axis from 1 to 2) shows the number of DNS names that have 0 or 1 parts of information and so on. The figure spikes at at least one information retrieved (second bin on x axis). It shows that Verizon has at least one information in a large number of DNS names.

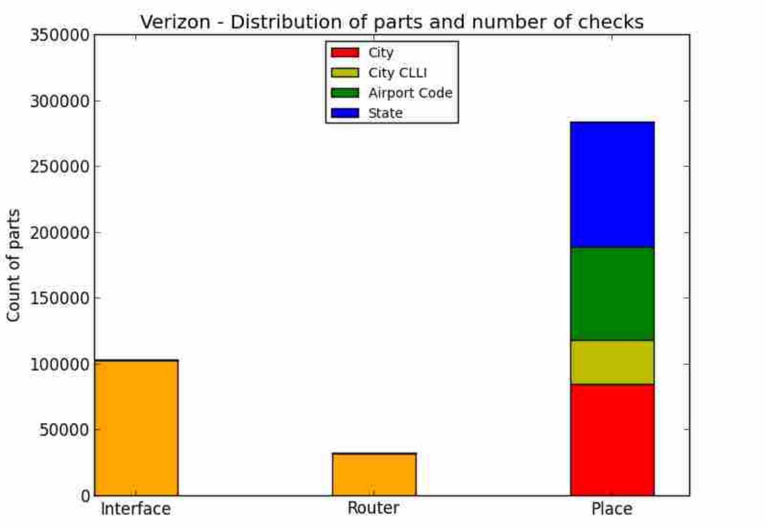

FIGURE 14. Verizon - Names Distribution

TABLE 20. Verizon - CDF of Information Parsed

Type	**Count**	**Percentage**
No inference	21,192	0.78 %
At least one	26,198,97	96.9 %
At least two	49,039	1.81 %
At least three	13,297	0.49 %
At least four	138	0.0051 %
Five and above	19	0.0007 %

Cogent

Table 21. shows Cogent's DNS names and the distribution of the components in the names such as the interface, router, city names, state codes and others. It also shows the dictionary words present in the names which aren't categorized as any of the prior categories mentioned. The number of checks shown in the last column shows the number of times a segment of a name was checked against a component type and the number of times each of them is found is shown in the second column.

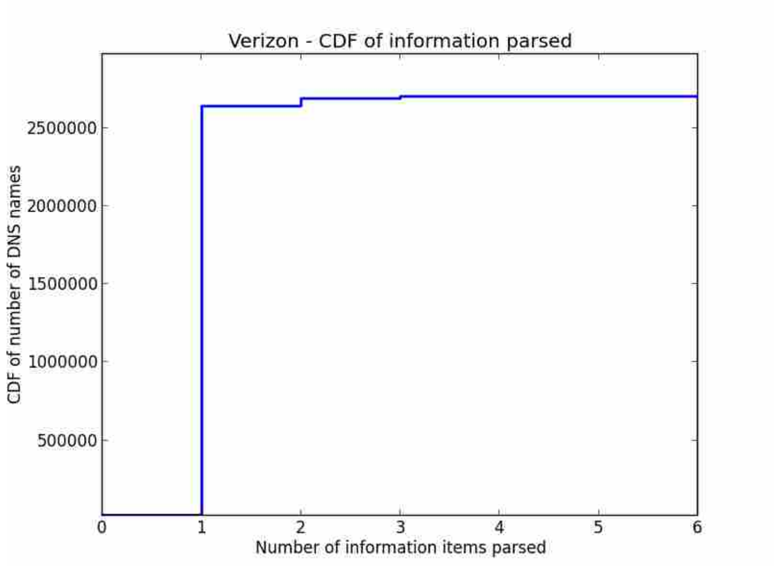

FIGURE 15. Verizon - CDF of Extracted Information

TABLE 21. Cogent - Parsed DNS Names

Information gathered	**Count**	**Percentage**	**Number of Checks**
Total no.of DNS Names	585,674	100 %	0
Interface	117,187	20 %	1,340,222
Router Function	10,261	1.75%	1,873,504
City Names	24,884	4.24 %	1,863,316
City CLLI	304	0.05 %	1,862,845
Airport Codes	90,370	15.43 %	1,865,624
State Codes	24,106	4.11 %	1,862,844

TABLE 22. Cogent - Parsed DNS Names(Others)

Type of Info	**Count**	**Percentage**
Number of Segments	1,872,904	-
Others	45,523	2.43 %
Dictionary Words	180,224	9.6 %

Fig 16. shows the pictorial representation of table 21.. The number of checks is scaled down to 10% of its original size for scaling purposes. In the figure 17., the ”others” bar shows the number of name segments that couldn’t be classified as either of interface, router, city categories. There are dictionary words which could tell something more about the dns names upon further investigation. A sample list of most occurring dictionary words are shown later. There can be more than one dictionary words present in a DNS name. Hence we calculated the number of

name segments we find in all the DNS names that we encounter and calculate the percentage of positive results in finding the dictionary words. The grey bar shows the number of segments for comparison of the success rate in finding the dictionary words.

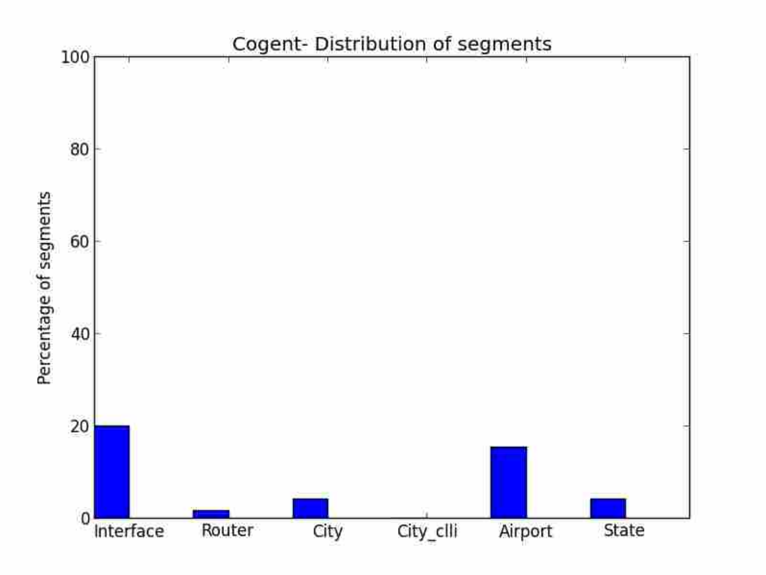

FIGURE 16. Cogent - Names Distribution

Fig. 18. shows Cogent's region information clearer along with interface and router function categories.

Table 23. shows Cogent's CDF of information parsed. The x-axis shows the number of items of information parsed. Since this is a cdf, the x-axis shows bins. The first bin is the number of DNS names that have no items of information(x axis from 0 to 1) . The second bin (x axis from 1 to 2) shows the number of DNS names that have 0 or 1 parts of information and so on. A spike in bin 1 shows that there are a lot of DNS names that don't have any specific information.

TABLE 23. Cogent - CDF of Information Parsed

Type	**Count**	**Percentage**
No inference	416,346	71.08 %
At least one	92,224	15.74 %
At least two	60,321	10.29%
At least three	14,493	2.47 %
At least four	2,290	0.39 %
Five and above	0	0 %

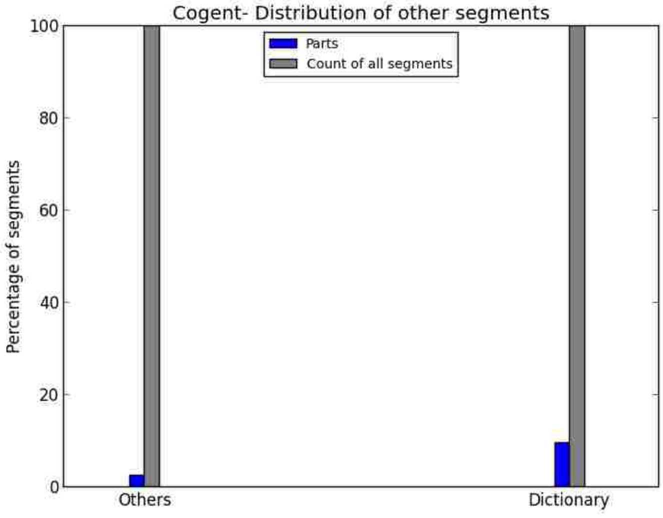

FIGURE 17. Cogent - Names Distribution (Others)

Table 24. shows Cogent's most occurring dictionary words, their number of occurrences and an example name.

TABLE 24. Cogent - Most Occurring Dictionary Words

ASN	Word	Number of occurrences	DNS name
174	atlas	56,531	gi0-0-0-18.202.nr11.b022073-0.ord01.atlas.cogentco.com.
174	static	19,630	153.38-89-161.static.servergrove.com.
174	host	13,140	host-38.80.71.016.mmcm.com.
174	mail	10,825	mail.amnow.com.
174	dynamic	8,420	dynamic-capital-management.demarc.cogentco.com.
174	cable	7,753	38-82-64-141-cable.cybercable.net.mx.
174	wireless	7,311	wireless.telebright.com.
174	unassigned	5,654	38.69.129.164.unassigned.neptunetg.com.129.69.38.in-addr.arpa.
174	reverse	3,467	181-18-68-38-static.reverse.queryfoundry.net.
174	domain	3,255	domain.not.configured.
174	customer	3,024	customer.hostiserver.com.
174	user	2,345	a.user.bayweb.com.
174	red	1,955	red.rentpayment.com.
174	net	1,021	38.89.246.0.cirbn.net.246.89.38.in-addr.arpa.
174	sac	988	sac-capital-adviser-llc.demarc.cogentco.com.
174	tnt	979	tnt-38-113-28-191.worldpath.net.
174	port	962	port-chan-1-23.core1.cvg1.zimcom.net.

Table 25. shows the comparison of Cogent's region information with that of IP2Location. The first column shows the number of times it matches with IP2Location data and the second column shows the number of times it doesn't match.

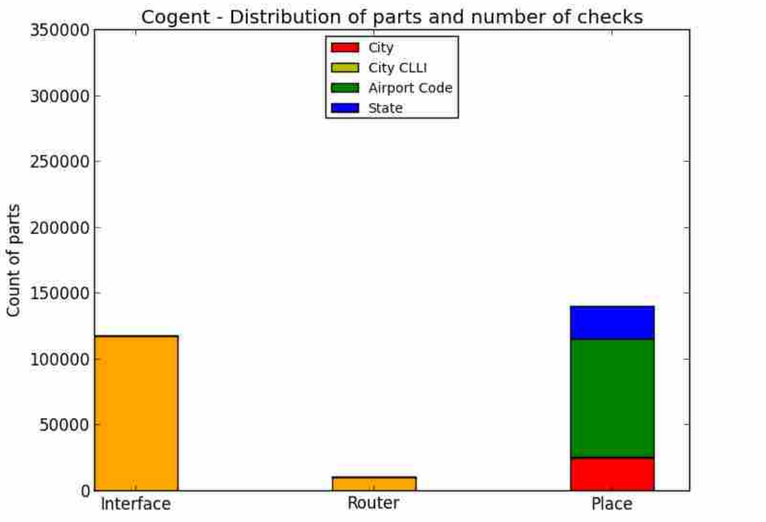

FIGURE 18. Cogent - Parts Distribution

TABLE 25. Cogent - Place Matches and Mismatches with IP2Location Data

Type	Match Count	Mismatch Count
airport code	10,796	72,245
city	1,564	23,320
state	1,645	6,599

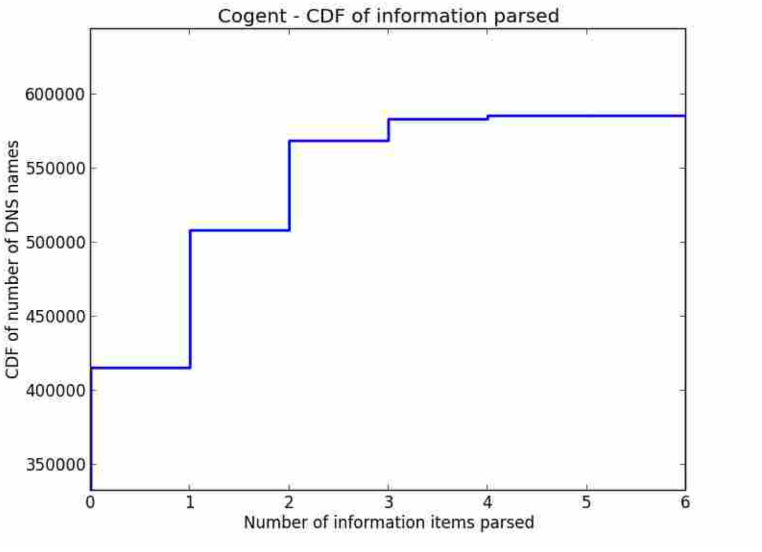

FIGURE 19. Cogent - CDF of Extracted Information

CHAPTER V

CROSS ISP VS CAIDA DATASET ANALYSIS

Center for Applied internet Data Analysis (CAIDA) runs many projects which do internet active and passive measurement projects. One of them is the CAIDA DNS [2] lookup. They perform DNS lookups everyday from a managed central server at CAIDA. It performs millions of DNS lookups everyday. They have other projects which use alias resolution techniques to find the topology of the network. Soon after they perform the topology trace, they perform the DNS-lookups. This is because it is assumed that doing so maintains the same state topology during DNS name resolution as well. They also don't lookup an IP address if they have successfully looked up that address in the last 7 days.

Several teams of monitors produce the IPv4 Routed /24 Topology Dataset from which they derive this DNS Names data. These teams independently probe every routed /24 in the IPv4 address space (one pass through every routed /24 is called a cycle). Because different teams have different members, locations, and capabilities, each team completes a cycle at a different rate.

The DNS Names data is collected on a per-day basis. Only a loose connection exists between the topology traces and DNS names exist because the topology data exists on a per-team and per-cycle basis.

Observations of The CAIDA Dataset

We work on two datasets. One is an old dataset collected on 08-31-2012. Another is a newer dataset. The hostname which are successful are shown in lowercase and those IP addresses which result in errors are in the uppercase. Here are some examples of those.

FAIL.NON-AUTHORITATIVE.in-addr.arpa : Equivalent to NXDOMAIN we encounter with `dig`

FAIL.SERVER-FAILURE.in-addr.arpa : Equivalent to SERVFAIL with `dig`

FAIL.TIMEOUT.in-addr.arpa : Equivalent to TIMEOUT with `dig`

Table 26. gives the basic observations made.

TABLE 26. Observations of CAIDA dataset

Observation	Value
Date of data collection	08-31-2012
Total number of IP addresses	1,880,374
Total number of SERVFAILs	63,478
Total number of NXDOMAINs	712,537
Total number of TIMEOUTs	6,133
Total number of DNS names	1,098,226

58.4% of the IP addresses have DNS names. 0.32% of the names have TIMEOUTs . 3.38% of the names have SERVFAIL error. 37.9% of the names are resolved but don't have DNS names.

Table 27. shows the CDF of the names found in CAIDA dataset. The x-axis shows the number of items of information parsed. Since this is a cdf, the x-axis shows bins. The first bin is the number of DNS names that have no items of information(x axis from 0 to 1) . The second bin (x axis from 1 to 2) shows the number of DNS names that have 0 or 1 parts of information and so on. Considerably high number of DNS names have at least one field of information such as interface, router function, city or state etc.

TABLE 27. CAIDA - CDF of Information parsed

Type	Count	Percentage
No inference	661,602	60.24 %
At least one	350,482	31.91 %
At least two	75,352	6.86 %
At least three	9,656	0.87 %
At least four	1,102	0.1 %
Five and above	32	0.0029 %

Fig 20. shows the cdf of CAIDA names that we found. Fig 21. shows the cdf of CAIDA along with the CDF of other ISPs namely Level3, Verizon and Cogent. It shows only the percentages of number of items found so as to scale them equally. From this figure, we can see that Verizon has the highest percentage of names with at least one part of information in it. It is significantly higher than the percentage we see for CAIDA names which is collected from multiple ISPs. Level3 and Cogent have lesser information than CAIDA.

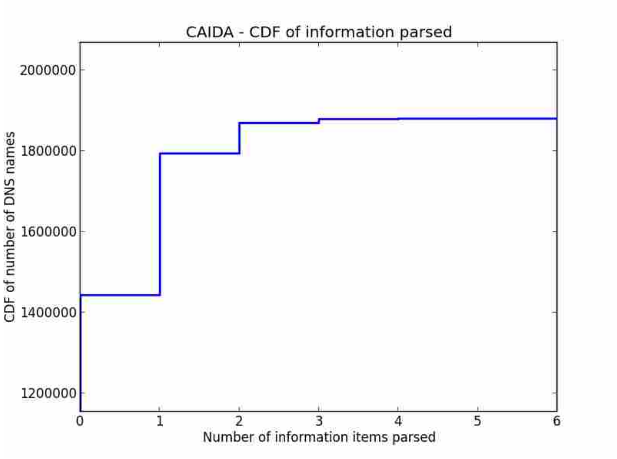

FIGURE 20. CAIDA - CDF of Extracted Information

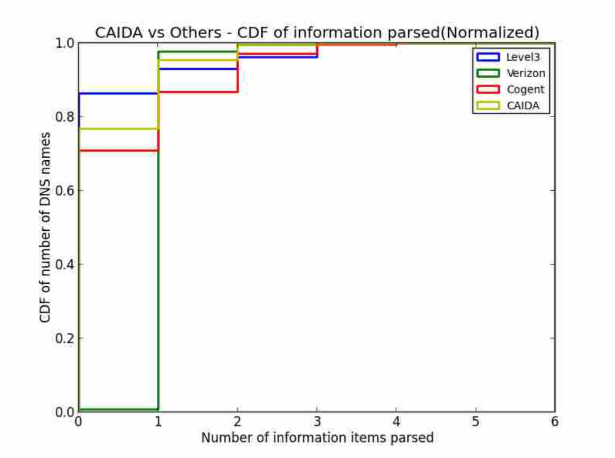

FIGURE 21. CAIDA vs Others CDF of Extracted Information

TABLE 28. CAIDA - Parsed DNS Names

Information gathered	**Count**	**Percentage**	**Number of Checks**
Total no.of DNS Names	1,098,226	100 %	0
Interface	169,645	15.44 %	5,222,736
Router Function	18,045	1.64%	8,039,110
City Names	45,112	4.1 %	8,023,746
City CLLI	60,584	5.51 %	8,023,189
Airport Codes	154,583	14.07 %	8,024,147
State Codes	86753	7.89 %	8,023,101

TABLE 29. CAIDA - Parsed DNS names(Others)

Type of Info	**Count**	**Percentage**
Number of Segments	8,038,088	-
Others	537,044	6.68 %
Dictionary Words	1,264,707	15.73 %

Figure 22. shows the distribution of the names in CAIDA dataset. Figure 24. shows the name parts and their distribution. Figure 21. shows a comparison of the names and entities found out in the names and that of Level3, Verizon and Cogent. It shows only the percentages of names/parts found in the ISPs and CAIDA to compare against all the other ISPs. we see that our analysis follows the general observations across multiple ISPs collected by CAIDA.

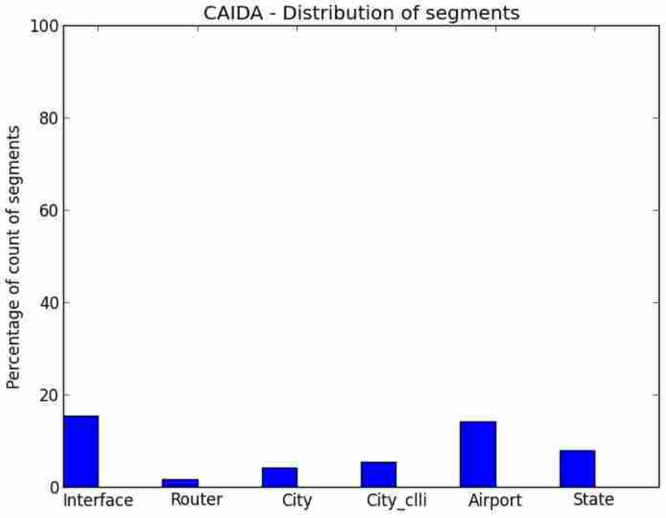

FIGURE 22. CAIDA Segment Distribution

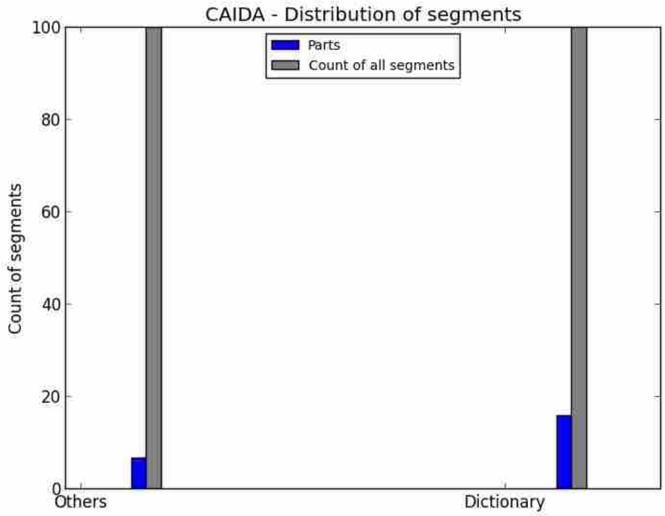

FIGURE 23. CAIDA Others Distribution

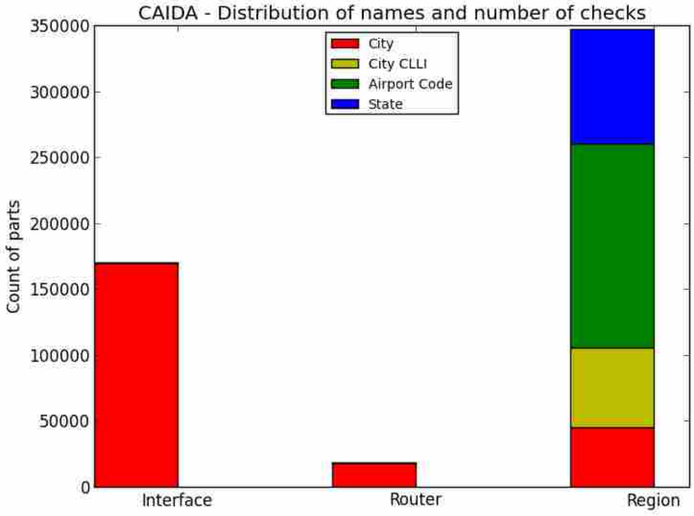

FIGURE 24. CAIDA vs Others CDF of Extracted Information

CHAPTER VI

TOPOLOGY MAPPING FROM XNET AND IFFINDER

xnet [25] is a tool which is used for subnet inference. It works by sending IP probe packets to hypothetical subnets of size /31 along the path of the target IP address and records the nodes that respond with ICMP port unreachable messages. When the destination IP is reached, it uses the hop count to determine the possible subnet the target IP address belongs to. More explanation is provided in [24]. It gives alias information as well that it encountered during the process.

iffinder [18] is another alias resolution tool. It works by sending IP probe packets to the destination IP address on high numbered ports. The target IPs are likely to respond with ICMP port unreachable messages. Sometimes they send this ICMP message from a different interface than the interface at which it was received. Hence, we have a tuple of interface IP addresses that belong to the same router. We can configure it to run multiple times as the tuple might possibly grow as we find more aliases of the same router.

Databank

I ran xnet on a small domain named databank.com. It has 1110 IP addresses assigned as part of 3356 ASN (level3). Finding the topology of these small components which make up the larger ASN would be reasonable since the structure of the ASN is structured by these smaller companies. Out of 1110 IP addresses, I found 82 IP addresses which responded to the xnet requests. Alias resolution on this particular dataset didn't yield any aliases in the same IP address space. So we are assuming that each interface to be a router. Based on the List of IP addresses within the target subnet and their hop distances from the vantage point, we are able to find the topology of the 82 nodes that we found. Figure 25. shows the topology that we found using xnet for databank.com. We used Gephi [6] graph visualization tool for visualization. We use force atlas layout with a weak attraction strength to show which routers are connected to each other. This graph has 108 nodes and 76 edges but clearly it is disconnected. This is because xnet doesn't respond to all the queries.

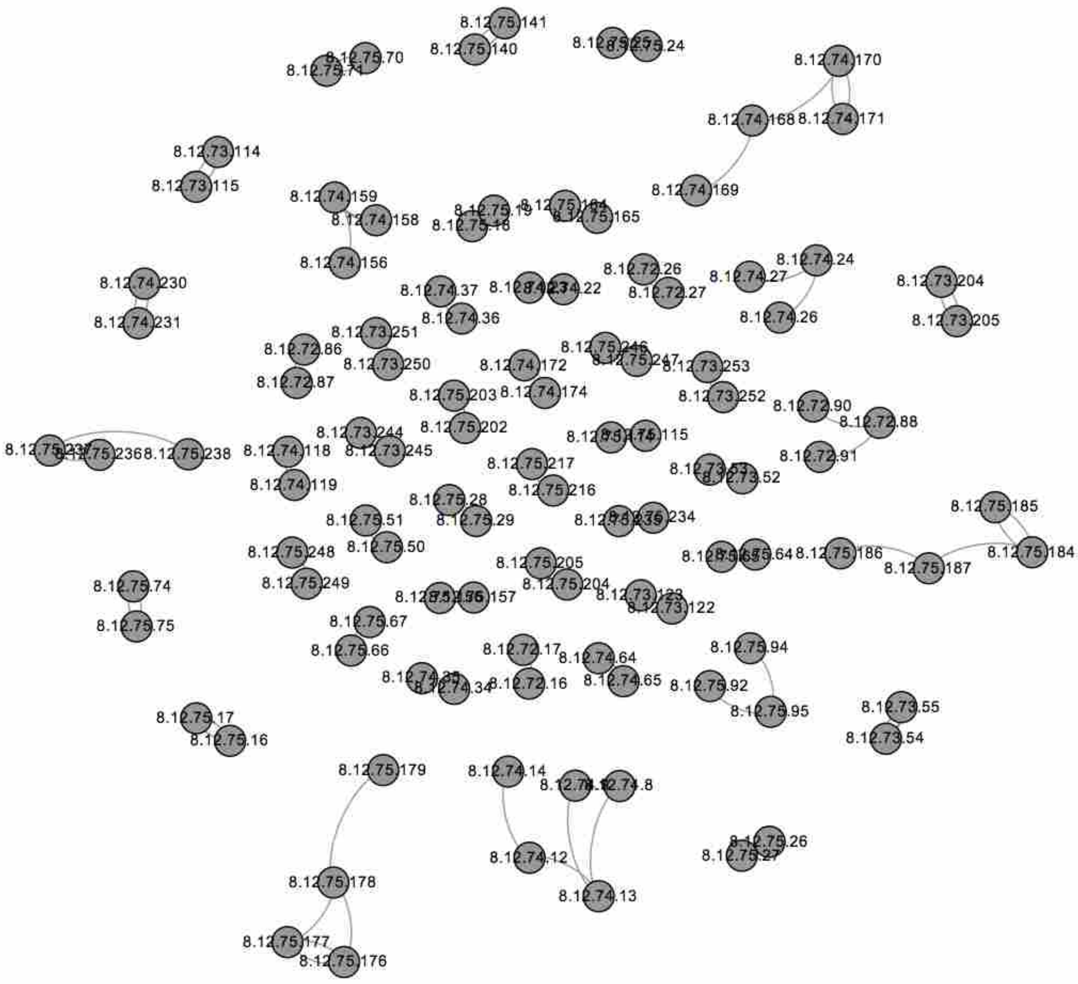

FIGURE 25. Topology of Databank

Yahoo

A similar analysis is done on yahoo.com domain which also belongs to the same level3 ASN. The domain has 5,788 IP addresses. Out of these 426 IP addresses responded to `xnet`. When we plotting it in Gephi we found 559 edges and 421 nodes. Fig. 26. shows the router level topology of yahoo.

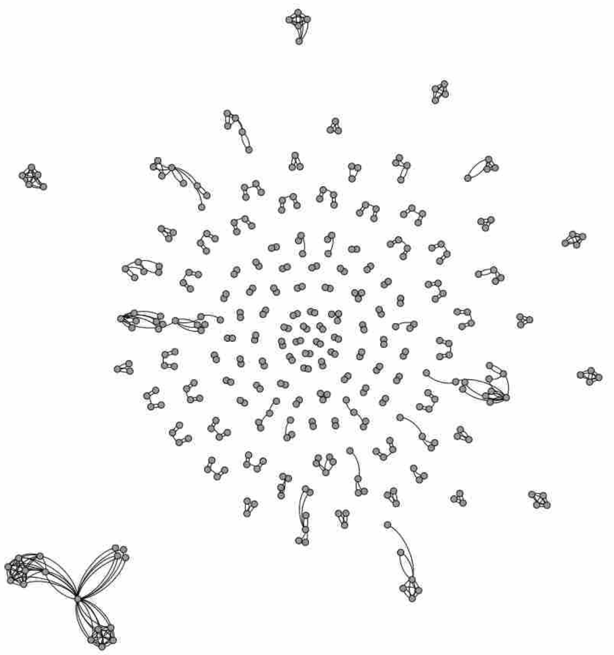

FIGURE 26. Topology of Yahoo

Graph Based on City Information

As discussed in the paper Growth Analysis of ISPs [15], /31 subnets are highly likely to be connected provided they are physically interfaces. Since we have DNS names for these IP addresses, we know that they are physical nodes. And using `xnet` and `iffinder`, we have alias information of these IP addresses too. Hence we use a similar methodology where we assume /31 addresses of an IP to be connected. Building on the router level topology mapping process described above, we also check the region information from the DNS name of the IP address and we group them together as nodes. When a router from city1 connects to a router in city2, we add a link. We know that every domain like verizon-gni has its routers in different cities. Hence, we came up with the region-based graph of a domain like verizon-gni shown in fig. 27.. The size of the nodes depends on the number of nodes in that region. The color gradient depends on the degree of the nodes. The nodes which have a degree of 1 are ignored.

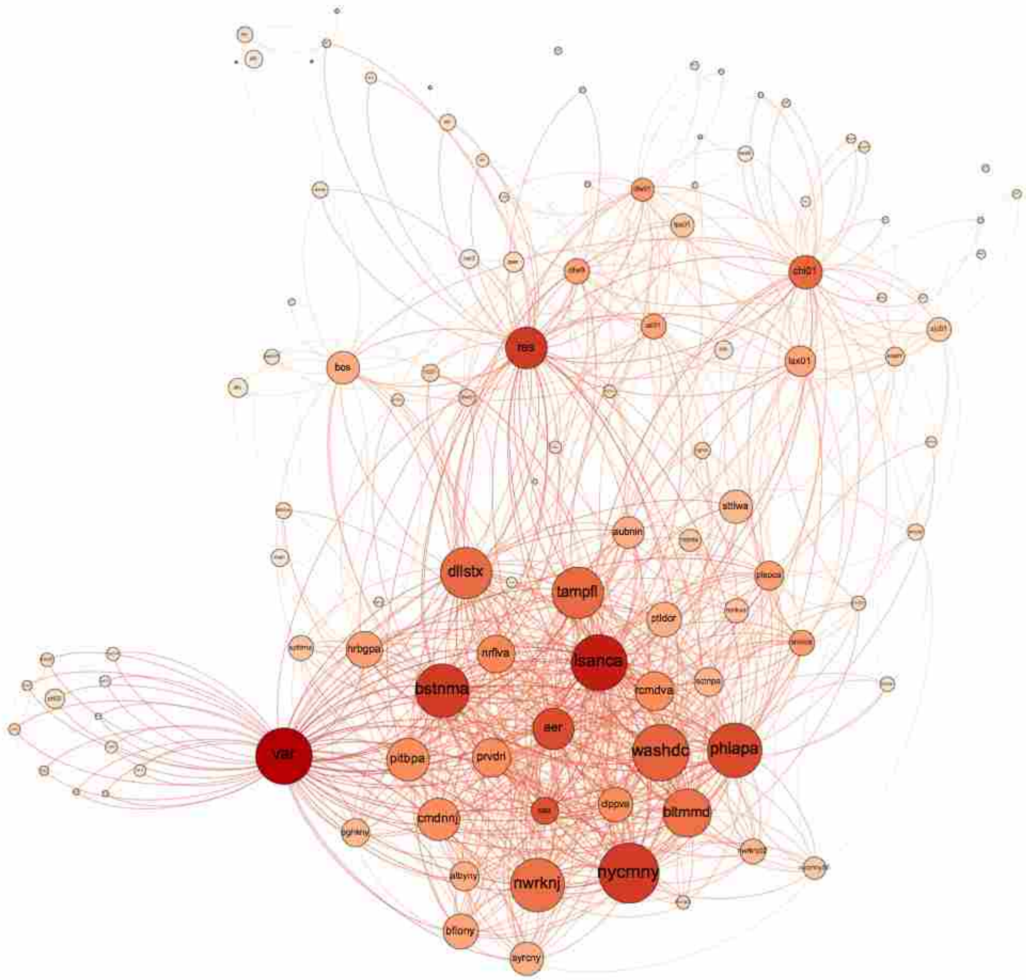

FIGURE 27. Region Level Topology of Verizon-gni

Another example of such a graph for Level.net domain belonging to level3 ASN (3356) is shown in fig. 28.. The size of the nodes depends on the number of nodes in that region. The color gradient depends on the degree of the nodes. The nodes which have a degree of 1 are ignored.

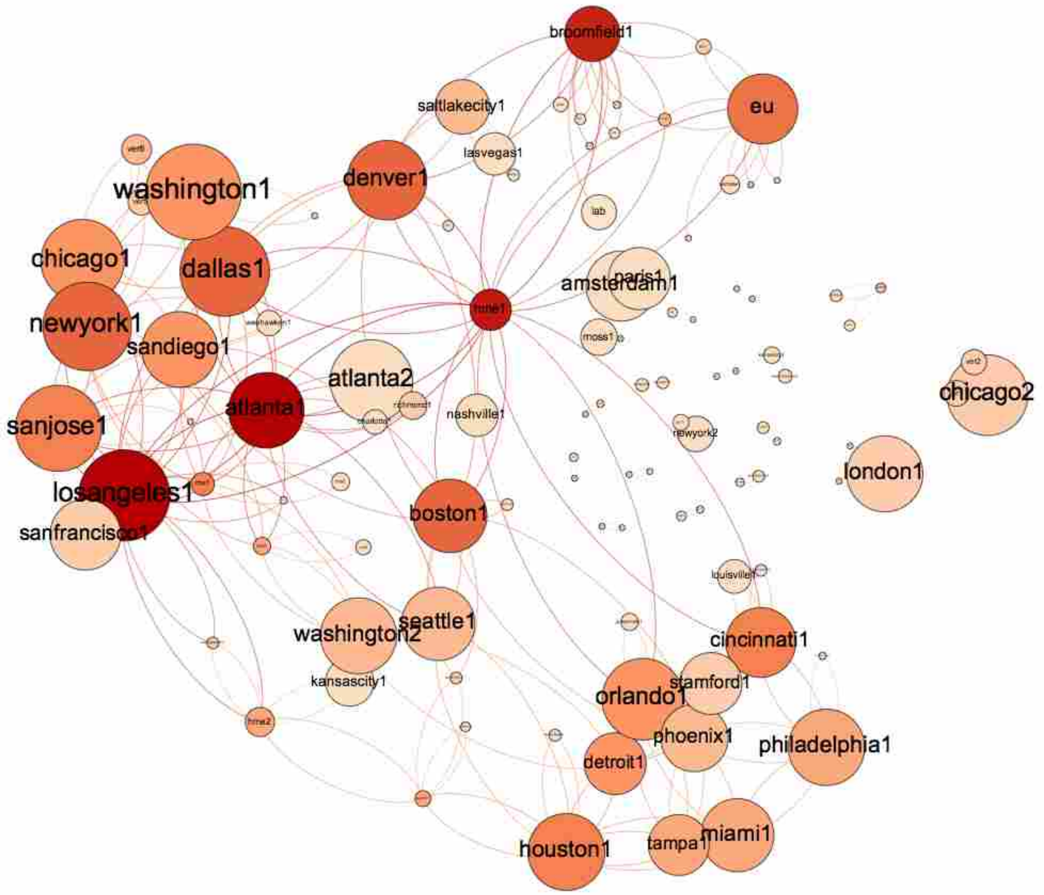

FIGURE 28. Region Level Topology of Level3

REFERENCES CITED

[1] Airport codes. https://www.airportcodes.org. Accessed: 2014-09-05.

[2] The caida ucsd ipv4 routed /24 dns names dataset. http://www.caida.org/data/active/ipv4_dnsnames_dataset.xml. Accessed: 2014-09-09.

[3] Clli code. http://en.wikipedia.org/wiki/CLLI_code. Accessed: 2014-09-05.

[4] Domain names - implementation and specification. https://www.ietf.org/rfc/rfc1035.txt. Accessed: 2014-09-16.

[5] Geonames. https://www.geonames.org. Accessed: 2014-09-05.

[6] Gephi. "http://gephi.github.io".

[7] Ip2location. http://www.ip2location.com. Accessed: 2014-09-05.

[8] "mrinfo". "http://technet.microsoft.com/en-us/library/cc957933.aspx".

[9] "pathaudit". "https://github.com/jc-wail/WAIL/tree/master/PathAudit".

[10] Public dns server list. http://www.public-dns.tk. Accessed: 2014-09-05.

[11] "state codes of us". "http://en.wikipedia.org/wiki/List_of_U.S._state_abbreviations".

[12] Telcodata.us telecommunications database. https://www.telcodata.us. Accessed: 2014-09-05.

[13] "undns". "http://www.scriptroute.org/source/".

[14] Joseph Chabarek and Paul Barford. What's in a name?: Decoding router interface names. In *Proceedings of the 5th ACM Workshop on HotPlanet*, HotPlanet '13, pages 3–8, New York, NY, USA, 2013. ACM.

[15] Andrew D. Ferguson, Jordan Place, and Rodrigo Fonseca. Growth analysis of a large isp. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 347–352, New York, NY, USA, 2013. ACM.

[16] Huanetwork. The naming conventions of huawei ar routers. "http://www.huanetwork.com/blog/the-naming-conventions-of-huawei-ar-routers/".

[17] Inc Juniper Systems. Interface naming overview. "http://www.juniper.net/techpubs/en_US/junos12.3/topics/concept/interfaces-interface-naming-overview.html".

[18] CAIDA Ken Keys. iffinder. "http://www.caida.org/tools/measurement/iffinder/".

[19] P. Mrindol, B. Donnet, J. Pansiot, M. Luckie, and Y. Hyun. MERLIN: MEasure the Router Level of the INternet. In *Conference on Next Generation Internet*, Jun 2011.

[20] University of Oregon. University of oregon route views project. http://www.routeviews.org. Accessed: 2014-09-05.

[21] Neil Spring, Ratul Mahajan, David Wetherall, and Thomas Anderson. Measuring isp topologies with rocketfuel. *IEEE/ACM Trans. Netw.*, 12(1):2–16, February 2004.

[22] ”Cisco Systems”. ”configuring router interfaces”. "http://www.cisco.com/c/en/us/td/docs/security/security_management/cisco_security_manager/security_manager/4-1/user/guide/CSMUserGuide_wrapper/rtintf.pdf".

[23] Team-Cymru. Ip to asn mapping. http://www.team-cymru.org/Services/ip-to-asn.html. Accessed: 2014-09-05.

[24] M.E. Tozal and K. Sarac. Subnet level network topology mapping. In *Performance Computing and Communications Conference (IPCCC), 2011 IEEE 30th International*, pages 1–8, Nov 2011.

[25] Mehmet Engin Tozal. Ntmaps - network mapping & modeling. "http://nsrg.louisiana.edu/project/ntmaps/output/explorenet.html".

[26] Ming Zhang, Yaoping Ruan, Vivek Pai, and Jennifer Rexford. How dns misnaming distorts internet topology mapping. In *Proceedings of the Annual Conference on USENIX '06 Annual Technical Conference*, ATEC '06, pages 34–34, Berkeley, CA, USA, 2006. USENIX Association.