UNIVERSITY OF CALIFORNIA, SAN DIEGO

Leveraging Internet Background Radiation for Opportunistic Network Analysis

A dissertation submitted in partial satisfaction of the
requirements for the degree of Doctor of Philosophy

in

Computer Science

by

Karyn Benson

Committee in charge:

kc claffy, Chair
Alex C. Snoeren, Co-Chair
Alberto Daniotti
George Papen
Stefan Savage
Geoffrey M. Voelker

2016

The Dissertation of Karyn Benson is approved and is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Co-Chair

_____

Chair

University of California, San Diego

2016

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Growing up, my family had a tradition where every evening we would share a thing we were thankful for. As an adult, I have intermittently continued this tradition. However, at a rate of one per night, it would take me years to fully express my gratitude for all the people who supported me in completing my PhD.

First, I am extremely indebted to my advisors. I went through a series of other research groups before joining CAIDA. I am glad kc and Alberto took a chance on a confused graduate student. It was comforting to learn that kc had also switched research groups as a graduate student, and I was impressed with her role in shaping the Internet measurement field and communications policy. Alberto led by example, in both his remarkable work ethic and contagious excitement about the darknet. Alex, a later addition to my advising team, provided much appreciated insight, structure, and experience. Above all, I am thankful for my advisors' patience. Although there is still plenty of room for improvement, I am a much better presenter, writer, and researcher due to their detailed and honest feedback. I am very fortunate to have worked with a set of wonderful mentors and world-class researchers.

I am also appreciative of the feedback from the rest of my committee. Helpful comments from Stefan and Geoff during my Syslunch presentations frequently resulted in new research directions.

My co-authors taught me a lot about working with different types of people, and articulating my ideas and findings. Emile Aben's packet loss idea was the basis for my first paper on IBR. Xenofontas Dimitropoulos, Alessandro Finamore, Eduard Glatz, and Philipp Richter were accommodating of my ideas to compare our different passive datasets. Michalis Kallitsis would always provide thoughtful feedback. I am excited to complete works-in-progress with Amogh Dhamdhere, Ahmed Elmokashfi, Ioana Livadariu, and Ramakrishna Padmanabhan. I am thankful for Hovav Shacham's

guidance my first year of graduate school.

My research would not have been possible without the help of the people who provided me with data and tools. Alistair King's program, corsaro, was integral in efficiently analyzing IBR. Brian Kantor answered my questions about the IBR collection process. Alistair and Daniel Andersen, who maintain the systems for storing and processing of UCSD's IBR, were very understanding whenever I inadvertently used more than my fair share of resources. Michalis assisted me in replicating UCSD-NT findings with MERIT's darknet. The mystery of the Qihoo 360 bug was solved (and fixed) with the help of Brian, Cooper Nelson, Nevil Brownlee, Louis DeKoven, and Vern Paxson. Paul Pearce provided me with BitTorrent data. Danny Huang's translation of a Chinese web site allowed me to associate traffic with Mythware software. Chiara Orsini's help with BGPStream allowed me to quickly compare IBR-inferred path changes with AS-level routes. The UCSD Writing Center and Steve Checkoway's template were valuable in writing my dissertation.

Moving to a new city and starting graduate school is exciting and stressful. Fortunately, I rarely ate dinner alone thanks to a cooking collective, comprised of Alan, Bridgette, Dan, Greg, Jason, Marlena, Ming, and Ryan. Additionally, I met other people in my cohort with whom I frequently socialized: Akshay, Dan, Manish, Matt, board game master Sheeraz, Bay Area co-explorer Vicky, and Wilson; as well as new (female!) graduate students in other departments: Amabel, Catherine, Irene, Jenny, Sakina, Sam, Saralin, and Sohini. Jenny and Sam, also Mercer County, NJ natives, were invaluable in assuring me that I was not alone in my struggles to find the right research topic.

The amazing people in CAIDA, CSE 4240, and Sysnet made the time spent at school fun and interesting. I will miss cheering for Chiara and Ioana at lunchtime ping-pong matches, and justifying American-style Italian food with Alessandro. CAIDA potlucks were a highlight thanks to Brad, Matthew, Young, Vasilis, and many of the

people previously mentioned. Dan, Qiushi, Igors, Joseph, and honorary officemate Julie, provided a daily chorus of hellos upon arrival in 4240, conversations that lasted the entire afternoon, and a knowledge of Magic cards (despite my never actually playing). The Sysnet group is, in my opinion, the friendliest in UCSD's computer science department. From that group I have made many friends: Alex (please carry on the CAIDA/Sysnet legacy), Arjun (thanks for the witty conversations), Brown & Louis (thanks for the fun nights out in PB), Danny (thanks for the tour of Google's campus), David (thanks for your efforts to encourage women in CS), Joe (thanks for planning group dinners), Karl (thanks for encouraging everyone to hack), Kirill (thanks for the reminders that I study trash), Malveeka (your kindness and intelligence are refreshing to be around), Sunjay (please continue the poster session social hour), Tristan (it was fun hanging out in Japan), and Wilson (thanks for convincing me to go to UCSD, at Georgia Tech's visit day).

Outside of my lab mates, the CSE department has many other wonderful people. Dustin was an excellent listener, cook, and Bananagrams opponent. Jessica indulged my desire to visit as many taco shops in San Diego as humanly possible. Neha always offered me good life advice and brought me to more Japanese stationary stores than one would imagine existed. Zach encouraged my excessive excitement in mundane things, including grunions, continuous time zones, and state quarters. It was fun to plan social hour with Dimo, John, and Valentin; though most of our socializing was outside Friday's 4pm hour. Additionally, I am thankful for my interactions with Ailie, Alex, Niki, Michael, Nima, Sam, Panos, and Yasmine.

It is hard not to enjoy the outdoors in San Diego, and I was fortunate to meet many active people. Joining the UCSD Club Field Hockey team my first year resulted many weekend adventures. I'm grateful for my friendships with Amanda, Joyce, and Sarika which have continued long since we stopped playing together. The CSE Run Club brought together many students across the department for exercise and conversa-

tion. Through the CSE Run Club, I am happy to have gotten to know: Brian, Janani, Kawa, Matteo, Ming, Natalie, Olivia (an excellent Girls on the Run coach), Qiushi, and Zhaomo. Aaron was a great training partner for the Chicago marathon, where I qualified for the Boston marathon. The San Diego Track Club's Boston group contains so many wonderful and inspiring people. With this group, I welcomed waking up at 6 am for 20 mile jaunts up Mount Soledad.

My roommates made living in San Diego extremely fun: Alan (and Wai-San), Alex, Augusta (and Jesus), Kai, Qiushi, Rhiannon, Sarah, Thomas, and Yajaria. I met a number of wonderful women, including Emily, Laura, and Stephanie, through the Wellesley's Young Alum Club of San Diego.

I am also indebted to many people outside of UCSD. My Wellesley CS professors, especially Lyn Turbak and Randy Shull, made me excited about computer science as an undergraduate. Susan Hohenberger encouraged me to apply for PhD programs. During my time at ARL, I would frequently brainstorm with Kerry Long and Olisa Stephensbaily about the things we could do with passively collected traffic, which was excellent practice for graduate school.

On a more personal level, Karla has been an indispensable sounding board on the entire PhD journey, from applying to graduating. Shelley encouraged me to participate in network forensic contests and attend DEF CON. Courtney, Erika, Lisa, Rachel, and Varun were always a phone call or text away. Finally, my family has been extremely supportive. My mom has been integral in helping me silence my imposter syndrome. My dad reminds to do the important things in life (eating tacos, running, going to the beach, etc.). My siblings, who are also in PhD programs, may tease me about being a nerd, but were happy to listen to me ramble about "cryptographic accelerators."

As per UCSD's guidelines, I would like to acknowledge the projects that have

been incorporated into this dissertation:

Sections 3.2 and 6.1.4, in full, are adapted from material as it appears in SIGCOMM Computer Communication Review. Dainotti, Alberto; Benson, Karyn; King, Alistair; Kallitsis, Michael; Glatz, Eduard; Dimitropoulos, Xenofontas; ACM, 2013. The dissertation author was one of the primary investigators and authors of this paper.

Sections 4.3.2, 6.2.1, 6.3.1, and 7.1, as well as Chapter 5, in part, are adapted from material as it appears in the proceedings of the Internet Measurement Conference (IMC 2015). Benson, Karyn; Dainotti, Alberto; claffy, kc; Snoeren, Alex C; Kallitsis, Michael; ACM, 2015. The dissertation author was the primary investigator and author of this paper.

Section 6.1, in part, is adapted from material as it appears in the Journal on Selected Areas in Communications (JSAC). Dainotti, Alberto; Benson, Karyn; King, Alistair; Huffaker, Bradley; Glatz, Eduard; Dimitropoulos, Xenofontas; Richter, Philipp; Finamore, Alessandro; Snoeren, Alex C.; IEEE, 2016. The dissertation author was one of the primary investigators and authors of this paper.

Section 7.2, in full, is adapted from material as it appears in the proceedings of 2013 Traffic Monitoring and Analysis Workshop (TMA 2013). Benson, Karyn; Dainotti, Alberto; claffy, kc; Aben, Emile. IEEE, 2013. The dissertation author was the primary investigator and author of this paper.

Section 6.4.2, in full, is currently being prepared for submission for publication of the material. Livadariu, Ioana; Benson, Karyn; Elmokashfi, Ahmed; Dhamdhere, Amogh; Dainotti, Alberto. The dissertation author was one of the primary investigators and authors of this material.

Section 6.4.3, in full, is currently being prepared for submission for publication of the material. Padmanabhan, Ramakrishna; Benson, Karyn; Dainotti, Alberto. The dissertation author was one of the primary investigators and authors of this material.

VITA

2006	Bachelor of Arts, Wellesley College
2008	Masters of Science, The Johns Hopkins University
2008–2010	Computer Scientist, U.S. Army Research Lab
2016	Doctor of Philosophy, University of California, San Diego

PUBLICATIONS

[1] Giuseppe Ateniese, Karyn Benson, and Susan Hohenberger. Key-Private Proxy Re-Encryption. In, *Topics in Cryptology (CT-RSA 2009)*, pages 279–294. Springer, 2009.

[2] K. Benson, A. Dainotti, k. claffy, and E. Aben. Gaining Insight into AS-level Outages through Analysis of Internet Background Radiation. In *Proceedings of the International Workshop on Traffic Monitoring and Analysis (TMA '13)*, 2013.

[3] Karyn Benson, Benjamin Birnbaum, Esteban Molina-Estolano, and Ran Libeskind-Hadas. Competitive Analysis of Online Traffic Grooming in WDM Rings. *IEEE/ACM Transactions on Networking (ToN)*, 16(4):984–997, 2008.

[4] Karyn Benson and Manuel Cebrian. Searching with Cooperators. *Chaos, Solitons & Fractals*, 56:45–52, 2013.

[5] Karyn Benson, Alberto Dainotti, Alex C Snoeren, Michael Kallitsis, and kc claffy. Leveraging Internet Background Radiation for Opportunistic Network Analysis. In *Proceedings of the 15th ACM SIGCOMM Conference on Internet Measurement (IMC '15)*, 2015.

[6] Karyn Benson, Rafael Dowsley, and Hovav Shacham. Do You Know Where Your Cloud Files Are? In *Proceedings of the 3rd ACM Workshop on Cloud Computing Security (CCSW 2011)*. ACM, 2011, pages 73–82.

[7] Karyn Benson and Lisa M Marvel. Using Adaptive Lossless Compression to Characterize Network Traffic. In *Proceedings of 43rd Annual Conference on Information Sciences and Systems (CISS 2009)*, 2009.

[8] Karyn Benson, Hovav Shacham, and Brent Waters. The k-BDH Assumption Family: Bilinear Map Cryptography from Progressively Weaker Assumptions. In, *Topics in Cryptology (CT-RSA 2013)*, pages 310–325. Springer, 2013.

[9] A. Dainotti, K. Benson, A. King, k. claffy, M. Kallitsis, E. Glatz, and X. Dimitropoulos. Estimating Internet Address Space Usage through Passive Measurements. *ACM SIGCOMM Computer Communication Review (CCR)*, 44(1), December 2013.

[10] A. Dainotti, K. Benson, A. King, B. Huffaker, E. Glatz, X. Dimitropoulos, P. Richter, A. Finamore, and A. Snoeren. Lost in Space: Improving Inference of IPv4 Address Space Utilization. *IEEE Journal on Selected Areas in Communications (JSAC)*, April 2016.

ABSTRACT OF THE DISSERTATION

Leveraging Internet Background Radiation for Opportunistic Network Analysis

by

Karyn Benson

Doctor of Philosophy in Computer Science

University of California, San Diego, 2016

kc claffy, Chair
Alex C. Snoeren, Co-Chair

In this dissertation, we evaluate the potential of unsolicited Internet traffic, called Internet Background Radiation (IBR), to provide insights into address space usage and network conditions. IBR is primarily collected through darknets, which are blocks of IP addresses dedicated to collecting unsolicited traffic resulting from scans, backscatter, misconfigurations, and bugs. We expect these pervasively sourced components to yield visibility into networks that are hard to measure (e.g., hosts behind firewalls or not appearing in logs) with traditional active and passive techniques. Using the largest collections of IBR available to academic researchers, we test this hypothesis by: (1) iden-

tifying the phenomena that induce many hosts to send IBR, (2) characterizing the factors that influence our visibility, including aspects of the traffic itself and measurement infrastructure, and (3) extracting insights from 11 diverse case studies, after excluding obvious cases of sender inauthenticity.

Through IBR, we observe traffic from nearly every country, most ASes with routable prefixes, and millions of /24 blocks. Misconfigurations and bugs, often involving P2P networks, result in the widest coverage in terms of visible networks, though scanning traffic is applicable for in-depth and repeated analysis due to its large volume. We find, notwithstanding the extraordinary popularity of some IP addresses, similar observations using IBR collected in different darknets, and a predictable degradation using smaller darknets. Although the mix of IBR components evolves, our observations are consistent over time.

Our case studies highlight the versatility of IBR and help establish guidelines for when researchers should consider using unsolicited traffic for opportunistic network analysis. Based on our experience, IBR may assist in: corroborating inferences made through other datasets (e.g., DHCP lease durations) supplementing current state-of-the art techniques (e.g., IPv4 address space utilization), exposing weaknesses in other datasets (e.g., missing router interfaces), identifying abused resources (e.g., open resolvers), testing Internet tools by acting as a diverse traffic sample (e.g., uptime heuristics), and reducing the number of required active probes (e.g., path change inferences). In nearly every case study, IBR improves our analysis of an Internet-wide behavior. We expect future studies to reap similar benefits by including IBR.

Chapter 1

Introduction

Unlike in the natural sciences, where scientists use measurements to test hypotheses about the phenomena governing the universe, researchers and practitioners designed the forces controlling the Internet. These designs, which are publicly available, ensure Internet functionality and interoperability between hosts. Thus, the existence of a subfield of computer science dedicated to measuring the Internet may seem unnecessary. However, measuring the Internet — perhaps the most complex man-made system ever — is imperative for studying its growth, usage and state. While Internet protocols are well-specified, the users, infrastructure, and their interactions are intricate and dynamic.

Analyses of Internet growth, usage, and state are useful from engineering, economic, and scientific perspectives. In each case, measurements can verify the effectiveness of a large system, and plan for improvements. For example, when deploying a new technology (e.g., firewall) or protocol (e.g., SPDY), engineers use measurements to get insight into improvements over existing solutions, the performance in a variety of situations, and adoption rates. A company may decide where economically to place a data center based on their current and expanding markets, as well as the frequency in which failures occur. As computer scientists, we are interested in making similar generalizations about the Internet as a whole, as such insight can help develop best-practices, new protocols, and an understanding of online behaviors.

Measurements alone are often not sufficient to draw conclusions about Internet dynamics. We also need a concrete understanding of the collection processes and a sound interpretation of the data. For example, it is difficult to glean the number of unique people visiting a website [57]. Cookies associate a unique identifier with a browser; however, people may have multiple cookies by using more than one browser or Internet-connected device, surfing the web in incognito mode, or deleting their cookies.

Internet data analysis becomes even more muddled when we do not have control over the end points we are measuring. And, many interesting Internet phenomena fall in this category: the size of a malware-infected population, routing policies across the Internet, the frequency of service disruptions, etc. For these phenomena, researchers must decide which measurement technique, possibly developing their own, will yield the best insight. These techniques fall into two broad classes: *active probing* and *passive collection*.

Active probing consists of the injection of packets into the Internet to elicit a response. At first glance, the ability to communicate with hosts world-wide, appears to provide unrestricted insight into all Internet hosts. Unfortunately, this is not the case. Violations of the end-to-end principle, such as firewalls and Network Address Translation (NAT), render certain hosts unreachable. Moreover, conscientious probing of measurable networks should not induce an excessive load on remote resources; as a result, it is challenging to design an effective probing strategy.

The alternative to active probing is to passively capture traffic at strategic locations, such as webservers or Internet Exchange points. The effectiveness of a passive technique is contingent on vantage point placement. Regrettably, academic researchers are unlikely to obtain data from strategic locations (e.g., at a popular web site, or an important transit link) due to laws, expense, as well as privacy and proprietary concerns.

As a result, due limitations in measurement techniques and infrastructure, both

active and passively collected datasets have inherent biases or provide incomplete views of the Internet. To overcome these limitations, researchers commonly use multiple datasets. Corroborated findings across datasets increase the confidence in inference correctness. Differences in the results can help quantify the limitations of either dataset.

To make these more accurate inferences, researchers may use multiple traditional datasets, such as scans and web logs. Alternatively, Casado *et al.* proposed using "spurious" network traffic, such as Internet Background Radiation (IBR) and spam emails, to illuminate regions of the address space where traditional techniques fail to provide visibility [39]. The insight is that malicious and inadvertent traffic likely contains information relevant to Internet-wide network analysis, though it may require some ingenuity to tease the data into a usable format. ***This dissertation rigorously evaluates the potential for IBR to improve our understanding of address space usage and characterize the state of networks and hosts on an Internet-wide scale.***

IBR is unsolicited Internet traffic, composed of scans (e.g., searching for hosts running a vulnerable service), misconfigurations (e.g., a typo in the IP address for a mail server), backscatter (responses to packets with forged source IP addresses, including spoofed Denial of Service (DoS) attacks), bugs, etc. Though unsolicited traffic can be collected in any network, researchers frequently collect it using darknets, regions of the address space dedicated to collecting IBR (i.e., without any active hosts).[1] Historically, researchers have collected and used IBR to study worms [142, 140, 16, 5], DoS attacks [141], and scanning activity [54, 62]. IBR has many properties that suggest that it may be a good Internet-wide data source: it is of considerable volume, incessant, and originates from a variety of services [156, 217].

Recently, instead of studying malicious activities, researchers have leveraged IBR to learn about hosts and networks generating unsolicited traffic [112, 56, 55, 180]

[1] An alternative name for a darknet is a network telescope.

— a concept proposed by Casado *et al.* [39]. The pervasively sourced components of IBR makes a darknet the potential recipient of traffic from all networks connected to the global Internet. Botnets employ machines worldwide to perform scans; misconfigurations can occur in any network; many networks host services that are potential victims of DoS attacks. These sources enable constant analysis of networks Internet-wide. Moreover IBR may provide insight beyond traditional measurement techniques, such as when a censorship event visible through IBR, was not entirely reflected in BGP messages [56].

However, previous uses of IBR for opportunistic network analysis often focused on isolated events or specific components of the IBR. It is unclear if the same analysis techniques will work on similar events or with different collections of IBR (e.g., at different times or across different IBR vantage points). More broadly, these studies do not provide insight into which properties of IBR are amenable to Internet-wide analysis and whether the networks themselves must have certain characteristics to allow IBR-based inferences.

These gaps in previous research guide the first part of this dissertation. Intuitively, a data source can provide insight into properties of a network, when it legitimately sends a sufficient volume of relevant information. We then turn our intuition into a scientific investigation, by eliminating cases of sender inauthenticity, examining which networks send IBR, identifying components of IBR that enable opportunistic network inferences, characterizing the frequency and granularity of traffic sources, and analyzing sensitivity to time of collection and position of the darknet in the address space. We find: IBR originates from most countries, large ASes and a non-trivial number of prefixes, /24 blocks and IP addresses; the composition of IBR enables Internet-wide measurements through a variety of traffic types; repeated and long-term measurements are possible with IBR; and these findings are consistent in time (at least recently) and space.

The second part of this dissertation examines when it is appropriate to use IBR

Table 1.1. Example inferences with IBR. Inferences made through IBR vary in the number of observations required and the type of packet-level information used.

Number of Observations	Packet Layer: Internet (IP)	Transport (TCP/UDP/ICMP)	Application
One	Ascertaining IPv4 Utilization † (through source IP) Section 6.1	Discovering HTTP Servers† (through TCP ports and flags) Section 6.2	Locating Open Resolvers † (through DNS responses) Section 6.2
		Finding Routers † (through ICMP codes) Section 6.2	Determining Filtering Policy † (through Conficker) Previous work:[180]
Two	Identifying Path Changes ◊ (through TTL) Section 7.1	Determining Uptime † (through TCP timestamps) Section 6.3.1	Evaluating Patch Efficacy † (through Qihoo 360 traffic reduction) Section 6.3.3
Many	Deducing Number Packets Sent † (through IPID) Apply non-IBR method:[40, 120]	Detecting NAT Usage † (through TCP options and TTL) Section 6.4.1	Assessing Software Popularity † (through BitTorrent client) Section 6.3.2
			Analyzing IP Address Sharing† (through BitTorrent ID) Sections 6.4.2 and 6.4.3
Predictable	Detecting Outages ◊ (through number sources) Previous work:[56, 55]	Recognizing Packet-loss ◊ (through packets per connection) Section 7.2	Determining Number of Disks † (through re-seeding of Witty's PRNG) Previous work:[112]

† = Address space usage and attributes ◊ = Network state

as an Internet-wide data source. We assess IBR's applicability in a series of case studies that illuminate IPv4 address space usage and characterize the state of hosts and networks. Table 1.1 shows 15 types of IBR-based inferences, which vary in type of measurement task and along the dimensions of packet-level information (Chapter 4) and number of required observations of the source (Chapter 5). These inferences include both previous studies, and novel uses of IBR. While itself not an exhaustive list, Table 1.1 suggests the versatility of IBR in terms of the number and range of inferences it may be able to support.

This dissertation studies the 11 novel inferences. For each inference, we apply (or extend) the technique to IBR data, examine the results, validate the technique, and discuss implications in the greater context of using IBR for opportunistic Internet analysis. We look to demonstrate the versatility of IBR, expose strengths and weakness of using IBR for opportunistic network analysis, and increase the community's understanding of macroscopic properties of Internet hosts and networks

1.1 IBR Datasets

Our primary datasets are collections of IBR. Both UCSD and Merit Network operate large darknets, which we call UCSD-NT [205] and MERIT-NT [138] respectively. UCSD-NT observes traffic destined to more than 99% of IP addresses in a contiguous /8 block. MERIT-NT covers about 67% of a different /8 block. With access to traffic reaching 0.65% of all IPv4 addresses, this dissertation uses the largest collections of IBR available to academic researchers.

Unless otherwise specified, we study packet traces captured during the time periods of July 31 to September 2, 2012 and July 23, 2013 to August 25, 2013. We chose these time periods since they align with the ICMP-ping based census conduced by ISI [86]. We refer to these 34-day periods as the *2012 census* and *2013 census* respectively. We label our datasets based on the collection site and the year: `UCSD-12`, `UCSD-13`, `MERIT-12`, and `MERIT-13`. To reduce the impact of forged traffic, we sanitize the IBR datasets using the technique described in Chapter 3.2.

1.2 Contributions

This dissertation provides a rigorous assessment of IBR as a data source for inferring network properties on an Internet-wide scale. To this end, we further our knowledge of IBR, develop, test, and refine measurement techniques, and advance our understanding of the Internet. Specifically, the contributions of this dissertation include:

- A method for removing spoofed traffic from IBR. Using this method, we excluded traffic from over 10M /24 blocks in `UCSD-12`.

- A modern characterization of IBR components, which illuminates the complexity of current networked systems and malicious Internet activities. For example, we

identify a new DNS-based attack involving 1.5M open resolvers, uncover large-scale BitTorrent index poisoning attacks that induce hosts in over 2M /24 blocks to send IBR, and detect a byte-order bug in a Chinese security product associated with over 100M IP addresses. We characterize IBR along the dimensions relevant to opportunistic network analysis. In particular, we study the *sources* and the *frequency* in which they send IBR, as opposed to the total *volume* of traffic.

- A template for assessing a data source's visibility. Specifically, researchers can compare IBR to any other data source by performing the series of analyses we present in Chapter 5.

- New methods for detecting IPv4 address utilization, routers, open DNS resolvers, Carrier Grade NAT (CGN) and DHCP lease durations. Notably, our new methods:

  - Increased the number of known used /24 blocks from 4.59M to 5.30M by using multiple data sources, including IBR. In particular, the inclusion of IBR reveals 2.7M /24 blocks primarily used by end users.

  - Discovered 71k /24 blocks with router interfaces that did not send ICMP Destination Unreachable messages in a traceroute dataset.

  - Revealed almost 900 ASes with repeated evidence of CGN deployment, more than two times more than a concurrent Internet-wide study of CGN [172].

  - Corroborated, with beaconing RIPE Atlas probes, the characteristic DHCP address durations for 9 ASes. This method is potentially extensible to any of the 27k ASes sending BitTorrent IBR in that dataset.

- Evaluation and refinement of techniques used by popular network analysis tools to determine system uptime and NAT usage. In both cases, we discover that the

tool uses heuristics that produce false positives. By refining the uptime heuristic, we accurately determine the uptime for tens of thousands of hosts in each of our four 34-day datasets.

- Evidence that using IBR can provide insight into large network events beyond macroscopic outages [56]. Our IBR-based method of detecting path changes provides a picture of routing dynamics consistent with traceroute measurements, without sending any probes. We used our packet loss metric to explore two known cases of congested links, and one case of packet filtering.

1.3 Organization

This dissertation is organized as follows:

In Chapter 2 we discuss related work, including outcomes of comparing two datasets, opportunistic measurement with other data sources, and previous uses of IBR.

In Chapter 3 we analyze the effects spoofed traffic within IBR could have on our IBR-based inferences. We propose and validate a method for removing spoofed traffic from IBR.

Enumerating all types of IBR-derivable information is a daunting, and probably impossible task. Instead, we characterize IBR along dimensions relevant to network measurement. In Chapter 4 we identify the phenomena that induce many sources to send IBR. In Chapter 5 we ask the questions: who sends IBR? How often do they send traffic? We also evaluate our characterization's sensitivity to the time of collection, the position of the darknet in the IPv4 address space, and the size of the darknet.

Next, we conduct a series of case studies that leverage IBR to infer network properties of the Internet. We broadly divide our case studies into inferences revealing information about address space utilization (Chapter 6) and network conditions (Chap-

ter 7).

Based on our characterization of IBR, and the successes and failures of our case studies, we arrive at guidelines for when it is prudent to leverage IBR for Internet-wide inference. In Chapter 8 we summarize these insights and suggest future directions.

Chapter 2

Related work

This dissertation evaluates IBR's potential as a data source to examine properties of networks Internet-wide. We build upon previous work in the areas of (1) evalutating and comparing comparing data sources, (2) the "opportunistic" usage of data sources and (3) understanding Internet Background Radiation. In Section 2.1, we consider previous work that compares multiple data sources, which results in deciding to use one data source over another, scrutinizing the differences to learn new information, or leveraging their collective power — all viable options when comparing IBR to other datasets. In Section 2.2, we examine data sources, other than IBR, which researchers used to opportunistically measure Internet phenomenon. This section provides insight into when it is appropriate to use such measurements. Finally, in Section 2.3, we discuss the relationship between opportunistic usage of IBR and previous work designing darknets, characterizing IBR composition, and investigating malicious events. Also in Section 2.3, we summarize previous examples of using IBR to learn network properties. Our summaries identify gaps in the analysis of IBR's usage as general data source for Internet-wide measurement, which we address in this dissertation.

2.1 Outcomes when using or comparing multiple data sources

In this dissertation, we evaluate IBR's potential to improve visibility into networks. This requires (a) being aware of limitations of other methodologies, and (b) making recommendations on how to best use IBR for a particular measurement problem. For each of our case studies, we discuss limitations of existing techniques in their respective chapters. In this section, we examine the outcomes of other measurement studies with multiple data sources.

Many previous measurement studies use multiple types of data. For brevity, we only discuss a handful of approaches. We find that there are three possible outcomes when analyzing the effectiveness of multiple measurement data sources to answer the same problem: one data source is superior and it is unnecessary to use other sources, the differences between results from the data sources provide unique insight, and combining the data sources provides greater visibility than the individual sources themselves.

2.1.1 Outcome: One data source is superior

Heidemann *et al.* use active probing to study IPv4 address space utilization [86]. To justify their choice, they determined how many USC IP addresses were observed through active probing and passive monitoring within the USC network. Active probing revealed a significant number of used addresses (72% of USC IP addresses inferred as used by either method). Passive monitoring found more used addresses (93% of USC IP addresses inferred as used by either method) than active probing, but the passive technique (measuring USC from USC) was not scalable for the entire Internet — it was impossible to deploy the monitoring in all networks.

Heidemann *et al.* also justified their use of ICMP probes over TCP probes. First,

they inferred more addresses as used through ICMP probes (62% of all known used IP addresses at USC) than TCP probes (54% of all known used IP addresses at USC). This result held for 1M randomly selected IPv4 addresses. Additionally, they received thirty times less complaints from probing surveys conducted with ICMP than TCP.

Heidemann *et al.* had a limited probing budget and it was unclear how to scale their passive technique. As a result, they choose ICMP probes because they provided the best information. In Section 6.1, we use multiple data sources (ICMP, TCP, passive) to study IPv4 address space utilization. We observe slightly more used addresses than ICMP probes alone. However, the straightforward method of sending only ICMP probes resulted a successful on-going effort, with publicly available data dating back to 2003 [95].

2.1.2 Outcome: Interesting differences between the data sources

Dainotti *et al.* showed that IBR can provide valuable insight into country-wide outages [56]. Their analysis compared IBR, traceroute data, and BGP announcements.

For a government-induced censorship event in Egypt, all three data sources signaled an outage. This event showed it was possible to use IBR to detect outages, but it was unclear if there was any advantage to using IBR as opposed to traditional methods.

For similar events in Libya, the data sources painted different pictures. Historical traceroutes were insufficient: there were not enough traceroutes to Libya to infer two short outages. Had the authors known there was going to be an outage in Libya, they could have launched targeted probes. Some outages were visible in both BGP announcements and IBR. However, one known outage was inferable with IBR but not visible in BGP. Since BGP messages and IBR provide insight into different aspects of the Internet (namely, control plane vs. data plane), the authors inferred that this outage was implemented via packet filtering (which only affects the data plane).

The different accounts of the Libyan outages provided valuable information: namely the censorship technique. In this case, it makes sense to use IBR in addition to traceroute data and BGP announcements, so that we can scrutinize the differences in results. Similarly, in Section 6.2.1, the differences in results when actively scanning for open resolvers and discovering open resolvers via IBR, provide insight into which open resolvers are actively used by attackers.

2.1.3 Outcome: Combining the data sources yields improved visibility

Active probing is a natural choice for diagnosing failures, such as path anomalies or reachability problems. The success of detecting an event is tied to probe frequency. However, responsible Internet measurement should not place undue stress on remote networks with excessive probing. At least two works have improved their failure diagnosing systems by combining passive and active data sources.

PlanetSeer characterized failures experienced by clients connecting the CoDeeN Content Distribution Network [223]. Specifically, when a client connected to CoDeeN, PlanetSeer issued a baseline traceroute. As the client downloaded content, PlanetSeer passively monitored the TTL field and the number of timeouts. If these metrics indicated an anomaly PlanetSeer sent additional probes to confirm and analyze the event, including probing the anomalous path from other vantage points. The authors found that PlanetSeer was very effective at identifying short-lived anomalies. Additionally, they could further reduce the amount of active probing by using fewer vantage points, with a small decrease in number of detected events.

Hubble detected black holes across the entire Internet through traceroutes, active ping monitoring, and passive BGP monitoring [107]. It was not feasible to constantly send traceroute probes to every BGP-announced prefix, so Hubble updated topology

maps daily. For constant insight across the Internet, Hubble used pings and BGP messages. These lightweight components triggered additional analysis, including traceroutes from multiple vantage points and reverse path traceroutes. In their analysis of the system, the authors found that BGP monitoring was not sufficient to detect all blackholes, but BGP monitoring supplemented active probing. In particular, for short reachability problems affecting only a few active probing sites Hubble often observed the outage via BGP updates and not ping.

Similarly darknets can help decide when and where to probe for path changes (Section 7.1). By passively monitoring TTL values, we can determine when a path change is likely. Traceroute measurements can then potentially pinpoint the location of the change. This union of data has the potential to greatly reduce the number of probes required to analyze path change dynamics.

2.2 Opportunistic measurement with other data sources

Casado *et al.* considered IBR as one of the many possible sources of spurious traffic to leverage for opportunistic network measurement [39]. They surmised that traffic from worms, scanning, email spam, and misconfigurations would also provide valuable insight into hard-to-measure hosts, citing HTTP traces of Code Red II traffic collected from many different subnets, a daily Eurasian scan, an IP address located at the University of Wisconsin hard-coded into NetGear routers, and a heavily spammed domain as respective examples. In this section, we discuss other non-traditional data sources used indirectly to extract network properties. The most common drawbacks of these opportunistic techniques are either limited visibility or the inability to generalize the findings to the entire Internet — drawbacks they share with IBR.

We restrict our discussion to studies inferring properties of the senders and their

networks used for Internet connectivity, omitting works that uncover malicious infrastructure via spurious traffic [10, 119, 133, 191, 87].

2.2.1 Web traffic

Productive web traffic reveals many insights into Internet properties: NAT, DHCP, and middle box deployment [38], in-transit modification of web pages (e.g., pop-up blockers, advertisements) [169], etc. Another example of leveraging productive web traffic is in Section 6.1, where we compare observed IPv4 address space utilization from darknets and an academic network — where web sites are the largest attractor of used /24 blocks. With web traffic, researchers have the means to attract additional users to their sites, control the content — and measurements — served, and extract detailed information from the two-way connection (not possible with IBR), e.g., user agent and requested page. Web traffic can also contribute to IBR, including scans and requests to unreachable web servers (as studied though one-way traffic by Glatz and Dimitropoulos [79]).

Casado *et al.* detected NAT usage from HTTP traces of Code Red II traffic [39]. They compared the number of packets destined to subnets in 192.0.0.0/8 to packets destined to subnets in other /8 network blocks. Due to Code Red II's preferential scanning of nearby networks, NAT'ed hosts using a private IP address in the 192.168.0.0/16 subnet were more likely to send packets to subnets in 192.0.0.0/8. This technique worked because they had visibility into a subnet near 192.168.0.0/16, they were unable to detect NAT using addresses in the 10.0.0.0/8 and 172.16.0.0/12 subnets since they did not have access to subnets near these private address ranges. Additionally, they warned that their results may not generalize to the entire Internet, as it is less likely that hosts behind a NAT will be infected with Code Red II than hosts using a public IP address. Similarly, as we show in Section 5.3.1, when using IBR, darknet position in the address space

influences our capabilities and results.

2.2.2 Spam

Spam is a useful opportunistic data source due to long-lived TCP flows with mail servers, and insight into hard-to-measure hosts (such as proxies) [39].

Ramachandran and Feamster leveraged spam to understand the network-level behaviors of spammers [168]. They reasoned that understanding the ISP, IP address space, or botnet sending spam is more valuable than analyzing the easily modifiable content of spam emails. They used spam emails reaching a domain with no legitimate addresses to characterize several network-level behaviors including the network (IP addresses, ASes, and countries) of origin and operating system. They cautioned that their traffic may not be reflective of all Internet spam, but claimed it was still interesting as their dataset contained all spam emails reaching a domain.

In general, comparing unproductive data to other measurement sources can provide some confidence that the methodology using unproductive data is sound and provides insight beyond that provided by either source individually. To determine which network-level characteristics can filter out spam, Ramachandran and Feamster compared their spam emails to legitimate emails and BGP measurements [168]. To investigate the spam traffic originated from the Bobax worm, they hijacked one of the worm's DNS servers to identify infected machines [168]. We also evaluate IBR-based techniques by comparing the results to inferences made through other data sources.

The lessons learned when evaluating spam correspond to advice for evaluating IBR. We enrich our analysis by examining who sends malicious traffic. We heed warnings about the accuracy of extrapolating our findings to the entire Internet. Finally, we compare IBR to other datasets to gain additional insight.

2.2.3 Logs

Logs, though almost universally collected, are seldom used to analyze network events [202]. The barriers to wide-spread usage of logs include the volume of irrelevant data, and inability to collect logs at diverse vantage points. However, researchers have used logs conducted to successful longitudinal, in-depth studies, including the work of Labovitz *et al.* [113] and Turner *et al.* [202] which we describe in this Section.

Labovitz *et al.* combined routing tables with failure logs of a medium sized regional network to categorize Internet stability [113]. The proprietary nature of failure data prevented them from obtaining similar data from other providers. With one vantage point, their study of intra-domain routing failures was not comprehensive, but provided valuable insight into the differences between inter-domain routing failures and intra-domain routing failures.

More recently, Turner *et al.* collected router configuration files, syslog archives, and operational mailing list announcements to understand the causes and impacts of network failures in a large regional network over a period of five years [202]. Their logs contained an abundance of information, but reconstructing events was "painful." Additionally, since logs were a lossy data source they recovered instantaneous link state only 90% of the time. Similarly, mailing list announcements contained valuable information, but were difficult to analyze since a non-deterministic social process generated them.

IBR and logs are similar in that they contain excessive data. We need to filter out irrelevant IBR before we can leverage the useful components for opportunistic measurement. In some cases, we remove spoofed traffic (Chapter 3); in other cases, we extract a single class of IBR (e.g., Conficker traffic). Extracting relevant data is one of the biggest challenges of widespread-usage for both logs and IBR.

2.2.4 BitTorrent

BitTorrent, while an unconventional data source, provides insight into many edge networks due to its popularity as a distributed file-sharing application. Since it is a P2P network, hosts connect to other hosts throughout the world. The main challenge for BitTorrent-based measurement is attracting many clients to connect (or peer) with the machine performing measurements. Another challenge is that, some ISPs, like corporate networks, do not permit P2P applications — a source of potential bias in the resulting BitTorrent-based inferences.

In 2006, Isdal *et al.* connected to BitTorrent file-sharing swarms and examined the large number of visible hosts [100]. They observed ≈500k IP addresses with only eight BitTorrent vantage points over a span of a week. They extracted upload capacity, latency, network topology, and bandwidth using standard BitTorrent messages. Their study covered over 20k BGP prefixes in almost 4k ASes; they note that BitTorrent visibility into prefixes and ASes was greater than data sources involving (1) an academic CDN, (2) downloading robots.txt files, and (3) a worm outbreak. Additionally, BitTorrent traffic did not trigger Intrusion Detection System alarms.

Isdal *et al.* found swarms by harvesting popular web sites for statistics on BitTorrent trackers [100]. Since their 2006 study, BitTorrent evolved to use a Distributed Hash Table (DHT) — a decentralized method of providing "trackerless" torrents [124]. To obtain BitTorrent measurements, researchers at Northwestern created an extension for the Vuze BitTorrent client, Ono, which contained about 3,000 lines of measurement related code [42]. As an incentive for end users to install Ono, the extension improved peer selection — by leveraging its measurements to select the best peers and reducing cross-ISP traffic (with some tricks involving CDNs).

Over 200k hosts installed Ono and collected DNS redirections, transfer rates,

path latencies, and traceroute measurements [42]. The resulting dataset was huge: they observed over 2.3M peers per day, and collected over 1.2M traceroutes per day. Researchers also leveraged Ono to characterize ISPs [32] and analyze outages during natural disasters [31].

BitTorrent traffic is a major component of IBR, which we leverage to study IP address space utilization (Section 6.1), as well as NAT and DHCP (Section 6.4.1 and 6.4.2). We rely on erroneous entries in clients' hash tables (e.g., from index poisoning attacks) to obtain IBR with a BitTorrent payload. Consequently, the amount of BitTorrent traffic observed in a darknet is erratic.

2.2.5 Discussion

In this section we have described a number of unconventional data sources, which researchers leveraged to learn network characteristics. Many of the challenges in unproductive HTTP traffic, spam, logs, and BitTorrent also apply to IBR. We are concerned with: extrapolation of our findings to the entire Internet, validation and comparison to other data sources, easy identification of relevant components, and source diversity. Compared to the other data sources, especially web traffic and logs, IBR is easily obtainable. Additionally, IBR is an evolving mix of traffic (including several of these unproductive data sources), while the other sources are more homogeneous.

2.3 Previous studies using IBR

Analyzing malicious Internet activity is crucial to understanding the attack methods and vulnerable services. Security researchers often collect information about threats at the end host (e.g., through antivirus software) and the network (e.g., with snort [176]). However, there are a number of technical and privacy issues with disseminating knowledge of attacks. Differentiating between legitimate and malicious activity (in particu-

lar, new attacks) requires expertise in computer operations and intimate knowledge of the monitored network. Additionally, administrators are reluctant to share information about the success of past attacks, or the extent to which they are vulnerable to future attacks (e.g., network resources, configurations). Due to these issues, researchers often lack an Internet-wide view of attack techniques and the scale of malicious activities.

One solution is to use a darknet (also called a network telescope) [144]. A darknet is a large collection of routed, but unused regions of the address space. Since there is no legitimate traffic associated with these regions of the address space, there are relatively few privacy concerns with sharing IBR (although IBR may reveal hosts infected with malware). Additionally, most traffic reaching a darknet, is malicious in nature: unsolicited scans, backscatter from Denial-of-Service (DoS) attacks, etc. As a result, darknets are well-suited to capture many Internet-wide security phenomena (e.g., scanning techniques, the size of an infected Worm population).

Understanding the historical context of darknets and IBR provides insight into the challenges of using the data source and the potential uses of IBR. We discuss design decisions in deploying a darknet (Section 2.3.1), the type of traffic observed in a darknet (Section 2.3.2), and the success stories of using IBR to characterize Internet-wide security events (Section 2.3.3); each of these topics have implications for opportunistic studies of IBR. Additionally, we summarize previous work in the area of network analysis via IBR, noting gaps that this dissertation addresses (Section 2.3.4.

2.3.1 Questions to consider when constructing a darknet

Many parameters affect the traffic a darknet observes. This section addresses tradeoffs associated with responding to IBR, the number of unused addresses comprising a darknet (its size), and darknet placement.

Should we respond to unsolicited traffic?

In this section, we discuss technologies related to darknets. For the purposes of this dissertation, we consider a darknet to be a purely passive method of collecting unsolicited traffic. Application-layer responders are custom pieces of software that mimic an application for the duration of a flow, e.g., they complete a TCP handshake. Honeypots are a collection of resources (physical or virtual) that exist for the sole purpose of being infected with malware.

Responding to attack traffic can reveal complex attack methods and malicious activities after infection. A number of researchers have built lightweight application-layer responders to IBR [156, 18, 17, 218]. To be functional, these lightweight responders needed to make assumptions about IBR. For example, to tame the traffic volume, Pang *et al.* assumed all traffic from a source IP address results in the same activity (i.e., there was no need to respond to traffic destined two darknet IP addresses from the source, as the source used the same exploit) [156]. To save on storage space, the Internet Motion Sensor responded only to traffic where the initial payload did not match any previously observed signature [17].

It is difficult to accurately mimic many applications; as an alternative, researchers set up machines with the expectation that they will be compromised, called honeypots [163]. Vrable *et al.* showed, through virtualization, that it is possible to have both high-quality responses (i.e., they execute the kernel or application code permitting the researchers to witness the infection and subsequent actions made by the compromised machine) and to track infections [210]. Unfortunately, honeypots are resource intensive and complex. There is an inherent tradeoff between the quality of responses and the number of monitored IP addresses. Both the University of California, San Diego (UCSD) and the Merit Network operate /8 darknets – a factor of 256 more addresses than

Vrable *et al.*'s deployment. Thus, choosing to use IBR to analyze security events, is a choice to study breadth rather than depth.

Outside of the security realm, responding to IBR may provide additional insights. A response may help differentiate between spoofed and non-spoofed traffic (Chapter 3). Additionally, NATed hosts and hosts behind firewalls are hard to measure with active probes; however, responding when these hosts send IBR may provide insight into hard-to-measure networks.

How many IP addresses are required for accurate inference?

Moore *et al.* found that the size of the darknet (the number of IP addresses used to collect IBR) influenced its ability to detect network events, as well its precision in event duration and rate [143]. Specifically, they showed scenarios in which a /8 darknet has a more accurate view of security events than a /16 darknet, under assumptions of uniform selection of IP addresses. For example, a DoS lasting 1 minute at a rate of 500pps had more than a 95% chance of sending at least 100 packets to a /8 darknet, but only a 36.7% chance of sending one packet to a /16 darknet. With the smaller darknet, it was difficult to accurately characterize DoS attack magnitude and volume. In a simulation of a Code-Red-like worm, a /8 darknet observed the true infection rate, while the curve for a /16 darknet was distorted. Moore *et al.* also noted some practical limitations of large darknets, including: overloading the links used to collect IBR, insufficient storage and processing capacity, routing instabilities, and difficulties differentiating simultaneous IBR events.

Are some IP addresses more desirable than others?

Location of the darknet within the address space influences the observed traffic. The Internet Motion Sensor used a collection of darknets in academic networks, enterprise networks, tier 1 ISPs, national ISPs, regional ISPs and broadband providers [47,

17, 18]. A distributed architecture is preferable because the composition and magnitude of IBR varies across collection points. The darknets used in the Internet Motion Sensor varied in size from many /24 blocks to a single /8 block. The traffic received by the darknets varied in normalized packet rate, amount of local preference (fraction of traffic from same /8 network), and top ports. In particular, although a few hosts were responsible for the majority of traffic volume reaching an individual darknet (10% of IP addresses were responsible for 90% of packets [18]), the same source IP addresses were not observed across darknets. The researchers attributed to the targeted nature of attacks and episodic nature scanning activities.

Hotspots, or deviation from uniform targeting, are one cause of discrepancies across darknets. Cooke *et al.* [48] characterized the cause of hotspots as either algorithmic (host-centric, programmatic) or environmental. Analyzing data collected by the Internet Motion Sensor they found examples of algorithmic discrepancies: botnets using a hit-list, bad entropy in PRNGs, poorly designed PRNGs; and environmental discrepancies: Network Address Translation (NAT), and filtering by Fortune 100 companies. Wustrow *et al.* [217] studied five /8 darknets to determine the effect of general allocation of unused address space. They found that, in terms of traffic volume, environmental factors were a significant source of non-uniformity, especially to the 1.0.0.0/8 block. The environmental factors included traffic to IP address with patterns (1.1.1.1 and 1.2.3.4), byte order bugs (1.*.168.192), discrepancies between hexadecimal and base 10 numbers, and false information about a server on an eMule forum.

How does position relative to live networks influence observed traffic?

There is also a concern that scanners or malware may blacklist darknets once the region is known to be unused [20]. While this hypothesis has not been fully tested, some malware binaries have avoided regions of the address space used by researchers [224].

Additionally, in a 2004 study, Shinoda *et al.* easily located passive Internet threat monitoring systems that publicly released aggregate statistics [184]. Instead, researchers may use greynets, or a region of the address space sparsely populated with unused IP addresses to monitor IBR reaching live networks. Cooke *et al.* developed a method to find unused IP addresses for a local network [46].

Previous studies in this area extracted valuable information from greynets. Harrop *et al.* found little difference in inferred scanning activity (Sasser infection as well as linear scanning) from contiguous dark IP addresses (i.e., the configuration of the previously mentioned darknets), and distributed IP addresses [85]. Glatz and Dimitropoulos monitored a regional academic backbone network and inspected all one-way flows, i.e. traffic without a response [79]. They captured activity commonly found in darknets (most flows were associated with malicious scanning), as well as traffic associated with live hosts such as temporary service disruptions, P2P applications attempting to access unavailable peers, and applications using a separate connections for data transfer and control. In conclusion, if blacklisting is a legitimate concern, a plausible solution is to monitor IBR from live regions of the address space.

Are darknets feasible as we transition to IPv6?

Although primarily deployed using IPv4 addresses, the concept of a darknet easily extends to IPv6. Czyz *et al.* used covering prefixes (i.e., they announced prefixes which include used networks, and collected traffic to prefixes without more specific BGP announcements) that encompassed about 86% of all allocated IPv6 networks outside of 6to4 [51]. They found that the rate of traffic reaching a large IPv6 darknet was about 500 times less than a /8 IPv4 darknet, and that there was little evidence of malicious activity in IPv6 IBR. As IPv6 deployment grows, we expect IPv6 darknets to collect larger volumes of IBR, and to provide similar types of insights into Internet phenomena

as current IPv4 darknets.

Using traditional techniques, scanning the IPv6 address space will take significantly longer than scanning the IPv4 address space [43]. Scanners may be able to reduce the time to scan the address space by exploiting spatial patterns and densities of used IPv6 addresses [160, 204]. This implies that passive techniques, including IBR analysis, may provide crucial information for measuring IPv6 networks.

Discussion

Although previous research studied darknet parameters in the context of analyzing malicious traffic, the parameters also impact our ability to infer network properties of remote networks. Although, with a large, unresponsive darknet, our analysis is limited to the initial communication attempts, we expect to be able to broadly and accurately characterize both malicious traffic and the networks generating IBR. We are aware of potential differences between our darknet and (1) other darknets due to IBR's non-uniformity, as well as (2) live regions of the address space due to their different use cases and blacklisting. In this dissertation, we use traffic from multiple darknets and analyze the effect of darknet size and position (Chapter 5).

2.3.2 Previous work characterizing IBR composition

At least three studies examined the composition of IBR as a whole. Studying this aspect of IBR can lead to the discovery of new attacks and bugs. Understanding the origins of unsolicited activity is beneficial, not only to the security community, but also for the insights it provides into the nature of IBR. For example, general studies of IBR non-uniformity provided operational insight into the placement of darknets to collect the most worm traffic [47, 48] and the allocation of used IP addresses [217]. More generally, as a source of measurement data, we are interested in the number of hosts

associated with an IBR phenomenon, the likelihood that an event continues (permitting repeated analysis), and the overall diversity in IBR phenomena (e.g., our dependence on a single activity).

Moreover, the composition of IBR influences the types of opportunistic network analysis we can perform. For example, backscatter from a DoS attack results in many packets in a short burst. As a result, it is unlikely that we can conduct long-term analysis of sources sending only backscatter. Some types of network inferences are application-level specific. If the application's usage declines, our ability to apply that application-level specific inference techniques declines. For example, since the machines infected with Witty were patched, Kumar *et al.*'s techniques to analyze infected hosts are no longer applicable [112]. For other network inferences, the type of traffic is inconsequential (e.g., In Chapter 7.1, we detect path changes using the TTL field, which is present in every IP packet). However, even in these cases, IBR composition may influence the scope, volume, or temporal aspects of the traffic used in opportunistic analysis.

In this section, we describe papers that broadly characterize the make up of IBR. These papers describe IBR as an evolving, unpredictable data source. We discuss the current composition of IBR in Chapter 4.

In their seminal 2004 paper, Pang *et al.* analyzed IBR at three locations, with the help of application layer responders [156]. Although TCP was the dominant protocol at all three locations, the authors did not observe a consistent composition in part due to site-specific scans and filtering. Across locations, they found: known exploits, old worms, malformed DNS queries, and empty connections; and only a small percentage of sources contacted both a /8 network and a smaller network at a national laboratory. A key finding was about the dynamism of IBR: sources sending IBR were unlikely to be observed the next day or month, due the mix of traffic changing on a near-daily basis.

Wustrow *et al.* studied IBR in the context of allocating new IP addresses: an

allocated address should receive a reasonable amount of IBR [217]. In particular, they examined the number of bytes and packets reaching five /8 darknets over three weeks, and a single /8 darknet over five years (2006-2010). They found that the amount of IBR was increasing over time, faster than productive Internet traffic (as compared to work measuring commercial Internet traffic at diverse vantage points [114]); TCP was the dominant protocol, except in 2008 due to the Slammer worm; Conficker accounted for significant non-uniformity; the 1.0.0.0/8 address block received more traffic than other monitored blocks; scanning accounted for most packets except in 1.0.0.0/8 where misconfigurations dominated. In some cases, only a few thousand hosts (or less) were responsible for large discrepancies in IBR volume.

In 2012, Brownlee noted the composition of IBR was evolving such that is difficult to discern new activities based on packet and byte counts [35]. As a solution, he proposed *iatmon* to detect new classes of IBR. iatmon classified traffic by its type (e.g., to a single source IP address and port, to many ports on the same IP address, to many IP addresses on the same port; as well as TCP, UDP, backscatter, etc.) and inter-arrival time (e.g., three seconds between retransmits, stealthy, DoS). To detect new classes of IBR, Brownlee aggregated type and inter-arrival statistics over volume and number of sources. During a period of six months in 2011, the volume of traffic did not change; however, UDP probes increased while TCP scans of many IP addresses on the same port decreased. Additionally, over a period of half a month the number of sources decreased by about a factor of two. Brownlee's classification can serve as a starting point to investigate new types of IBR.

While these studies provide insight into the diversity and persistent nature of IBR, we require an updated analysis of IBR, with attention to properties relevant to making opportunistic Internet-wide inferences. Pang *et al.*'s study is over ten years old [156]; and since this time the Internet (and IBR) has evolved significantly. Wustrow

et al.'s study does not provide detailed insight into the sources that send IBR [217]. A phenomenon generated by a handful of hosts may produce many packets and/or bytes, but provide little insight into Internet-wide behavior. Brownlee studied how certain properties of sources change over time, but did not identify the phenomena responsible for the changes [35]. Moreover, these studies did not address spoofed packets (traffic where the source IP address is forged), which will likely lead to incorrect inferences of network properties.

2.3.3 Previous work using IBR to analyze malicious activities

Analysis of IBR has led to a better understanding of DoS attacks, many worms, and scanning techniques. The benefits, concerns, limitations and assumptions made by the studies of malicious activity extend to opportunistic network analysis. IBR provides the ability to study a phenomenon across many networks without deploying measurement infrastructure in each network, and archived IBR permits long-term analysis and reuse for diverse purposes. To accurately study both malicious activity and infer network properties, we must analyze the effect of darknet size and placement, and be aware of limitations of darknet infrastructure (e.g., packet-loss during an outbreak of IBR) and our ability to extract specific traffic components. Furthermore, with IBR we may have an incomplete picture of network activities; as a result we may require additional data or make assumptions about our data.

Backscatter

Moore *et al.* studied DoS attacks by examining backscatter reaching a darknet [141]. With a long-standing collection of IBR collected at UCSD, they analyzed attacks of over four years (2001-2004). They studied the protocols used (mostly TCP), the attack rate (65% of attacks could overwhelm a server), attack duration (very few attacks

last more than an hour), victim type (most victims were home users or small businesses), victim domains (`.com` and `.ro` were the most popular "three-letter" and "country-code" TLDs respectively), and repeat attacks (relatively uncommon). Although they used a /8 darknet to characterize DoS attacks, they observed many of the same attacks in smaller darknets. This analysis required making assumptions about attack techniques (attackers spoof addresses uniformly), the network (low packet loss), and IBR (unsolicited responses are from attacks).

Worms

The first worm studied with a darknet was Code Red [142]. Darknet data led to an analysis of the number of infected hosts, the rate at which machines were patched or rebooted and characteristics of the infected machines. Similar observations have been made for other worms, including: Slammer [140], Witty [112], Blaster [16], and Conficker [5]. The ease and breadth of collection provided numerous analysis benefits. Since darknets captured IBR over a long period of time, researchers could study the persistence and origins of infection. In particular, the Blaster worm showed evidence of infection in over 90k /24 blocks two years after the outbreak [16]. Kumar *et al.* provided strong evidence that a certain IP address acted as "Patient Zero" of the Witty worm and that the target of the attack was a US Military base [112].

There are limitations to using IBR to study worms. Darknets do not respond to probes, so it may be difficult to identify worm traffic. Wei and Mirkovic noted that dropped packets, Network Address Translation, non-uniform scanning, short lifetimes of infectees, and equipment errors may lead to incorrect inferences about worm dynamics [213].

Slammer and Witty use a single UDP packet to spread. Worms using multiple packets, including worms that spread via TCP, are less straightforward to detect.

- The Code Red worm spreads over TCP port 80. One analysis considered a host to be infected if it sent at least two packets to two unique darknet IP addresses on TCP port 80 [142]. Although seemingly effective in 2001, this technique is unlikely to differentiate Code Red from other types of TCP port 80 scanning. Responding to TCP port 80 probes would reveal which are associated with Code Red infections.

- The Conficker worm spreads via TCP; however, a bug in its pseudorandom number generator (PRNG) means that only one-quarter of IP addresses in a /8 darknet are targets [41]. Thus, we can differentiate Conficker from other scans based on the set of destinations scanned (see Appendix B).

- Some worms, such as certain versions of Code Red [142], preferentially probe hosts on a local network. But because contiguous darknets do not have active hosts, local preference is not observed through IBR. (i.e., Moore *et al.* use a /8 darknet to study Code Red; by definition their darknet not contain any hosts that preferentially scan the /8 network).

- Researchers used IBR to study extremely fast spreading worms, including the Slammer worm [140]. Fast spreading worms may congest links on the paths from infected machines to their destinations, including darknets. As a result of congestion, routers will drop packets and inferences about worm dynamics may be incorrect. Wei and Mirkovic analyzed worm behaviors correcting for this type of error [213]. They found that the standard analysis of a /8 darknet (multiplying the number of scans by the darknet's size) underestimated the number of scans per second by a factor of three.

Scanning

In addition to studying the scanning activity of worms, darknets are well suited to perform general studies of scanning. Darknets have been used to assess frequently scanned protocols, ports (in many recent studies Conficker resulted in TCP port 445 having the most packets [217, 62]), as well as attributes of the originating sources and scan dynamics.

Durumeric *et al.* studied scans that target a large darknet at a rate of 10 pps or faster [62]. In particular, they studied the amount of scanning that occurs after major vulnerability disclosures e.g., announcement of the Heartbleed bug [199]. They found that scanning often commenced within 48 hours of the announcement; many scans did not appear to be for research purposes; and large scans originated from bullet-proof hosting providers, and not botnets.

Dainotti *et al.* [54] studied a scan from the Sality botnet. Most sources involved in the scan would not reach Durumeric *et al.*'s threshold of 10 pps: over a span of 16 days 3M IP addresses scanned 86.6% of a /8 darknet, but sources sent, on average, 6.85 probes each. The scanners in the Sality botnet were stealthy and highly coordinated, as evidenced by their approach (reverse byte order), coverage, and adaptivity of scan rate.

Researcher also use live networks to conduct scan analysis. One longitudinal study of scanning used traffic reaching Lawrence Berkeley National Laboratory [9]. Traffic from live networks will capture scans from attackers that blacklist darknets, but since live networks are typically smaller than darknets, it will be harder to capture scan dynamics (i.e., the reverse byte order scanning found by Dainotti *et al.* [54]). Yegneswaran *et al.* compared port scan behavior collected by 1600 firewall administrators to scan behavior in a /16 darknet [219]. They found that both datasets captured large scan dynamics (i.e., traffic from worms), but for smaller scans they observed fewer scans

and more variability with their darknet. Yegneswaran *et al.*'s study implies that larger or distributed darknets are necessary for comprehensive scan analysis.

Discussion

The benefits and challenges of using IBR to study malicious activities often map directly to benefits and challenges of using IBR to study network properties.

In terms of benefits, IBR can capture traffic from a variety of networks, long-term behavior, and complex behavior. Moore *et al.* found that many networks were victims of DoS attacks [141]; we leverage the variety of networks sending IBR to study IPv4 address space utilization (Section 6.1). Researchers studied worm persistence via IBR; by analyzing 8 years worth of IBR, we conclude darknets are a sustainable source of measurement data (Section 5.1.1). Dainotti *et al.* explored stealthy, coordinated scanning behavior in IBR [54]; we investigate complex BitTorrent behavior to study CGN and DHCP (Sections 6.4.2 and 6.4.3).

In terms of challenges, we may need to: make assumptions about attack techniques (e.g., who are targets of DoS attacks), the network (e.g., amount of packet loss) and the nature of IBR (e.g., why we receive traffic); adjust models to account for measurement error; and consider how darknet size and position affect our results. Moore assumed reliable delivery of attack packets to the victim and darknet when studying DoS attacks [141]; we assume that the network has near constant speed when studying uptime (Section 6.3.1). Wei and Mirkovic adjusted their model of worm propagation to account for packet loss [213]; we caveat that we find a *lower bound* on the total number of open DNS resolvers (Section 6.2.1). Yegneswaran *et al.* compared scan findings from live networks to a /16 darknet [219]; we generally examine darknet placement and size on our ability to perform opportunistic network inferences with IBR (Section 5.3.1).

2.3.4 Previous opportunistic uses of IBR

Instead of using IBR to study malicious activity, we can use IBR to study the machines and networks generating the traffic. Casado *et al.* [39] formalized this idea in 2005, citing the difficulties in measuring Internet growth in size and complexity due to NAT and firewalls. They proposed that spurious traffic (IBR and other traffic such as SPAM emails) could provide insight due to: the large number of sources, diversity in sources (not just academic), and social acceptability (does not consume large amounts of bandwidth, and few privacy concerns). In order to use spurious traffic for opportunistic measurement analysis, they stated as requirements (a) measurement-specific specifications (e.g., long-term and predictable for path characterization), (b) enough traffic to be statistically significant, and (c) visible to the researcher. Casado *et al.* provided examples of using spurious traffic to infer network properties, but did not fully evaluate the potential of IBR datasets to meet these specifications.

Casado *et al.* cited Kumar *et al.*'s analysis of the Witty worm [112] as an example of leveraging IBR to infer network characteristics. Kumar *et al.* reverse engineered the Witty worm's PRNG, which generated a sequence of IP addresses to scan. The PRNG's seed was system uptime. As a result, Kumar *et al.* inferred the system uptime for infected machines. Witty was a destructive worm that would attempt to overwrite a randomly chosen disk; if the randomly chosen disk existed, the PRNG was reseeded. As a result, Kumar *et al.* inferred the number of disks a machine had based on whether or not reseeding occurred. Witty used a blocking system call to send packets. As a result, Kumar *et al.* inferred the access bandwidth of infected machines based on the timing between packets. The Witty worm is no longer spreading, so the analysis of uptime, number of disks, and bandwidth cannot be applied to the current composition of IBR.

A less application-specific use of IBR is to detect outages. Dainotti *et al.* ana-

lyzed the number of packets per second from Egypt and Libya during censorship events, resulting in outages in the respective countries [56]. They found that total volume of IBR sharply decreased during the outages. By component, only backscatter did not reflect the outages (although packets revealed DoS attacks on sites belonging to the Egyptian government). As we discuss in Section 2.1.2, this study showed that IBR can provide insight into macroscopic events even when more traditional methods fail.

In a follow up study, Dainotti *et al.* used IBR to analyze the impact geophysical events on computer networks [55]. They used a metric (ratio of packets received prior to the event versus after the event) to determine if an earthquake affected an geographic area (on the order of kilometers). They determined the magnitude (the impact/fraction of machines affected) and radius (how far away were the effected machines) of the earthquake. An earthquake of stronger magnitude (Tohoku) on the Richter scale had a larger impact on computer networks than a smaller earthquake (Christchurch), as shown by the magnitude and radius metrics. This method showed that IBR is applicable to smaller portions of the address space than entire countries; however, there are still gaps in the general applicability of IBR. It is unclear if outage analysis will work for other countries, smaller networks or geographic areas, or at shorter time scales (e.g., outages lasting only a few minutes).

Sargent *et al.* showed that, in conjunction with other data, IBR can provide insight into filtering policies [180]. Specifically, they checked that sources sending traffic to Conficker sinkholes (domain names registered by researchers because Conficker will eventually use the domain name as a rendezvous point) are also observed in darknet traffic on TCP port 445. (Conficker spreads by randomly scanning the Internet.) If Sargent *et al.* observed at least five sources from an AS in sinkhole data, but none in darknet data then they inferred the existence of TCP port 445 filtering; observing a source in both datasets implied no filtering. Their technique characterized the TCP port 445

routing policy for 28% of the routed IPv4 address space – much larger than any current technique. Unfortunately, they did not believe that their technique would extend to other ports due to a low number of sources.

Discussion

The previous literature used IBR to make a handful of inference types (uptime, number of disks, bandwidth, outages, filtering). This dissertation increases this list substantially, and provides a foundation to use IBR-techniques in additional settings.

Chapter 3

Santization: Removing spoofed traffic

IPv4 lacks a mechanism to verify a sender's authenticity. Specifically, the sender sets the source address field of every packet. For malicious or inadvertent reasons, the sender may forge the source IP address. We call this act of IP address forgery "packet spoofing." There are a wide range of malicious packet spoofing attacks, including: spoofed Denial of Service (DoS) attacks, reflective DoS attacks, TCP connection spoofing, decoy scans[1], idle scans (also called bounce scans)[2], and zombie control (the spoofed packets act as one-way control messages) [196, 127, 129]. Inadvertent reasons for spoofed packets include bit flips, misdirected malicious attacks, traffic emulations escaping a local network, and software errors such as byte-order bugs.

The central idea of this dissertation is to convert packets into measurements of the sender's network and spoofing can grossly distort our analysis. For example, when studying IPv4 address space utilization (Section 6.1), spoofed traffic, if not re-

[1] In a decoy scan, the attacker scans a remote host with both non-spoofed and spoofed packets. It is difficult for the victim to determine which of the IP addresses is the non-spoofed origin of the packets [127].

[2] An idle scan can determine if a port is open without sending a packet to the targeted machine with their own IP address. The method exploits the fact that, on many operating systems, the IPID field increases by one for each connection. An attacker can determine if a port is open by (i) sending a legitimate packet to a "zombie" machine to get its current IPID; (ii) sending a packet spoofed with the zombie's IP address to the targeted machine and port; (iii) sending another legitimate packet to the zombie to get its IPID. If the IPID increased by one between (i) and (iii), the scanned machine did not respond to the spoofed packet, and the port is not open; if the IPID increased by two between (i) and (iii), the scanned machine likely responded to the spoofed packet, implying that the port is open [196, 129].

moved, results in a threefold over-estimation of used /24 blocks. Previously, researchers sidestepped this problem by selecting only packets exhibiting properties of known non-spoofed traffic (e.g., in studies to infer filtering policy [180] and outages [56] researchers selected only Conficker packets). This selective method discards many packets. In our study of IPv4 address space utilization just using Conficker traffic results in a five-fold under-estimation of used /24 blocks. Consequently, including all traffic will result in incorrect inferences, while including only certain types of traffic will underutilize the signal IBR provides. To fully and accurately utilize the signal provided by IBR, it is imperative that we selectively remove spoofed traffic from our datasets.

In general, detecting spoofed packets is difficult. A highly motivated attacker can craft packets to evade spoofing detection by forging many packet headers and using carefully selected IP addresses. Fortunately, since darknets are not associated with services and end users, such an attacker is unlikely to send traffic to a darknet. We find that darknets are rarely the targets of attacks, and that most spoofed packets reach the darknet inadvertently (i.e., not maliciously). Therefore, our focus is on detecting the inadvertent spoofing of many IP addresses.

In this chapter, we first discuss related work identifying and removing spoofed traffic (Section 3.1). The focus of many of these related works is mitigating spoofed attacks on live networks, and are not directly applicable to IBR. We then propose and evaluate a new method to remove spoofed packets from IBR (Sections 3.2). Our method looks for abnormalities in the distribution of IP addresses and is applicable to large, historical datasets like IBR.

3.1 Related work

Previously, researchers studied spoofed traffic in the context of mitigating the effects of spoofing attacks [196, 104, 27], understanding threats [218, 151], and creat-

ing accurate Internet models [21, 222]. This dissertation is most aligned with creating an accurate Internet model; however, related work in this context requires responding to traffic [21] or relies on an assumption (that spoofed addresses are uniformly distributed) [222] that does not appear to hold in IBR (as shown in Section 3.2.5). Another unique aspect of this dissertation is the use of IBR; most of the related work used traffic from live networks which are often targets of malicious spoofing. The works that studied spoofed IBR required responding [218, 21] (we currently do not have an infrastructure to respond) or small collections of IBR [151].

The remainder of this section discusses these methods and their relevance to the specific problem of removing inadvertent spoofing from IBR. We taxonomize work in this domain along two dimensions: the packet fields analyzed and the technique used to infer spoofing.

3.1.1 Packet fields indicative of spoofing

To identify and remove spoofed traffic we can leverage abnormalities in three packet fields, including time-to-live (TTL), IP identification (IPID), and source IP address. In this section, we explain the behavior of these fields under normal circumstances and during spoofing episodes.

Time-to-live (TTL) field

A host's operating system generally sets the time-to-live (TTL) field to a default value [182]. Each router that forwards the packet decrements the TTL to prevent routing loops.[3] Under the assumption that all packets sent by a host to the darknet take the same route, the TTL's final value should remain constant at the receiver. As a result, a TTL different than previously observed packets from the IP address may indicate spoofing.

[3]Router drop packets if the TTL is zero.

There are several circumstances where the TTL value is not a reliable indicator of spoofing. First, many hosts take the same number of hops to reach a destination. As a result, it is possible that a spoofing source is the same distance away from the darknet as the address it forges. Second, multiple hosts, due to Network Address Translation (NAT) or DHCP, may use a single IP address. As a result, if the hosts use different operating systems (e.g., Windows hosts Linux hosts use different default TTL values), the TTLs of packets they send will differ upon entering the global Internet. Alternatively, large NAT deployments may include hosts that use a varying number of hops to reach the gateway. Finally, Internet routes are constantly changing and there is no guarantee packets take the same route (e.g., load balancing paths).

Additionally, TTLs may be set arbitrarily. For example, in IBR, we observe spoofed traffic where all TTL values appear with the same frequency — likely caused by the sender uniformly selecting a random initial TTL value. An attacker n hops from a destination can set the initial TTL such that the TTL at the recipient is any value smaller than or equal to $255 - n$.

A similar metric to TTL is the inferred hop count (the number of routers that forward a packet). To calculate hop count, we first infer the starting TTL (normally the next highest power of 2). We then set hop count to starting TTL minus observed TTL. Although inference of the starting TTL can cause some errors, hop count is preferable over TTL when there is some variance in starting TTL, e.g., due to NAT or traffic generated by multiple applications which use different starting TTLs.

IP identification (IPID) field

The IPID field assists in fragmentation and reassembly of IP packets [201]. Many operating systems implement the IPID field as a counter. As a result, consecutive connections from the same host will have consecutive IPID values. From a remote

observation point, if we receive two packets from the same host in a short period of time, we expect the IPID values to be close to each other. For example, when a host randomly scans the Internet, at a constant rate, the IPID will appear to increase linearly with time.

There are a number of exceptions to this method's assumptions. Due to NAT, multiple hosts may use the same IP address, resulting in IPID values that appear non-linear over time.[4] More importantly, some operating systems do not implement the IPID field as a counter [214]. RFC 6274 states that counter implementation are inappropriate due to excessive wrapping and security concerns, including idle/bounce scans; many operating systems follow this guideline [81]. Thus, it is possible to observe non-linear IPID values for both non-malicious (i.e., non-counter implementations of the IPID field) and malicious (i.e., to evade spoofing detection) reasons.

Source IP address

IP-address-based methods work best when the source spoofs a significant number of addresses. More sources increases the likelihood that the spoofed addresses include ones that should not originate global Internet traffic. It is unlikely that we can detect small spoofing events with IP address-based methods.

IP address-based methods assume that the process generating the spoofed packets selects the spoofed IP addresses indiscriminately, e.g., randomly.[5] In particular, there are a number of addresses we should not observe in global Internet traffic:

- reserved addresses [49], including:
  - private addresses (e.g., 10.0.0.0/8 used on local area networks)
  - loopback addresses (e.g., 127.0.0.1 for this machine)

[4] In the case of NAT, it is possible to extract linear subcomponents, which may reveal properties of the hosts sharing the same IP address (e.g., number of hosts) [40].

[5] With inadvertent spoofing, it is generally a safe to assume indiscriminate selection of spoofed addresses. In other situations, such as reflective DoS attacks, attackers may pick the spoofed addresses purposefully.

  - multicast addresses (i.e., 224.0.0.0/4)
  - test addresses
  - future use addresses

- dark addresses (e.g., from UCSD-NT)
- addresses unannounced in BGP (intermediate routers are unable to route responses)

Observing these addresses may indicate spoofing. Zander *et al.* leverage the number of unexpected addresses to estimate the amount of random spoofing in an entire dataset [222]. In our method, described in Section 3.2, we use traffic with unexpected addresses to obtain signatures for spoofed packets.

IP address spoofing may not be random. For example, in our 2012 dataset, we find a spoofing episode where all the source addresses are from the 88.0.0.0/8 block. Additionally, it is a best current practice for ISPs to drop outbound packets when the source address is not in a range they originate [68]. Although this practice is not universally adopted, many hosts spoof only IP addresses in their local network [28]. Non-random address generation and filtering can restrict the ranges of addresses visible in our datasets.

Moreover, not all traffic with an unexpected address is spoofed. A small fraction of observations result from legitimate packets escaping a private network. Not only do we observe legitimate packets from IP ranges reserved for private use (e.g., 10.0.0.0/8), we also observe "IP squatting," or using non-private IP addresses on a local network [1]. Another potential source of misclassification is partial BGP visibility (we use RIPE's suggested threshold to consider a /24 block routed if it is covered by prefixes announced by at least 10 BGP peers [215]).

Table 3.1. Summary of techniques to identify spoofed traffic. The methods to identify spoofed traffic use a variety of packet fields and techniques to obtain reference data.

Method	Packet Field	Technique		
		Active	Comparative	Aggregate
Respond and wait for reply [218, 21]	N/A	X		
Probe and check response TTL [196]	TTL	X		
Probe and check response IPID [196]	IPID	X		
Prove and check response OS [196]	multiple	X		
Respond with TCP window = 0 [196]	TCP window	X		
Respond with invalid ACK [196]	ACK Number	X		
Compare to historical values [196]	TTL, OS		X	
Compare to expected hop count [104, 27]	TTL		X	
Look for abnormal distribution [151]	TTL			X
Identify linear IPIDs [151]	IPID			X
Look for many dark sources [222]	IP address			X
Look for many total/unrouted sources (Section 3.2)	IP address			X
Look for increase in total/unrouted sources (Section 3.2)	IP address			X

We use a method based on source IP address to remove spoofed traffic from our datasets, which we describe in Section 3.2. We believe using addresses unannounced in BGP is preferable to using only darknet addresses to detect spoofing. First, it is difficult to know if subnets of darknets are in use, since both used and unused subnets are announced in BGP (while we can easily extract the networks that are unannounced in BGP). Assuming uniformly random spoofing, detecting forgery using only a handful of known dark blocks will, in expectation, require significantly more spoofing than using all unrouted blocks. For example, Zander *et al.* [222] restricted their analysis to six dark /8 blocks — an order of magnitude fewer than the number of unannounced IP addresses. Additionally, with unannounced addresses we are more likely to catch non-uniform spoofing, as the addresses are spread throughout the IPv4 space. However, it seems reasonable to use both unrouted and dark addresses for spoofed traffic identification.

3.1.2 Techniques for identifying spoofed traffic

We categorize previous work into three categories: techniques that actively respond to verify a packet's authenticity, techniques that compare packets to other previously collected data, and techniques that identify anomalies in large collections of data.

In general, there is a tradeoff between overhead and our ability to capture targeted spoofing. For example, active techniques require significant infrastructure to respond to all unsolicited traffic but can potentially identify even a single spoofed packet; aggregate techniques do not require additional measurement overhead, but can only detect events with many packets. Table 3.1 summarizes work used by other authors and our proposed method.

Active techniques

Sustained bidirectional communication implies that the hosts are not forging packet headers. For example, analysis of traffic at an academic backbone traffic reveals that selecting bidirectional TCP flows with at least five packets and an average of 80 bytes/packet is an accurate heuristic for removing spoofed traffic [52]. With IBR, we can also respond to unsolicited traffic to elicit a response. In this section, we use the following terminology to discuss the three packets involved in an active technique: (1) the unsolicited traffic, including IBR which is unsolicited traffic sent to darknet, (2) the probe we generate to check if the unsolicited traffic is spoofed, and (3) the response to our probe.

Several projects generated probes from IBR, including the iSink architecture [218]. In particular, using 2004 data, Barford *et al.* compared the distribution of sources throughout the IPv4 address space from iSink (spoofed and non-spoofed) and Dshield (raw data without spoofed traffic identification/removal) [21]. Barford *et al.* found that spoofing did not have a significant impact on the density of malicious IP addresses. However, we show this finding does not hold in more recent collections of IBR in Section 3.2.5.

A response to our probe ensures that there is a host associated with the IP address; however, a response is not sufficient to ensure that the host that sent the unso-

licited traffic is also the one that responded to the active probe. For example, if we send a TCP SYN-ACK probe in response to a spoofed TCP SYN packet, the legitimate host with the spoofed IP address could respond with a TCP RST. Templeton and Levitt outlined a number of basic checks on the response to the active probe to substantiate that the same host sent the original packet: (1) the TTL is the same; (2) the IPID is slightly higher than the initial packet; (3) the packets appear to be from the same operating system [196]. Additionally, Templeton and Levitt pointed out that the active probe could induce the remote host to alter the flow, such as (4) setting the TCP window size to 0 (host should stop sending packets) or (5) setting the ACK-number to a lower value than expected (forces the host to send a resynchronization ACK). A spoofing source will not receive the flow-altering packets and will continue uninterrupted, while an authentic host will respond appropriately to the active probes.

Generating a probe to check IBR's authenticity can be challenging. While it is trivial to respond to a single TCP packet sent to a darknet, it is often not straightforward how to construct responses involving the application layer, including constructing UDP responses and receiving multiple TCP packets (e.g., to increase our confidence such as the "five packets and 80 bytes" heuristic in [52]). Pang *et al.* built software to generate application-level probes to classify the origins of IBR [156]. However, they note that building this type of software is "difficult due to the lack of detailed documentation on services;" their implementation generated probes for 10 ports. In our datasets, a non-trivial fraction of traffic is unclassified because we are uncertain of the packet's encoding (i.e., the application protocol used). Moreover, it is impossible to generate a valid probe for some types of IBR (e.g., byte-order bugs).

Applicability to IBR: We did not apply active techniques to IBR. First, we currently do not have the proper infrastructure to respond. Second, we often analyze IBR in a historical context and the attributes of IPID, TTL, and OS may change between time

of probing and time of collection. Finally, although responding to IBR seems plausible (over 10 years ago, iSink responded to 20k connection requests per second [218]), there is no formal assessment of these techniques' success.

Comparative techniques

Active techniques generate probes to confirm that a packet is not spoofed. The next set of techniques seeks confirmation from other sources, including historical data [196] and models created from previously collected probes [104, 27]. These techniques are effective with much lower overhead than active techniques, but are less accurate due to Internet dynamics. In particular, instead of reflecting spoofing, packet attributes may reflect changes to the Internet that occurred between the time of the reference data and the time of the data in question. For example, TTL may not be an effective attribute during path changes that alter the number of hops it takes for authentic packets to reach a host. Similarly, the OS may be unreliable, as multiple hosts may use an IP address due to sharing (i.e., NAT) or reassignment (i.e., DHCP).

Templeton and Levitt examined TTL predictability and applicability for spoofed packet detection [196]. They examined traffic reaching their lab's network over a period of two weeks and calculated the conditional entropy. The conditional entropy was low (i.e., the TTLs were highly predictable) across many dimensions: protocol, internal/external, and number of packets per IP address. This suggests high predictability in TTLs from non-spoofed packets. However, they observed behaviors, besides spoofed traffic, that resulted in differing TTLs: UDP packets took a different path than TCP and ICMP packets; traceroute resulted in a source sending packets with varying TTLs.

Jin *et al.* created an accurate IP-address-to-hop-count mapping [104], i.e., a list of the number of hops packets from each IP address take to reach a destination. The basic idea is to mark as spoofed any packet where the hop count, inferred from

the TTL, does not match the expected value in the mapping. Jin *et al.* outlined the technique's robustness to a number of attacks (single spoofing source, multiple spoofing sources, randomly setting the TTL). To obtain the expected hop count value, the authors suggested using TCP connections with a completed 3-way handshake. Unfortunately, it is unreasonable to expect an Internet server to receive traffic from all sources. Jin *et al.* proposed three methods for estimating the expected hop count of unseen addresses based on other hop counts in the same /24 block: the same as the minimum observed hop count, within one or two hops of the minimum observed hop count, and based on clustering smaller prefixes of the /24 block. The authors extracted a IP-address-to-hop-count mapping from traceroutes from 47 sites, and evaluated their methods with randomly selected spoofed and legitimate packets. The clustering approach provided high coverage with low false positive and false negative rates.

Beverly implemented a machine learning agent, Raskol, that improved upon the basic IP-address-to-hop-count mapping approach [27]. Instead of using data passively collected during 3-way handshakes, Beverly obtained training data from responses to random "pinging" of routable IP addresses. Raskol estimated the hop count of unobserved addresses by creating a model of the Interent's complex peering relationships and the routes packets traverse from real topology measurements. To test Raskol, Beverly varied the amount of legitimate traffic, placement of the sensor detecting spoofed traffic, and spoofing strategies (e.g., randomly, reflective, worms). Raskol was also implemented in hardware, highlighting its ability to process packets in a real system.

These comparative techniques were designed to mitigate spoofed attacks in real time on live networks. In live networks, accidentally dropping legitimate connections because they are believed to be spoofed, will upset users. As a result, comparative techniques should default to permitting packets through a firewall. Thus, the small number of spoofed packets that have a proper TTL are permissible, and have limited effect on

communication. Another option is to use a tunable technique, such as Raskol [27], which allows the application to determine a permissible level of spoofing.

Applicability to IBR: We did not apply comparative techniques to IBR to conduct opportunistic network analysis. The main reason is a mismatch between the problem comparative techniques solve and our goals. Comparative techniques make decisions in real time on a per-packet basis. With IBR, we need to make decisions on historical data to detect large-scale spoofing. Aggregate techniques (described in the next section) are more straightforward for offline, large-scale spoofing identification. In the future, we could apply comparative techniques to IBR to catch additional small-scale spoofing.

Aggregate techniques

Both the active and comparative techniques infer if traffic is spoofed on a per-packet basis. Aggregate techniques infer if spoofing is present in a set of packets. With a substantial amount of traffic, these techniques can easily identify large spoofing events without complex models or the overhead of responsive methods. However, aggregate techniques require additional processing to isolate and remove the spoofed traffic from the authentic data. In this section, we summarize the identification and removal steps of two aggregate techniques [151, 222].

Identification of spoofed traffic: Previous aggregate techniques used the TTL [151] or the source IP address [222] to identify large-scale spoofing. We discuss the specifics of the related works and potential sources of error when applying the techniques to IBR.

Ohta *et al.* examined the TTL field of darknet data for statistical abnormalities [151]. Specifically, based on previous observations, they calculated the expected number of packets received for each TTL value, as well as the standard deviation. Then, if the number of packets received with a given TTL value was more than two standard

deviations away, they investigated the resulting traffic to determine which traffic should be removed.

We believe there are a few shortcomings of Ohta *et al.*'s technique. First, the amount of spoofed traffic must be a significant fraction of the total packets. Secondly, it is unclear if this approach is applicable when multiple hosts generate spoofed traffic (the distribution of TTLs may mimic non-spoofed traffic). Finally, fluctuations in IBR may result in abnormal packet volumes.

Zander *et al.* [222] examined examined the number of IP addresses in their datasets from six dark or almost dark /8 blocks to determine the fraction of spoofed source addresses. They used this fraction to interpolate the expected number of spoofed addresses in a network block, assuming uniform spoofing. Although their data supported the uniform-spoofing assumption, we are hesitant to make this assumption due to our observations of non-uniform source address spoofing, related work that found destination addresses receive varying volumes of IBR [217], and a history of poor random number generation [48, 5].

Removal of spoofed traffic: Active and comparative techniques operate on a per-packet basis, which naturally leads to combining the identification and removal steps (i.e., these techniques remove all packets identified as spoofed). With active techniques, once we determine that a set of packets contains some spoofed traffic, we must do some additional work to remove the non-authentic components.

Ohta *et al.* looked for packets with IPID values that increased linearly over time, which is indicative of originating from the same machine; if the packets had different source addresses then they inferred spoofing [151]. They believed, but did not implement, a progressive probabilistic Hough transform [74] would extract linear components. Instead, they relied on visual inspection of IPID values, which is unlikely to scale to larger darknets. For example, graphing (IPID, timestamp) yields a completely col-

ored image: in 2015 UCSD-NT typically received over 5M packets per minute, which means that about 75 packets have the same IPID per minute. Analyzing the relationship between IPID and timestamp for a given TTL value (the identification step of Otha *et al.*'s technique) will yield a sparser graph, but will still be difficult (a) when analyzing common TTL values or (b) when IPID increases at an unknown or slightly variable rate.

Zander *et al.* removed spoofed traffic from NetFlow records to estimate the number of used IPv4 addresses [222]. Their concern was not with the contents of the spoofed traffic (nor do they have access to the spoofed payloads), but rather the error on their estimate. Several of their datasets did not contain spoofing (e.g., Wikipedia edits), which they leveraged to remove traffic from their NetFlow datasets with spoofing. Specifically, they removed traffic from /24 blocks using the following rules: if the block was unused in the spoof-free datasets they counted the block as unused; otherwise, they removed IP addresses from their dataset based on the popularity of the address' last byte. Zander *et al.* did not verify the accuracy of their approach, but found (1) after removing traffic, the NetFlow datasets captured approximately the same number of /24 blocks as other datasets, (2) that their estimate of the used number of /24 blocks was similar to an inference that excluded the NetFlow datasets, while using unfiltered NetFlow data resulted in a much higher estimate. This technique would decrease the estimate of used IPv4 addresses when the population of users captured via NetFlow differs from their other datasets.

Applicability to IBR: We use an aggregate method to identify and remove spoofed traffic from IBR. We choose this identification approach because it applies to large and historical datasets. Moreover, we currently do not run the infrastructure to respond to IBR. Aggregate methods work only on large spoofing events, and err on the side of excluding traffic — limitations that are acceptable for IBR.

3.2 Our technique for identifying and removing spoofed traffic from IBR

To mitigate the effects of spoofing on inferences based on darknet measurements, we devise a method that aggregates IP addresses to identify abnormal behaviors. We then build signatures to filter out suspicious components by manually isolating and analyzing anomalous traffic. We focus on spoofed traffic that appears to originate from many sources (such as randomly spoofed traffic), which we call *large-scale* spoofing. We assume that the remainder of spoofing (called *small-scale* spoofing) is not only difficult to detect without responding to received packets, but has a much smaller impact on our inferences, which we confirm at the end of this section.

In search of large-scale spoofing, we look for both bursts of spoofed traffic, and long-term consistent spoofing. We examine many aspects of the traffic:

1. *burstiness in number of unique sources:* we find sudden spikes in the number of unique source IP addresses and unique source /24 blocks;

2. *burstiness in number of newly observed sources:* we look for a large number of newly observed source IP addresses (source /24 blocks) per hour;

3. *burstiness in number of unrouted networks:* we find the same type of bursty events with only source addresses in unrouted network blocks;

4. *long-term consistent observation of unrouted sources by port:* we aggregate packets over the entire measurement window into traffic classes by protocol and port (when applicable) and investigate classes with many originating unrouted /24 blocks;

5. *long-term consistent least significant byte behavior:* we aggregate packets over the entire measurement window based on the least significant byte of the source

address to look for inconsistencies in address utilization (e.g., Fan and Heidemann found that IP addresses ending in 1, 129, 65, 33, and 2 respond to ICMP probes most frequently [66]).

We start our sanitization process by investigating bursty behaviors. We perform our analysis iteratively. After identifying a class of spoofed packets, we create a filter and remove the traffic from the entire dataset. We then look for additional bursty behaviors. Once our sanitized data no longer exhibits bursty spoofing behavior we examine our datasets for long-term consistent spoofing behavior.

Our approach is a time-consuming, manual process that requires in-depth knowledge of networking and packet analysis techniques. However, our approach seems to be an accurate method of removing spoofed IBR. One benefit of manual inspection is the insight into the properties of spoofed traffic. With more automated approaches (e.g., active techniques, removing traffic based on least significant byte order popularity) we would need to perform additional analysis to develop filters for common spoofing events.

3.2.1 Identifying bursty spoofing behavior

For bursty traffic, we apply a simple spike-detection algorithm, flagging hour-long time bins when we observed more than 25% more sources (or unrouted sources) than the average value observed over the last ten hourly time bins. We tried different values for the parameters — time-bin duration, spoofed threshold, and time window — without observing significant changes in what was detected as spoofed, since most events of interest cause large traffic variations.

Figure 3.1 shows that some bursty spoofing events are not visible when considering packets from all sources, but they become easily detectable when looking only at source addresses of unrouted networks. In some cases, this phenomenon is due to the nonuniform distribution of unrouted networks over the address space, e.g., the tempo-

rary popularity of some address blocks as source addresses despite little change in total number of spoofed sources.

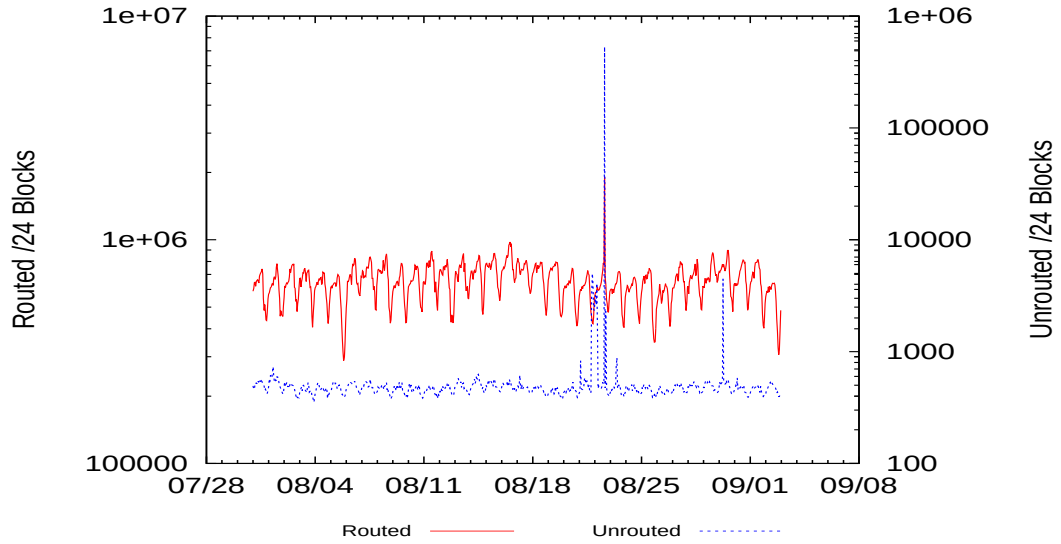


Figure 3.1. Routed and unrouted networks by hour `UCSD-13`. We observe significant increases in the number of unrouted source networks for some hours, which we inspect to discover and exclude spoofed traffic.

In some cases, we find that bursts of routed IP addresses (or /24 blocks) are due to changes in the composition of IBR. Typically, in these cases, the number of unrouted IP addresses and never-seen-before /24 blocks is still low. We also examine the distribution of TTLs and countries to determine if they match the non-bursty periods. Examples of bursts of non-spoofed routed IP addresses include: an increase in traffic due to a BitTorrent index poisoning attack, and a misdirected reflective DDoS from Quake servers (i.e., the Quake servers received spoofed packets, and responded using their legitimate source address).

Spoofed IBR is often caused by buggy software or simulation traffic escaping a local network, and is rarely malicious in nature. Our technique is versatile, as it detected a variety of spoofing types. Examples of bursty spoofing events include:

- UDP packets with payload "zzzzzzzzzz" to darknet IP X.X.X.X where X is

the /8 network corresponding to UCSD-NT. A likely explanation is that packet-generating software inadvertently sent these packets on the Internet.

- Packets with protocol 0, and source address A.B.0.0. These packets could be generated by buggy software that incorrectly writes certain packet fields.
- TCP SYN packets to a single darknet IP address on port 80. This is likely a misdirected, malicious spoofed denial of service attack.
- BitTorrent traffic with abnormally large TTL values (above 230). A possible explanation is that traffic from a simulation escaped a local network.
- TCP SYN traffic appearing to originate from almost all networks in 88.0.0.0/8 to several darknet IP addresses on port 44.

3.2.2 Identifying consistent spoofing behavior

To identify consistent spoofing behavior, we aggregated the entire dataset by protocol and port, and then examined classes of traffic with either more than 10 unrouted /24s or a percentage of unrouted /24s greater than 0.4% (0.4% was approximately two orders of magnitude lower than the 39% of /24s that were not announced on BGP during our measurements). These thresholds are orders of magnitude less than abnormalities discovered via the bursty spoofing events. Classes with traffic below these thresholds were difficult to infer as spoofed based on traffic patterns, but these thresholds are sufficient to remove large-scale events.

We also aggregated based on the least-significant byte of the source address. Our intuition is two-fold: (1) we expect the probability of observing IBR to depend on the last octet of the IP address [66], and (2) packets with least-significant byte 0 and least-significant byte 99 were symptomatic of several spoofing spikes. We investigated cases where a certain value of the least-significant byte rarely occurred in routed networks yet

Table 3.2. Summary of filtering heuristics used in darknet measurements and their impact in terms of source /24 blocks. We defined filters that captured general characteristics of spoofing, but in some cases we eliminated spoofing traffic specific to our darknets. For each general filter and the aggregate of all the specific filters, we report the total number of /24 blocks used as sources in packets captured by the darknets, as well as the number that are unrouted and dark.

	Num. /24s			Num. Unrouted /24s			Num. Dark /24s		
	UCSD 2012	MERIT 2012	UCSD 2013	UCSD 2012	MERIT 2012	UCSD 2013	UCSD 2012	MERIT 2012	UCSD 2013
TTL> 200 and not ICMP	10M	9.7M	11M	31k	69k	1.3M	120k	68k	110k
Least significant byte source address 0	660k	430k	45k	22k	1.7k	7	3.9k	1.0k	540
Least significant byte source address 255	328k	270k	44k	300	1.3k	6.7k	34	54	1.6k
Non-traditional protocol	61k	61k	57k	16k	17k	2.3k	512	509	720
Same source and destination address	630	1	96	0	0	0	630	1	96
No TCP flags			3.5k			640			29
UDP without payload			550			110			0
All specific filters	1.9M	980k	11M	530k	290k	1.3M	16k	7.5k	110k

commonly occurred in unrouted networks. For example, we excluded the broadcast address (least significant byte of 255) because it is used the second least when the network is routed; but in routed networks it is the 26th least used (after removing spoofed traffic identified by spike detection). In this per-least-significant-byte aggregation, we compared the relative popularity of an address' least-significant byte in routed and unrouted networks.

We added only two new filters as the result of aggregate analysis. The low number of new filters added from aggregate analysis implies most large-scale spoofing traffic is bursty, and does not exhibit not long-term consistent behavior.

3.2.3 Removing spoofed traffic

Each spoofing behavior we identified exhibited distinctive properties, which we synthesized into a set of filtering heuristics. Our goal is to come up with general heuristics that characterize a spoofing behavior. For example, we found spoofed packets with a BitTorrent payload. Our options, from least preferential to most are: (1) exclude all

BitTorrent traffic, (2) exclude BitTorrent traffic for the duration of the spoofing episode, or (3) look for other aspects of the traffic indicative of spoofing. Many legitimate sources send BitTorrent traffic, including packets with a similar payload. Fortunately, we found that the spoofed BitTorrent packets had an abnormal TTL value. We call a filter to exclude abnormal TTL values "general," while options (1) and (2) are "specific" to our datasets.

Table 3.2 lists how many /24 blocks (respectively total, unrouted-only, dark-only) originated traffic matching each heuristic. The first seven lines of Table 3.2 describe "general" filters, which other researchers can readily apply to other datasets. The last line of Table 3.2, "All specific filters," aggregates results for a set of filtering criteria specifically crafted for abnormal events observed in one of the darknets. They do not seem generally applicable, so we only report the aggregate effect these filters on our datasets.

The first heuristic, based on the value of the TTL IP header field, filters out by far the largest number of /24 blocks. We found 20 spikes (11 in UCSD data and 9 in MERIT data) where a significant number of UDP packets with unrouted sources had the same destination port and TTL above 200. Our filter excludes traffic based on the large TTL since it indicates a general abnormality: most operating systems use a default TTL of 128 or less [182] (although, several switch to a TTL of 255 when sending ICMP packets).

Other significant portions of spoofed traffic use uncommon or unassigned protocols, but such behavior could also be legitimately experimental so we do not exclude traffic solely for this reason. But when many packets with an uncommon protocol appear to originate from unrouted addresses, it is more likely they are the result of bit-flips during transmission or programming errors when writing packets. Similarly, TCP packets without flags and UDP packets without a payload indicate that there were errors writing

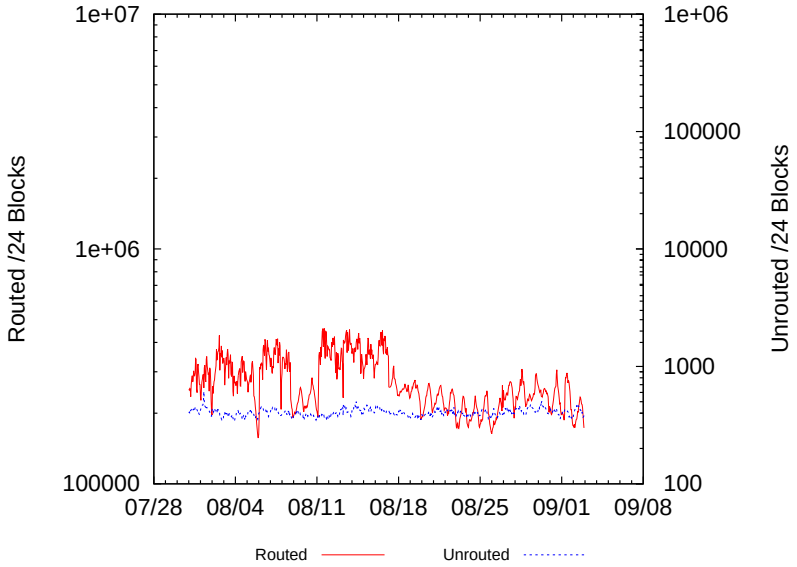


Figure 3.2. /24 blocks observed per hour after removing spoofed traffic. The resulting dataset has very few unrouted /24 blocks per hour. Additionally, compared to Figure 3.1, there is a significant decrease in both routed and unrouted /24 blocks.

or transmitting the packet. We exclude packets with source address ending in .0 or .255 since traffic should not originate from these addresses (when part of a /24 subnet). We also identified small spoofing events where the source and destination were in the same darknet.

Our heuristics based on traffic spikes filter out not only the likely spoofed traffic during the spike, but also traffic matching this filter outside of the spike. We find that the application of a filter to entire dataset is a necessary step. In Figure 3.2, we plot the number of routed and unrouted /24 blocks observed in UCSD-12 after applying our filters. The peak in routed /24 blocks is around 300k per hour for each day throughout the entire 34-day period; for comparison in Figure 3.1 the daily maximum is about 700k /24 blocks in a single hour. This is a substantial drop. The fluctuations in Figure 3.2 are due to changes in IBR composition.

While we believe we could confidently apply the general filters to any traffic to reduce spoofing, they are not a complete set of rules to exclude spoofed data. In

Section 3.2.6 we provide evidence that this set of filters removes spoofed traffic for the time period studied; for a different time period or network, repeating the identification and removal steps is necessary to exclude data-specific spoofing.

3.2.4 Need for a multi-faceted approach

Our methodology found and removed over 10M /24 blocks. We use a multi-faceted approach to examine many aspects of traffic. While it is tempting to cut down analysis time by analyzing only a single aspect of the traffic, we caution that this will likely lead to incomplete removal of spoofed traffic. The spoofed BitTorrent traffic and traffic from 88.0.0.0/8 show some potential weaknesses of analyzing a single aspect.

The spoofed BitTorrent traffic appeared throughout our 2012 datasets, but only triggered an abnormal event in about 10 hourly bins out of more than 800. We flagged these hours as abnormal due to the traffic's fluctuating volume. Specifically, we detected the spoofing at periods of high volume. This suggests the threshold of acceptable unrouted /24 blocks must be low, and that observing the start of a spoofing event is important, otherwise the spoofed traffic may become part of the baseline. Without fluctuations, we would detect the BitTorrent traffic with our long-term consistent technique: the class of traffic using port 65535 appeared to originate from over 10M /24 unrouted blocks as a result of spoofed BitTorrent traffic.

The traffic from 88.0.0.0/8 did not appear as a spike in unrouted data, but as a large increase in the number of new /24 blocks. This event shows that processes generating spoofed traffic may not choose address randomly from the IPv4 address space. It is possible that the originator of the traffic attempted to spoof the entire IPv4 address space but egress filtering dropped all packets not in 88.0.0.0/8. This spoofing behavior suggests that it is beneficial to use ranges throughout the IPv4 address space as spoofing indicators. For instance, most /8 blocks include some unrouted addresses; had the

forged packets appeared to originate from a different /8 block, it is likely the event would cause a spike in unrouted networks. More generally, using similar addresses as spoofing indicators (e.g., from a single ISP) restricts the classes of spoofing that we can detect.

These events also highlight the benefits of using multiple methods to identify spoofed traffic. In both cases we corroborate that the traffic was spoofed by examining the TTL (e.g., the technique of [151]). For the spoofed BitTorrent traffic the TTL was above 230, and for the 88.0.0.0/8 traffic the TTL was either 94 or 95 — a significant deviation from the expected distribution.

3.2.5 Influence of spoofing on network analysis

Darknets observe so much spoofed traffic that neglecting it would invalidate our inferences. The influence of spoofing is pronounced in our study of IPv4 address space utilization, where we classify a /24 blocks as used if it appears in a IBR dataset. An example of the potential detrimental effects of unmitigated spoofing is the first heuristic in Table 3.2, which covers approximately 10M /24s, whereas our final estimates of active /24 blocks are around 3M per darknet. In total, our filters reduce the number of active /24 blocks by 7.2M /24.

We visually show the pronounced impact of spoofed traffic on our study of IPv4 address space utilization in Figure 3.3. The two Hilbert curves show observed /24 blocks in our UCSD 2012 dataset: one with the raw data, and the second after removing spoofed traffic. Visually, it appears as though large network blocks (e.g., /8 blocks) with a large number of observed /24 blocks before removing spoofed traffic (Figure 3.3a) also have a large number of observed /24 blocks after removing spoofed traffic (Figure 3.3b). We find a correlation the number of /24 blocks before and after removing spoofed traffic: at the /8 granularity the Pearson correlation coefficient is 0.66[6]. However, there are

[6]The Pearson correlation coefficient varies from -1 (complete negative correlation) to 1 (complete positive correlation), with 0 representing no correlation.

(a) Before. **(b)** After.

Figure 3.3. Influence of spoofing on number of observed /24 blocks. We color /24 blocks observed in the UCSD 2012 dataset before and after applying our algorithm to remove spoofed traffic. The large reduction in observed blocks shows that spoofed traffic removal is a necessary step for network analysis with IBR.

exceptions. For example, as shown in Figure 3.3, with the raw data, we receive traffic from most /24 blocks in both 4.0.0.0/8 and 98.0.0.0/8; but after applying our heuristics 4.0.0.0/8 is almost completely unused, while 98.0.0.0/8 is almost completely used. Therefore, unlike the previous finding by Barford *et al.* [21], we find that spoofing can have a significant impact on the density of observed IP addresses.

The majority of packets are not spoofed (about 3% of packets are spoofed in `UCSD-12`; about 4% in `UCSD-13`). Consequently, we expect spoofing to have less of an impact on studies that require multiple packets from an IP address or /24 block. For example, the path change detection case study requires that a host send packets in consecutive time bins for analysis. For this inference we frequently use scanning traffic, and it is unlikely spoofed traffic will meet the requirement of appearing in consecutive time bins. If in the off chance that spoofed traffic meets the consecutive time bin requirement then TTL, which we use to infer a path change, is an unreliable indicator.

Table 3.3. Validation of our technique to remove spoofed traffic. Our filtering in the darknet datasets dramatically reduces the percentage of /24 blocks originating from darknets and unused blocks of SWITCH. These blocks originally appear as up to 98.9% active; filtering lowers their inferred usage to 0.038% or less.

			Number of /24 blocks (sources)	
Monitored destination	UCSD		MERIT	SWITCH-DARK
		before filtering	54210 (98.4%)	4522 (98.9%)
		after filtering	21 (0.038%)	0 (0%)
	MERIT		UCSD	SWITCH-DARK
		before filtering	57769 (91.5%)	4379 (95.7%)
		after filtering	8 (0.013%)	1 (0.022%)

In general, our technique is appropriate for Internet-wide measurement. Additional small-scale spoofing may exist, but should have a low impact on the number of networks observed. While it is future work to consider the full extent of small-scale spoofing, our technique removes traffic that would greatly skew the analysis of Internet-wide properties we examine in this dissertation.

3.2.6 Validation of our technique

In this section, we validate that our technique removes the majority of spoofed sources. We analyze sources from dark blocks — routed, but known to be unused ranges. Our method was not optimized for these ranges.

We examine the portion of the remaining filtered traffic that had source addresses we have ground truth that they are unused. Specifically, we know this traffic is spoofed because it originates from (i) UCSD and MERIT darknet IP addresses, or (ii) /24 blocks monitored at the border of an academic network, SWITCH, from which we never observed a bidirectional flow, which included 4574 /24 blocks out of the 9343 total /24 blocks monitored at SWITCH during the `UCSD-12` time period (49%). Since spoofing may be more likely to forge nearby addresses[7], we do not report combinations with source and destination addresses in the same darknet (e.g., UCSD-to-UCSD); our final

[7] Many researchers observed local preference in IBR [48, 142]. Additionally, we observe many packets with the same source and destination IP address.

algorithm excludes packets with sources from known darknets. Table 3.3 summarizes this analysis, showing that our filters captured most traffic using source addresses that we know to be spoofed. The substantial reduction suggests the remaining spoofing is low.

We could improve our validation in two ways. First, ground truth labeling of used/unused blocks for additional networks would increase our confidence that we remove the majority of spoofed traffic. The ground truth we have obtained involved analyzing traffic exiting SWITCH; repeating this analysis requires cooperation from additional network administrators. However, since we use different types of networks (completely dark, and unused portions of a live network), we believe our approach works Internet-wide. The second way in which we could improve our validation is with packet capture at a live network. Our technique does not differentiate between a legitimate host sending traffic to the darknet and packets spoofed to have a legitimate host's IP addresses; our analysis assumes that processes generating spoofed traffic forge packets indiscriminately. Obtaining such data is difficult due to privacy concerns.

3.2.7 Conclusion

In this chapter we have argued that removing spoofed traffic is a necessary step of opportunistic network analysis. We have taxonomized previous work in identifying and removing spoofed traffic. For our problem, aggregate methods are most appropriate as we make offline decisions on large collections of data. We have supplied a new method that utilizes IP addresses, instead of TTL or IPID, the primary packet fields used in prior work. Our analysis of the method provides insight into the types of spoofing observed in IBR. Finally, we have validated our technique using real data, which is preferential to the simulation-based analysis provided by other works [27, 104].

Acknowledgements

Section 3.2, in full, is adapted from material as it appears in SIGCOMM Computer Communication Review. Dainotti, Alberto; Benson, Karyn; King, Alistair; Kallitsis, Michael; Glatz, Eduard; Dimitropoulos, Xenofontas; ACM, 2013. The dissertation author was one of the primary investigators and authors of this paper.

Chapter 4

IBR Composition: Phenomena responsible for IBR

IBR is a complex mixture of many Internet phenomena. These phenomena influence our ability to make Internet-wide inferences. Obviously, the success of inference techniques using only a certain component is directly related to the volume of that component in IBR. For example, inferences leveraging Witty worm traffic [112] are no longer applicable. IBR composition also affects inference techniques that aggregate over all IBR (e.g., inferring IPv4 address space utilization in Section 6.1, or path changes in Section 7.1), or a sizable portion of IBR (e.g., we leverage TCP's retransmission behavior to infer packet loss in Section 7.2).

In this chapter we investigate phenomena that induce many sources to send traffic to our darknets. This exercise in analyzing components inspired some of our case studies: an increase in DNS traffic prompted us to investigate the percentage of open resolvers that send IBR (Section 6.2.1); we opted to use machine identifiers present the in large volume of BitTorrent IBR as opposed to machine identifiers from less influential components to study Carrier Grade NAT deployment (Section 6.4.2) and DHCP dynamics (Section 6.4.3). More generally, for each extracted component, we are interested in the long-term usability and the types of well-suited inferences. We evaluate

the long-term usability of a component by investigating why it reaches the darknet. We determine applicable inferences by characterizing the traffic.

We use a multi-step approach to characterize IBR. First, we extracting the following well-known classes of IBR: Conficker, Bro Scanner, Encrypted and Backscatter. Next, we isolate the traffic causing these temporal (Section 5.1.1) and spatial differences (Section 5.3.1). We derive a packet or flow-level filter matching the traffic responsible for each anomaly through manual analysis of the protocols, ports, UDP payloads, packet lengths, TCP flags, and number of packets. We perform this analysis iteratively: once we identify a component, we remove it from our data and find additional components causing abnormalities. Our method may not identify all components of IBR, but we find that the size (in observed /24 blocks) of components causing the abnormality decreased across iterations. Finally, attributing packets to the process that generated them is non-trivial. We discuss our methodology and findings in Appendix A. This process reassigns some traffic from the encrypted class to BitTorrent, Sality, and ZeroAccess.

Table 4.1 shows, using `UCSD-13`, the results of our manual decomposition and attribution of IBR. These results highlight the changing, complex composition of IBR. Scans (Section 4.1) and backscatter (Section 4.2) — the traditional sources of IBR — contribute more than 80% of packets. However, most sources send IBR as the result of misconfigurations and bugs (Section 4.3). These misconfigurations are often caused by complex P2P activity.

4.1 Scanning

Scanning is a preliminary step taken in many network attacks. Unless explicitly blacklisted, darknets capture all Internet-wide scanning efforts. Although scanning methodologies have changed overtime, our darknet continually receives a large volume of scanning traffic. We can leverage this large traffic volume to make inferences requir-

Table 4.1. Discovered IBR Components (`UCSD-13`)**.** Many of the largest IBR components in terms of sources are the result of bugs and misconfigurations; these components do not necessarily contribute the most IBR packets.

Classification	Type	Subtype	/24 blks	(%)	Packets	(%)
Misconfiguration	BitTorrent		2,210k	(70.2%)	5,480M	(5.48%)
		KRPC ping	1,720k	(54.8%)	187M	(0.19%)
		KRPC find_node	1,270k	(40.4%)	23M	(0.02%)
		KRPC get_peers	378k	(12.0%)	18M	(0.02%)
		KRPC announce_peer	2k	(0.07%)	0.02M	(0.00%)
		uTP	1,390k	(44.0%)	4,630M	(4.63%)
		TCP	1,320k	(41.8%)	615M	(0.61%)
		Encrypted	589k	(18.7%)	5M	(0.01%)
Unknown	Encrypted		1,340k	(42.5%)	318M	(0.32%)
		length = 96	546k	(17.3%)	47M	(0.05%)
		length = 256	421k	(13.4%)	6M	(0.01%)
		length = 57	353k	(11.2%)	13M	(0.01%)
		length = 41	353k	(11.2%)	3M	(0.00%)
Bug	Qihoo 360		1,340k	(42.5%)	2,470M	(2.46%)
Misconfiguration	eMule		838k	(26.6%)	1,380M	(1.38%)
		UDP	831k	(26.4%)	1,380M	(1.38%)
		TCP/4662	58k	(1.85%)	3M	(0.00%)
Misconfiguration	Encapsulated IPv6		744k	(23.6%)	485M	(0.48%)
		Teredo	570k	(18.1%)	327M	(0.33%)
		6in4	268k	(8.50%)	159M	(0.16%)
Scan	Conficker		579k	(18.3%)	27,400M	(27.3%)
Backscatter	All		392k	(12.5%)	25,600M	(25.5%)
		ICMP	264k	(8.38%)	1,700M	(1.70%)
		TCP	149k	(4.72%)	17,700M	(17.7%)
		UDP source port 53	4k	(0.14%)	6,160M	(6.15%)
Misconfiguration	Steam		341k	(10.8%)	96M	(0.10%)
Scan	Bro Scanner (nonConficker)		197k	(6.26%)	30,300M	(30.3%)
Misconfiguration	Xbox		172k	(5.45%)	3M	(0.00%)
Misconfiguration	qqlive		156k	(4.96%)	4M	(0.00%)
Misconfiguration	Sality		108k	(3.43%)	3M	(0.00%)
Unknown	udp[12:6]=0x000400000000		92k	(2.92%)	0.1M	(0.00%)
Misconfiguration	ZeroAccess		83k	(2.65%)	36M	(0.04%)
		UDP	16k	(0.51%)	10M	(0.01%)
		TCP/22292	40k	(1.28%)	16M	(0.02%)
		TCP/34354	31k	(1.00%)	9M	(0.01%)
Unknown	udp[9:2]=0xe10b		82k	(2.69%)	3M	(0.00%)
Unknown	len=53; |byte3-byte4|=16		76k	(2.41%)	1M	(0.00%)
Unknown	payload=.flv file name		44k	(1.42%)	1M	(0.00%)
Bug	Mythware		16k	(0.52%)	31M	(0.03%)
Popular Protocols	ICMP Echo Requests		218k	(6.91%)	195M	(2.17%)
Popular Ports	Destination TCP/80		523k	(16.6%)	333M	(0.33%)
Popular Ports	Destination UDP/53		406k	(12.9%)	624M	(0.62%)
Popular Ports	Destination TCP/3389		289k	(9.18%)	1,110M	(1.11%)
Popular Ports	Destination UDP/137		180k	(5.72%)	32M	(0.03%)
Popular Ports	Destination TCP/443		93k	(2.96%)	48M	(0.05%)
Popular Ports	Destination TCP/445		70k	(2.23%)	210M	(0.21%)
Popular Ports	Destination TCP/7111		44k	(1.41%)	1M	(0.00%)
Popular Ports	Destination TCP/3128		37k	(1.16%)	2M	(0.00%)
Popular Ports	Destination TCP/29947		35k	(1.10%)	0.3M	(0.00%)
Popular Ports	Destination TCP/1433		32k	(1.01%)	1M	(0.00%)
Popular Ports	Destination TCP/8080		30k	(0.96%)	3M	(0.00%)
Popular Ports	Destination TCP/25		23k	(0.74%)	6M	(0.01%)
Popular Ports	Destination TCP/389		23k	(0.72%)	8M	(0.01%)
Popular Ports	Destination TCP/110		17k	(0.52%)	1M	(0.00%)
Popular Ports	Destination TCP/139		15k	(0.49%)	6M	(0.01%)
Popular Ports	Destination TCP/53		15k	(0.47%)	1M	(0.00%)
Others			1,460k	(46.3%)	3,180M	(3.18%)
Total			**3,150k**		**100,000M**	

ing multiple observations.

Our analysis focuses on sources meeting Bro's scanner criteria of traffic sent to at least 25 different source IP with the same protocol and destination port.[1] This criterion misses some stealthy scans, such as botnet scan analyzed by Dainotti *et al.* [54]. It is out of the scope of this dissertation to fully identify stealthy scans. However, Table 4.2, which covers the unclassified traffic to the frequently targeted ports (identified in Table 4.1), shows that there is probably low-volume scanning missed by the Bro parameters. First, we remove traffic that is the result of a misconfiguration that induces at least 250 IP addresses to send traffic to a single darknet address. We then run the scan detection with the parameters of targeting at least 5 darknet IP over the entire *2013 census* time frame. This relaxed criterion shows that there are many scanners sending ICMP echo requests, TCP/80, TCP/445, or TCP/3389 packets. These misconfigurations and small scans account for most of the unclassifed activity in `UCSD-13`.

In the remainder of this section, we discuss the applicability of scanning traffic to opportunistic network analysis. In Section 4.1.1 we analyze trends in scanning that suggest that the traffic will continue to provide valuable measurements. While, in Section 4.1.2 we discuss when it is appropriate to use scanning traffic to make Internet-wide inferences.

4.1.1 Properties of scans reaching UCSD-NT

Longitudinally, many sources scan UCSD-NT, and the volume of scanning packets has increased. While the properties of the scans have changed over time, our findings suggest that scanning activity will continue to originate from many hosts. Specifically, there is evidence that scans are split among many machines, and we observe bursts of

[1] In Table 4.1 we separate out Conficker, the largest scanning component in `UCSD-13`. We first identify Conficker sources (sources sending TCP/445 packets to at least four darknet IP addresses in three different /16 blocks, and at least 95% of scans are to Conficker-targeted IP addresses). Then, we run a scan detection on the remaining traffic using the Bro parameters.

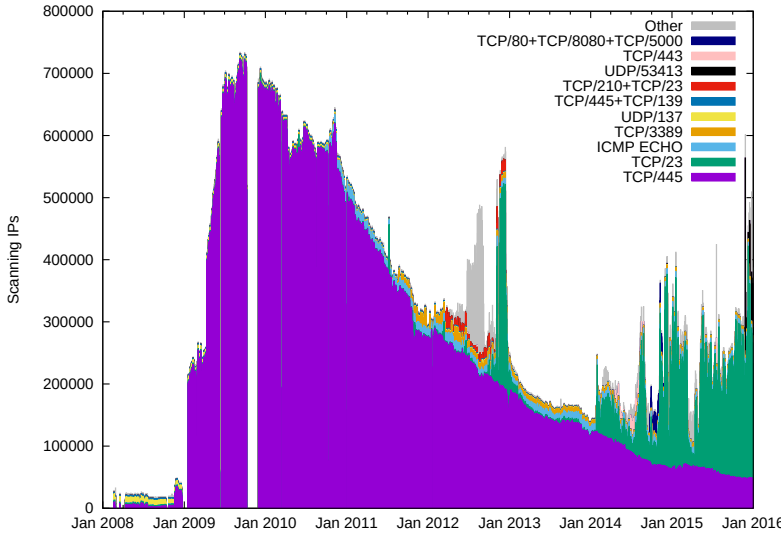
Table 4.2. Scale of misconfigurations and small scans in `UCSD-13`. Of the activity not captured by other filters, a large portion of traffic to the top ports is due to misconfigurations. On some ports (TCP/3389, TCP/80, TCP/445) and with ICMP echo requests, we observe hosts in over 20k /24 blocks conducting small scans — often collectively targeting large portions of the address space.

	Misconfigurations			Small Scans				Other	
	Hotspots	/24 Blocks	Packets	Scanners IPs	Scanners (/24s)	Coverage	Packets	/24 Blocks	Packets
ICMP Echo Requests	116	34k	7.6M	272k	(74k)	96.6%	160M	100k	32M
Destination TCP/80	26,685	450k	260M	170k	(21k)	4.6%	46M	120k	24M
Destination UDP/53	399	390k	570M	3.2k	(2.5k)	0.55%	0.50M	26k	58M
Destination TCP/3389	16,899	120k	120M	330k	(180k)	99%	990M	88k	1.2M
Destination UDP/137	54	140k	22M	11k	(10k)	9.7%	4.9M	32k	5.3M
Destination TCP/443	101	58k	25M	8.1k	(6.3k)	1.6%	12M	38k	11M
Destination TCP/445	22,431	49k	50M	110k	(54k)	32.3%	150M	260k	7.2M
Destination TCP/7111	1	44k	1.0M	8	(8)	-	-	840	0.42M
Destination TCP/3128	26	34k	1.1M	250	(229)	0.57%	0.37M	3.3k	0.28M
Destination TCP/29947	12	34k	0.34M	11	(9)	-	-	630	-
Destination TCP/1433	28	3.0k1	0.32M	530	(390)	0.85%	0.39M	3.8k	0.06M
Destination TCP/8080	17	24k	0.64M	770	(537)	1.9%	1.0M	7.9k	1.5M
Destination TCP/25	26	13k	0.96M	700	(580)	0.44%	1.2M	12k	3.7M
Destination TCP/389	26	6.2k	6.5M	47	(32)	0.09%	0.068M	660	1.4M
Destination TCP/110	13	9.1k	0.40M	8.1k	(4.3k)	1.8%	0.91M	1.0k	0.15M
Destination TCP/139	1	600	1.1M	6.7k	(5.3k)	4.3%	3.2M	9.9k	1.6M
Destination TCP/53	3	14k	0.44M	59	(39)	0.09%	0.51M	1.0k	0.07M

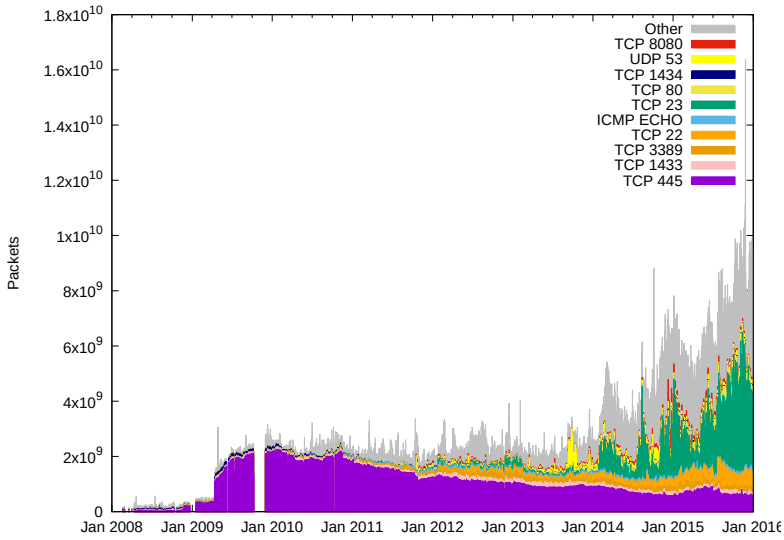
activity after vulnerability announcements.

As shown in Figure 4.1a, the number of scanning IP addresses per day drastically increased in late 2008/early 2009 due to activity on TCP/445 (due to Conficker). The number of scanning IP addresses generally decreased from 2010 to 2014, with two large increases in sources due to traffic on other ports (many different ports due to the Carna botnet) and TCP/23. More recently, there is significant activity on TCP/23 — bringing the number of scanners closer to the number observed in 2010.

As shown in Figure 4.1b, the aggregate number of packets originating from scanners has increased over time. Many previous studies reported that TCP/445 was the most commonly targeted port in IBR [217, 62], but this is no longer true due to an increase in TCP/23 traffic. Interestingly, many of the top ports in terms of packets differs from the top ports in terms of IP addresses. One possible explanation is that distributed ef-



(a) Number of Bro scanners per day.



(b) Number of packets from Bro scanners per day.

Figure 4.1. Frequently scanned ports. From 2009 until 2014, TCP/445, primarily due to Conficker, dominated the scanning traffic reaching UCSD-NT. However, more recently there has been an increase in TCP/23 in terms of both sources and packets; scanning packets destined to non-top 10 ports have also increased.

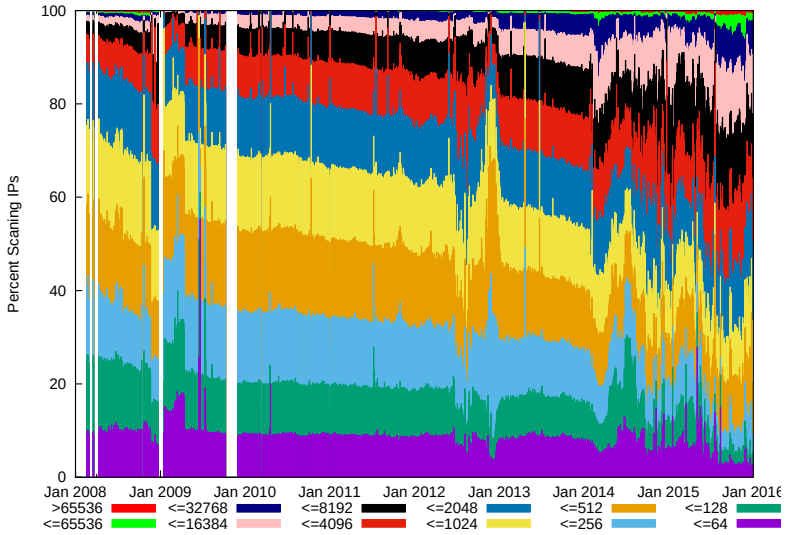


Figure 4.2. Distribution of scanning IP addresses by number of scanning packets sent to UCSD-NT per day.

forts (scans using many IP addresses) target different ports than individual actors (large scanning events originating from a few addresses).

In addition to the scale of scanning packets, the behavior of individual scanners is relevant to our evaluation of the traffic's utility. We examine the number of packets sent by scanners, as it provides insight into how many measurements we can extract from a source. We also hypothesize about the processes generating scans, as this can establish expectations for future scans.

Number of packets per scanner

Our visibility into a source depends on the volume of traffic from that source. We plot the distribution of number of scanning packets from each scanner, grouped by powers of 2, in Figure 4.2. This figure compares the fraction of scanners that send few packets versus those that send many packets.

The recent release of fast scanning software [63, 83] has facilitated the quick discovery of vulnerable hosts with a single machine. One concern for opportunisitic

measurements is that this type of software will eliminate the need to distribute the scan among many machines. This transformation would result in excellent visibility into a handful of hosts instead of good visibility of many hosts (e.g., from a botnet).

Fortunately, we still observe many smaller scans. Despite a trend towards larger scans, half of all scanning IP addresses (meeting the Bro detection criterion) send UCSD-NT less than 2k packets per day. In other words, half of all scanners are sending packets to UCSD-NT at a rate of about 1.5 packets per minute or less. Collectively, these smaller scanners account for a non-trivial portion of scanning packets (not shown in Figure 4.2). In January 2016, about 5% of scanning packets originated from sources sending less than 2k packets per day; and about 40% of scanning packets originated from scanners that scanned the equivalent of a /16 block or more per day. Note, relaxing the scan criteria to include stealthier scans will only increase the influence of small scans.

Processes generating scans

A future direction in IBR research is to fully understand the mechanisms by which the darknet is scanned. This type of analysis may shed insight into the scale and number of processes generating scans. Inference methods are more robust when multiple, diverse process transmit applicable traffic, e.g., we hope to not lose our ability to make certain types of inferences because one entity stops scanning.

In this section, we provide some evidence that there are distributed efforts to scan the IPv4 address space. Additionally, we identify cases where scanning appears to be in response to the discovery of a vulnerability.

Is scanning distributed or conducted by individual actors? While the techniques for coordinated scan detection [75, 120, 33, 54] could be applied to IBR, it is out of the scope of this dissertation to determine the full extent of coordination among scans. However, it is worth providing some preliminary evidence of distributed scanning efforts. In

particular, recent work suggests that scanning botnets have been replaced by scanning from bullet-proof hosting providers [62]. While this appears to be true for the largest scans, it does not seem to hold for smaller scans.

Durumeric *et al.*'s analysis suggests that scanning is primarily the result of individual actors [62]. Based on January 2014 data, Durumeric *et al.* found that 68% of non-Conficker probes (unique combinations of source IP address, destination IP address, and destination port) are the result of scans each targeting 10% of the address space or more (i.e., not distributed botnets). However, the analysis required that sources send a significant number of packets (100) to MERIT's darknet at a moderate scan rate (interpolated to 10 packets per second Internet-wide). These thresholds miss many scans that we consider with the Bro criteria. Additionally, both the Bro criterion and Durumeric *et al.*'s criterion miss a known stealthy scan by a botnet [54], when applied to the appropriate time frame.

We developed heuristics for determining a host's scanning technique, which we detail in Appendix B. Specifically, we develop heuristics to identify the following techniques:

- Originating from Conficker (Conficker)
- Originating from the Carna botnet (Carna)
- Random scanning of the entire darknet (Random)
- Selecting /24 blocks and randomly scanning most addresses (Random /24)
- Selecting /16 blocks and randomly scanning most addresses (Random /16)
- Selecting the last byte of the address and randomly scanning addresses with that byte (Random .X)

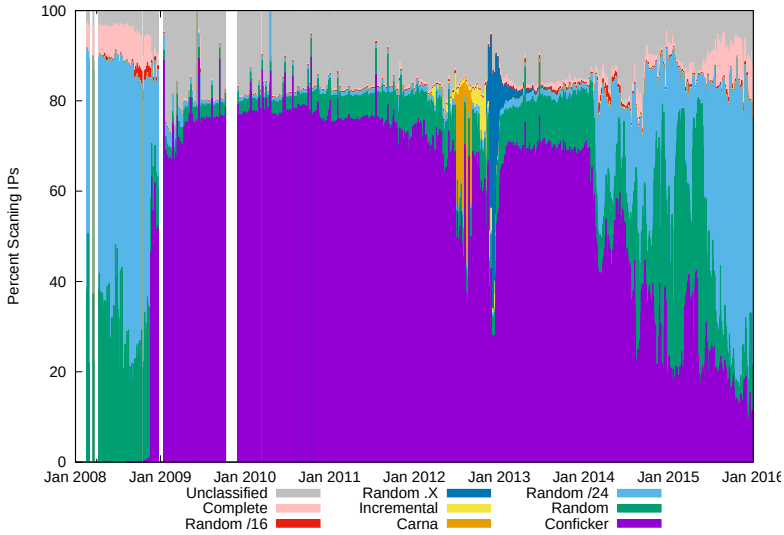


Figure 4.3. Scanning strategy heuristics

- Scanning most addresses in a contiguous block (Complete)
- Incrementing a counter and scanning the corresponding address (Incremental), though increments of 1 are classified as Complete.

We assume that scanners participating in a distributed scan will use the same technique, during the same period. In 2012 and early 2013, we see evidence of a series of long-lasting distributed scans: Carna, Incremental (scanning TCP/210 and TCP/23), and Random .X (scanning TCP/23). Like the Conficker worm, these campaigns account for a large fraction of IP addresses scanning UCSD-NT. The recent activity on TCP/23 suggests that there are multiple entities scanning this port: one (or more) randomly scanning the entire address space, and another (or more) using the Random /24 strategy. To fully investigate coordinated scans, we need to consider the collective coverage and overlap provided by the scanners.

Consequently, our data provides preliminary evidence that distributed scans are present in IBR. These distributed scans are useful for making Internet-wide inferences because many sources send packets at the same time. It is possible that relaxing the

parameters for scan identification (e.g., to include stealthier scans like [54]), will reveal additional distributed scans. However, with fewer packets, these scans would provide less visibility into the sources generating the traffic.

Does scanning increase in response to vulnerability discoveries? In Figure 4.1a, we notice periods of time of increased activity on TCP/443, TCP/5000 and web ports, and UDP/53413. Each of these ports likely became popular due to the discovery of a vulnerability, including: Heartbleed [199], the use of UPnP devices in DDoS attacks [7], and a backdoor in Netis routers [220]. We expect hackers and researchers to continue to discover new vulnerabilities, and that UCSD-NT will be scanned in response. These scans may provide a period of increased visibility into the networks hosting the scanners.

4.1.2 Appropriate inferences with scanning traffic

Scanning is the largest contributor of packets in `UCSD-13`. The Conficker worm, whose outbreak occurred in 2008, still accounts for 27.3% of all packets in `UCSD-13`. Consequently, scanning is useful for inferences that require a large number of packets. We use the large number of scanning packets in our uptime (Section 6.3.1) and NAT (Section 6.4.1) case studies.

Internet-wide scans can take a longtime to complete (e.g., even Zmap takes about 45 minutes [63]). Hosts participating in scans will repeatedly send packets to UCSD-NT. As a result, scanning is useful for inferences that require repeated observations. We use repeated measurements from scanning in our path change (Section 7.1) and packet loss (Section 7.2) case studies.

Scanning is fairly predictable. We know the number of packets sent to each address, and may be able to infer the next host to be scanned. We leverage the predictability of scanners in our packet loss (Section 7.2) case studies. Moreover, knowing

the exact order in which darknet IP addresses are scanned could be leveraged for outage detection; using similar information, Kumar *et al.* determined the number of disks on Witty-infected machines [112].

4.2 Backscatter

Normally, we think of backscatter from spoofed DoS attacks as coming from a small number of attacked machines or networks. In this section, we show that the number of sources sending backscatter can actually be large, and therefore suitable for Internet-wide measurement. The Spamhaus attack [162] targeted Spamhaus' network, the networks carrying Spamhaus' traffic and strategically selected Internet exchange points. An increase in DNS traffic is caused by responses to spoofed queries — from many open resolvers simultaneously.

4.2.1 Spamhaus attack

In the basic spoofed DoS attack, the attacker construct a packet with a randomly spoofed source IP address (possibly in our darknets) and sends the packet to the victim. However, it is more effective to conduct a reflective DoS attack, that makes use of an amplifier. In a reflective DoS attack, the attacker sends a forged packet with the victim's source address to the amplifier; the amplifier responds by sending a larger packet or multiple packets to the victim. Notably, in the reflective DoS attack, the darknet does not receive any traffic.

In March 2013, Spamhaus, a provider of anti-spam filters, experienced one of the largest known DoS attacks [162]. Although the attack was most effective due to their usage of DNS amplifiers (not visible in IBR), the attackers used additional DoS methods (visible in IBR). Once under attack, Spamhaus hired CloudFlare to distribute Spamhaus' content using anycast routing. In response to this mitigation, the attackers

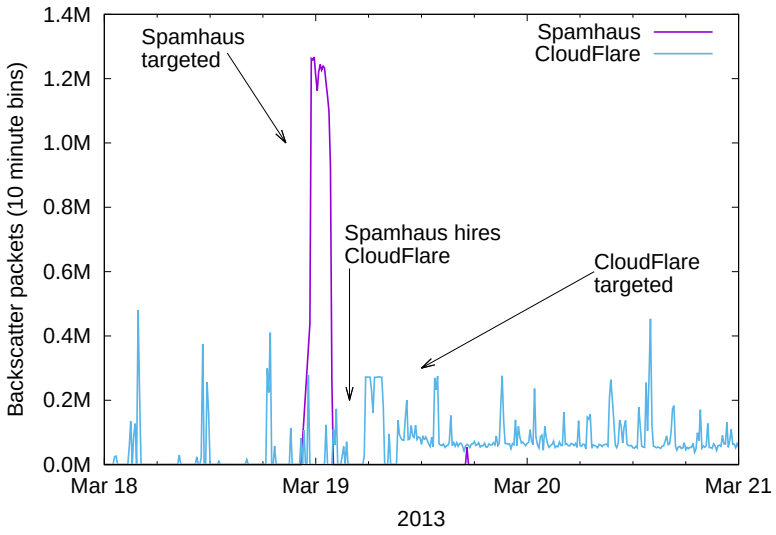


Figure 4.4. Sources from select ASes during Spamhaus attack. One of the largest DoS known attacks, targeted Spamhaus. Spamhaus used CloudFlare's anycast routing to mitigate the attack. In response, the attackers targeted CloudFlare, CloudFlare's peers, and Internet exchange points. The switch in the target is visible with IBR.

switched their target, not only to CloudFlare, but CloudFlare's peers, and Internet exchange points. As shown in Figure 4.4, this change in target is visible in the sustained volume of backscatter packets reaching UCSD-NT from hosts in both Spamhaus and CloudFlare's ASes.

Although, there are many services that attackers can use as amplifiers [177], we still observe traditional spoofed DoS attacks in IBR. The Spamhaus attack, which used both traditional and amplification techniques, is relevant to Internet-wide inferences with IBR because of the temporary increase in traffic. Since the attack targeted many ASes, we received traffic from sources all over the Internet.

4.2.2 DNS backscatter

If a darknet receives a DNS response, its most likely cause is a DNS server responding to a spoofed query. For the purpose of this analysis, we label an IP address as an *open resolver* if the Recursion-Available flag is set in an UDP source port 53 packet

arriving from that source, as it indicates the resolver's willingness to resolve recursive queries.[2]

Starting around February 2014, we observe a sustained increase in DNS responses. we investigate why we observe this sustained DNS traffic using a dataset, `UCSD-14-DNS`, containing all DNS packets received between January 20, 2014 and March 1, 2014. Our analysis suggests that this traffic is from DoS attacks on authoritative name servers, i.e., the traffic is backscatter from an attack that inhibits the DNS lookup of the domains served by an authoritative name server. Van Nice reports on this type of attack from the perspective of Nominum, a DNS analytics company [208].

Traditional amplification attacks using open resolvers do not result in IBR; consequently, studying the significance of open resolvers is difficult unless you are involved in attacks (i.e., under attack or running an open resolver). However, this specific type of attack causes open resolvers to repeatedly send packets to our darknets. We compare the number open resolvers visible in IBR to an active scanning technique (which finds open resolvers not used in this attack) in Section 6.2.1. More generally, since this is an ongoing attack that leverages the same open resolvers, this traffic is useful for inferences requring multiple packets using the UDP and IP layers.

Attack specifics

It is common for DoS attacks use open resolvers to amplify their effectiveness. Since the response to a DNS query is typically larger than the query itself, the attacker queries the open resolver with a packet spoofed with the victim's IP address. Darknets do not receive packets in the common attack. Consequently, we want to understand why this new attack causes many open resolvers to send packets to darknets.

[2]The IP address may be an open resolver or one that recursively resolves domains on behalf of a forwarding open DNS server [181]. In other words, the machine we receive a response from may not be the one that initially received the spoofed packet.

DNS responses contain the query which they answer (or fail to answer). As a result, we can examine the domains that were queried to induce the backscatter packets. If sources in at least 50k /24 blocks send traffic for the same second-level domain to the darknet, we consider it part of an "attack." 462 second-level domains met this criteria. These domains account for most of the DNS behavior in `UCSD-14-DNS`: less than 1% of source addresses were not associated with any of these second-level domains. We find suspicious behaviors associated with the second-level domains:

- baidu.com was the first second-level domain used in the attack, six days prior the second domain reaching the 50k threshold. This was likely a testing phase.
- Besides baidu.com, the other observed domains likely exist for the purpose of these attacks. According to WHOIS data, 60% of domains were created less than 6 months prior to the scans.
- Unsurprisingly, the registration contact information is often obviously fake (e.g., a phone number of 11111111 and street address of hkjhkjhjkhjk).
- The queried domains are registered through a variety of registrars, most commonly: GoDaddy (74 domains), eName (70), eNom (40), HiChina Zhicheng (39). This may indicate that attackers have multiple resources for obtaining domain names.
- The 462 second-level domains often share name servers, most commonly: *.dnspod.net (56 domains), *.dnsabc-[b|d|f|g].com (38), *.iidns.com (37), *.hichina.com (25), *.zndns.com (24).

In `UCSD-14-DNS`, many open resolvers sent UCSD-NT packets with errors (RCODE$\neq$0). This was not the case for previous collections of IBR or in the responses of weekly scans of the entire IPv4 address space for open resolvers by the Open Resolver

Table 4.3. Comparison of number of observed open DNS resolvers across datasets. DNS responses reaching the darknet with the Recursion-Available bit set indicate an open resolver. The number of open resolvers sending IBR increased in 2014, allowing us to infer their existence and provide insight into an attack on authoritative name servers.

	UCSD-12	UCSD-13	MERIT-13	UCSD-14-DNS	Open Resolver Project [152]
Unique IPs	49,111	3,401	834	1,561,324	37,607,402
Recursion-Avail.	42,312	2,298	591	1,518,360	32,917,724
RCODE=0 (OK)	48,746	2,991	329	1,437,310	32,595,867
RCODE=1 (FORMERR)	43	7	5	1,422	841
RCODE=2 (SERVFAIL)	317	148	41	1,445,276	919,899
RCODE=3 (NXDOMAIN)	215	200	518	1,349,092	153,466
RCODE=4 (NOTIMP)	7	8	1	64	166
RCODE=5 (REFUSED)	173	241	34	136,328	4,433,126

Project [152] conducted during the same time period. In particular, compared to the Open Resolver Project, UCSD-14-DNS observes more sources with SERVFAIL errors, which indicates that the authoritative name server did not answer the query. Moreover, many open resolvers in UCSD-14-DNS respond with non-errors and errors for queries for the same second-level domain, suggesting that authoritative name servers are inundated with queries — a characteristic of DoS attacks. By contrast, the Open Resolver Project scans the Internet at a rate sustainable by authoritative name servers.

It is unclear clear why the attack uses open resolvers. If an attacker (with the ability to send spoofed packets) wanted to perform a DoS attack or poison the cache of an authoritative name server they could send spoofed queries directly to the authoritative name server. A likely explanation is that using open resolvers reduces the code complexity of the malware launching the attack, as writing code to correctly resolve a domain names is non-trivial [58]. Another hypothesis is that filtering policy may necessitate the use of open resolvers. The networks where the spoofing machines reside may only permit port 53 packets destined to certain DNS servers to leave their networks. Alternatively, these open resolvers may actually be local DNS servers in networks that use egress filtering (i.e., the spoofing machine cannot send spoofed packets to the authoritative name server).

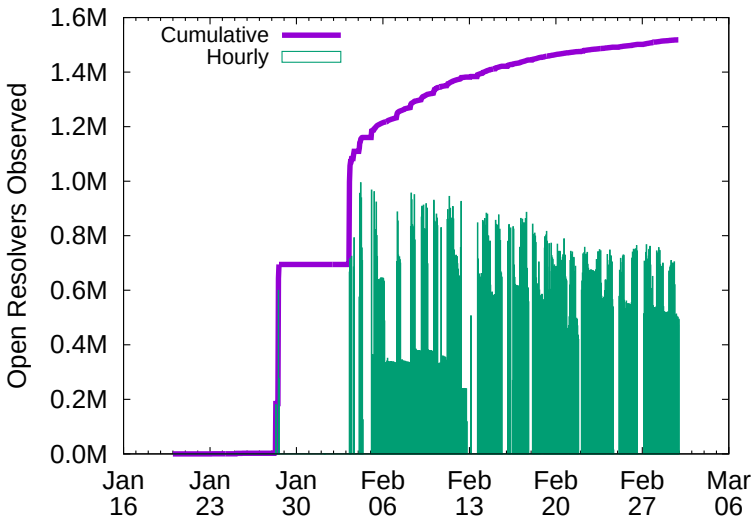


Figure 4.5. DNS backscatter (early 2014). After February 6, 2014, almost every hour, over a quarter of a million open resolvers send DNS packets to UCSD-NT. The slow growth in cumulative open resolvers implies that the same open resolvers repeatedly send this type of backscatter.

Utility of DNS backscatter

Unlike the Spamhaus event, which was a short-lived event, DNS backscatter is an on-going phenomenon. We are still observing this attack as of May 2016. Consequently, we can use this traffic for sustained insight into open resolvers and their networks, with two caveats. First, there are periods of inactivity. Second, we observe the same open resolvers repeatedly as opposed to all open resolvers.

We illustrate both of these caveats in Figure 4.5, which shows the number of open resolvers observed per hour, as well as cumulatively. We notice that the number of open resolvers observed per hour varies, including consecutive hours without traffic. After an initial spike, the cumulative number of open resolvers grows slowly, despite observing over 200k open resolvers in many hour bins. We observe only a fraction known open resolvers: during this time period we capture 1.5M open resolvers through IBR, while the Open Resolver Project collected responses (with the Recursion-Available

flag set) from 32.9M IP addresses through their weekly IPv4 scans [152]. As a result, this behavior indicates reuse, i.e., the phenomenon generating the DNS responses is repeatedly sending spoofed packets to the same set of open resolvers.

4.2.3 Appropriate inferences with backscatter traffic

Backscatter events provide a period of increased visibility of remote networks, and it may be advantageous to infer network properties during these events. The increase in traffic during the Spamhaus attack leads to better coverage in our IPv4 address space utilization study (Section 6.1). The increase in DNS traffic inspired us to analyze open resolvers when assessing the ability of IBR to provide insight into server architecture (Section 6.2.1). This window of opportunity may vary: the Spamhaus attack lasted a couple days, while DNS backscatter is an on-going phenomenon.

4.3 Bugs and Misconfigurations

The ability to leverage traffic resulting from bugs and misconfigurations depends on the popularity of the affected software and the nature of the bug. In this section, we discuss a bug in Qihoo 360, and misconfigurations in BitTorrent that affected millions of hosts. Both Qihoo 360 and BitTorrent are popular software, but their use of P2P communications exacerbates the scale of the traffic reaching the darknet. While these bugs and misconfigurations do not result in a high-volume stream of packets from a host (e.g., like scanning or backscatter), the traffic originates from many hosts and persists for a long period of time. Consequently, this traffic provides excellent, though potentially biased, coverage for inference possible with few packets.

```
6:00:00.083796 IP 123.4.253.107.8090 > XX.179.58.115.42501: UDP, length 30
        0x0000:  4500 003a df4b 0000 2e11 ---- 7b04 fd6b  E..:.K......{..k
        0x0010:  XXb3 3a73 1f9a a605 0026 c0cf 0000 0000  ..:s.....&......
        0x0020:  0000 0000 3100 3d57 0000 0000 0000 0000  ....1.=W........
        0x0030:  0000 0000 287e 02c7 0000
```

Figure 4.6. Example Qihoo 360 packet. Some bytes are fixed, while others appear to increment or be set based on the connection.

4.3.1 Qihoo 360

Figure 4.6 shows a packet originating from over 100M IP addresses in UCSD-13. By capturing two-way traffic at a live-network, we determined that this traffic was the result in a bug in Chinese security software, Qihoo 360. Since this bug persisted for over five years, the resulting traffic seems useful for longitudinal analysis of many hosts. However, this traffic has a Chinese bias, and has been nearly eliminated upon reporting the bug to Qihoo.

Bug specifics

Due to the widespread use of this protocol, we could coordinate with UCSD CSE researchers monitoring live networks to capture additional traffic from some IP addresses and ports sending the payload. With bidirectional traffic, we find that a byte order bug causes the hosts to contact the darknet. As shown in Figure 4.7, external IP address 113.70.40.122 sent traffic to a UCSD CSE host. The UCSD CSE host then responded to 122.40.70.113. Therefore, our /8 darknet (X.0.0.0/8) receives packets whenever a host responds to a legitimate IP address whose last byte is X (e.g., 1.2.3.X).

The use of P2P technology to update the software [4] explains the magnitude of this traffic. After contacting an address in the 360.cn domain, a host receives a packet with a list of peers. The host then contacts all of the peers to attempt to download the update. Consequently, each host updating triggers many connections and potentially many packets written in reverse byte order.

```
04:40:46.877858 IP 113.70.40.122.5437 > CS.239.95.102.10102: UDP, length 30
        0x0000:  4500 003a 6213 0000 2f11 ---- 7146 287a  E..:b.../...qF(z
        0x0010:  CSef 5f66 153d 2776 0026 8a67 0000 0000  ..._f.='v.&.g....
        0x0020:  a800 0d13 2100 55e1 0149 f488 0134 9733  ....!.U..I...4.3
        0x0030:  0038 0000 0005 0006 0000
04:40:46.878016 IP CS.239.95.102.10102 > 122.40.70.113.15637: UDP, length 30
        0x0000:  4500 003a 552d 0000 3f11 ---- CSef 5f66  E..:U-..?....._f
        0x0010:  7a28 4671 2776 3d15 0026 2c6b 0000 0000  z(Fq'v=..&,k....
        0x0020:  0000 0000 3100 55e1 0000 0000 0000 0000  ....1.U.........
        0x0030:  0000 0000 42d6 0005 0000                 ....B.....
```

Figure 4.7. Example of Qihoo 360 byte-order bug (captured in a live network: UCSD CSE) The UCSD CSE machine (CS.239.95.102) receives a packet from IP address 113.70.40.122 but due to a byte order bug responds to 122.40.70.113.

Table 4.4. Country of origin for Qihoo 360 traffic. We show the top 10 countries in terms of IP addresses in UCSD-13. Many IP addresses in China and nearby countries are associated with Qihoo 360.

	IPs	% BGP Announced Address Space
China	101,240k	36.26%
Taiwan	505k	1.45%
Malaysia	442k	7.65%
USA	324k	0.03%
Hong Kong	280k	2.75%
Japan	186k	0.11%
Canada	129k	0.26%
Thailand	126k	1.55%
Australia	126k	0.31%
Singapore	116k	2.16%

Usability of traffic originating from Qihoo 360's byte order bug

Two aspects of Qihoo 360 IBR greatly influence the usability of the traffic: the large number of Chinese sources, and the nature of repeated contact from individual clients. According to their website, Qihoo 360 is the top provider of Internet and mobile security products in China [164]. IBR confirms Qihoo's popularity in China. The breakdown of IP addresses for the top countries in UCSD-13 is shown in Table 4.4. We also report the magnitude of the IP addresses as a percentage of BGP space announced belonging to the country. Nearly a third of all Chinese IP addresses announced in BGP sent a packet matching our signature. Consequently, this traffic has a significant bias towards Chinese hosts.

In Section 5.2.2 we examine repeated contact of sources sending IBR. Qihoo

360 is unique in that, generally, there are few packets associated with each IP address sending Qihoo 360 traffic but a long time between observations. This is primarily due to the extreme diurnal patterns: in UCSD-13, the average number of source IP addresses sending this traffic per hour varies between 165k at 20:00 UTC to 2.31M at 0:00 UTC. Unfortunately, we do not observe individual IP addresses sending Qihoo 360 traffic at predictable intervals. Consequently, this traffic is primarily useful in aggregate, or when for individual machines we do not need many packets or high predictability.

We contacted Qihoo to report the byte order bug. They confirmed the issue, and planned to deploy fixes around January 12, 2016. About a month later we observed a substantial decrease in the number of packets reaching the darknet matching our Qihoo 360 filter. Other than our analysis of the time to fix the Qihoo 360 bug (Section 6.3.3), the Internet-wide inferences made in this dissertation do not appear to be dependent on the existence of the Qihoo 360 bug. However, this bug fix illustrates how a small change can drastically change the composition of IBR.

4.3.2 BitTorrent

BitTorrent traffic accounts for a sizable portion of IBR. Hosts in over 2M /24 blocks sent over 8 billion BitTorrent packets in UCSD-12 (we observe similar statistics in UCSD-13). While voluminous, this traffic may be difficult to leverage for opportunistic inferences due to its unpredictability, which is likely due to the reason BitTorrent traffic reached our darknets. We find evidence that this traffic is, in part, due to index poisoning attacks — the intentional pollution of the Distributed Hash Table (DHT) with erroneous data.

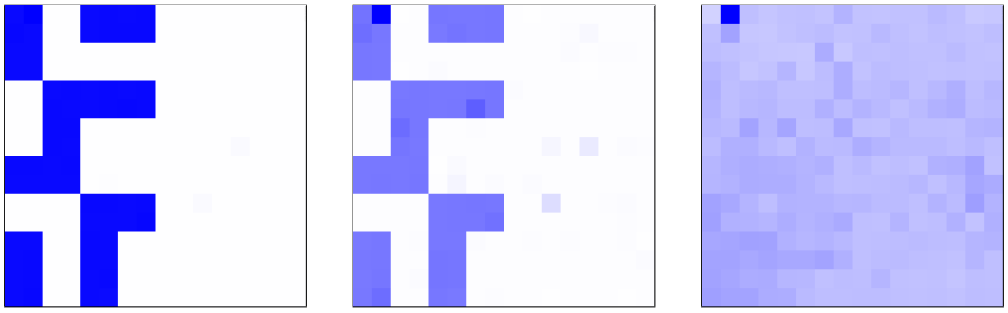
Evidence of index poisoning attacks

Since darknets contain no active hosts, all BitTorrent traffic in IBR is the result of implementation bugs or incorrect information in the DHT. It is difficult to conclusively determine if BitTorrent packets reach our darknet intentionally or unintentionally, but we provide evidence that there is systematic falsification of data (and not a bug).

The purposeful pollution of the DHT with false locations is called an index poisoning attack [121]. The goal of this attack is to thwart a user's ability to download torrents. In this attack, malicious BitTorrent clients share incorrect torrent locations (i.e, the IP address and port of a peer), or falsify information about other BitTorrent clients, claiming they are likely to know the torrent's location. In particular, the malicious BitTorrent clients may advertise that a darknet IP address has information relevant to downloading the torrent. Incorrect information will cause a legitimate client to send excessive packets and lengthen the time to find a target. In theory, if a legitimate client receives a large amount of false information, it may be impossible to actually obtain the desired torrent.

In this section, we look for evidence that BitTorrent clients are purposefully directed to incorrect locations, including our darknets. In our 2012 and 2013 datasets, the darknet IP address receiving traffic appear to be generated randomly (albeit with a poor PRNG). Additionally, in this data, we observe a large volume of packets requesting torrents whose name includes the word "China." Starting in 2015, traffic to one darknet IP address resulted in a ten-fold increase in sources sending BitTorrent IBR per minute. This traffic also has peculiarities that suggest that the false information is generated programmatically but multiple hosts.

Evidence of random address selection for index poisoning attacks As described in Section 5.3.1, BitTorrent traffic collected in 2012 and 2013 has a preference for certain



(a) get_peers (2.1M /24 blocks) **(b)** find_node (1.7M /24 blocks) **(c)** ping (1.3M /24 blocks)

Figure 4.8. Targeting patterns by BitTorrent packet type in `UCSD-12`. We shade, on a linear scale, each /16 in the Hilbert curve of UCSD-NT's address space. White corresponds to zero /24 blocks and dark blue corresponds to the most /24 blocks. Sources sending get_peers and find_node packets exhibit a preference for certin /13 blocks.

/13 blocks. Specifically, darknet IP addresses (X.B.C.D) satisfying: B & 0x88 = 0x00 and D & 0x09 = 0x01 are likely to receive BitTorrent traffic from many sources. Additionally, there is an attraction to destination ports satisfying: destination port & 0x2081 = 0x0080. Figure 4.8 shows the preferential targeting on the /16 granularity. The visible preferential targeting in get_peers and find_node packets is consistent with obtaining erroneous information when traversing the DHT (as is the case with get_peers and find_node packets), and not when checking if known hosts are still alive (with ping packets). We observe the same preferential targeting in `UCSD-12`, `UCSD-13`, and `MERIT-13`; though, the targeting is more pronounced in `UCSD-12` than the 2013 datasets.

Our data suggests that the addresses and ports are selected uniformly from the targeted ranges, as each address and port receives traffic from approximately the same number of sources. For get_peers packets in `UCSD-12`, 99.9% of targeted IP addresses receive traffic from between 35 and 85 /24 blocks; 99.6% of targeted ports receive traffic from between 6,700 and 7,400 /24 blocks. A plausible explanation for the spatial abnormalities is a bug in a pseudo-random number generator (PRNG) that determines which addresses and ports to target.

Table 4.5. Top 10 infohashes (by number of packets) in UCSD-12. Four of the top ten most requested infohases have "China" in their name, suggesting that certain content is more likely to be targeted in this index poisoning attack.

Infohash	Torrent	Packets	/24 blocks
48484fab5754055fc530fcb5de5564651c4ef28f	Grand Theft Auto - *China*town Wars	450k	32k
5b5e1ffa9390fff13f4af2aef9f5861c4fbf46eb	Modern Family S3E22	398k	30k
d90c1110a5812d9a4bf3c28e279653a5c4f78dd1	CSI S12E22	204k	6.5k
2ecce214e48feca39e32bb50dfcf8151c1b166cc	Coldplay Ft. Rhianna Princess of *China*	187k	18k
79f771ec436f09982fc345015fa1c1d0d8c38b48	-	129k	53k
b9be9fc1db584145407422b0907d6a09b734a206	Parks and Recreation S4E22	127k	7.5k
99a837efde41d35c283e2d9d7e0a1d4a7cd996dd	Missing 2012 S1E9	106k	6.6k
7b05b6b6db6c66e7bb8fa5aa70a185c7cfcd3d07	-	104k	44k
c0841cf3196a83d1d08ae4a9eaf10fcfc6c7ba66	Big Trouble Little *China*	99k	6.1k
99dfae74641d0ca29ef523860713a6270daefc6e	36 *China* Town	91k	1.5k

Evidence of targeted content in index poisoning attacks Get_peers packets include a hash of the torrent. We analyze this traffic to determine frequently requested content for the 2012 and 2013 data. We then search the web for these hashes to determine the content. As shown in Table 4.5, hashes for content with "China" in the name are among the most popular in terms of number of packets in UCSD-12. Not shown in the table is a similar observation in UCSD-13; for example, "Sette Anni in Tibet" is the most requested torrent in terms of packets. This behavior suggests that certain content is targeted for index poisoning.

Evidence of a distributed poisoning effort Starting in July 2015 the number of IP addresses sending BitTorrent traffic increased from about 20k per minute to 200k per minute. This ten-fold increase, which has lasted for almost one year, is due to one darknet IP address, UCSD.235.104.34, on port 37547. Sustained traffic to a single darknet IP address is unexpected as we expect well-behaving clients to hinder the propagation of erroneous information.

We examine the BitTorrent clients and requested content to look for evidence of bugs, misconfigurations or index poisoning. We find a variety of clients. Unlike the 2012 and 2013 data, many web searches for the top infohashes did not return common torrent names. These findings suggest that there is neither a bug in a particular client

nor a particular type of content that induces a host to send BitTorrent IBR.

To further investigate why BitTorrent clients send IBR to UCSD.235.104.34 we installed two versions of BitTorrent on machine a machine in live network: uTorrent and a LibTorrent client (Deluged). Without requesting any BitTorrent content, we passively monitored all KRPC packets (uTorrent for 2.5 months, Deluged for 2 months). Both versions contacted UCSD.235.104.34 multiple times (uTorrent 112 times, Deluged 63 times).

Interestingly, there are patterns in the responses from the 54 nodes we analyzed. Almost all of these nodes associate UCSD.235.104.34 with different IDs. All but six nodes map UCSD.235.104.34 to different IDs, but the third byte is always `0x04`. Generally, responses to find_node messages include information about multiple nodes. The nodes that appear with UCSD.235.104.34 are often the same. For example, we received 125 packets from 27 nodes indicating that 212.246.161.63 is a BitTorrent node where the third byte of the ID is `0x06`. The patterns in the responses suggest a common process generated the fake IDs.

There are no obvious commonalities in the 54 nodes that included the darknet IP address in their responses (uTorrent 12 nodes, Deluged 42 nodes). Although the hosts are primarily in China, one address geolocates to the United States and another to Russia. The implies that the process generating the bogus information may be geographically distributed. The 54 nodes are in 18 different ASes. Most of the nodes (36 nodes) use LibTorrent, with version 1.00 being the most popular (33 nodes). However, we also observe uTorrent, A0, and packets without client information (i.e., non-LibTorrent based client). Since most of these clients are not buggy (e.g., our uTorrent client never propogates the UCSD address as a peer), it is unlikely that a software bug is responsible for this traffic.

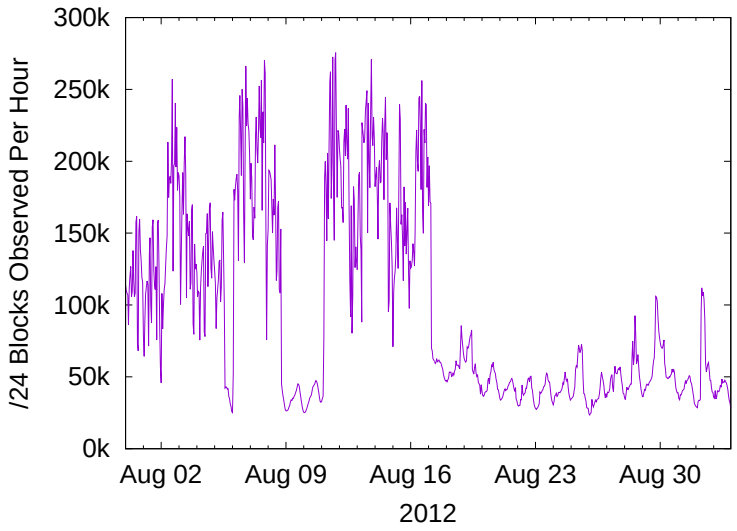


Figure 4.9. /24 blocks sending BitTorrent traffic per hour (`UCSD-12`). BitTorrent traffic is not well-suited to inferences requiring predictability due to its erratic source behavior.

Utility of BitTorrent traffic

Using BitTorrent traffic for opportunistic inferences is difficult due to its bursty temporal behavior. As shown in Figure 4.9, the aggregate number of hosts sending BitTorrent packets to the UCSD darknet is sustained but erratic over time. The popularity of downloaded content or occasional intensification of the attacks likely causes the impulses. The bursty behavior extends to individual BitTorrent clients. In most cases, an individual darknet IP address/port receives KRPC traffic in a single short burst. This behavior is consistent with (1) determining that the darknet address is an invalid node and (2) not propagating or storing the bad mapping. Consequently, this traffic is not well-suited for inferences requiring predictable traffic.

However, the wide-spread use of BitTorrent makes this traffic extremely useful in many of our inferences that do not require predictable behaviors. In particular, the contents of BitTorrent packets contain machine identifiers. With these identifiers we can associate a machine (and its user) with an IP addresses. We leverage this association

to study DHCP dynamics and CGN deployments. Additionally, this association could potentially reveal privacy sensitive information [117].

4.3.3 Appropriate inferences with traffic resulting from bugs or misconfigurations

Many sources send IBR to our darknets as a result of P2P misconfigurations and bugs. We can leverage these sources in inferences requiring many sources. In our study of IPv4 address space utilization in 2013 (Section 6.1), BitTorrent and Qihoo 360 are two of the top three sources of /24 blocks. New bugs or misconfigurations will likely improve our coverage. For example, traffic to UCSD.235.104.34 resulted in a ten-fold increase in sources sending BitTorrent traffic per minute.

If the bugs and processes generating misconfigurations persist, we can receive a continuous stream of traffic. For example, for more than five years, Qihoo 360 software sent IBR as a result of a byte-order bug. However, our ability to extract predictable subcomponents from the continuous stream may be limited. Per host, both Qihoo 360 and BitTorrent traffic are low-volume and unpredictable. This is not a limitation for some inferences. For example, we leverage BitTorrent identifiers to study DHCP leases (Section 6.4.3), and Carrier Grade NAT usage (Section 6.4.2).

One downside of leveraging misconfigurations or bugs is bias towards users of the responsible software. For example, we are unable to accurately assess BitTorrent client popularity due to uncertainty of the biases associated with BitTorrent traffic (Section 6.3.2). This bias is less of an issue when measuring network properties (e.g., DHCP deployment, CGN usage).

4.4 Conclusion

The composition of IBR clearly influences the types of traffic available for researchers to leverage. This chapter has shown that the sending patterns of the various types of IBR make them well-suited for different classes of inferences. Moreover, understanding the processes that generate IBR has provided insight into the outlook for long-term studies with IBR.

In terms of the utility of IBR components, we concluded Sections 4.1, 4.2, and 4.3 with discussions about when it is appropriate to use scanning, backscatter, and misconfigurations/bugs for opportunistic Internet-wide inferences. As a summary: scanning traffic is well-suited for inferences that require many packets, bursts of packets, or predictable behaviors; backscatter is well-suited for short-term visibility into a phenomenon; misconfigurations and bugs in P2P networks are well-suited for insight into many hosts, though there may be limited insight into individual hosts. These large classes of traffic are useful for complementary purposes. Consequently, we can leverage IBR for a variety of inference types.

The distinct behavior of the IBR components means that, for some application-agnostic inferences, we may end up using primarily a certain class of traffic. For example, our path-change algorithm (Section 7.1) uses information in the IP header when a source sends packets in consecutive time bins. This requirement on the traffic means that scanning is well-suited for the task, while P2P bugs and misconfigurations are unlikely to assist in inferring path changes.

The long-term outlook for IBR-based inferences is encouraging due to the abundance of phenomena we find generating IBR through our manual decomposition of the traffic (Section 4). The existence of many phenomena implies that inferences using a variety traffic types will likely continue to be applicable, even as the composition of

IBR evolves. Moreover, Internet-wide inferences based on information in the IP and transport layer headers can also leverage unclassified and unknown-origin traffic.

We expect to continue to observe traffic from hosts Internet-wide as a result of varying scanning methods, complex DoS attacks, and large-scale misconfigurations and bugs. Our outlook for each class of traffic is as follows:

- Although scanning is one of the first steps in a network attack, there has been limited research into the evolution of scanning mechanisms. These mechanisms are relevant to opportunistic inferences because they determine the nature of scanning traffic reaching darknets. Scans conducted by individuals on a handful of machines provide excellent visibility into the properties of the scanning hosts and networks. Alternatively, distributed scans (e.g., through a botnet) produce less packets per host but collectively have wider Internet coverage. We find evidence of both types of scans in IBR, and expect scanning traffic to continue to reach our darknet.

- Leveraging amplifiers is an effective technique for conducting reflective DoS attacks. Traditional reflective DoS attacks do not produce IBR; however, we found two examples of complex DoS attacks that resulted in IBR, and increased visibility into Internet-hosts. The Spamhaus attack used a multiple DoS techniques over a few days, while a new type of DoS on authoritative name servers using open resolvers is ongoing.

- In recent years, there has been an increase in number of sources due to bugs and misconfigurations in P2P networks (Qihoo 360 and BitTorrent). As a result, the number of networks we can obtain measurements for is increasing. However, sources sending P2P traffic generally produce few connection attempts at irregular intervals, so fine-grained analysis (repeated analysis on a short time scale, e.g.,

minutes) is difficult for these hosts.

Acknowledgements

Section 4.3.2, in part, is adapted from material as it appears in the proceedings of the Internet Measurement Conference (IMC 2015). Benson, Karyn; Dainotti, Alberto; claffy, kc; Snoeren, Alex C; Kallitsis, Michael; ACM, 2015. The dissertation author was the primary investigator and author of this paper.

Chapter 5

IBR Visibility: Factors influencing network measurability

Evaluating IBR's potential as an Internet-wide opportunistic data source involves two major questions:

1. What information can I extract from IBR?
2. How many sources (IP address, /24 block, prefix, AS, country) send that type of information?

With respect to Question 1, enumerating all types of IBR-derivable information is a daunting, and probably impossible task. In Chapter 4 we characterized the composition of IBR. This characterization can act as a starting point for formulating IBR-based inference techniques, or deciding to forgo using IBR. E.g., calculating web site popularity is an unlikely use case as we do not observe HTTP payloads. However, it is probable that both the evolving composition of IBR and the ingenuity of researchers will both lead to additional use cases for IBR.

Although we are limited in answering Question 1, we can provide some intuition for Question 2. To investigate the number of sources for which we can make inferences, we first consider IBR as a whole and report our overall visibility into networks Internet-wide (Section 5.1). We use the metric *coverage*, which we define as the number of

Table 5.1. Number of observed sources in darknet traffic. The number (and percentage of announced resources) of IP addresses, /24 blocks, prefixes, ASes, and countries observed in each dataset is consistent across sites (UCSD-NT vs. MERIT-NT) and years (2012 vs. 2013).

	Announced		UCSD-12		UCSD-13				MERIT-13	
							Partial			
	2012	2013								
IP addresses	2.61B	2.66B	148M	(5.7%)	133M	(5.0%)	109M	(4.1%)	111M	(4.2%)
/24 blocks	10.2M	10.4M	3.13M	(31%)	3.15M	(30%)	2.65M	(26%)	2.76M	(27%)
Prefixes	410k	452k	198k	(48%)	205k	(45%)	170k	(38%)	175k	(39%)
ASes	44k	46k	24.3k	(55%)	24.2k	(54%)	19.3k	(44%)	19.8k	(45%)
Countries	245	236	234	(96%)	233	(99%)	231	(98%)	232	(98%)

networks where at least one IP address from the network is captured in IBR. We then consider factors that would reduce our coverage. Some factors are related to properties of IBR (Section 5.2): for many IBR-based inferences only certain types of traffic, either due to payload or the frequency in which we observe it, is useful. Other factors stem from the collection infrastructure, such as darknet position in the address space, and darknet size (Section 6.1.6).

5.1 Overall visibility

We investigate how many and what type of networks send IBR. In all our datasets we observe traffic from a non-trivial number of IP addresses ($>$ 100M), /24 blocks ($>$ 2.6M) and prefixes ($>$ 170k), and traffic from almost all countries and most large networks (including non-enterprise ASes). As a result, we can potentially use IBR to characterize many hosts and /24 blocks, and provide Internet-wide analysis at the AS or country-code level.

5.1.1 How many sources are observed?

Table 5.1 reports the absolute number of sources (IP addresses, /24 blocks, prefixes, ASes and countries) observed through our datasets. To analyze coverage, we consider the ASes and prefixes announced in BGP (and number of IP addresses and

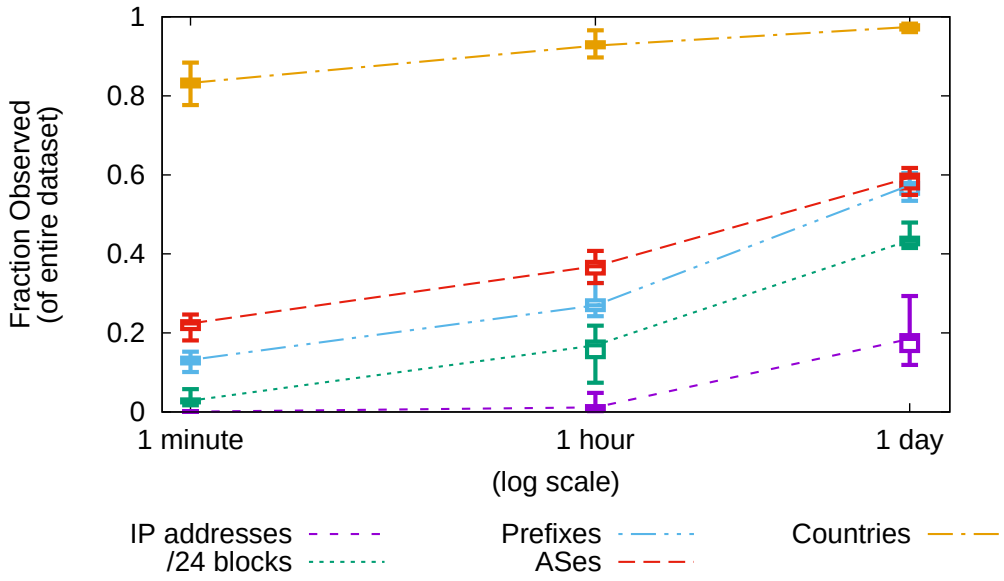


Figure 5.1. Fraction of sources observed per minute, hour and day (UCSD-13). The longer one observes, the more sources one can observe, especially at the IP address granularity.

/24 blocks within the announced prefixes). We observe a few IP addresses, more than a quarter of /24 blocks, and close to half of all prefixes and ASes. We also use MaxMind to geolocate the .0 IP address of all observed /24 blocks: we capture almost all country codes with announced IP addresses. However, a large fraction of address space announced in BGP may not actually generate traffic on the global Internet, which is often called "used" [222, 53]. Based on previous literature, we observe about half of the inferred used /24 blocks: using seven different data sources, Dainotti *et al.* found 5.3M actually used /24 blocks in 2013 [53], while Zander *et al.* estimated that a total of 6.2M to 6.3M /24 blocks were used in June 2014 [222].

While the numbers in Table 5.1 are consistent across all four, 34-day-long datasets, we find considerably fewer sources (except at country-level granularity) with shorter measurement intervals. Figure 5.1 shows statistics on the fraction of sources observed in a minute, hour, or day for UCSD-13. We omit Figures for UCSD-12,

partial-UCSD-13 and MERIT-13 since, for each source and time granularity, we capture approximately the same fraction of the respective dataset total. As expected, by lengthening the observation period, we capture additional sources. However, due to repeated contact, the growth in number of sources observed is less than linear. The exact growth rate depends on source granularity (i.e., IP address, /24 block, prefix, AS or country). For example, we observe over 80% of countries at all time granularities, while, for IP addresses, increasing the time granularity from an hour to a day results in about 20 times more sources. This result is intuitive. At the IP level, individual machines may stop transmitting IBR (due to properties of IBR, users turning off the machine, outages, etc.) or use multiple IP addresses (due to DHCP assignment, host portability, etc.) resulting in intermittent observation of the source. At the other extreme, many hosts contribute to a country's visibility, resulting in more frequent observation (e.g., not depicted in Figure 5.1 is the statistic that for the median country in UCSD-13, we observe 6,070 IP addresses throughout the entire 34-day period).

The number of observed sources can be highly variable, especially at the IP address and /24 block granularities. In Table 5.2, we report the average, standard deviation, and coefficient of variation [67] for a number of source and time granularities for each dataset. The coefficient of variation (ratio of standard deviation to mean) is a dimensionless measure of the fractional increase or decrease relative to the mean that is within a standard deviation. In UCSD-13, the coefficient of variation across days is large for IPs (0.19), but smaller for /24 blocks (0.029), prefixes (0.031), ASes (0.034) and country codes (0.0061). In other words, we can expect (i.e., it is within 1 standard deviation) to observe a 19% increase or decrease in the number of IP addresses captured in a single day.

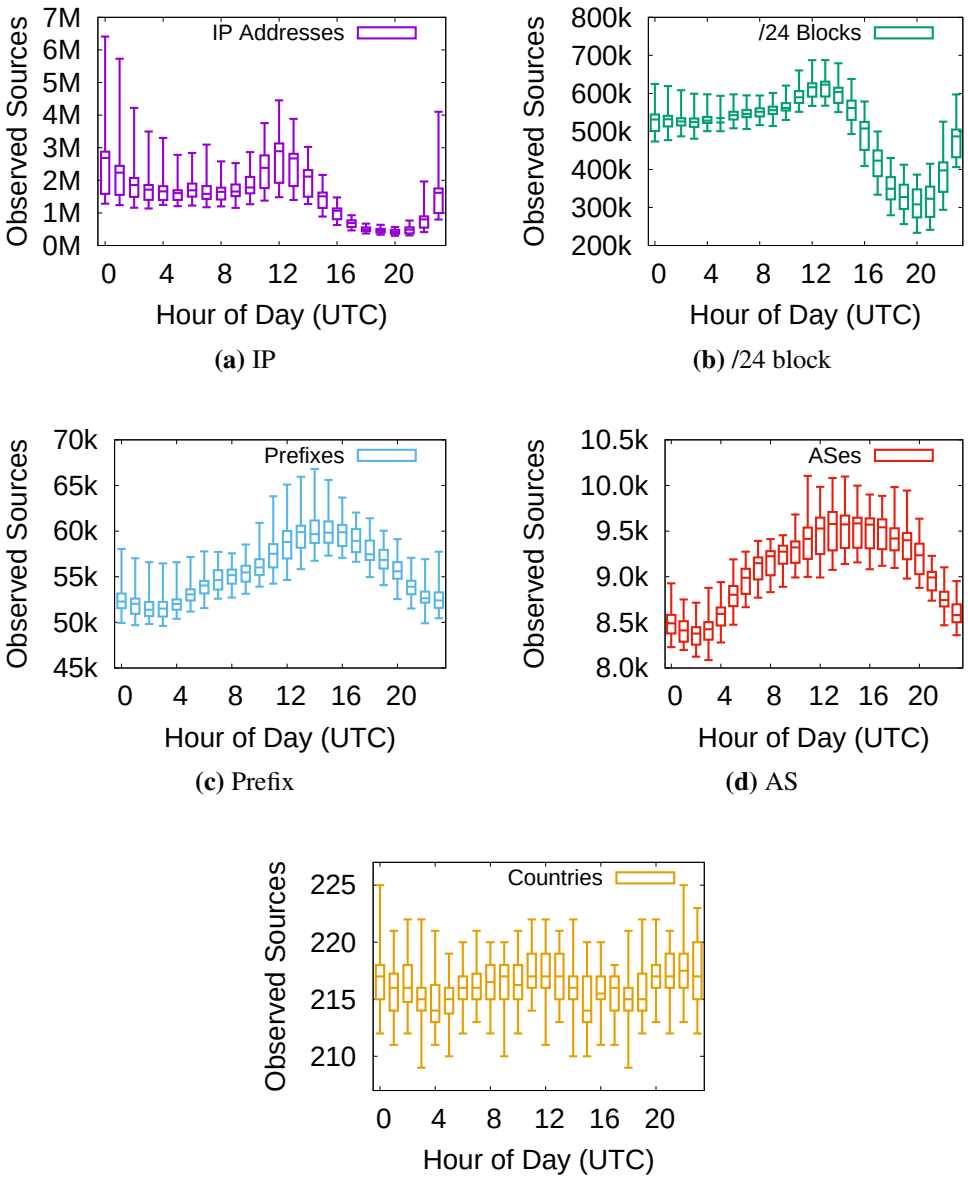
At least two factors — diurnal patterns and the changing composition of IBR — contribute to variance.

Table 5.2. Average and standard deviation of number of observed sources in each dataset. At the IP address and /24 granularities there is considerable variance in the number of observed sources. The number in parenthesis is the coefficient of variation (ratio of standard deviation to the mean), which permits us to compare variability across source granularities.

	IP Address		/24 Block		Prefix		AS		Country	
Time Period	**Avg.**	**Std. Dev.**	**Avg.**	**Std. Dev.**	**Avg.**	**Std. Dev.**	**Avg.**	**Std. Dev.**	**Avg.**	**Std. Dev.**
UCSD-12										
1 minute	120k	24k (0.20)	100k	20k (0.19)	28k	2.0k (0.072)	5.2k	320 (0.061)	186	3.9 (0.021)
1 hour	1.8M	820k (0.46)	560k	100k (0.18)	62k	6.4k (0.10)	9.0k	700 (0.078)	215	4.0 (0.019)
1 day	27M	4.3M (0.16)	1.6M	210k (0.13)	120k	9.7k (0.081)	14k	950k (0.068)	227	1.6 (0.0069)
Census	148M	-	3.14M	- 198k		-	24.3k	-	232	-
UCSD-13										
1 minute	100k	22k (0.22)	90k	18k (0.20)	27k	1.7k (0.063)	5.5k	270 (0.050)	194	2.9 (0.015)
1 hour	1.54M	810k (0.53)	500k	98k (0.20)	56k	3.2k (0.058)	9.1k	440 (0.049)	216	2.3 (0.011)
1 day	24M	4.5M (0.19)	1.4M	39k (0.029)	120k	3.6k (0.031)	15k	490 (0.034)	227	1.4 (0.0061)
Census	133M	-	3.15M	- 205k		-	24.2	-	233	-
partial-UCSD-13										
1 minute	62k	9.6k (0.15)	54k	8.4 (0.15)	21k	1.4k (0.067)	4.5k	230 (0.051)	185	3.0 (0.016)
1 hour	690k	340k (0.49)	360k	97k (0.27)	43k	2.3k (0.056)	7.7k	360 (0.046)	208	2.9 (0.014)
1 day	11M	2.4M (0.21)	990k	34k (0.034)	88k	3.3k (0.037)	12k	370 (0.031)	223	1.4 (0.0060)
Census	109M	-	2.65M	-	170k	-	19.3k	-	231	-
MERIT-13										
1 minute	70k	10k (0.14)	62k	8.8k (0.14)	22k	1.4k (0.063)	4.6k	230 (0.049)	189	3.0 (0.016)
1 hour	730k	350k (0.47)	380k	94k (0.25)	45k	2.3k (0.051)	8.0k	340 (0.043)	211	2.5 (0.012)
1 day	12M	2.4M (0.21)	1.1M	29k (0.027)	94k	2.7 (0.028)	12k	310 (0.025)	224	1.5 (0.0068)
Census	111M	-	2.76M	-	175k	-	19.8k	-	232	-

Diurnal patterns

Diurnal patterns in IBR[217] are one cause of variability, especially for small source granularity (i.e., IP addresses and /24 blocks). As shown in Figure 5.2, with small source granularity, diurnal differences result in about a six-fold increase in observed IP addresses per hour and a two-fold increase in observed /24 blocks per hour (based on the median observation). Previous work showed that diurnal patterns are prevalent; in particular, China had both a high number of /24 blocks and a high fraction of blocks exhibiting diurnal activity [166]. Our peak at 12:00 UTC corresponds to 8 PM in China, and the low point at 20:00 corresponds to 4AM in China. At the IP-address level there is also an extremely large maximum value from 0:00 UTC to 2:00 UTC – more than twice all other values during these hours. Traffic associated with a software update caused the abnormally high values (Section 4.3.1); we believe the vendor pushes updates to all



(a) IP **(b)** /24 block **(c)** Prefix **(d)** AS **(e)** Country

Figure 5.2. Diurnal Patterns. Minimum, first quartile, median, third quartile and maximum number of sources observed for each hour of the day in UCSD-13. Especially for small source granularity (i.e., IP addresses and /24 blocks), the number of sources observed depends on the time of day.

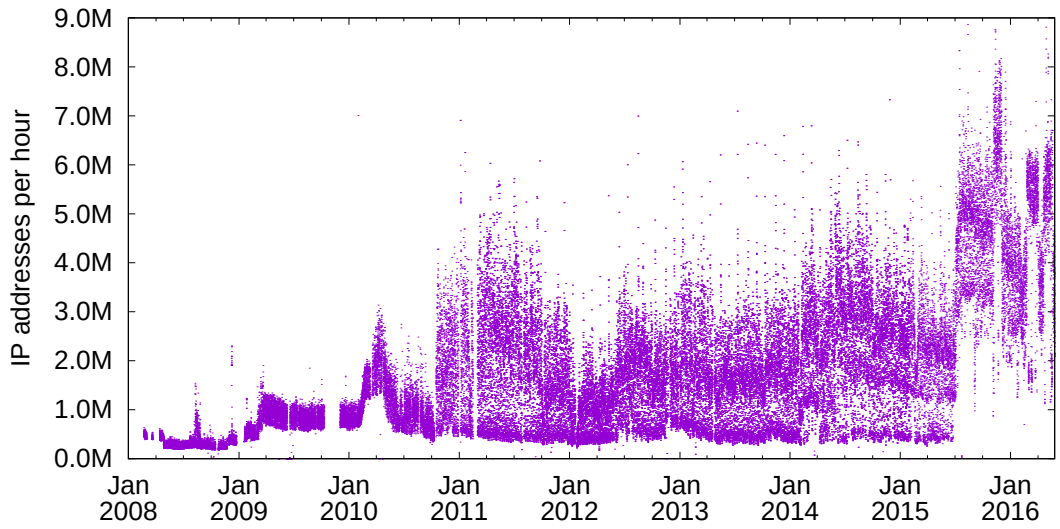


Figure 5.3. IP addresses observed per hour over a 8-year period (UCSD-NT).

clients during this time. There is less diurnal variation for prefixes, ASes and countries. The small peaks at the prefix and AS levels occur at a different time of day than for IP addresses and /24 blocks. This difference is likely due to the non-uniform distribution of ASes: the United States accounts for almost one-third of observed ASes.[1]

Temporal differences caused by the changing composition of IBR

Over time, IBR evolves. Not just in terms of its constituent packets and bytes, as studied by Wustrow *et al.* [217], but also in terms of the number of sources sending IBR. The changing composition of IBR contributes to the variance in number of observed sources on longer time scales.

To identify times when significant changes occurred, we consider: (1) the number of IP addresses observed per hour for most of 2008–2016 (Figure 5.3); (2) the per-day contribution of the major components[2] over the period of January 2012 to May 2016

[1] We geolocate an AS to the most common location of /24 blocks in the AS.

[2] We extract some IBR components with a pcap signature. When operating on flow-level data, we use heuristics instead. E.g., for BitTorrent traffic we use popular message lengths (with low false positive rate) instead of examining the payload.

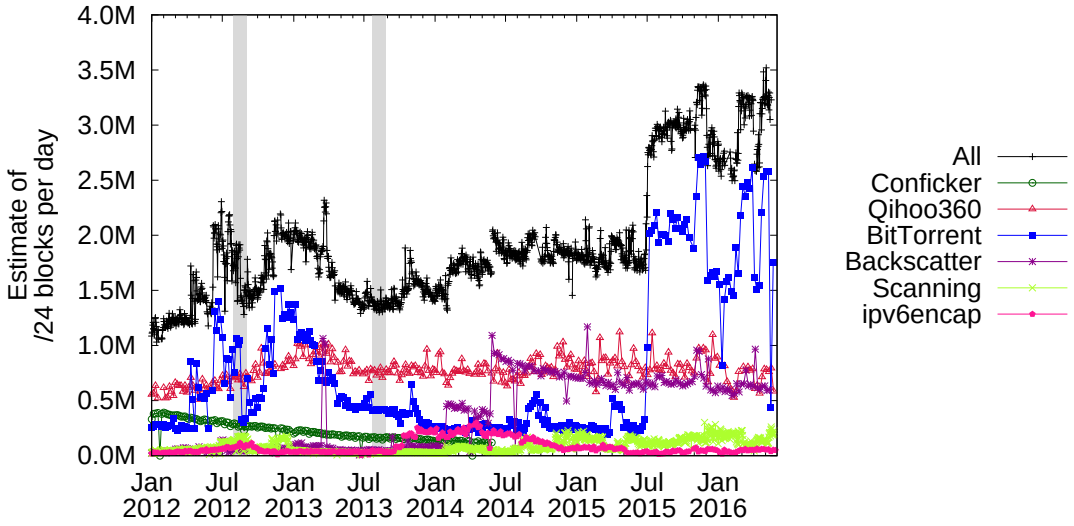


Figure 5.4. /24 blocks observed every 6th day at UCSD-NT by IBR component. Fluctuations in IBR influence the total number of /24 blocks observed. In this graph, we use heuristics for components with pcap-level signatures. The shaded portions indicate the *2012* and *2013 census*

(Figure 5.4); and (3) the total number of /24 blocks per component during the *2012* and *2013 census*. The following events produced large changes in the number of of IBR sources:

- *November 2008*: Conficker worm outbreak
- *March 2010, July 2015*: Significant BitTorrent traffic observed
- *October 2010*: Start of traffic from Qihoo 360 bug
- *March 2013*: A spike in Backscatter traffic as the result of a DoS on Spamhaus
- *February 2014, June 2014*: Increase in backscatter containing responses to DNS queries
- *2012 census vs 2013 census*: Due to activity by the Carna Botnet in 2012, the number of /24 blocks labeled as Bro Scanners in `UCSD-12` is three times the amount in `UCSD-13`

Figure 5.4 shows that the number of observed sources per day is correlated with the number of BitTorrent sources. This correlation is due to the fact that misconfigurations in BitTorrent's DHT induce many sources to send IBR to our darknet. Backscatter events, such as the Spamhaus attack and traffic from open resolvers, can also cause many sources to send IBR. Qihoo 360 traffic, which is the result a long-lasting bug, has little effect on the per-day number of sources.

Our ability to make network inferences is influenced by both the trends and erratic nature of IBR. Some inferences use a specific type of traffic; thus fluctuations of that specific component will diminish or improve our ability to make the corresponding inferences. Additionally, inferences that aggregate many traffic components can also be influenced by changes in an individual components. For example, from Figure 5.4, we see that BitTorrent traffic was highly variable during the *2012 census* compared to the *2013 census*. This variance is reflected in Table 5.2 where the coefficient of variation is large per day at the /24 granularity in `UCSD-12` (0.13), but smaller for all 2013 datasets (about 0.03).

5.1.2 What types of networks are observed?

We analyze which countries and which autonomous systems have at least one host that sends IBR. Since we find hosts located in almost all countries and most large autonomous systems there is not an obvious bias in IBR.

Country-level coverage

We use historical MaxMind country-level databases to geolocate the .0 address of each /24 block in our IBR datasets. Since MaxMind updates the database regularly (to reflect changes in the address space), we use the databases produced on August 1, 2012 and August 16, 2013 for the *2012 census* and *2013 census* periods, respectively.

We observe traffic from diverse locations. During the *2013 census*, of the 249 ISO-3166-2 country codes, thirteen do not have an address announced in BGP. 11 are islands or collections of islands with populations under 7,500. The remaining two of the country codes are located in Africa: the disputed territory of Western Sahara (population 555k, possibly using addresses that geolocate to Morocco) and South Sudan (population 11.5M, which has an Internet Penetration of 100 users [99]). Of the remaining 236 country codes (those with an IP address announced in BGP), we miss only three with the `UCSD-13` dataset. All three countries are small islands or collections of islands, each with a population of under 4,000 people [44].

AS-level coverage

We use CAIDA's Prefix-to-AS mapping dataset (pfx2as) to map IPv4 addresses to AS numbers [179]. CAIDA extracts this dataset from BGP announcements captured by Routeviews. Specifically, we use the mapping produced on the first day of the IBR datasets. To label ASes as transit/access providers, content providers, or enterprise networks, we use a dataset provided by CAIDA developed using a scheme similar to that proposed by Dhamdhere and Dovrolis [59].

Many ASes do not send IBR to our darknets: we observe about half of ASes announced in BGP. However, most missed ASes are small. Figure 5.5 shows, for `UCSD-13`, the distribution of observed ASes in terms of /24 blocks announced. Of the 20.6k unobserved ASes in `UCSD-13`, almost half announce a single /24 block, and 90% announce the equivalent of 8 or fewer /24 blocks. Conversely, we observe 86% of ASes that advertise the equivalent of at least a /16 block – we call these ASes large. ASes belonging to the US Department of Defense account for a fifth of unobserved large ASes, which appears to have many routed but "unused" /24 blocks [53]. In terms of AS type, we miss 26% of large ASes classified as enterprise, and about 4% of the large ASes classified as

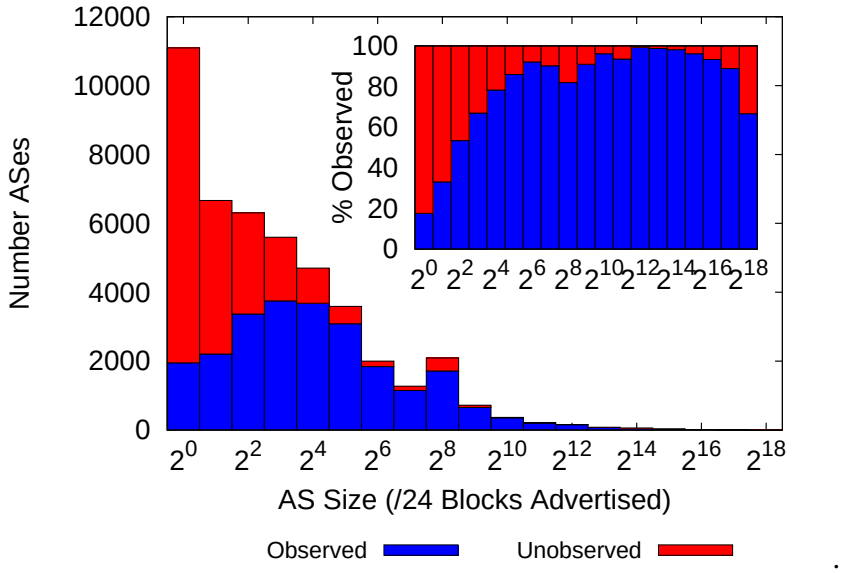


Figure 5.5. Number and percentage of observed ASes by the number of /24 blocks announced. Although we observe only half of announced ASes in `UCSD-13`, most missed ASes announce few /24 blocks.

transit/access or content.

5.1.3 Implications of overall IBR visibility

The number of sources captured by a discussion is dependent on the duration of observation, the time of day, and the size of the network. Across our datasets, we consistently observe a significant fraction of the observably "used" IPv4 address space, and in particular nearly all large transit/access and content ASes. As a result, IBR has the potential to provide an Internet-wide view.

5.2 Properties of IBR influencing visibility

Most often, only a certain type of traffic (based on the specific information that its behavior or content brings) is helpful in inferring a property of a network. For example, in Section 7.2 we use the retransmission behavior of TCP to infer packet-loss. To make packet-loss inferences with this methodology, we need certain types of packets

(TCP) and sending behavior (retransmissions). It is thus important to understand how the properties of IBR influence potential information content.

We lack control over both what type and how often traffic reaches our darknets. However, we can measure coverage as a function of common categorizations. In Section 5.2.1, we study IBR's coverage when we restrict our analysis to certain types of packets; in Section 5.2.2, we study IBR's coverage when we have requirements on the frequency in which we observe a source (IP addresses, /24 block, prefix, AS, or country).

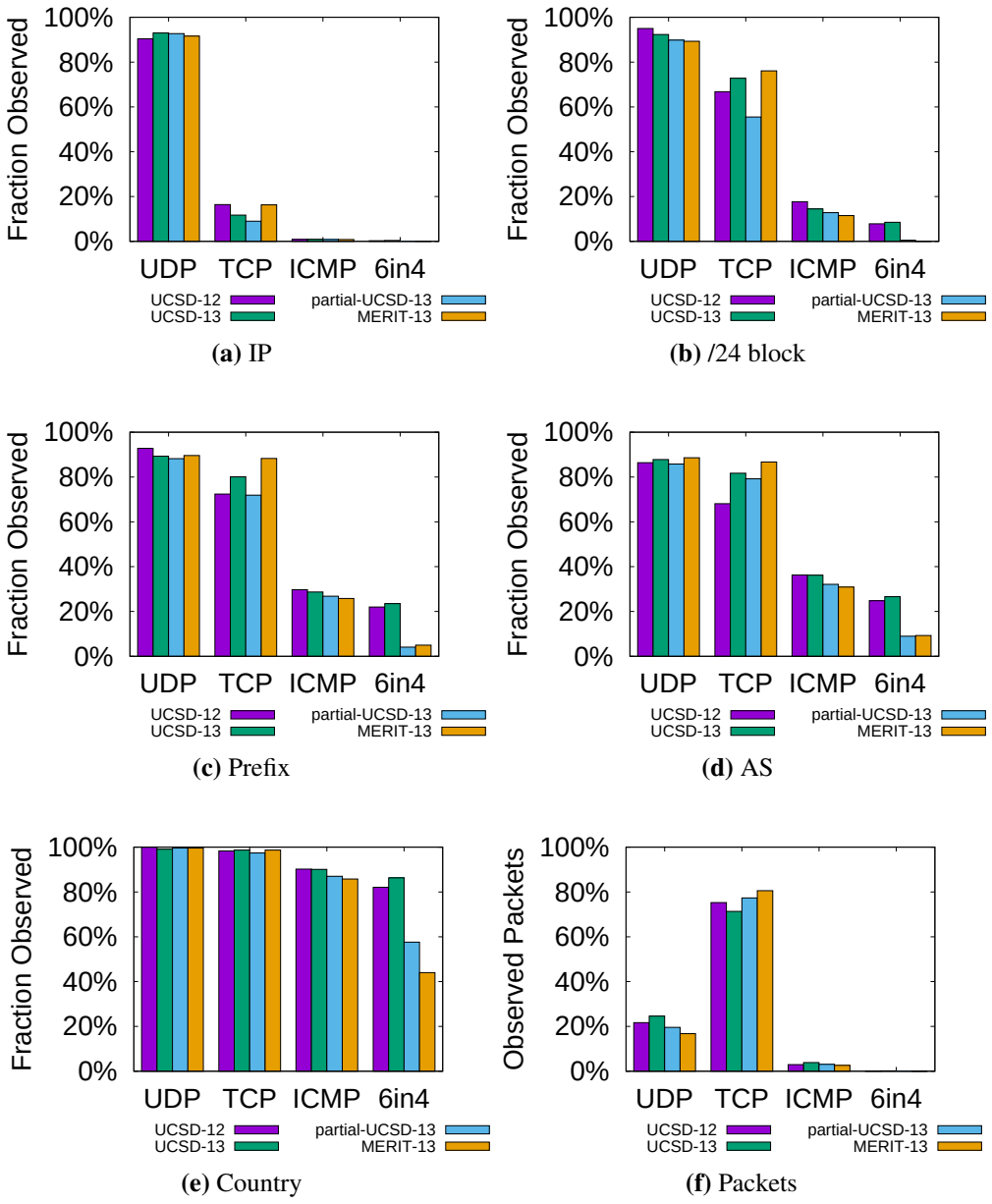
5.2.1 Impact of IBR components

In this Section, we look at how IBR components (extracted in Chapter 4) influence our visibility into remote networks. Our main goal is to determine if IBR's Internet-wide coverage is dependent on a single phenomenon the result of a mix of traffic types. To make this determination, we characterize IBR along two basic dimensions: transport layer protocol and application, since the information encoded in IBR is a function of them.

How many sources use TCP vs. UDP?

Figure 5.6 reports the fraction (out of the total observed in the respective dataset) of IP addresses, /24 blocks, prefixes, ASes, and country codes observed through the most popular transport layer protocols. We observe most IP addresses via UDP traffic. Both TCP and UDP packets provide high visibility into /24 blocks and ASes, although neither provides complete coverage. All transport layer protocols provide excellent coverage of countries.

Wustrow *et al.* [217] characterize IBR based on the volume of *packets*, and not the number of *sources*. They find that from 2006–2010 TCP was the dominant protocol (above 75% of packets) for all years except 2008. Although our datasets are not directly



(a) IP **(b)** /24 block **(c)** Prefix **(d)** AS **(e)** Country **(f)** Packets

Figure 5.6. Top protocols by source granularity and packets. Most IP addresses send UDP traffic. At the /24 block, prefix, AS and country levels we observe a similar percentage of sources sending TCP and UDP. TCP accounts for most packets.

Table 5.3. /24 blocks observed by IBR component. IBR is composed of many different types of traffic. File-sharing traffic contributes the highest number of /24 blocks in all datasets, but there are variations based on time (`UCSD-13` vs `UCSD-12`) and position (`partial-UCSD-13` vs `MERIT-13`). We observe most /24 blocks through multiple IBR components, implying that insight into a network is not dependent on a single type of traffic.

Component	**UCSD-12**		**UCSD-13**			**MERIT-13**		
	Total	Unique	Total	Unique	Partial Total	Total	Unique	∩ UCSD-13
Bugs & Misconfigurations								
File Sharing (BitTorrent, eMule, QQLive) [124, 150, 122]	2,640k	284k	2,490k	344k	1,910k	2,090k	377k	1,980k
Qihoo 360 Safe Bug [4]	1,450k	98.5k	1,340k	117k	1,110k	1,110k	138k	1,050k
Encapsulated IPv6 (6in4, Teredo) [197]	1,080k	9.48k	744k	11.5k	392k	368k	5.94k	312k
Gaming (Xbox, Steam) [183, 139]	503k	4.50k	490k	14.3k	258k	185k	11.9k	131k
Botnet C&C (ZeroAccess, Sality) [65, 136]	551k	17.3k	184k	4.97k	51.7k	51.6k	2.37k	25.7k
Scanning								
Conficker [41]	642k	24.4k	579k	58.1k	573k	568k	96.9k	563k
Bro Scanner [198]	597k	8.48k	197k	4.57k	104k	99.1k	4.06k	91.8k
Backscatter								
Backscatter [141]	394k	45.3k	392k	51.6k	247k	246k	21.3k	219k
Unclassified								
Encrypted [82]	1,450k	98.5k	1,340k	117k	819k	755k	29.8k	667k
Other	1,980k	73.8k	1,910k	127k	1,440k	1,70k	135k	1,410k
All Components	3,130k		3,150k		2,650k	2,760k		2,670k

comparable (they do not remove spoofed packets), we also find that TCP is the dominant protocol by number of packets (Figure 5.6f). Since UDP is the dominant protocol in terms of source IP addresses and TCP is the dominant protocol in terms of packets, the protocols may have different strengths when inferring network properties: UDP is more likely to provide wide coverage, while TCP is more likely to support analyses requiring repeated contact (Section 5.2.2).

Which applications contribute the most sources?

Table 5.3 shows that multiple components contribute a significant number of source /24 blocks (we included all components contributing over 300k /24 blocks in a dataset, except classes we derived because they were popular ports or UDP packet lengths). We aggregate some small components and all unclassified components into the "Other" category. We group the components based on the reason they appear in

IBR: accidentally (i.e., due to bugs or misconfigurations), as part of a scan, backscatter from spoofed traffic (such as DoS attacks), and for unknown reasons. In Section 6.1.6, we link trends of the individual components to changes in IBR properties over time.

When studying 2010-era IBR reaching four /8 networks, Wustrow *et al.* find that scanning accounts for the majority of packets in all but 1.0.0.0/8 [217]. In our datasets, many well-studied, malicious IBR phenomena— Bro Scanner, Conficker, and Backscatter—also account for most of the packets (collectively contributing about 83% of all packets in `UCSD-13`). But, surprisingly, malicious traffic is not the largest component of IBR in terms of sources. Packets with a P2P file-sharing payload contribute over 1.9M /24 blocks in all datasets, accounting for over two-thirds of all /24 blocks observed; Qihoo 360 traffic alone contributes about 100M IP addresses.

We observe most /24 blocks through multiple IBR components, implying that many types of IBR can provide insight into the same networks. In particular, even without the top IBR components, the "Other" component alone contributes 1.4M /24 blocks. The "Unique" column of Table 5.3 reports the number of /24 blocks observed through a single IBR component. For each component, the number of unique /24 blocks is at least an order of magnitude smaller than the total number of /24 blocks observed through that component. As a result, if the composition of IBR changes slightly we would still observe many of the same networks.

Moreover, we expect little causality between IBR components: e.g., playing Xbox games does not result in a Conficker infection. Exceptions to causality include: many Encapsulated IPv6 (6in4) packets have a BitTorrent payload, and botnets may coordinate scans of the Internet.

Implications of IBR components on visibility

Some IBR-based inferences require a certain type of traffic; other network properties can be inferred regardless of the underlying application, but their success is dependent on the composition of IBR. Fortunately, IBR is made up of many components, each of which contributes relatively few unique /24 blocks (implying some analyses may be robust to fluctuations in IBR composition). While most packets are TCP (due to scanning and backscatter), we observe more IP addresses from UDP traffic (due to P2P and bugs). IBR is commonly known as malicious traffic. However, we find that the phenomena that contribute the highest number of sources (over 1M /24 blocks) appear to be of benign nature.

5.2.2 How often do we receive IBR?

In this section, we consider inferences that require multiple observations of a given host/network. For example, in Section 7.1, we determine that the path from hosts in an AS to a darknet changed by observing the behavior of the TTL field. In addition to looking for changes in given fields, we can leverage the timing between packets (e.g., to infer uptime [112]) and the predictability of repeated contacts (e.g., to infer outages [56]).

To study repeated contact from IBR sources, we report (1) how often we observe a source, (2) the length of time between the first and last observation of a source, and (3) the timing between contacts. Our approach is to partition our dataset into 1-minute, 1-hour, and 1-day time bins and record the sources sending IBR in each bin. In mathematical notation, let S and T be the set of all sources and time bins at given granularities, and $I_s(t)$ be an indicator function for a source s for a time bin t that is 1 if the source is

observed, and 0 otherwise. For property (1) we compute, for each $s \in S$:

$$\sum_{\{t \in T\}} I_s(t);$$

for property (2) we determine, for each $s \in S$:

$$\max_t \{t \in T | I_s(t) = 1\} - \min_{t'} \{t' \in T | I_s(t') = 1\};$$

and property (3) can be expressed as a multiset, where we include for each $s \in S$ and $\{t \in T | I_s(t) = 1\}$ the value (if it exists)

$$t - \max_{t'} \{t' \in T | I_s(t') = 1 \wedge t' < t\}.$$

Host contacting the darknet frequently retransmit the packet after the initial packet fails to elicit a response. These *communication attempts* could span multiple time bins, which would lead to inadvertently skewing properties (1), (2) and (3). To check that our partitioning confines most communication attempts to a single time bin, we determine the typical number of packets per communication attempt and the timing between these packets. Table 5.4 reports statistics on communication attempts (packets with the same {source ip, destination ip, protocol, source port, destination port}) with hour bins by IBR component from `UCSD-13`. The number of communication attempts varies depending on the IBR component, as does the behavior of the hosts sending each type of traffic (as evidenced by the median number of attempts per source IP address). However, for all components, the average number of packets per communication attempt is small. Manual investigation reveals that the timing between packets is also small (e.g., 3 seconds between retransmission of Conficker packets). As a result, binning does not significantly skew our calculations in the following sections.

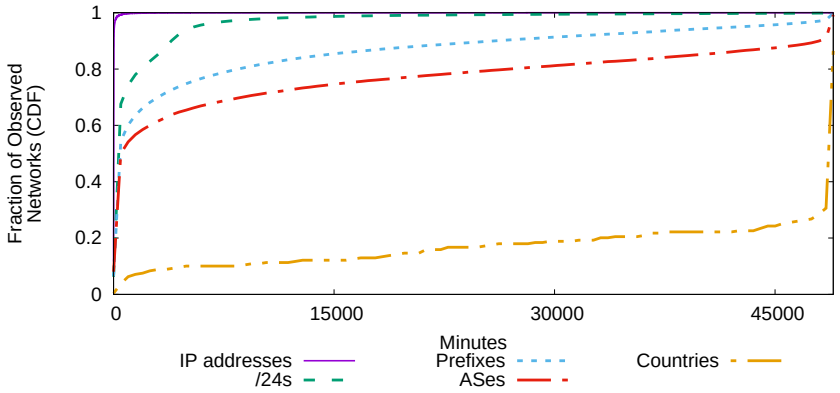
Table 5.4. Communication attempts by IBR component for `UCSD-13`. IBR components vary in the number of communication attempts made, and the median attempts made per source IP addresses. But, all components have a low number of packets per attempt, which suggests binning the data will not result in significant double counting.

Component	Communication Attempts	Avg. Pkts per Attempt	Median Attempts per Source IP
File Sharing	1,120M	6.13	2
Qihoo 360	1,520M	1.62	11
Encap. IPv6	108M	4.49	2
Gaming	95.4M	1.04	1
Botnet C&C	13.3M	2.95	3
Conficker	13,800M	1.98	109
Bro Scanner	27,400M	1.10	684
Backscatter	20,700M	1.23	6
Encrypted	137M	2.33	1
Other	1,740M	3.33	3
Total	66,700M	1.50	11

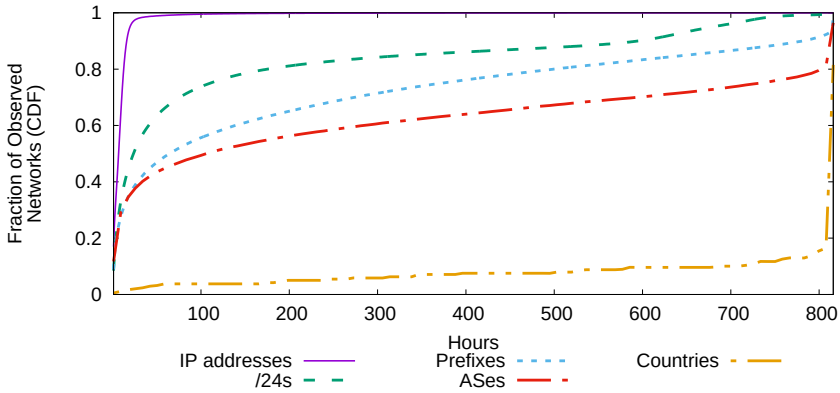
How often do sources send IBR?

The frequency with which we can infer properties of a remote network depends on how often we receive traffic from that network. Figure 5.7 shows the cumulative distribution function of sources observed using 1-minute, 1-hour, and 1-day time bins in `UCSD-13`. We observe frequent contact at coarse source granularities, e.g., countries and some ASes. The values on the far right of the subfigures in Figure 5.7 indicate the number of networks that we observed in every time bin of `UCSD-13`. Figures 5.7a and 5.7b suggest that inferences requiring near-constant traffic samples are only possible for ≈80% of countries and ≈20% of ASes. As expected, the CDF curves shift towards more frequent contact as we move to larger time bins.

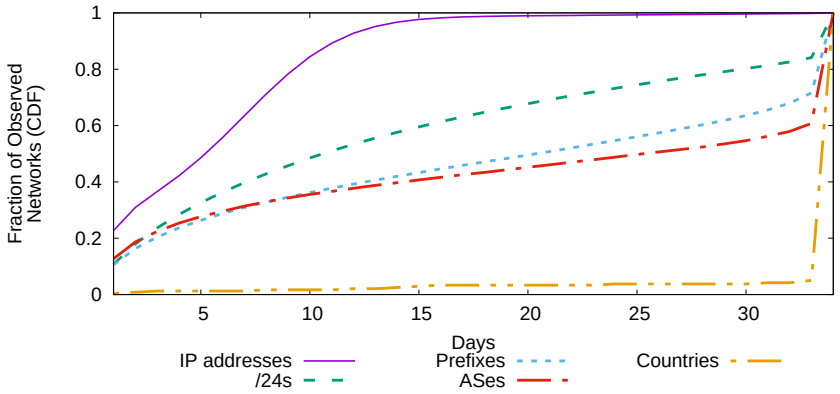
Finer source granularities, such as IP addresses, are unsuited for inferences requiring frequent observations. We cannot conduct repeated measurements for approximately 12% of IP addresses because we observe them in only one 1-minute time bin; we observe most IP addresses in less than 11 1-minute time bins. But as the size of the time bin increases to hours or days, the number of contacts per source increases. For example, we observe traffic from over 75% of IP addresses, /24 blocks, prefixes ASes and countries in multiple days.



(a) Minutes.



(b) Hours.



(c) Days.

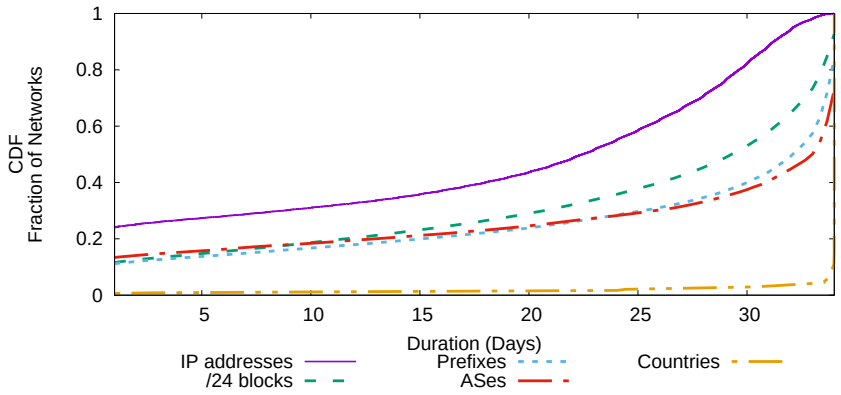
Figure 5.7. CDF of fraction of sources observed with minute, hour and day time bin granularities (`UCSD-13`)**.** To make repeated inferences, we need to observe a source in multiple time bins. While inferences requiring observations in every time bin seems possible for countries and some ASes, we observe most sources — even at the IP address granularity — more than once.

At the IP address granularity, our results depend on the current composition of IBR. we observe 50% of IP addresses in more than 6 distinct days. However, Qihoo 360 traffic strongly influences the distribution of IP address over days. Excluding this traffic component results in 75% of IP addresses appearing only in a single day. Consequently, after the Qihoo 360 bug is fixed we expect substantially different results. Qihoo 360 does not impact significantly statistics at the other source granularities.

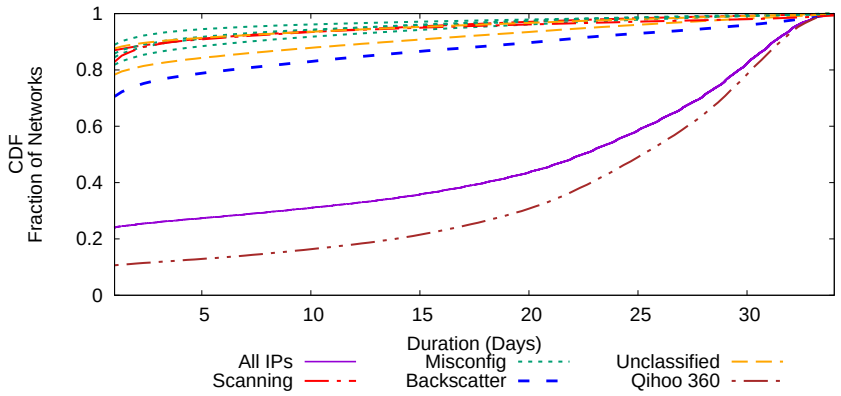
What is the total duration of contact?

To conclude if our observations are the result of a single bursty event, or if sources are visible throughout our datasets, we investigate the range of times that we observe a source. We calculate each source's duration of contact (time of last contact minus time of first contact). Figure 5.8a shows the CDF of this distribution. The total duration of contact is long (over 29 days out of 34) for most /24 blocks, prefixes, ASes, and countries. Despite observing most IP addresses in only a few 1-minute or 1-hour time bins, the duration of contact is also long for IP addresses (50% IP addresses had a duration of contact longer than 22.5 days), implying that there is a long time between consecutive observations of a source IP address.

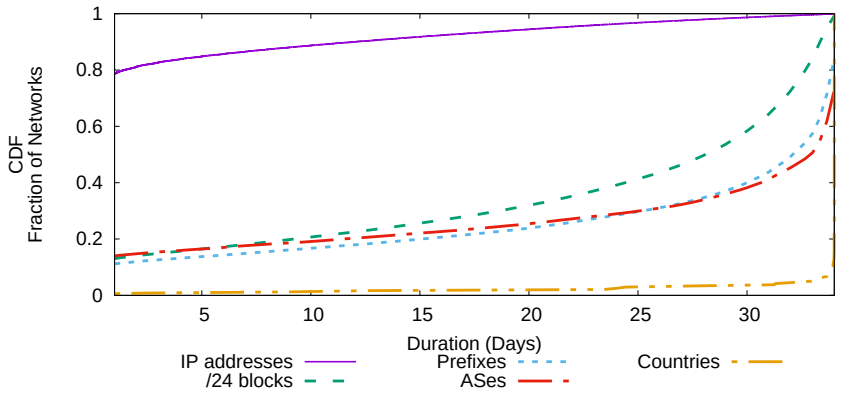
Figure 5.8b shows, at the IP-address-level, the duration of contact broken down by IBR component. All components besides Qihoo 360 traffic have relatively short duration of contact: for each component except Qihoo 360, over 70% of IP addresses have a duration of contact of less than 1 day. This implies that most scanning events and misconfigurations are short lived. Although most sources sending backscatter have short durations of contact, for about 130k IP addresses the contact duration is greater than 15 days. This is surprising as we generally think of backscatter as a byproduct of DoS attacks — which are typically short events [141]. We investigate why many IP addresses sending either backscatter or Qihoo 360 traffic have relatively long contact



(a) All traffic.



(b) By IBR component (IP addresses).



(c) Without Qihoo 360 traffic.

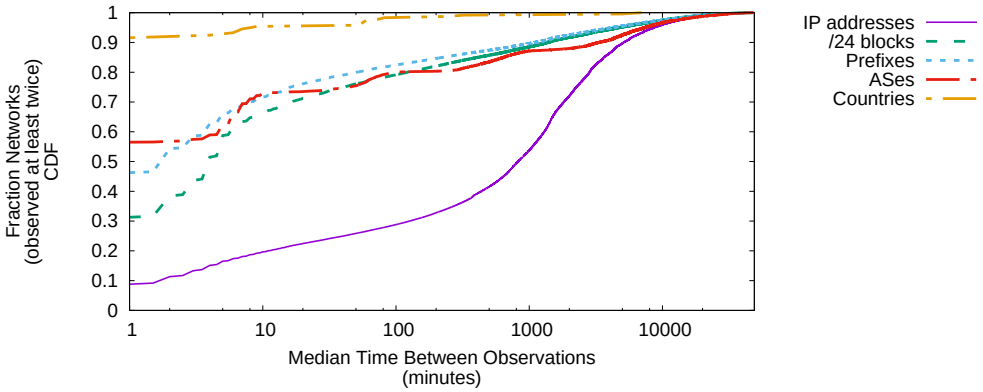
Figure 5.8. CDF of contact duration (`UCSD-13`)**.** At all source granularities the contact duration is long, which is desirable for analysis throughout the datasets.

durations.

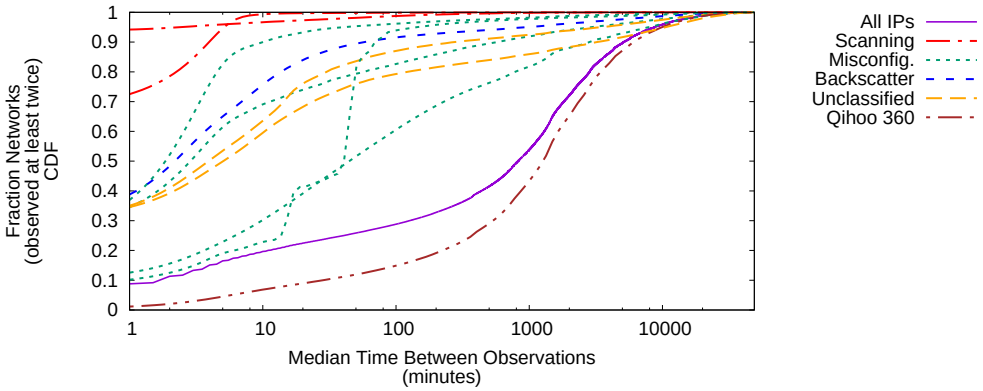
Backscatter: Often packets classified as backscatter reach the darknet for reasons other than DoS attacks. Only 3,600 of the 130k IP addresses appear to be repeat targets of DoS attacks as they sent more than 1,000 backscatter packets per hour more than one day apart. A large number of web servers (61k of the 130k IP addresses on TCP port 80 and 24k on TCP port 443) appear to repeatedly receive and respond to a low volume spoofed packets. Similarly, over 38k IP addresses sent a low volume of ICMP backscatter messages throughout `UCSD-13`. Additionally, there is a small amount of packet misclassification. 2,400 IP addresses appear to be conducting stealthy scans of TCP destination port 3389.[3]

Qihoo 360: We attribute the long duration of contact at the IP level to Qihoo 360 traffic, which has a diurnal cycle. Since about 70% of IP addresses send Qihoo 360 traffic in `UCSD-13`, it strongly influences the overall duration at the IP address granularity. As observed in Figure 5.8c, without Qihoo 360 traffic 80% of IP addresses have a contact duration of less than one day. However, there is only a small influence on the duration of contact at the /24 block, prefix, AS, and country granularities. The signal for these aggregated granularities is comprised of a mix of traffic components and is not dependent on Qihoo 360.

This analysis shows the potential to make IBR-based inferences at the /24 blocks, prefixes, ASes and countries granularities for the duration of the datasets. At the IP-address granularity, we observe the sources throughout the datasets, but this is mostly due to Qihoo 360 traffic.



(a) All traffic.



(b) By IBR component (IP addresses).

Figure 5.9. Median time between observations (`UCSD-13`). Most /24 blocks, prefixes, ASes, and countries observed multiple times have a short time between observations (less than 10 minutes), which is desirable for fine-grained analysis. By component, scanning traffic has the shortest median time between observations.

Time between communication attempts?

If the time between observations is short, we can pinpoint the precise moment network changes occur (e.g., the exact moment a path change occurs in Section 7.1). To evaluate our ability to perform this "fine-grained analysis" with IBR, we study the time between observations of traffic from a source. Figure 5.9a shows the median time between all sources that we observe in at least two 1-minute time bins. We observe most countries all the time: the median time between observations is 1 minute for 92% of countries. At the /24 block and AS levels, the time between observations is often longer, although the time between contacts at these granularities is often within 10 minutes. There is a longer period of time between observations of an IP address: half of IP addresses have a median inter-observation time of more than 13.7 hours. However, for some IP addresses the inter-observation time is still short (27% of IP addresses have a median inter-observation time of less then 1 hour).

Figure 5.9b shows the breakdown of median time between observations for IP addresses by IBR component. Qihoo 360 traffic heavily influences the overall behavior of IP addresses: 50% of IP addresses associated with Qihoo 360 have a median time between observations of greater than 21.2 hours (presumably because they receive updates about once per day). The median time between observations is substantially shorter for the other IBR components. As a result, our ability to conduct fine-grained analysis comes from IBR components other than Qihoo 360. Scanning traffic has the shortest time between observations: for over 90% of IP addresses the median time between observations is less than 4 minutes. One type of misconfiguration causes hosts infected by a botnet to send command and control traffic to the UCSD darknet and wait

[3]These scanners send SYN packets as well as either SYN-ACK or RST packets to darknet IP addresses. In general, the scan is conducted at a lower rate than our scan detection parameters (sends TCP destination port 3389 packets to at least 25 unique darknet IP addresses in a 5 minute period). A handful of these IP addresses (114) do reach the scan threshold at least once during the 2013 census.

either 15 minutes or 1 hour between communication attempts. Qihoo 360 traffic does not heavily influence the time between observations at the /24 block, AS or country levels.

Implications of repeated contact on IBR visibility

We find that many sources repeatedly contact our darknets. We almost always observe traffic from most countries and many ASes, e.g., we observe them in nearly all time bins, throughout the entire observation period, and with a short time between observations. We continually, but not constantly, observe most /24 blocks and prefixes, e.g., they have a long contact duration but the median time between observations is often over an hour. At the IP level, a diurnal bug in Qihoo 360 generates traffic that heavily influences the contact duration and time between intervals. When we exclude the Qihoo 360 traffic, three-quarters of IP addresses have a contact duration of less than one day (i.e., we observe the source in a single day of our 34-day observation period). As a result, IBR is not well suited for long-term inferences at the IP address granularity.

5.3 Properties of collection infrastructure influencing visibility

In this section, we examine the dependence of IBR on the site of data collection. We discover a number of differences, which can be attributed to the properties of influential IBR components. These results, in conjunction with Section 4.4, (1) confirm that the findings presented in the rest of this chapter are representative in terms of number of IBR sources, the mix of components and visibility, (2) identify aspects of IBR that limit its ability to make inferences about remote networks, and (3) set expectations for the performance of other darknets.

5.3.1 Dependence on position in IPv4 space

In this section we analyze how the IP addresses that collect IBR influence the number of observed /24 blocks. Specifically, we consider hotspots, darknets in different /8 blocks (UCSD-NT and MERIT-NT) as well as non-contiguous darknets — similar to the greynets studied by Harrop *et al.* [85].

Hotspots

Hotspots are IP addresses or groups of IP addresses that receive traffic from a disproportionate number of sources. To study IP hotspots, we first determine which IP addresses are not hotspots because they receive traffic from a typical number of sources. Figure 5.10 shows the distribution of observed /24 blocks for 99.5% of darknet IP addresses in `UCSD-13`. The equivalent graph for the 2012 data exhibits a similar distribution. The graph has three obvious modes. The highest mode consists of addresses targeted by Conficker. The remainder of darknet IP addresses capture between 1,000 and 2,000 /24 blocks, with a slight increase when the last byte of the darknet IP address is less than 128. Not visible in the graph is a small peak corresponding to the IP addresses in UCSD.175.0.0/16.

We are interested in the remaining 0.5% of darknet IP addresses: those receiving traffic from the most /24 blocks. In both datasets there were over 75 darknet IP addresses receiving traffic from over 100k /24 blocks — 60 times more than the median darknet IP. We show the magnitude of the top 10 IP hotspots in `UCSD-12` and `UCSD-13` in Table 5.5.[4] We also note the number of unique /24 blocks, those block that are only observed (in that dataset) through the hotspot. The number of unique /24 blocks can help us analyze a hotspot's exclusivity. For example, while numerous, Qihoo 360 hotspots produce

[4]Starting in July 2015, one darknet IP address received BitTorrent traffic from over 3M /24 blocks per month (Section 4.2.2) — an order of magnitude more than those in Table 5.5.

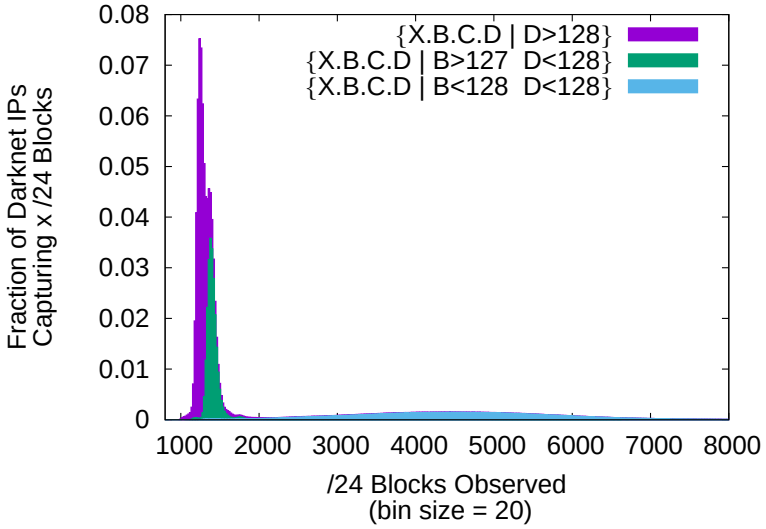


Figure 5.10. Distribution of observed /24 blocks by darknet IP addresses (`UCSD-13`). The typical darknet IP address receives traffic from 1k to 7k /24 blocks, depending on address properties. This graph excludes 0.5% of UCSD-NT addresses, which are IP hotspots.

550 or fewer /24 blocks. That is, hosts sending Qihoo 360 traffic tend to send packets to multiple darknet IP addresses.

Examining /24 blocks observed over time (both per hour and cumulatively) indicates if hotspots are due to flash events, or ongoing phenomena. Spikes in /24 blocks indicate a flash event. Examples of spikes include: a temporary misconfiguration where hosts in 26k /24 blocks sent TCP SYN packets to a single host in a span of 12 hours and

Table 5.5. Top 10 IP hotspots in `UCSD-12` and `UCSD-13`. Due to a variety of reasons, the top 10 IP hotspots in both datasets receive traffic from over 200k /24 blocks.

UCSD-12				UCSD-13			
IP Address	**/24 blocks**	**(Unique)**	**Payload**	**IP Address**	**/24 blocks**	**(Unique)**	**Payload**
X.48.59.58	395k	(61)	Qihoo360	X.0.0.253	307k	(3k)	DNS Queries
X.136.65.114	332k	(1k)	BitTorrent	X.28.192.20	291k	(1)	Qihoo360
X.32.204.14	248k	(3)	Qihoo360	X.187.203.223	286k	(207)	Qihoo360
X.238.254.254	247k	(8k)	ZeroAccess	X.0.0.3	283k	(753)	DNS Queries
X.150.105.113	247k	(2)	Qihoo360	X.176.120.106	226k	(550)	Qihoo360
X.200.7.9	236k	(1k)	Unknown 13 byte encrypted	X.15.97.82	211k	(8k)	eMule
X.205.184.61	224k	(3)	Qihoo360	X.197.150.123	211k	(134)	Qihoo360
X.9.233.27	216k	(6)	Qihoo360	X.83.230.87	209k	(11k)	eMule
X.86.0.1	216k	(1k)	BitTorrent	X.195.138.123	204k	(73)	Qihoo360
X.124.94.218	200k	(4)	Qihoo360	X.112.177.163	201k	(370)	Qihoo360

Table 5.6. Phenomena associated with IP hotspots receiving traffic from 100 /24 blocks in `UCSD-12` and `UCSD-13`. Qihoo 360 and other P2P activity (indicated with a *) cause many IP hotspots.

Payload	UCSD-12		UCSD-13	
Qihoo 360*	63	(84%)	98	(90%)
eMule*	3	(4%)	2	(2%)
Unknown 13 Byte Encrypted	3	(4%)	0	(0%)
BitTorrent*	2	(3%)	1	(1%)
DNS Queries	0	(0%)	2	(2%)
Port 80 SYN	0	(0%)	2	(2%)
Sality*	1	(1%)	1	(1%)
ZeroAccess*	1	(1%)	0	(0%)
Port 3906 SYN	1	(1%)	0	(0%)
Steam	0	(0%)	1	(1%)
NetBIOS	0	(0%)	1	(1%)
Multiple	1: Port 51536 SYN, eMule	(1%)	1: eMule, Mythware	(1%)

a misdirected amplification attack using 20k Quake servers.

Hotspots occur for varying reasons. However, we can attribute the cause of most hotspots to a single phenomenon, typically a type of P2P activity. We show the breakdown of IP hotspots attracting over 100k /24 blocks in Table 5.6.

Two IP hotspots in Table 5.6, X.70.0.0 in 2012 and X.0.0.0 in 2013, reached the list due to multiple IBR components. Both of these addresses correspond to the first address of a /16 block. This suggests that certain addresses will receive more unsolicited traffic just by the numeric properties of the address. Several IP hotspots identified by Wustrow *et al.* also exhibited patterns (e.g., 1.1.1.1 and 1.2.3.4) [217].

Most hotspots do not persist over a period of year. Only seventeen hotspots attract over 100k /24 blocks in both the 2012 and 2013 datasets. Sixteen are Qihoo 360 hotspots and the remaining address is a Sality hotspot. Since reversing the bytes of the Qihoo 360 hotspots reveals another host running Qihoo 360 software (Section 4.3.1), it is likely that the reversed address is assigned to a host for over a year (e.g., statically assigned).

We conduct similar analysis for larger aggregations of darknet IP addresses. Encapsulated IPv6 traffic and BitTorrent traffic cause the largest subnet hotspots; and, as

shown in Figure 5.10, Conficker does not target IP addresses in UCSD.128.0.0/9. Since these subnet hotspots are the result of on-going, longterm phenomena, they exhibit slow growth in cumulative number of /24 blocks observed.

UCSD-NT vs MERIT-NT

Wustrow *et al.* [217] find significant non-uniformity in the number of bytes and packets received by four /8 darknets in March 2010. However, we find more uniformity when considering the number of sources sending non-spoofed traffic to our /8 darknets. Intuitively, filtering out spoofed traffic removes some irregularities, and many IBR components target UCSD-NT and MERIT-NT with equal probability (e.g., scanning, backscatter, P2P misconfigurations).

We observe a similar number of /24 blocks through `partial-UCSD-13` and `MERIT-13` (2.65M and 2.76M respectively). Table 5.3, in Section 5.2.1, shows that `partial-UCSD-13` and `MERIT-13` also have a similar traffic composition. All components, except Gaming and Other, contribute approximately the same number of /24 blocks to each dataset. The Gaming difference can be explained by a misconfiguration: a single UCSD-NT IP observes 115k /24 blocks sending Steam traffic. In the Other category, 10 times as many /24 networks send TCP traffic destined to IP addresses matching {A.B.C.D | A=MERIT & C=13} than {A.B.C.D | A=UCSD & C=13}.

Additionally, many source /24 blocks send traffic to both UCSD-NT and MERIT-NT. The ∩UCSD-13 column of Table 5.3 shows the overlap—the number of /24 blocks observed in both `MERIT-13` and `UCSD-13` accounts for more than 84% of /24 blocks. We also observe an overlap of at least 49% in individual IBR components (Conficker produces the highest overlap, 99%) which implies that sources sending IBR likely target multiple /8 networks. Thus, it is likely that other portions of the address space receive packets from these sources.

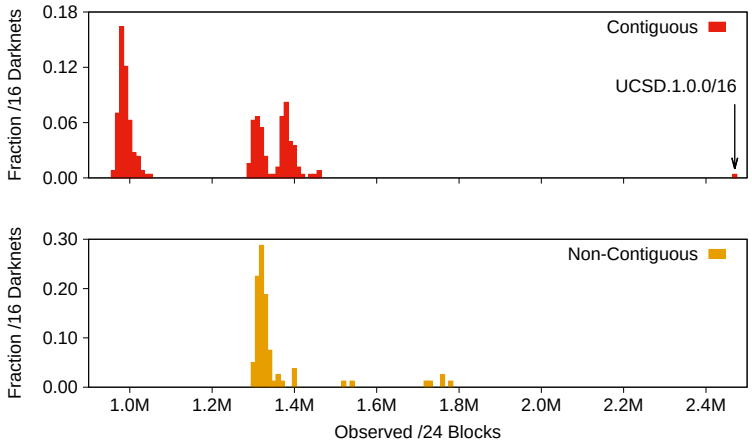


Figure 5.11. Effect of using non-contiguous darknets. The top graph shows the distribution of /24 blocks captured through the contiguous /16 darknets in `UCSD-13`. The bottom graph shows the distribution of /24 blocks captured from non-contiguous /16 darknets (constructed by randomly selecting dark /24 subnets of UCSD-NT). The non-contiguous darknets perform better than the worst-performing contiguous darknets, but never exceed the best contiguous /16 darknet which received over 2.5M /24 blocks.

However, we cannot examine all /8 darknets to understand the full effect of position. The non-uniform nature of IBR may cause variance when examining other darknets. Wustrow *et al.* find that many misconfigurations affect only the 1.0.0.0/8 block (e.g., traffic to 1.2.3.4) [217]; these misconfigurations may also influence the number of sources sending traffic to 1.0.0.0/8, in addition to bytes and packets. As we show in Section 5.3.2 sources often do not target all subnets within a /8 darknet.

Non-contiguous darknets

To test if a more distributed darknet would provide better coverage, we construct 80 non-contiguous /16 darknets by randomly selecting 256 /24 blocks within UCSD-

NT's addresses.[5,6] These non-contiguous /16 darknets observed an average of 1.36M /24 blocks, which is more than the average for contiguous /16 block from UCSD-NT (1.17M /24 blocks). In particular, in our samples, all the non-contiguous /16 blocks typically had better coverage than the median-performing contiguous /16 blocks.

Figure 5.11 compares the distribution of observed /24 blocks by all contiguous /16 darknets in `UCSD-13` to distribution from the 80 randomly selected non-contiguous /16 darknets. We see that for contiguous /16 blocks the distribution is multi-modal: one /16, UCSD.1.0.0/16, is an extreme outlier; targets of Conficker and BitTorrent traffic; targets of Conficker but not BitTorrent traffic; and targets of neither Conficker nor BitTorrent traffic. For the non-contiguous /16 distribution, the number of observed /24 blocks is almost always at least as much as the second smallest mode in the contiguous distribution. This mode corresponds to Conficker traffic: because Conficker-infected hosts send traffic to many networks, we can get similar coverage of the infected hosts with a subset of Conficker-targeted dark /24 blocks. There are five outliers in the non-contiguous distribution, each capturing over 1.7M /24 blocks; these non-contiguous darknets each contained at least one of the /24 blocks that caused UCSD.1.0.0/16 to be an extreme outlier in the contiguous distribution. The mode corresponding to BitTorrent traffic has a minimal effect on the non-contiguous distribution because BitTorrent hosts individually send traffic to only a handful of destinations; consequently, a large number of BitTorrent targets are required to observe significantly more /24 blocks.

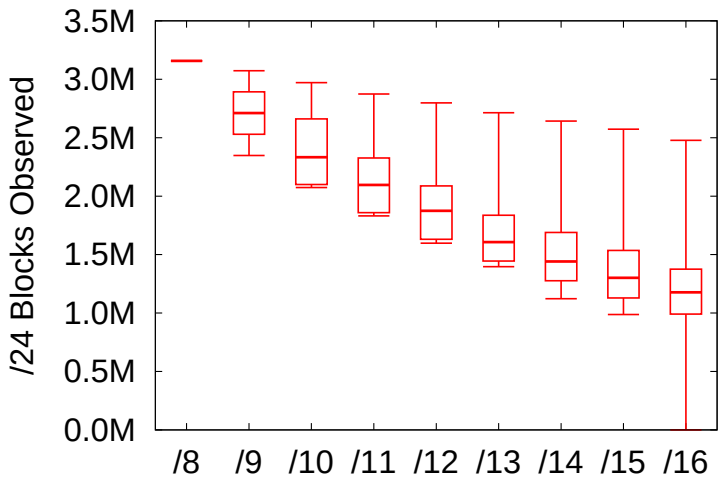


Figure 5.12. /24 blocks observed by contiguous blocks of the UCSD darknet. There is a power-law relationship between number of /24 blocks observed and size of the darknet.

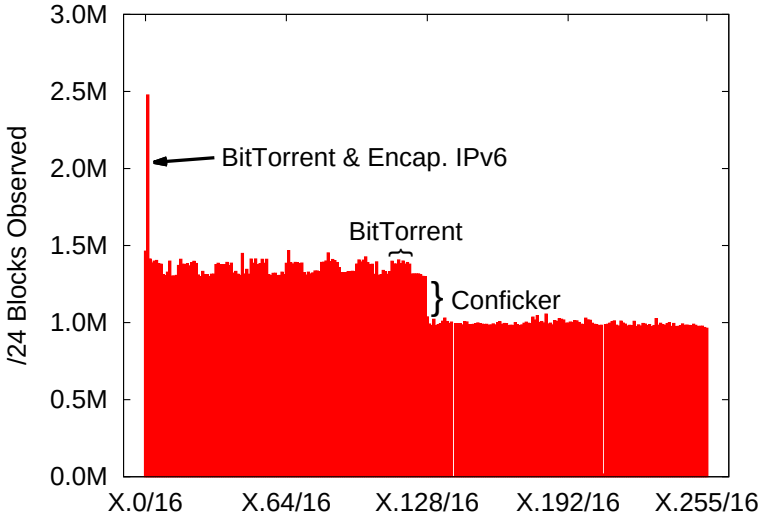


Figure 5.13. /24 blocks observed by each /16 in UCSD-NT. There is significant variance in the number of observed based on position within the UCSD darknet

5.3.2 Dependence on darknet size

With smaller darknets, we expect to observe fewer sources and observe those sources less frequently. To study the effect of using a smaller darknet, we vary darknet size, from a /16 to a /8, by considering contiguous subnets of UCSD-NT as their own mini-darknet. Figure 5.12 reports for each darknet size, the range of source /24 blocks captured by these contiguous subnets in `UCSD-13`. We find, due to the non-uniform nature of IBR, significant differences in the number of sources captured by subnets of the same size.

Despite these differences, based on median observations, the marginal utility of a single darknet IP address decreases as the size of the darknet increases (e.g., doubling the size of the darknet results in fewer than a 2x increase in the number of /24 blocks observed). In the /8 to /16 range, we observe a power-law relationship between the median number of /24 blocks observed, y, and the number of darknet IP addresses monitored x. Through linear regression we estimate the parameters to be:

$$y = 1.84 \times 10^5 \times x^{0.168}. \tag{5.1}$$

This relationship implies, in the /8 to /16 range of UCSD-NT, reducing darknet size by a factor of two should yield about 89% of the original /24 blocks. As a result, we expect small darknets to also observe many /24 blocks. But this power-law relationship does not hold for all darknet sizes: the median number of /24 blocks observed by an IP in `UCSD-13` is an order of magnitude less than the number implied by the power law

[5]There are two /16 blocks in UCSD-NT announced by entities other than UCSD, which we exclude in our non-contiguous experiment. The number of /24 blocks captured through these /16 blocks is non-zero due to routing properties, e.g., we receive traffic for these blocks when the routes are down.

[6]We chose 80 non-contiguous /16 blocks using a margin of error calculation [186]. Using the standard deviation from the contiguous /16 blocks, a sample size of 80 provides 95% confidence that the average number of /24 blocks observed by all similarly selected non-contiguous /16 darknets (from UCSD-NT's address space) is within 50k of the sample average (1.363M).

relationship projected by Equation 5.1.

For /16 or larger subnets of UCSD-NT, most variations can be attributed to: (1) the bug in Conficker's PRNG, (2) BitTorrent's RPC mechanism, KRPC, and (3) Encapsulated IPv6 traffic. We show the effect of these phenomena in Figure 5.13, which reports for each /16 within UCSD-NT the number of /24 blocks captured during *2013 census*. Individual IP hotspots are observed as little spikes in Figure 5.13, but create small discrepancies compared to the differences caused by the Conficker, BitTorrent and IPv6 components (for /16 or larger darknets).

5.4 Conclusion

In this Chapter, we have looked at the feasibility of using IBR to conduct Internet-wide opportunistic network inferences. We have found that on a whole, many IP addresses, /24 blocks, prefixes, ASes, and countries send IBR. Naturally, if we restrict our analysis to certain subsets of IBR — due either to the type traffic relevant to an inference or because of properties of the collection infrastructure — we observe fewer sources.

We have examined two aspects of the traffic type: the protocols used and the frequency in which we receive packets from a source. Many protocols comprise IBR, and each may contribute different types of inferences (e.g., BitTorrent has wide coverage, while scanning sends many packets). We continually receive IBR, though traffic analyses requiring continual observation seems feasible for large collections of IP addresses, such as the AS and country granularities.

The collection infrastructure includes which and how many IP addresses comprise the darknet. There are few differences in the quality of IBR collected from UCSD-NT and MERIT-NT's large network telescopes. Our analysis has shown, with smaller darknets, e.g., a /16, we still expect to observe many networks. However, there were considerable differences in coverage across the /16 subnets within UCSD-NT.

We feel that our data is representative of IBR, and that researchers using other darknets will experience similar results when using IBR to make Internet-wide inferences. We base this speculation on the minor differences in space when considering large darknets (`partial-UCSD-13` vs `MERIT-13`), as well as the widespread observation of the IBR components causing the largest discrepancies in time and space.

Acknowledgements

This chapter, in part, is adapted from material as it appears in the proceedings of the Internet Measurement Conference (IMC 2015). Benson, Karyn; Dainotti, Alberto; claffy, kc; Snoeren, Alex C; Kallitsis, Michael; ACM, 2015. The dissertation author was the primary investigator and author of this paper.

Chapter 6

Inferences with IBR: Using IBR to learn about address space usage

As discussed in Chapter 5, IBR originates from many hosts and networks. This bodes well for our goal of inferring network properties on an Internet-wide scale. However, our actual success depends on both the properties of IBR itself and the inference type. In Chapter 5, we provided intuition on the properties of IBR that influence our (in)ability to extract measurements, e.g., the frequency at which source IP addresses contact our darknets. In this Chapter and Chapter 7, we shift our focus to determining the types of inferences for which IBR is well-suited to provide insight. In this chapter, we consider inferences that reveal information about IPv4 address space usage. We break our analysis into four questions:

- Is an IP address or network used? I.e., does a machine use this IP address to communicate on the global Internet? (Section 6.1)
- How is the IP address or network used? E.g., web server, end host. (Section 6.2)
- What are basic attributes of Internet hosts or networks? E.g., uptime, installed software. (Section 6.3)
- How is the machine or network configured? E.g., using Network Address Trans-

lation (NAT). (Section 6.4)

Using many types of Internet data, including IBR, we directly answer the question "Is an IP address or network used?" [52, 53]. It is difficult to throughly answer the remaining questions because they are very broad. However, through a series of case studies, we can extract insight into the usability of IBR in answering these broad questions.

Researchers often use dedicated probing to study IPv4 address space usage [86, 97, 61, 145, 98]. We compare IBR's coverage to dedicated probing and remark on the differences between the datasets. Typically IBR provides less coverage, but can provide more complete results and reveal additional information about the host or network. For example, in our IPv4 address space utilization case study supplementing active probing with IBR exposes additional used /24 blocks. Moreover, the networks in the IBR dataset but not in active probing datasets likely filter unsolicited probes.

We find that it is not always straightforward apply existing techniques to IBR. There may be subtle differences between IBR and the live traffic used to develop a technique. For example, we find that SYN retransmits, which are more prevalent in IBR than live traffic, can induce false positives in p0f's NAT detection heuristics (Section 6.4.1). Additionally, some passive techniques are not sufficiently validated. With packets from hundreds of thousands of hosts, we find that a common method for inferring uptime is invalid for certain operating systems (Section 6.3.1). We encourage tool developers to use the large volume of traffic available in IBR to develop and refine passive techniques.

We caution that IBR sources may introduce bias into inferences about Internet-wide behavior. For example, we are hesitant to make inferences about BitTorrent client popularity since a 2012 index poisoning attack targeted specific content (Section 4.3.2). While we could add this information as a caveat to the data — other studies on BitTorrent client popularity depend on the swarms crawled [207] or which users downloaded a plugin [154] — we do not fully understand the phenomena resulting in IBR.

6.1 Inferring IPv4 address space utilization

For decades, researchers and administrators have anticipated that the number of devices connected to the Internet would be greater than the number of IPv4 addresses [185]. Despite changes to how IP addresses are allocated [72, 73], efforts to adopt IPv6 [216], and widespread NAT usage, IPv4 address space exhaustion remains a concern. Of the Regional Internet Registries (RIRs) that allocate IPv4 addresses, only AFRINIC has more than one /8 pool of addresses remaining [77]. Consequently, individuals are forced to acquire new IPv4 addresses through transfer markets [123], or use addresses designated for other geographic regions [187].

Researchers have extensively studied IPv4 address space exhaustion from the viewpoint of allocation and BGP-announced prefixes [171, 77, 90, 92, 46]. However, only one measurement effort, ISI's longstanding IP census [86], has tackled whether allocated addresses are actually used to communicate on the global Internet. ISI's IP census discovers used addresses by sending ICMP echo requests and analyzing the responses. Unfortunately, many hosts do not respond to ICMP echo requests [125], likely due to firewall policies (which may blacklist measurement infrastructure).

In a collaborative effort, we supplemented ISI's measurements by combining many different passive and active datasets [52, 53]. We performed our analysis on the /24 block granularity. We called a /24 block "used" if we observe at least one of its IP addresses in the header of a packet exchanged on the public Internet. We found 5.3M used /24 blocks by combining multiple datasets, which is 15.6% more than ISI discovered during the same time period [53].

In this Section, we highlight IBR's role in improving our understanding of IPv4 address space utilization.

6.1.1 Related work

Besides the aforementioned work in studying IPv4 address space exhaustion [77, 90, 92, 46, 86], projects with alternative goals have also identified used or unused IPv4 addresses blocks. Internet-wide scanning projects, including malicious scanning, typically aim to find vulnerable hosts (e.g., [98, 145, 61]), which are in used portions of the address space. At least two papers studied unused portions of the address space in the context of building better darknets or honeypots. Cooke *et al.* identified unused regions of a local network that could be used for passively monitoring unsolicited traffic [46]. Shinoda *et al.* evaluated the feasibility of malicious actors avoiding Internet threat monitoring systems [184].

Of particular relevance is Barford *et al.*'s work modeling malicious traffic from IBR [21]. Barford *et al.* reported the number of IP addresses sending IBR over seven days in 2004: 450k from the unused portions of two /16 blocks, and 2.4M from a /8 block. Consequently, these numbers are a darknet's view of IPv4 address space utilization in 2004. The authors did not generalize the results to comment on IPv4 address space utilization. This is an important distinction, as by itself IBR is likely to have biases (e.g., from malicious hosts). We limit the effect of dataset bias by combining IBR with multiple, diverse datasets.

Around the same time we published our first results identifying used portions of the address space, Zander *et al.* combined serveral passive and active datasets to estimate the total number of used IPv4 addresses and /24 blocks [222]. Zander *et al.* used a capture-recapture model to arrive at their estimate of about 6.2 to 6.3M used /24 blocks, which includes some /24 blocks that were unobserved in any of their datasets. The magnitude of /24 blocks in their estimate is consistent with our findings of 5.3M actually used /24 blocks.

Table 6.1. Census datasets. We infer used /24 blocks from passively collected traffic (UCSD-NT, SWITCH, IXP, R-ISP) and active probing (ISI, HTTP, ARK-TTL).

Dataset	Source type	Data format	Period
UCSD-NT [205]	Traffic: Darknet	full packet traces	July 23 to Aug. 25, 2013
SWITCH [193]	Traffic: Live Academic Net.	Netflow logs	July 23 to Aug. 25, 2013
IXP [6]	Traffic: IXP	sFlow packet samples	July 8 to July 28, Aug. 12 to Sept. 8, 2013
R-ISP [76]	Traffic: Residential ISP	Tstat[70] logs	July 1 to Sept. 31, 2013
ISI [2]	Active Probing: ICMP ping	logs	July 23 to Aug. 25, 2013
HTTP [145]	Active Probing: HTTP GET	logs	Oct. 29, 2013
ARK-TTL [93]	Active Probing: traceroute	logs	July to Sept., 2013

6.1.2 Methodology

The key insight of our approach is that, on their own, datasets individually provide partial information on whether or not IP address are used. However, collectively the datasets yield a more complete view of IPv4 address space utilization. Thus our methodology is to first identify diverse datasets (Section 6.1.3). Then, we validate that our methodology distinguishes between used and unused portions of the address space (Section 6.1.4). Next, we comment on IPv4 address space utilization based on the combined results (Section 6.1.5). Finally, for our passive datasets, we discuss our sensitivity to properties of the data and the collection methodology (Section 6.1.6).

6.1.3 Dataset selection

We combine diverse passive and active datasets, which we summarize in Table 6.1 and describe in detail in the remainder of this section.

Passive Measurements

In addition to traffic collected at UCSD-NT, we discover used /24 blocks with the following three vantage points. Each vantage points captures traffic data in a different format and thus requires a different approach of removing spoofed traffic, which we detail in our collaborative publications [52, 53].

SWITCH We collected unsampled NetFlow records from all the border routers of a national academic backbone network serving 46 single-homed universities and research institutes in Switzerland [193]. The monitored address range of SWITCH contains 2.2 million IP addresses, which correspond to a contiguous block slightly larger than a /11.

R-ISP We collected per-flow logs from a vantage point monitoring traffic of about 25,000 residential ADSL customers of a major European ISP [76]. The vantage point is instrumented to run Tstat, an open source passive traffic flow analyzer [70] that stores transport-level statistics of bidirectional flows.

IXP Our final passive vantage point is one of the largest Internet exchange points (IXPs) in the world, which is located in Europe, interconnects $O(100)$ networks, and exchanges more than 400 PB monthly [6]. We have access to randomly sampled (1 out of 16K) packets, capturing the first 128 bytes of each sampled Ethernet frame exchanged via the public switching infrastructure of this IXP.

Active Measurements

Many measurement projects send probes to the entire IPv4 address space. We use the results of three different measurement efforts in our analysis.

ISI We used the ISI Internet Census dataset *it55w-20130723* [2], obtained by probing the routed IPv4 address space with ICMP echo requests and retaining only those probes that received an ICMP echo reply from an address that matched the one probed (as recommended [96]). Note that the ISI Census experiment was designed to report at a /32 (host) rather than /24 (subnet) granularity, but we apply the resulting data set to a /24 granularity analysis.

HTTP We extracted IP addresses from logs of Project Sonar's HTTP (TCP port 80) scan of the entire IPv4 address space on October 29, 2013 [145].

ARK-TTL We processed ICMP traceroutes performed by CAIDA's Archipelago (Ark) to each /24 in the routed IPv4 address space between July and September 2013 [93]. Specifically, we extracted the ICMP Time Exceeded replies sent by hops along the traceroute path.

6.1.4 Validation

Before conducting an IP census leveraging many different data sources, we check that we can accurately distinguish between active and inactive areas of the IPv4 address space. From Section 3.2.6, where we validated our technique for removing spoofed traffic from IBR, we know that we rarely label unused addresses as used. For our other datasets, we describe our techniques to remove spoofed traffic in other papers [52, 53]. In this section, we assess our ability to identify used /24 blocks, and we evaluate the effectiveness of using multiple passive datasets to study IPv4 address space utilization.

We validate our methodology using datasets collected during the *2012 census*: `UCSD-12`, `MERIT-12` and `SWITCH-12`. These datasets respectively captured 3.14M, 2.98M, and 3.63M (unspoofed) /24 blocks.

Our assessment of `UCSD-12`, `MERIT-12` and `SWITCH-12` is based on limited ground truth data extracted from BGP and ISI data. We label as "inactive" all unrouted /24 blocks[1] (≈6.5M), assuming they should not appear in (unspoofed) traffic.[2]

[1] To establish a set of unrouted IPv4 address blocks, we take a list of BGP prefixes announced and captured by the *route-views2.routeviews.org* [206] collector between July 31 and September 2, 2012, and assume all other address blocks are unrouted.

[2] Manual analysis of darknet traffic revealed a few unrouted address blocks that seem to be used internally (but not globally advertised) by some organizations.

We label as "active" all the /24 blocks found responsive in the *it49c-20120731* ISI census dataset [3] (≈4.3M). This ground truth dataset accounts for about 60% of all IPv4 addresses. We are limited by our inability to classify hosts that do not respond to active probing. Though we compiled a large labeled dataset of several million prefixes, the inactive prefixes are based on information about *unrouted* networks, which are not representative of *routed but unused* networks in the Internet. Additionally, our active networks are based on destinations that respond to ISI Census probes, which may include border routers that respond on behalf of end hosts [52]; these will induce false positives in the ISI data and our labeled data set, which may induce underestimation of performance of our method.

With our ground truth data, we can infer for each dataset (`UCSD-12`, `MERIT-12`, `SWITCH-12`): true positives (tp), false positives (fp), true negatives (tn) and false negatives (fn). I.e., a tp is a /24 block observed in a passive dataset which contains a host that responded to an ISI probe. We evaluate each dataset's performance using four standard metrics:

- *Precision* $= \frac{tp}{tp+fp}$: fraction of positives that are true positives.
- *Recall* $= \frac{tp}{tp+fn}$: fraction of "active"-labeled networks correctly reported as active.
- *True negative rate* $= \frac{tn}{tn+fp}$: fraction of "inactive"-labeled networks correctly reported as inactive.[3]
- *Accuracy* $= \frac{tp+tn}{tp+tn+fp+fn}$: fraction of correctly classified positives and negatives.

Table 6.2 summarizes our validation results. These results use the ground truth dataset constructed from BGP and ISI, which only accounts for about 60% of all IPv4 addresses. Our technique performs well in terms of precision, true negative rate, and

[3]This metric is also known as specificity.

Table 6.2. Validation of passive census techniques. Using unrouted regions of the address space (inactive networks) and /24 blocks that responded to ICMP pings (active networks) we partially validate that our passive technique accurately differentiates between used and unused regions of the address space.

	Precision	Recall	True Negative Rate	Accuracy
UCSD-12	0.998	0.672	0.999	0.869
MERIT-12	0.999	0.645	0.999	0.859
SWITCH-12	0.999	0.756	0.999	0.903
Total	0.998	0.811	0.999	0.924

accuracy. High precision (the blocks we infer as used are actually used) and high true negative rate (identifying unrouted networks as unused) are unsurprising given that we developed our heuristics to remove spoofed traffic based on unrouted networks; however, it is reassuring that combining the datasets does not diminish the validity of our results. The lower values for recall show that our techniques do not capture all active /24 blocks, which is consistent with the fact that each of our measurement sources sees only a fraction (64.5% - 75.6%) of the labeled positives. Combining measurements from all three sources increases recall to 0.811. Finally, the last column of Table 6.2 shows that the overall accuracy, including negative and positive samples, is between 0.859 and 0.903 and improves when combining our three data sources to 0.924. Improved recall and accuracy with the aggregate data is promising for our method of combining multiple, diverse datasets.

6.1.5 Results of combining multiple datasets

Figure 6.1 shows a taxonomy of the IPv4 address space based on our collaborative effort [53]. We found that nearly a quarter of available addresses are "unrouted assigned," meaning that many organizations do not announce assigned prefixes in BGP. Of the /24 blocks announced in BGP, we find evidence that about half are actually used. This finding is consistent with Zander *et al.*'s estimate of 6.3M /24 used blocks [222].

IBR contributes to this study by identifying used /24 blocks. Table 6.3 shows that

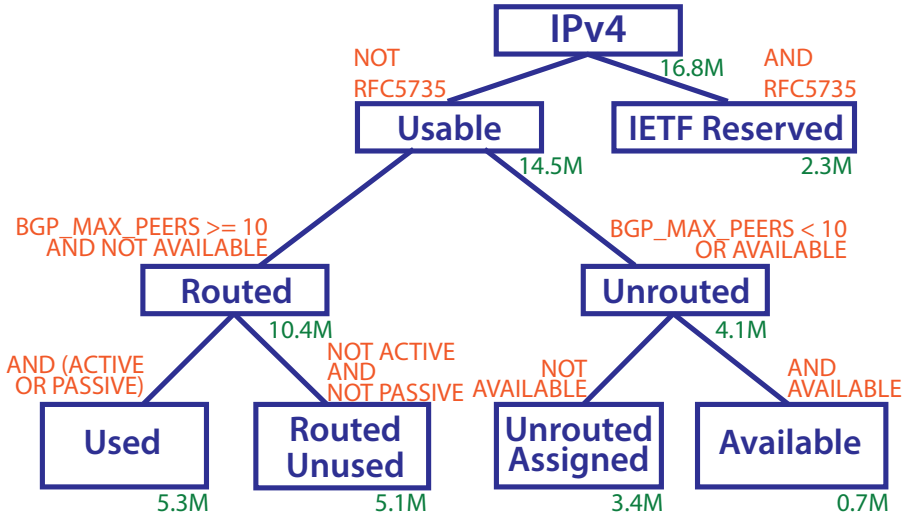


Figure 6.1. IPv4 address space taxonomy [53]. Our datasets, including IBR, help differentiate between "used" and "routed unused" addresses.

Table 6.3. Contributions to census by dataset. Individually, each of our datasets provides additional insight into address space utilization when combined with the state of the art (third column). Moreover, each dataset reveals /24 blocks unobserved through similar measurements (fourth column) and in all other datasets (fifth column).

Dataset	# /24s	# /24s not found with state of the art (ISI)	# Unique /24s within same family	# Unique /24s among active + passive
Active				
ISI	4,589,213	-	1,319,283	398,334
HTTP	3,161,064	207,578	189,831	76,189
ARK-TTL	1,627,363	58,021	40,284	24,533
All Active	4,837,056			
Passive				
SWITCH	3,599,380	350,132	147,220	54,905
UCSD-NT	3,149,944	241,676	61,443	24,134
R-ISP	3,797,273	361,539	176,721	59,278
IXP	3,090,645	345,062	195,328	55,155
All Passive	4,468,096			
Total	5,306,935			

the number of /24 blocks visible in IBR is less than ISI's census, but similar to other passive datasets and the HTTP dataset. If using only a single dataset to supplement the ISI data, our passive datasets provide more new /24 blocks than other forms of active probing (second column), highlighting the importance of dataset diversity. For all sources, most of the observed used /24 blocks are also observed with other datasets. However, IBR has the most overlap with other passive datasets (other passive datasets capture 98% of IBR-discovered /24 blocks) and collectively (combined, all other datasets capture 99.23% of IBR-discovered /24 blocks).

6.1.6 Sensitivity analysis

In Chapter 5 we analyzed how properties of IBR and the infrastructure used to collect IBR influenced our coverage. We conduct a similar analysis for other passive vantage points to determine the effect on our study of IPv4 address space utilization. We comment on how IBR compares to other forms of passive analysis.

Traffic characterization

Characterizing traffic at our vantage points assists with two objectives: (i) highlighting how the vantage point contributes to the census; and (ii) ensuring that traffic components specific to a vantage point do not skew our findings or make them not generally applicable. That is, to legitimately use passive traffic data for a census, we need to convince ourselves that a given vantage point is not observing a special set of /24 blocks.

In Section 4 we found that many sources send IBR to UCSD-NT due to misconfigurations and bugs in P2P software. In Section 5.2.1 we found most components had a low fraction of /24 blocks visible only through that component. In total, 80% of /24 blocks are visible through multiple IBR components, suggesting that no one component

Table 6.4. Effect of top attractors at each vantage point. The entries in the *Using Only Top n* columns show the percentage of /24 blocks (observed at that vantage point) that reach the *n* most popular destinations. The entries in the *Excluding Top n* columns show that many of the /24 blocks reaching the *n* most popular destinations are often observable by other monitored IPs.

n	Using Only Top *n*			Excluding Top *n*		
	UCSD	SWITCH	R-ISP	UCSD	SWITCH	R-ISP
1	9.8%	52.2%	31.2%	99.95%	96.7%	99.98%
10	27.9%	83.8%	63.6%	99.7%	89.7%	99.8%
100	42.8%	91.2%	82.2%	99.0%	84.9%	98.7%
1000	70.3%	96.9%	95.6%	97.5%	69.9%	89.7%
25%	98.9%	99.96%	99.7%	77.3%	13.1%	24.6%
50%	99.3%	99.99%	99.98%	63.6%	3.2%	1.5%
75%	99.9%	99.998%	99.998%	51.9%	0.5%	0.04%

skews our findings.

While HTTP and HTTPS account for 57.7% of the traffic volume, they contribute only 6.8% of the /24 blocks observed at R-ISP. Instead, the largest source of /24 blocks comes from client-to-client communication (e.g., P2P and VoIP). P2P is a key contributor, as 610k /24 blocks are only observable through P2P traffic. These 610k /24 blocks account for only 16% of the /24 blocks observed at R-ISP, implying that this large component is not vital to R-ISP's contribution.

SWITCH hosts popular services that serve content to many end users, including: a website hosting medical information (exchanging traffic with hosts in 1.8M /24 blocks), a SourceForge mirror, PlanetLab nodes, university web pages, and mail servers. These services attract large varying client populations. Compared to the UCSD-NT and R-ISP vantage points, SWITCH's value as a vantage point depends more on popular IP addresses.

IP addresses receive traffic from a varying number of sources, due to the content they host and the presence of IP hotspots. Table 6.4 quantifies the influence, per vantage point, of the *n* IPs that attract the most traffic (which we call "top attractors"). We report the percentage of /24 blocks that would be observed (at that vantage point) if we considered only the top *n* attractors, as well as the percentage of /24 blocks that would

be captured if the top *n* attractors were not part of the census. We find that many of the /24 blocks observed at the top attractors are also observed elsewhere, e.g., although the top attractor at SWITCH is sent traffic from 52.2% of all /24 blocks observed at SWITCH, without this IP, 96.7% of blocks would still be observed via other IP addresses in SWITCH. For SWITCH, the quick decrease in percentage of /24 blocks observable when excluding the top *n* attractors show that we heavily rely on the collection of top attractors at this vantage point. In the darknet, the slower decrease indicates we are less dependent on the top *n* attractors.

Vantage point size

We analyze vantage point size (the number of IP addresses monitored) to determine the extent to which our census results depend on access to large datasets. Unfortunately, the analysis of vantage point size is not straightforward due to the non-uniform nature of the monitored address space. Notwithstanding the extraordinary popularity of some IP addresses, as well as non-uniform assignment of hosts within an address subnet, we found an interesting correlation: for each vantage point, the median number of /24 blocks observed is roughly proportional to the log of the number of monitored IP addresses. Consistent with this observation, the utility of a monitored IP address declines as the size of the vantage point increases. While our census results benefit from large datasets, halving or doubling the size of our vantage points is unlikely to have a substantial impact on the number of /24 blocks we infer as used.

Duration of collection

Figure 6.2 shows sublinear but varied growth of the number of /24 blocks collected over time for our four vantage points. For all vantage points, a period of few (e.g., 10) days is enough to capture a the majority of the sources that are observed at each vantage point within the considered time frame. SWITCH, which initially captures the

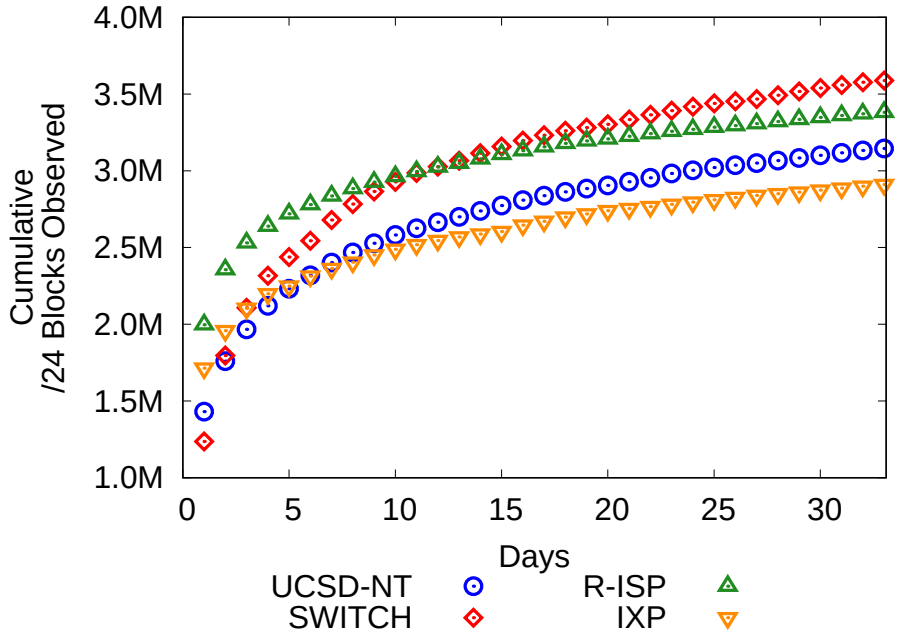


Figure 6.2. Cumulative /24 blocks observed. The cumulative number of /24 blocks observed grows sublinearly at each vantage point.

fewest /24 blocks, has the fastest growth rate; while the R-ISP and IXP vantage points capture more /24 blocks initially but they grow more slowly. Other factors that can influence inferences are strong changes in traffic composition, e.g., flash events. Our traffic datasets all had low (max 2%) standard deviation in the number of /24 blocks observed per week, with no abnormal events observed. However, when observing measurements from a broader time frame, we found evidence of flash events and changes in traffic. For example, in August 2012 (the year preceding our datasets), SWITCH web sites hosting content about shark protection experienced a sharp increase in visits (and thus observed /24 blocks); the Discovery Channel's Shark Week aired that month.

Time of collection

Figure 6.3 shows per-month sample measurements using our methodology over a period of two years. The SWITCH and IXP vantage points observed a similar number

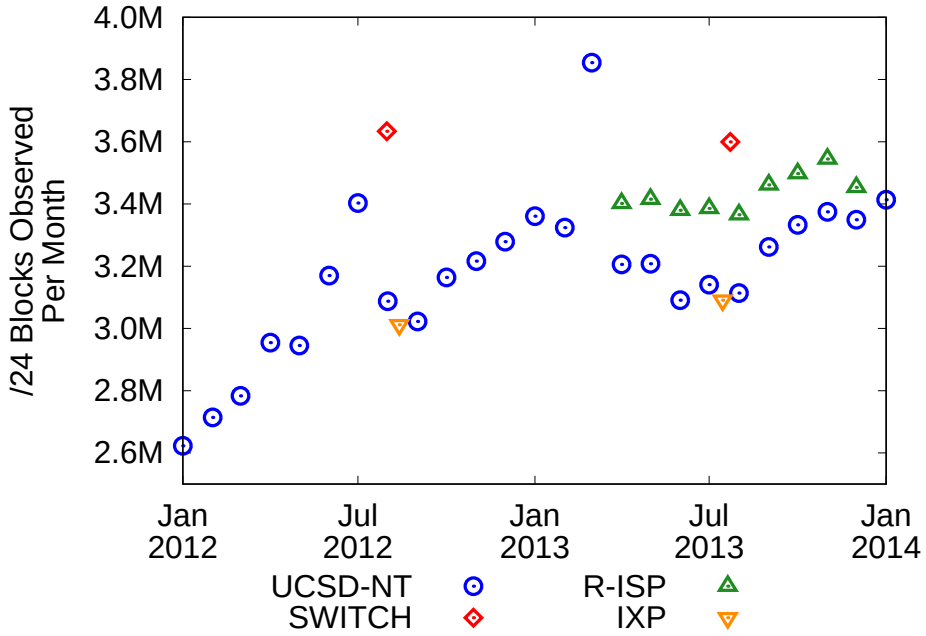


Figure 6.3. /24 blocks observed per month. In our data, taken over two years, every vantage point observed at least 2.6M /24 blocks per month. The fluctations in UCSD-NT data are the result of changes in the traffic components comprising IBR.

of /24 blocks approximately one year prior to our census. R-ISP consistently observed between 3.4M and 3.6M /24 blocks for nine consecutive months. At UCSD-NT, changes in the phenomena responsible for IBR resulted in increases in visible /24 blocks. Specifically, (i) in July 2012, there was an increase in BitTorrent traffic; (ii) in March 2013, there was a large increase in the darknet's backscatter category, likely related to the DDoS attacks on Spamhaus [162]. Such events may increase the number of /24 blocks inferred as used, but our technique does not appear to significantly depend on one-off events: in our data, every vantage point observed at least 2.6M /24 blocks per month.

6.1.7 Discussion

This case study has (i) shown that supplementing existing studies with IBR can improve coverage, and (ii) provided an in-depth of comparison of IBR to other data sources for the purposes of /24 block visibility.

Through this case study we have demonstrated that combining multiple data sources improves our understanding of IPv4 address space utilization over current state of the art (ICMP echo requests). Each dataset captured /24 blocks unobserved through other sources. IBR can be used to add diversity to measurement data: compared to using ICMP echo requests, adding one of our passive datasets would increase the number of used /24 blocks by at least 240k, whereas other our active datasets would result in smaller increases. While it was expected that ICMP echo requests would provide an incomplete view of used /24 blocks [125], it is promising that we can improve visibility with IBR (and other data sources). We are hopeful that other inferences made through active measurements can be supplemented with IBR.

We separately collected used /24 blocks from each dataset. However, active efforts to study IPv4 address space utilization can skip probing used /24 blocks identified through passive techniques. Such an effort would greatly reduce the number of measurement packets. For example, with `UCSD-13` the reduction is about 30%: active probing would only need to send packets to 7.3M /24 blocks instead of 10.4M /24 blocks.

Due to the collaborative nature of this work, we can compare the nature of UCSD-NT data to other passive data types. For all studied data sources, the benefit of using a bigger vantage point diminishes as the size of the vantage point increases. Similarly, for each analyzed data source, the benefit of using longer time periods diminishes as the duration of observation increases. Like large collections of clients (R-ISP), IBR observes many sources through P2P traffic; in contrast, SWITCH attracts traffic from many /24 blocks through its servers. IBR is less dependent on IP hotspots than SWITCH and R-ISP traffic, implying that, compared to IP addresses in live networks (where users and services are heterogeneous), IBR uniformly reaches darknet IP addresses.

Compared to the other datasets, IBR contributed on the low end of number of

unique /24 blocks during the *2013 census* (Table 6.3).[4] While this performance may result in researchers prioritizing other passive datasets over IBR, we believe IBR is still valuable to studies of address space utilization. IBR captured at least 2.6M /24 blocks per month over a two-year period, suggesting that IBR is likely to improve IPv4 address space utilization inferences regardless of the time of observation. Unlike the other data sources, IBR exhibits large fluctuations in observed /24 blocks over time. As a result, there are time periods where IBR is more valuable to utilization studies (e.g., if we repeated this study in July 2015, we would expect better coverage due to the increase in BitTorrent traffic described in Section 4.3.2).

6.2 Characterizing host functionality

In this section, we use IBR to infer how an IP address or network is used. IBR is a "one-stop shop." With IBR we can identify clients, servers, and routers. In comparison, dedicated probing typically only reveals targeted services or routing infrastructure.

We first compare the scale and quality of IBR results to traditional measurement techniques for extracting host functionality (Section 6.2.1). As expected, dedicated probing for servers and routers significantly outperforms IBR-based inferences; although, IBR can provide insight into which servers are being attacked or used maliciously. Scanning for client machines is more difficult. However, during the *2013 census*, passively collecting P2P traffic a residential ISP revealed a similar number of clients as IBR. As a result of the clients visible in IBR, we can combine IBR with traditional datasets to characterize host functionality for large network blocks (Section 6.2.2).

[4]The number of unique /24 blocks are not directly comparable across datasets due to the varying durations and times of collection.

Table 6.5. Number of servers, routers, and clients observed through IBR compared to other data sets.

Type	Traditional dataset	Date	Traditional /24s	IBR /24s	Traditional ∩ IBR
HTTP Server (Server)	Project Sonar [145]	*2013 census*	3,160k	52k	50k
Open Resolver (Server)	Open Resolver Project [152]	*2013 census* Jan. 20 – Mar. 1, 2014	2,520k	3.4k 454k	448k
Router	ARK-TTL [93]	*2013 census*	1,630k	133k	71k
P2P Users (Client)	R-ISP P2P	*2013 census* July 2015[a]	3,170k	2,490k 3,710k	

[a] BitTorrent traffic only.

6.2.1 Comparison to other data sources

Table 6.5 reports our comparison of the number of /24 blocks hosting HTTP servers, open DNS resolvers, routers, and P2P users. For servers (HTTP and open DNS resolvers), we observe significantly fewer /24 blocks, and nearly all the /24 blocks discovered through IBR are also discovered through dedicated probing. For routers, dedicated probing reveals an order of magnitude more /24 blocks; however, about half of the /24 blocks found in IBR were not discovered with active probing. For P2P traffic, we observe slightly fewer /24 blocks during the *2013 census* and slightly more /24 blocks after a large increase BitTorrent traffic in July 2015 (Section 4.3.2).

Locating HTTP servers

We identify HTTP servers through the backscatter component of IBR. We consider any source that sends UCSD-NT a TCP source port 80 packet with the SYN-ACK or RST flags set a HTTP server.[5] We compare our results to Project Sonar [145], which sends a HTTP GET request on TCP port 80 to all IPv4 addresses. We consider any host responding to the HTTP GET request to be a ground-truth HTTP server.

Project Sonar clearly outperforms IBR-based analysis in locating HTTP servers. Project Sonar data uncovers 60 times more /24 blocks hosting HTTP servers than our

[5] While ACK scans exist [128], darknets are rarely the target of these scans.

IBR-based inference; almost all /24 blocks we observe through IBR are also found by Project Sonar. We suspect the handful of /24 blocks we discover only through IBR are primarily due to a mismatch in the dates of analysis (we analyzed IBR data collected between July 23 and August 25, 2013; we used the Project Sonar scan from October 29, 2013). Moreover, IBR does not provide any insight into hosted web content. The response to the HTTP GET request includes headers such as when the content was modified and the type of server — information unavailable in IBR.

One benefit of using IBR over active scanning is that IBR may reveal which HTTP servers that are under attack [141]. Another potential benefit of IBR is its ability to provide insight in between scans. This benefit is diminishing. Recently, researchers created an architecture to collect and share frequent Zmap [63] scans of the IPv4 address space, including TCP port 80 scans [61].

Locating Open DNS Resolvers

As discussed in Section 4.2.2, the darknet receives packets from open DNS resolvers on UDP port 53 packets with the recursion-available bit set. Starting around February 2014, there was a significant increase in the number of open resolvers sending traffic to UCSD-NT and MERIT-NT. As a result, the number of IBR-visible /24 blocks with an open resolver increased from 3.4k to 454k.

We obtained the results of weekly scans for open resolvers conducted by The Open Resolver Project (ORP) [152] between January 26 and February 23, 2014. Compared to IBR collected during *2013 census*, the ORP data contains a factor of 750 more /24 blocks with open resolvers.[6] With the increase in open resolver traffic reaching UCSD-NT, ORP discovers 5.5 times the number of /24 blocks with open resolvers than

[6] In the six months between the *2013 census* and January/February 2014 (the date of the ORP data), the number of open resolvers decreased slightly [153]. Thus there are at least 750 times more open resolvers discovered through ORP than IBR.

IBR. This substantial increase in relative performance from IBR highlights how the changing composition of IBR can provide periods of increased visibility.

There are 6k /24 blocks visible through open resolver traffic to UCSD-NT that are unobserved in ORP data. This discrepancy is likely due to the continual monitoring through IBR versus the once-a-week probing by ORP. Like any Internet service, open resolvers may experience outages, or change IP addresses. Kührer *et al.* found that over 50% of open resolvers experience IP address churn in the first week [110]. Another possibility is that some ORP queries are not resolved: open resolvers may blacklist probes from ORP; an intermediate router's filtering policy may discard the ORP queries; some packets are lost (e.g., due to congestion).

Analysis of IBR reveals about a fifth of all /24 blocks that contain open resolvers. This is a sizable fraction of open resolvers to discover without scanning infrastructure. However, the biggest benefit of using darknets to identify open resolvers is the context of observing a source in IBR. Open resolvers visible through IBR are actively used in attacks — and should be the focus of clean up efforts (e.g., a campaign to alert administrators of NTP servers vulnerable to use in amplification attacks was highly successful [110]). Moreover, IBR associated with malicious activity may help researchers better understand attack vectors.

Locating routers

We observe ICMP messages generated by routers in IBR. ICMP time exceeded in transit messages appear in IBR in response to spoofed packets destined to live networks with small TTL values [161]. When spoofed packets to live networks have a sufficiently large TTL, but cannot reach the destination (e.g., the host or network is down, or filtering prohibits the communication), routers may generate a ICMP destina-

tion unreachable message [161, 34, 19].[7] In UCSD-13, we observed 107k /24 blocks as the result of ICMP host or network unreachable messages, 16k /24 blocks from ICMP time exceeded in transit messages, 13k /24 blocks because communication was administratively prohibited, and 2k /24 blocks from large packets reaching gateways.

We compare our results to the ARK-TTL dataset, which consists of routers that sent ICMP time exceeded messages during Ark's ICMP traceroute probes to all IPv4 destinations. With IBR, we observe an order of magnitude fewer /24 blocks with routers than ARK-TTL. Interestingly, only about half of the /24 blocks found in IBR also appear in the ARK-TTL dataset. There are three possible reasons for this discrepancy: First, the routers only visible in IBR may appear on paths not covered by Ark's probing. Second, some hosts (and presumably routers) will only respond to TCP or UDP probes [125]. Finally, we receive communication administratively prohibited messages whose source address is the same as the encapsulated packet.[8] These packets likely originate from home routers that share an external IP address with the clients in the home network; these routers were excluded in the ARK-TTL dataset.

Darknets are not the best source for enumerating IP addresses associated with routers. However, the ARK-TTL data missed about 50% of /24 blocks with routers observed in IBR, suggesting that ARK-TTL is also a partial enumeration of router interfaces.

Locating P2P users

Scanning the entire Internet for clients is difficult. Clients are often behind firewalls, which may prohibit external sources from initiating a connection with a host. Moreover, many client programs run on arbitrary ports (i.e., we cannot find all BitTorrent clients by scanning a single port). As a result, passive datasets are a reasonable

[7] We exclude ICMP destination unreachable messages with codes sent by hosts [161].

[8] ICMP encapsulates the packet that induced the transmission of the ICMP message.

method to discover regions of the address space used by clients.

For P2P traffic specifically, other researchers discovered BitTorrent clients by connecting to swarms [100], convincing users to install their extension [42], or crawling the DHT [190]. We also observe significant P2P activity through passive collection of both residential ISP traffic and IBR. We find a similar number of /24 blocks containing hosts sending P2P traffic in the R-ISP and UCSD-NT datasets during the *2013 census*.[9] As the result of an increase in BitTorrent traffic (Section 4.3.2), the number of /24 blocks originating P2P traffic in IBR increased by almost 50%, and surpasses the number of /24 blocks collected through BitTorrent at the residential ISP in 2013.

We have a list of all /24 blocks visible by the residential ISP, but not a breakdown by component. As a result, we cannot directly determine the /24 blocks discovered via P2P activity in both residential ISP and darknet traffic. However, of the 2.49k /24 blocks sending P2P traffic in `UCSD-13`, 2.46k are also visible in residential ISP traffic. This large overlap likely extends to the intersection of P2P activity in IBR and the residential ISP: of the extracted components from the residential ISP traffic, P2P activity contributed the most /24 blocks [53].

The magnitude of P2P users found in IBR suggests that darknet traffic is a good data source for identifying clients.

6.2.2 Combining with other data sources

In the previous section, we found that IBR reveals many /24 blocks used by clients. In this section, we combine these IBR-inferred client /24 blocks with inferences from active probing to classify the functionality of large network blocks. This information could be used to concentrate scanning efforts to certain types of hosts, and provide

[9] UCSD-NT data for the *2013 census* contains BitTorrent, eMule, and QQLive traffic [124, 150, 122]. BitTorrent traffic contributes the most /24 blocks (2.2M) to the *2013 census*. The residential ISP includes BitTorrent, eMule, ED2K, KAD, PPLive, SopCast, TVAnts, and PPStream traffic as identified by Tstat [70].

more specific inputs to scientific studies of Internet phenomena (e.g., it may be easier to classify ASes based on the roles of their hosts). Over time, this classification could provide detailed insight into Internet growth. To the best of our knowledge, this is the first map of the IPv4 address space to identify which regions function as clients.

Classification of /24 blocks

The main classification challenge is that some addresses function as clients, servers, and routing infrastructure. It is a common configuration to have a home router act at a NAT gateway for the rest of the home network [78]. In this case, clients share an external IP address with their router. Furthermore, some home routers are open resolvers [88] or host other services.

One option is to consider /24 blocks containing multiple inferred types as "unknown." Unfortunately, this method labels as "unknown" more than half of the used /24 blocks in the *2013 census*. Instead, we prioritize our inferences in the following order: clients, servers, routing infrastructure. Our reasoning is that while end users might host one-off services (or inadvertently have ports open), companies that host web content (e.g., Akamai, GoDaddy) will not exhibit characteristics of clients. Additionally, we expect routers to be interspersed with clients and servers (the .1 address of a BGP announced prefix is often a router [131]). We classify /24 blocks from the *2013 census* as follows:

- *unused:* We classify as unused: IETF reserved blocks, unrouted blocks, and routed unused blocks (unobserved by any dataset in our IPv4 utilization study).

- *client:* We construct a IBR-client dataset consisting of /24 blocks associated with P2P or Qihoo 360 traffic. These applications primarily run on clients.

- *server:* We designate any /24 block discovered by Project Sonar as a server. Fu-

ture versions of our analysis may include non-HTTP servers (mail, FTP, DNS, etc.). If a /24 block contains both clients and servers, we categorize the block as a client.

- *routing infrastructure:* We consider /24 blocks appearing the ARK-TTL dataset to be infrastructure. If a /24 block in the ARK-TTL contains clients or servers (as specified by the previous criteria) the client or server inference takes precedence.

- *used unclassified*: These /24 blocks are any that are found during the *2013 census* but do not appear in the HTTP, ARK-TTL or IBR-client datasets.

Results

Figure 6.4 shows the results of our classification. Certain regions of the address space are more likely to host clients (e.g., the bottom left quadrant) while other regions are more likely to host servers (e.g., many /8 blocks between 198.0.0.0/8 to 209.0.0.0/8). Inspection of several blocks verifies that the classification works as expected. Two of the most dense regions of servers, 23.0.0.0/8 and 54.0.0.0/8, correspond to Akamai and Amazon addresses. For a popular ISP, Time Warner Cable, we consider 75% of /24 blocks in its largest announced prefix (76.168.0.0/13) to be clients. Many of the regions with a high percentage of infrastructure /24 blocks are associated with ISPs (e.g., Cox, Vidéotron), which likely represent the IP addresses used by these ISPs for transit.

Furthermore, we can inspect how individual ASes use their address space. For example, the bottom right quadrant of the 98.0.0.0/8 block in Figure 6.4 (98.192.0.0/10) is a block assigned to Comcast that has many clients. Comcast is also assigned the top portion of the 50.0.0.0/8 block in Figure 6.4 (50.128.0.0/9), which has many server /24 blocks.[10] Many IP addresses in 50.128.0.0/9 have comcastbusiness.net domain names, while addresses in the 98.192.0.0/10 are comcast.net domain names.

[10]Comcast is also assigned 73.0.0.0/8, which is only observed as used through ISI data [52].

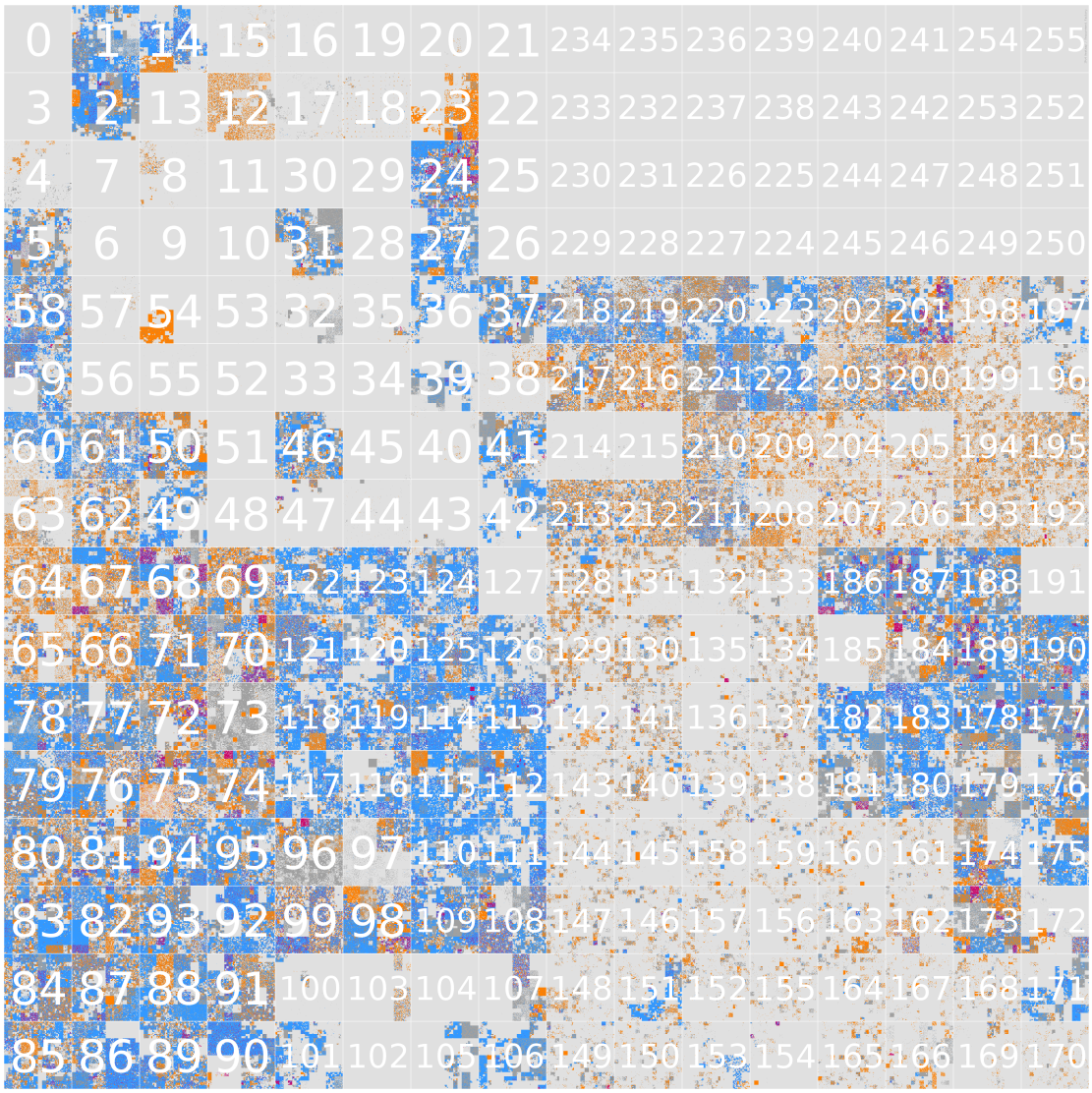


Figure 6.4. Hilbert curve of the IPv4 address space showing host functionality. *Light grey* indicates unused, *dark grey* used unclassified, *blue* client, *orange* server, *pink* routing infrastructure.

6.2.3 Discussion

This case study: (i) shows that the changing composition of IBR can provide an opportunity to learn about the Internet; (ii) exposes limitations in IBR's ability to determine the existence of network components; (iii) shows that IBR can supplement active probing techniques by providing additional information.

For finding (i), both the number of open resolvers and P2P users increased as a result of changes in IBR composition. The phenomena causing these changes in composition may cease, in which case IBR coverage of open resolvers or P2P users will decrease. This variability is partly due to a dependence on a specific application. However, end-users generate multiple IBR components. We can also use Qihoo 360 and certain botnet traffic (e.g., if it is known to primarily infect personal computers) to infer the client attribute.

For finding (ii), IBR is limited in the insight in can provide into servers and routing infrastructure. Dedicated probing clearly outperforms IBR in these categories. Moreover, we often obtain less information with IBR than active probing (e.g., we cannot analyze HTTP headers or web content) and at irregular intervals (e.g., we cannot determine the precise time a DNS server changes its configuration to no longer be an open resolver). For historical analysis, IBR could be used to provide information in the absence of active probing records; however, this scenario seems unlikely given the community's interest in continual probing projects [61, 145].

For finding (iii), IBR can supplement active probing in four ways. First, we capture some traffic from servers that are down during Internet-wide scans; though the number of such servers seems very low for HTTP servers, and somewhat low for open DNS servers. Second, observing a host in IBR provides additional context: we know that open resolvers appearing in IBR are used maliciously. Third, IBR can provide

hints as to when scans of the address space are not fully enumerating a resource. For routers, the high fraction of IBR-visible /24 blocks that are unobserved in traceroutes suggests that additional active probing may be necessary to fully enumerate Internet routing infrastructure. Finally, active probing is unlikely to enumerate a high number of clients, that are visible and identifiable in IBR. We used this insight to create a Hilbert curve of the IPv4 address space that indicates host functionality.

6.3 Extracting host attributes

In addition to determining a host's function we can also characterize the machine. For example, Kumar *et al.* determined the number of disks used by machines infected with the Witty worm [112]. With the current composition of IBR it is straightforward to collect attributes including: uptime (Section 6.3.1), used software (e.g., in Section 6.3.2 we examine BitTorrent client usage), and if the machine is patched (e.g., in Section 6.3.3 we infer the time to fix a software bug in Qihoo 360). Likely, further analysis of IBR will reveal additional attributes of hosts.

6.3.1 Determining uptime

We use IBR to infer the uptime of end hosts. Studying uptime can help understand human behavior [166], characterize availability [29], identify unpatched machines [22], and select resources with better availability (such as BitTorrent peers [36]).

Method

We use TCP timestamps to calculate uptime [134], a technique already implemented in Nmap [130] and p0f [221]. RFC 1323 specifies that TCP timestamps should be obtained from a clock that is approximately proportional to the real time [101]. Under the assumptions that (1) the OS zeros the counter at boot time, (2) the timestamp has

Table 6.6. Summary of the uptime validation results. We validate the TCP timestamps method for inferring uptime by checking the actual uptime on our machines and inspecting the distribution of uptimes for abnormalities.

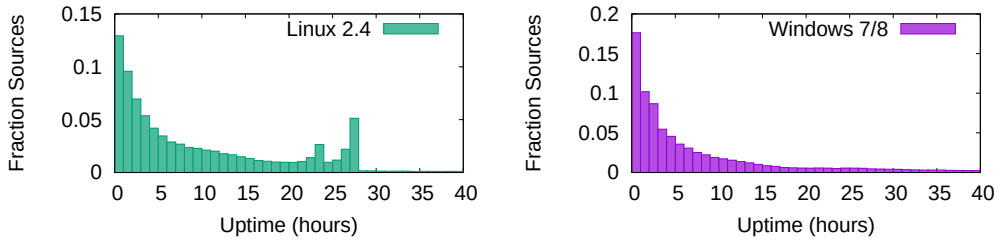
OS (from p0f)	# Srcs	Verified	Distribution	Include?
Linux 2.4.x	217,989		Wraps @27 hours	no
Windows 7 or 8	102,097	✓	70% up for less than 1 day	yes
Linux 3.x	52,200	✓	Longer uptimes less likely	yes
iOS iPhone/iPad	48,360	×	Most uptimes 3 to 13 days	no
Mac OS X 10.x	32,721	×	Most uptimes 3 to 13 days	no
Linux 2.2.x-3.x	28,034		Wraps @27 hours	no
FreeBSD	21,717	✓	Reboots for patch [30]	yes
Linux 2.6.x	17,290		Longer uptimes less likely	yes
Linux 2.4.x-2.6.x	14,800		Longer uptimes less likely	yes

not wrapped, and (3) network speeds are about constant, we can compute the frequency of the timestamp increments and total uptime. Specifically, for two packets j and k received at times r_j and r_k respectively with TCP timestamps t_j and t_k, the frequency of the timestamp increments is $f = \frac{t_k - t_j}{r_k - r_j}$, and the uptime (when packet k is sent) is $\frac{t_k}{f}$.

For each hour of data, we calculate frequency and uptime for each source IP sending TCP timestamps, and use p0f to determine the operating system that sent the packets. We then aggregate over all hours of data, excluding sources when either p0f reports conflicting OSes, or we determine that the OS violates assumption (1), or we receive packets that reveal conflicting uptimes (e.g., from two hosts behind a NAT). Additionally, we verify that the uptime is less than a year and that the frequency is close to a typically used value (e.g., one-third of IP addresses have a clock rate of 1000Hz) before including an IP address in our analysis.

Validation

To validate this technique, we analyze the accuracy of assumptions (1) and (2). Table 6.6 summarizes our findings in ensuring that the TCP timestamp is set to zero at boot time. First, we verify the accuracy of TCP timestamps on our own machines using Nmap and p0f. We found inconsistencies for iOS and Mac OS, and exclude IP addresses with these OSes from analysis. Additionally, we examine the distribution



(a) Linux 2.4.x **(b)** Windows 7 or 8

Figure 6.5. Wrapping of Linux 2.4.x TCP timestamp. We observe short uptimes for Linux 2.4.x and Windows 7/8 machines. The abrupt drop for Linux 2.4.x hosts at 27 hours is an artifact of wrapping timestamps. The Windows 7/8 hosts do not exhibit this behavior, and indicate actual short uptimes.

of uptimes in `UCSD-13` for each OS individually. We exclude two OSes, Linux 2.4.x and Linux 2.2.x-3.x, because the TCP timestamps appear to reset when the counter reaches 100M (at approximately 27 hours). We include Windows 7/8, which has a similar distribution from hour 0 to 24; but there is no evidence of a reset, implying that Windows 7/8 users frequently turn off their machines. Figure 6.5 depicts the difference in timestamp behavior between Linux 2.4.x and Windows 7/8 machines. It total, the TCP timestamp method accurately infers uptime for about 40% of the IP addresses we considered.

Another concern is that the TCP timestamp will wrap once it meets its maximal value. Less than 2% of timestamps we consider wrap more frequently than every 49 days. Since about 0.1% percent of hosts have an uptime of 49 days, which suggest the impact of a wrapping timestamp is minimal.

Results

In this section, we analyze the uptimes of machines associated with IP address that appear to originate from one host using a typical clock frequency and an operating system that resets the TCP timestamp counter to zero at boot time. In `UCSD-12`,

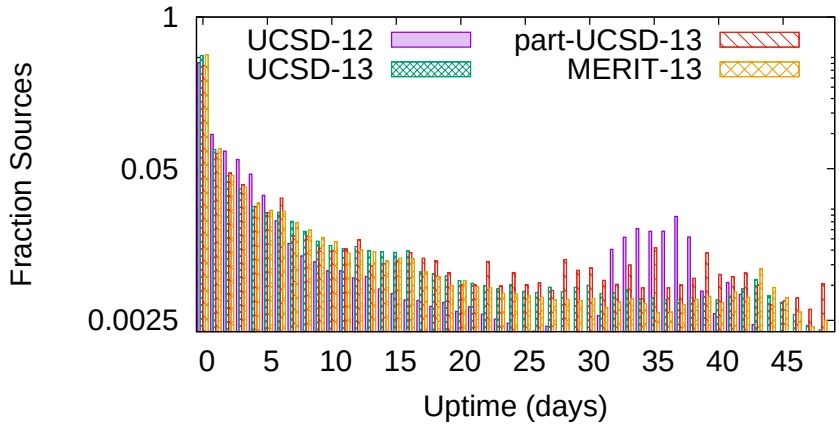
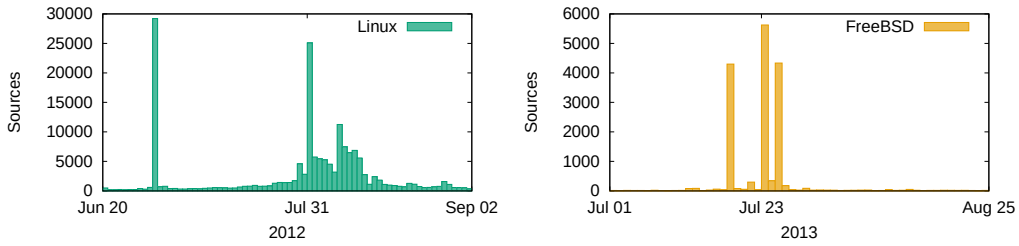


Figure 6.6. Distribution of days IP addresses with TCP timestamps. Most hosts have an uptime of one day or less. (Note the y axis uses the logarithmic scale.)

`UCSD-13`, `partial-UCSD-13`, and `MERIT-13`, we were able to infer uptimes associated with 290,697, 208,104, 57,990, and 47,122 IP addresses respectively. While these numbers represent less than 20% of IP addresses sending TCP timestamps, IBR still provides a large sample of uptimes. Both `partial-UCSD-13` and `MERIT-13` reveal significantly fewer uptimes than `UCSD-12` and `UCSD-13`, showing the influence of darknet size and temporal fluctuations (Sections 6.1.6).

Despite the differences in coverage, the datasets provide a consistent picture of uptime. Figure 6.6 shows that most hosts have short uptimes in all datasets, and a significant fraction have an uptime of less than 1 day. For the next three weeks, the fraction of up hosts decays exponentially, consistent with a constant probability of being turned off or rebooted.

We observe many hosts with an uptime of about 35 days for `UCSD-12`, caused by hosts running Linux. This spike in the data suggests that an external event may have caused a reboot for many Linux machines. Such events are hard to identify with Figure 6.6, as we are calculating the longest known uptime. Instead, Figure 6.7 shows the distribution of the reboot date for Linux hosts in `UCSD-12` and FreeBSD hosts in



(a) Linux: UCSD-12 **(b)** FreeBSD: UCSD-13

Figure 6.7. Distribution of reboot date. Spikes in the distribution of reboot date can be caused by bugs (Linux: July 1, 2012), patches (Linux: July 31, 2012; FreeBSD: July 23, 2013), and individual network behavior (FreeBSD: July 18, 2013).

UCSD-13. The large spikes likely correspond to: the addition of a leap second that caused many Linux machines to crash [135] (July 1, 2012); a Linux patch requiring a reboot [94] (July 31, 2012); FreeBSD machines belonging to a single company (Earthlink) (July 18, 2013); and a FreeBSD patch requiring a reboot [30] (July 23, 2013).

Discussion

This case study exemplifies the following findings: (i) IBR can provide insights into host behavior, which are likely unavailable through other data sources, (ii) IBR can provide a large sample of traffic to test inference techniques; (iii) and techniques using transport layer information are preferable to application-layer techniques.

The main benefit of using IBR to infer uptime is the diversity in end hosts analyzed. To the best of our knowledge, this is the first study to provide an Internet-wide analysis of uptime. Nmap [130] and p0f [221] both use the TCP timestamp technique, but are limited in the sources they can evaluate. Active probing (Nmap) will not reach end hosts behind a firewall or NAT, whereas passive observation (p0f) will be biased based on the population observed.

As discussed in the validation of the technique, some operating systems do not satisfy the assumptions necessary to apply the TCP timestamp method for inferring

uptime. The previous method of validation involved asking individuals to confirm the technique with their own machines [134]. Our method of examining uptime distribution is both preferable since it uses observations from many machines and easy to repeat when new operating systems are released.

Kumar et al. examined IBR from the Witty worm to extract host uptimes. However, since Witty targeted certain network security products, the number of networks they could analyze was limited (inferring uptime for only about 800 machines) and not diverse (about a quarter of the machines were from only two institutions) [112]. Inferring properties from information extracted at the transport layer of IBR expands our coverage.

6.3.2 Assessing BitTorrent client popularity

Many BitTorrent clients are erroneously directed to the darknet when attempting to torrent files. Both uTP [150] and KRPC [124] packets contain client identifying information. uTP packets indicate the client in the handshake messages. Libtorrent, the open source implementation of BitTorrent's DHT, implements an extension where each KRPC packet includes the client and version [149]; thus, with KRPC get_peers packets, we can either extract the client or the fact that it is not a Libtorrent-based client. We compare the breakdown of clients in IBR to previous studies [207, 154] in Table 6.7.

uTorrent is the most popular BitTorrent client in all datasets represented Table 6.7. However, we are hesitant to trust the darknet-based inferences. The percentage of uTorrent clients is much larger in the uTP data compared to previous studies [154, 207]. The inferences with darknet-based KRPC traffic yield a percentage of uTorrent clients consistent with the previous studies [154, 207], but there is an unknown client associated with 10.7% of IP addresses sending KRPC packets. One researcher hypothesized that this client could be associated with malware, private torrents, or software

Table 6.7. Popularity of BitTorrent clients. We show the percentage of source IP addresses sending handshake messages with the top BitTorrent clients in UCSD-12 and UCSD-13. We also show results from crawling BitTorrent swarms [207] and as observed through users of a Vuze plugin [154].

Client	Vuze Plugin [154] (2009)	Crawling BitTorrent Swarms [207] (2011)	UCSD-12 uTP	UCSD-13 uTP	UCSD-13 KRPC
BitComet	5.29%	1.01%	0.0012%	0.0007%	N/A
BitTorrent Mainline	9.28%	13.0 %	9.15 %	11.3 %	N/A
libtorrent	–	1.02%	0.0012%	0.498 %	4.78 %
qBittorrent	–	–	0.491 %	0.347 %	N/A
Transmission	2.68%	7.00%	2.11 %	2.21 %	0.241 %
uTorrent	50.6 %	48.0 %	84.1 %	80.7 %	53.8 %
uTorrent Mac	–	–	2.86 %	3.32 %	0.0436%
Vuze	22.5 %	22.5 %	1.38 %	0.924 %	N/A
Zo (unknown client)	–	–	–	–	10.7 %
Non-libtorrent	–	–	–	–	32.7 %

using BitTorrent as a P2P discovery system [211].

In Section 4.3.2, we provided evidence that some BitTorrent KRPC messages resulted from index poisoning attacks. Although uTP packets do not appear to have the same PRNG bug indicative of the index poisoning attack, they could stem from a different attack or bug. Consequently, our results may be biased due to the content targeted in an attack, or the implementation by the client (e.g., how often the client checks for stale peers). In general, it is hard to get an unbiased sample of BitTorrent clients. The 2011 study crawled BitTorrent swarms from English language sites [207], and likely overrepresents English speaking users. The 2009 study analyzed the peers of users of a Vuze plugin while downloading content [154], and may be biased towards populations that use Vuze and the plugin.

6.3.3 Time to patch the Qihoo 360 bug

In this case study, we analyze a change in application layer behavior. Specifically, we study how long it takes for Qihoo to patch a bug described in Section 4.3.1. We notified Qihoo of a reverse byte-order bug in their P2P update process, which they verified on January 5, 2016; Qihoo stated that they would start pushing out the fix the following week. In total, it took about one month from when we notified Qihoo to observe

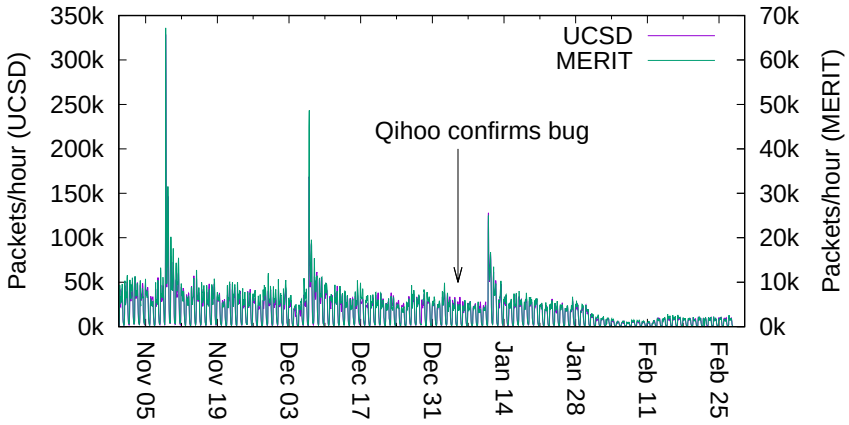


Figure 6.8. Time to fix Qihoo bug. Qihoo confirmed the byte order bug on January 5, 2015 and said that they would start to push fixes the following week. It took a month before we observed significantly fewer sources per hour sending Qihoo traffic to UCSD-NT and MERIT-NT.

a decrease in traffic resulting from this bug.

Unfortunately, Qihoo packets observed in darknets do not have client identifying information. However, we can examine the total number of packets from affected machines reaching UCSD-NT and MERIT-NT (Figure 6.8).[11] We look for a sustained decrease in packets, since patched machines should not send any traffic to our darknets.

Figure 6.8 shows that after notifying Qihoo there is a large spike in traffic on January 13, 2016.[12] These spikes occur every four to five weeks (on Wednesdays), and are likely the result of an automatic update pushed to many hosts. However, there is not a sustained decrease in packets immediately following the spike. So, it is unlikely that

[11] We look at both UCSD-NT and MERIT-NT to ensure that Qihoo's update mechanism not result in an abnormal amount of traffic reaching either darknet. Despite different magnitudes of traffic reaching UCSD-NT and MERIT-NT the relative number of packets reaching each darknet is the same.

[12] Although the spike on January 13, 2016 is smaller than the previous two spikes it is about the same magnitude of other spikes in 2015.

the update on January 13, 2016 included the bug fix.

The volume of Qihoo packets reaching the darknets per hour is not substantially smaller until early February 2016. According to Qihoo's version history, a new version of the product was released on February 1, 2016 [209]. It is probable that this new version fixed the byte-order bug. Although, Qihoo updates both automatically and by prompting the user [4], we hypothesize that an automatic update distributed the patch (manual updates generally have very slow patch rates [118, 192]). Since the decrease was not immediate, the patch was likely pushed to a small percentage of Qihoo users and then propagated via the P2P network.

We could easily extend this case study to examine the time-to-patch for individual networks. Though without client identifying information, we cannot study the host-level patch rate as we cannot quantify the effects of DHCP and NAT.

6.3.4 Discussion

In our experience, extracting host attributes requires intimate knowledge of network protocols and why machines using these network protocols transmit IBR. We generally understand why Qihoo 360 traffic reaches our darknets; consequently, we could infer when a large portion of machines start using a patched version of the software. It is out of the scope of this dissertation to fully understand details of BitTorrent client behavior as well as all index poisoning attacks; consequently, we could not characterize the biases IBR imposes on our study of BitTorrent client popularity. Even for well-defined protocols there may be differences in how individual operating systems or clients implement the protocol. For example, a common method to extract uptime only works for some operating systems as they predictably initalize TCP timestamp variables.

Our dependence on specific network protocols to extract host attributes may introduce bias. Although diverse hosts may send IBR, it is possible that only a subset send

the information required for our inferences (especially at the application-layer). We infer uptime, but for only certain operating systems; we infer BitTorrent client usage, but certain clients may be more susceptible to index poisoning attacks that generate IBR; we infer patching of Qihoo 360 software, but more security-conscious users may have turned off the feature that causes hosts to send IBR.

6.4 Inferring network configurations

In this section, we use IBR to gain insight on how networks, Internet-wide, are configured. The following case studies reveal how several ASes share IP addresses among their subscribers. In Section 6.4.1, we apply p0f's NAT detector to IBR. Unfortunately, Internet-wide measurement is not p0f's intended usage, and modifications are necessary to use this tool at scale. In Section 6.4.2, we focus our attention on a specific type of NAT, CGN. Although individual client data can be inconsistent with our expectations for a single machine, we can successfully identify CGN by widespread evidence of IP address sharing throughout an AS. We shift gears in Section 6.4.3, where we examine DHCP usage and corroborate Padmanabhan *et al.*'s findings on characteristic address durations for nine ASes.

6.4.1 Detecting NAT usage

Network address translation (NAT) is a technology that maps IP addresses in one network to IP addresses in another network. A common configuration is to map IP addresses in a private network, typically in the ranges defined in RFC 1918 [170], to globally routable IP addresses. By mapping multiple internal IP addresses to the same external IP address, a network can provide connectivity for more hosts than the number of globally routable addresses it owns. NAT provides security benefits (e.g., by filtering externally initiated connection [188]) and increases the flexibility that operators

have within their own networks (e.g., internal IPv6 hosts can communicate with external IPv4 hosts [15]) However, NAT can also be seen as violating the end-to-end principle and slowing the rate of IPv6 adoption [84].

Research analyzing NAT deployments tries to answer two main questions: (1) how many users share an IP address?; and (2) how widespread is NAT deployment? Work by Bellovin, and Beverly analyzed question (1) in an attempt to estimate the number of Internet hosts [23, 26]. For question (2), Maier *et al.* determined that, for a single ISP, 90% of DSL lines use NAT, and at least 10% have multiple hosts that are active at the same time [132]. Armitage conducted a study of Internet-wide NAT deployment using traffic reaching Quake servers [13]; however, this study was biased towards populations using Quake, and is almost 15 years old. Using Javascript to extract an internal address, Casado *et al.* discovered nearly 450k networks using NAT, and less than 0.03% had more than 10 hosts [38]; however this study is biased towards populations downloading content in the CoralCDN, and is almost 10 years old.

While there is limited academic research analyzing the extent of NAT Internet-wide, one freely available tool, p0f [221], includes NAT detection heuristics that can be used on any passively collected dataset. Each heuristic produces a score based on the current packet and previous packet from a host. To infer NAT multiple sets of packets must produce non-zero scores, which eliminates DHCP as a cause of the changes over short time scales.

Running p0f's NAT detector on IBR

We were hopeful that we could apply p0f's NAT detector directly to IBR. However, to the best of our knowledge, no formal evaluation of p0f's heuristics exists. Our examination with IBR finds three heuristics yield false positives. These heuristics generally produce low scores, but scanning sources can trigger the heuristics at least the 4 to

```
17:00:21.031492 IP 79.177.9.244.51317 > XX.246.83.27.21: Flags [S], seq 833462406, win 8192,
                                        options [mss 1360,nop,wscale 8,nop,nop,sackOK], length 0
    0x0000:  4500 0034 1ab2 4000 6f06 ---- 4fb1 09f4  E..4..@.o.......
    0x0010:  XXf6 531b c875 0015 31ad a086 0000 0000  ,.S..u..1.......
    0x0020:  8002 2000 daff 0000 0204 0550 0103 0308  ...........P....
    0x0030:  0101 0402                                ....
17:00:24.035639 IP 79.177.9.244.51317 > XX.246.83.27.21: Flags [S], seq 833462406, win 8192,
                                        options [mss 1360,nop,wscale 8,nop,nop,sackOK], length 0
    0x0000:  4500 0034 1ab3 4000 6f06 ---- 4fb1 09f4  E..4..@.o.......
    0x0010:  XXf6 531b c875 0015 31ad a086 0000 0000  ,.S..u..1.......
    0x0020:  8002 2000 daff 0000 0204 0550 0103 0308  ...........P....
    0x0030:  0101 0402                                ....
17:00:30.036256 IP 79.177.9.244.51317 > XX.246.83.27.21: Flags [S], seq 833462406, win 8192,
                                        options [mss 1360,nop,nop,sackOK], length 0
    0x0000:  4500 0030 1ab4 4000 6f06 ---- 4fb1 09f4  E..0..@.o.......
    0x0010:  XXf6 531b c875 0015 31ad a086 0000 0000  ,.S..u..1.......
    0x0020:  7002 2000 ef0e 0000 0204 0550 0101 0402  p.........P....
```

Figure 6.9. Example of packets stream where second retransmit differs. These packets appear to come from the same machine: same TCP sequence number, consecutive IPID values, typical exponential backoff behavior (3 seconds, 6 seconds). However, the third packet is shorter because it does not have the TCP window scale option.

8 times required to infer NAT. Subtle differences between IBR and live network traffic, an updated best practice, and incomplete data collected by p0f cause these issues.

In live networks, many communication attempts are successful, whereas in darknets all communication attempts are unsuccessful. As a result, IBR typically captures a significant number of TCP retransmits. There appears to be a quirk in TCP retransmits for certain Windows machines. As we show in Figure 6.9, the final retransmit excludes the TCP window scale option producing a smaller packet than first two packets in the communication attempt. p0f generally runs NAT detection on the first packet of a communication attempt; however, in packet loss situations, we may receive only the third packet in the sequence. These differences flag a source as using NAT since the initial SYN packets differ in their options. We suspect that p0f does not account for this quirk because it was developed using traffic targeting live networks, without many TCP retransmits.

p0f flags an IP address as potentially using NAT if the ephemeral (source) port decreases significantly in consecutive connection attempts. Like the IPID (discussed

in Section 3.1.1), some OSes select ephemeral ports by incrementing a counter. However, for security reasons, RFC 6056 recommends that hosts randomize the selection of ephemeral ports [116]. Furthermore, Linux and several BSD flavors follow this guideline for TCP connections [109]. As a result, this heuristic produces false positives.

Analysis of TCP timestamps is a promising technique for detecting NAT. Internet hosts set the TCP timestamps according to an internal counter, which increases proportionally to wall-time (e.g., 1k Hz). Thus, given two or more packets from a host, we can determine if a third packet came from the same host. Unfortunately, p0f's implementation only keeps track of one previous TCP timestamp. With only one stored TCP timestamp, p0f is unable to determine the rate at which TCP timestamps increase. Instead, p0f decides if the TCP timestamp is indicative of NAT using an expected clock rate. Though the expected rate does correctly capture the behavior of most hosts we find in Section 6.3.1, about 2% of machines send packets at a rate that results in false positives.

Discussion

Our partial evaluation of p0f's NAT detection heuristics, revealed that three heuristics produce false positives. While these false positives may preclude the tool's usage for an Internet-wide study of NAT, the heuristics are reasonable suggestions for which networks are using NAT. It is fairly straightforward to manually determine NAT usage from a set of flows tagged by p0f. Manual analysis is reasonable for p0f's common use cases: "reconnaissance during penetration tests; routine network monitoring; detection of unauthorized network interconnects in corporate environments; providing signals for abuse-prevention tools; and miscellanous [sic] forensics [221]." In the future, we could use IBR analysis to help modify p0f's heuristics to produce fewer false positives.

6.4.2 Detecting carrier grade NAT (CGN)

In this section, we use IBR to detect a specific type of NAT: carrier grade NAT (CGN).[13] CGN is NAT managed by an ISP, where the subscribers have limited or no control over the NAT deployment [159]. CGN is one of the biggest hindrances to IPv6 adoption as many users access the Internet using the same external IPv4 address.

Data

Like previous work in NAT detection, a technique to detect CGN must be able to fingerprint individual machines to discern when many hosts are sharing a IP address. Though any traffic with unique machine identifiers could be used to detect CGN, we use BitTorrent traffic. Clients using BitTorrent's DHT generate a random 160-bit node ID [124]. Using 160 bits means that there is an extremely high probability that well-behaved clients will generate unique node IDs.

We analyze our ability to detect CGN in January 2015 and July 2015. Notably, in July 2015, there was a large increase in BitTorrent traffic (Section 4.3.2), which should allow us to detect the presence of CGN in more networks. The January and July datasets contain BitTorrent IDs from clients in 15.6k and 27.3k ASes respectively.

Technique for identifying CGN with IBR

We use the following criteria for detecting CGN with BitTorrent IBR:

1. *Many node IDs per IP address*

2. *Many IP addresses in a /24 block meet the previous criteria*

At a minimum, we need to observe many node IDs associated with a single IP address (Criterion 1). We do not expect every host behind a NAT device to have a

[13] CGN is also called Large Scale NAT (LSN).

BitTorrent client that sends IBR. However, BitTorrent is a popular protocol. So, if many hosts share an external IP address it is probable that multiple hosts will send traffic to our darknet.

Criterion 1 is not sufficient for detecting CGN for two reasons. First, some BitTorrent clients do not follow the expected behavior and use multiple IDs (e.g., they are part of a Sybil attack or frequently changes their node ID[14]). Second, we need to differentiate CGN from NAT deployed by the end user.

Criterion 2 helps eliminate one-off behaviors and differentiate between NAT deployed by home users versus the ISP. We assume that an ISP deploying NAT will use a contiguous block of IP addresses. Although we expect the number of node IDs per IP address to be much higher for CGNs than home NATs, there may be some individual hosts that have an abnormally large number of IDs (e.g., a coffee shop); the IP addresses of these hosts are likely dispersed throughout the AS's address space. Specifically, we require that multiple IP address in the same /24 block show evidence of NAT to identify CGN. This criterion also helps eliminate BitTorrent clients that use multiple IDs, as they too are likely distributed throughout the AS's address space.

Validation

To validate our methodology, we gathered a list of networks that deploy CGN, and a list of networks that do not deploy CGN. We include the results of a CAIDA survey, email confirmation, online resumès, reverse DNS names, and the results of active measurements [126]. Table 6.8 summarizes our ground truth data.

We then check that CGN networks have at least one /24 block meeting our criteria. We do not specify an exact number for Criteria 1 and 2 (and what constitutes "many" depends on the volume of BitTorrent traffic). However, we expect that CGN

[14]From manual inspection, many of the clients that appear to frequently change their node ID are using LibTorrent with the DHT security extension [148].

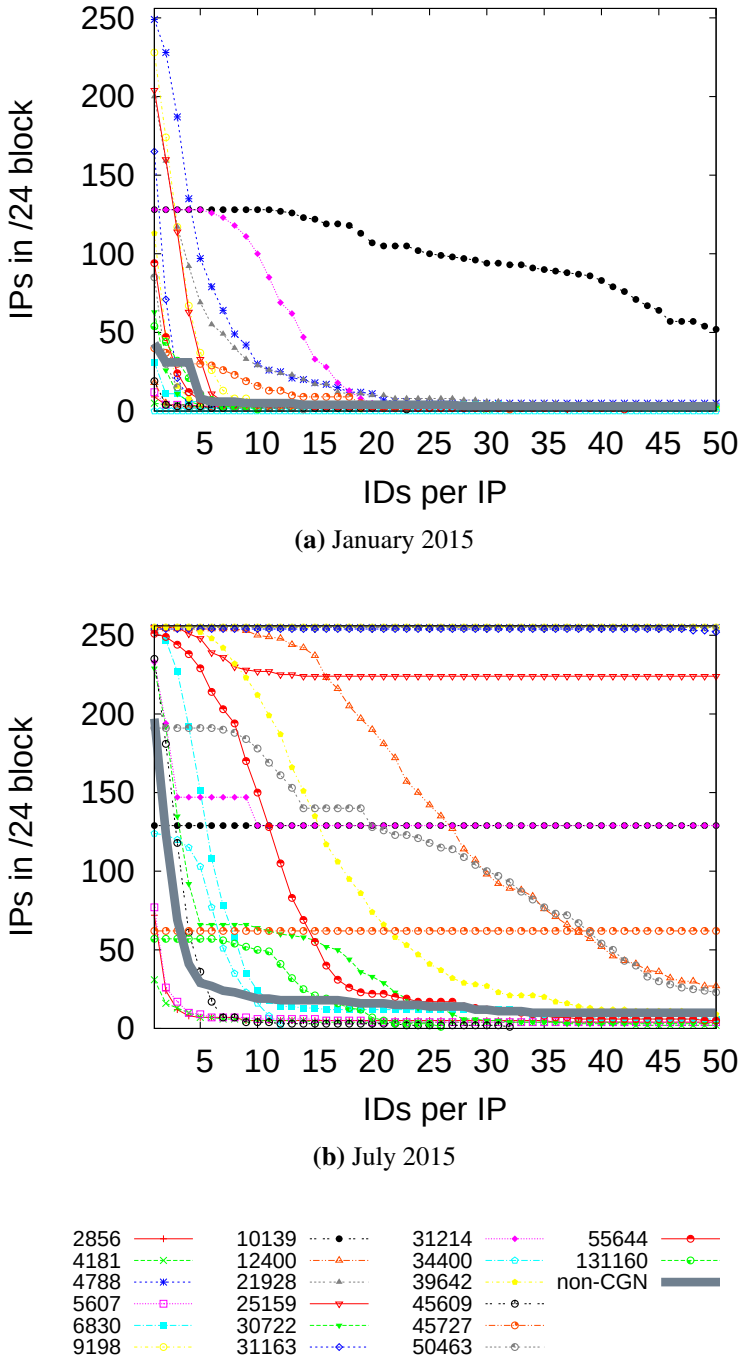
Table 6.8. CGN ground truth data.

CGN			Non-CGN		
AS	ASN	Method	AS	ASN	Method
British Telecommunications	2856	NATRevelio [126]	BELWUE	553	CAIDA survey
Malaysia Telecom	4788	Resumè	State of Oregon	1798	CAIDA survey
TDS TELECOM	4181	CAIDA survey	HUNGRARNET	1955	CAIDA survey
BSkyB	5607	Resumè	Orange S.A.[a]	3215	List Subscribers
Liberty Global Operations	6830	DNS names	Sandia National Laboratories	3562	CAIDA survey
Kazakhtelecom JSC	9198	Resumè	NYSERNet	4804	Email
Smart Broadband, Inc.	10139	CAIDA survey	Microplex PTY LTD	4804	List Subscribers
Viagénie	10566	Resumè	Spin SpA	6734	CAIDA survey
Partner Communications	12400	Resumè	UC San Diego	7377	
T-Mobile USA Inc.	21928	Web search	SEI Data	7871	CAIDA survey
PJSC MegaFon	25159	DNS names	Brasil Telecom S/A	8167	List Subscribers
Vodafone Omnitel B.V.	30722	NATRevelio [126]	Woosh Wireless	9737	Email
JSC MegaFon	31163	DNS names	IP-Only Networks	12552	CAIDA survey
TIS Dialog LLC	31214	DNS names	City West Cable	18988	CAIDA survey
Etihad Etisalat	34400	Resumè	AxisInternet	19104	CAIDA survey
Stofa A/S	39642	DNS names	Modesto Irrigation District	19621	CAIDA survey
Tech Mahindra	45432	Resumè	Bowdoin College	22847	CAIDA survey
Bharti Airtel Ltd.	45609	Resumè	Micronet Broadband (Pvt)	23674	Email
Hutchison CP	45727	Email	SafeNZ Networks LTD	24005	CAIDA survey
Triple C	50463	Resumè	Meanie	31019	CAIDA survey
Idea Cellular	55644	Resumè			
Empresa Brasileira	53128	NATRevelio [126]			
Wire and Wireless Co.	131160	Web search			

[a] We exclude the mobile portion of Orange's network.

networks will have more IP addresses with a high number of node IDs than non-CGN networks. Specifically, we consider all possible values of a "high number of node IDs" and find the /24 block in the non-CGN networks with the most IP addresses meeting this threshold. In Figure 6.10, we graph the maximum of the non-CGN networks as a thick solid line. We also graph (with thin dotted lines) the same relationship for each AS known to deploy CGN. In general, the number of node IDs per IP address is low in non-CGN networks and high in networks deploying CGN.

Figure 6.10b shows a large separation between the non-CGN threshold and many CGN networks during a high-volume scenario (July). In total, 17 of the 20 CGN ASes sending BitTorrent IBR have at least one /24 block that exceeds all non-CGN blocks for some values of Criteria 1 and 2. Figure 6.10b shows that our success is slightly worse in the low volume scenario (January). There are 14 ASes known to deploy CGN that exceed the non-CGN thresholds. All 14 ASes also exceeded the non-CGN thresholds in



(a) January 2015

(b) July 2015

Figure 6.10. Validation of CGN detection method. The *x*-axis corresponds to Criterion 1, and the *y*-axis corresponds to Criterion 2. Compared to non-CGN networks, most networks deploying CGN contain at least one /24 block where many IP addresses are associated with many BitTorrent node IDs. This is especially in the high volume scenario (July 2015).

July.

This method does not find all networks deploying CGN. Three ASes known to deploy CGN (10566, 45432, 53128) did not send any BitTorrent IBR to UCSD-NT in either January or July 2015. These ASes likely do not have many BitTorrent users. Furthermore, three ASes known to deploy CGN (2856, 5607, 4181) did not exceed the non-CGN threshold in any of our characterizations. All three ASes are large and likely deploy CGN only in a small portion of their network.

We can further increase our confidence by ensuring that we see interwoven node IDs, a continual stream of traffic from an IP address, or multiple BitTorrent clients. The former confidence measure helps eliminate the DHCP case where multiple subsequent BitTorrent clients using the same IP address; the later two confidence measures help eliminate the case of short-lived Sybil attacks. However, these confidence measures work best in high-volume situations. With sparse traffic we are unlikely to observe the same host multiple times (i.e., interwoven with other hosts) or client diversity (there are only a handful of popular clients).

In summary, our validation shows that CGN networks generally have a /24 block where many IP addresses have a large number of node IDs. In a high volume situation, we can increase our confidence by including additional criteria. Our method did not discover all of our ground truth networks, probably because they have a small population of BitTorrent users.

Findings

We find that a considerable number of ASes show evidence of CGN deployment, even in the low-volume scenario. There were 1,054 ASes in January and 2,930 ASes in July that exceeded the non-CGN thresholds from our validation step. In general, ASes behave consistently over the two months: about 85% of ASes above the non-CGN

thresholds in January were also above the non-CGN thresholds in July.

Interestingly, many of the ASes we find to have a CGN deployment also have regions of the address space that are unlikely to be part of a CGN deployment. A manual analysis of some of these regions yields that mobile networks frequently use CGN.

Geographically, many Eastern European ASes deploy CGN. Based on a whois lookup of the 2,930 ASNs meeting the non-interwoven CGN criteria for July 2015 about 75% (2212) were registered through RIPE NCC, followed by APNIC (418), LACNIC (115), AFRINIC (95) and ARIN (86). There were over one thousand ASes in Russia meeting the criteria, followed by Ukraine (531), India (100), Czech Republic (72), China (63) and the USA (59). The high number of ASes in Russia is partially due to the company Rostelecom, which had 70 ASes exceed the non-CGN threshold.

Discussion

This case study highlights (i) our ability to infer configuration information through traffic samples; and (ii) how changes in IBR can improve our ability to make inferences. Although we do not receive traffic from every Internet host, nor does every Internet host use BitTorrent, we are still able to infer CGN usage for about 1k-3k ASes. In particular, we identify many CGN networks in July 2015 due to the large increase in BitTorrent IBR. In lower traffic volume situations, we can still infer CGN usage, though our confidence decreases (as the extra criteria is non-applicable). We expect that other packet fields, less influenced by fluctuations in IBR, (e.g., TCP timestamps or IPID) could also reveal CGN.

Recently, using active measurements, Richter *et al.* found 421 ASes deploying CGN in an Internet-wide study of 3,166 ASes [172]. Although there is probably a lower confidence associated with the IBR data, IBR has wider visibility ($\approx$15.5k ASes sent BitTorrent IBR in our low volume month) and yields more ASes deploying CGN. Addi-

tionally, IBR seems better suited to provide a historical view of CGN deployment.[15] IBR could provide some insight into pooling behavior (i.e., do clients use the same external IP address in subsequent connections?). However, Richter *et al.*'s active measurements provide insight into how the CGN is deployed that seem impossible or difficult to glean with BitTorrent IBR (e.g., which internal IP addresses they use, port allocation strategies).

6.4.3 Analyzing DHCP lease dynamics

The Dynamic Host Configuration Protocol (DHCP) facilitates the assignment of IP addresses to hosts. In particular, DHCP supports *dynamic allocation* where a host uses an IP address for a limited period of time (or until the host explicitly returns the address) [60]. This means that over long periods of time a host may use multiple IP addresses and that multiple hosts may use the same IP address. The use of DHCP makes it difficult to count the number of machines infected with malware [167, 105], and may limit the effectiveness of blacklisting.

Despite DHCP's widespread deployment, very few measurement studies have analyzed the dynamics of DHCP, including lease duration, and the size addresses pools from which addresses are assigned. By tracking over 500k clients with HTTP cookies for a month in 2006, Casado *et al.* found that 72% of clients used a single IP addresses for more than two weeks [38]. More recently, Padmanabhan *et al.* presented preliminary findings on dynamic address durations using RIPE Atlas probes [155].

The findings in this section are a result of a collaboration with Padmanabhan *et al.* to corroborate the RIPE Atlas findings with IBR. RIPE Atlas [174] is an active measurement architecture that has over 9k nodes with a presence in over 3k ASes [80]. The nodes periodically report the measurements to a central server. The reports include ma-

[15] Richter *et al.* analyze 2016 data, which they find to be consistent with late 2015 data.

chine identifying information. Padmanabhan *et al.* leverage the pairing of (IP address, machine ID) to determine when a machine is using a new IP address — typically due to an expired or relinquished DHCP lease.

Technique for analyzing DHCP lease dynamics with IBR

Like CGN, we can analyze DHCP leases with any traffic that contains IP addresses and machine identifiers. We considered using IBR to extract identifiers from packets containing a BitTorrent payload, originating from the Sality and ZeroAccess botnets, or having TCP timestamps. We chose BitTorrent traffic due to the large increase in packets of this type starting in July 2015 (Section 4.3.2).

Clients using BitTorrent's DHT generate a random 160-bit node ID [124]. Using 160 bits means that there is an extremely high probability that well-behaved clients will generate unique node IDs. The node ID is included in all KRPC packets. Although clients could change their node ID at any time, they typically select a node ID to use until they rejoin the DHT (an unlikely event) [146]. The uniqueness and repeated use of the BitTorrent node ID, as well as the large volume of BitTorrent traffic in IBR, make it a good candidate for studying DHCP.

Hosts send BitTorrent IBR at irregular intervals. This is problematic as there are often long periods (e.g., days) in which we do not receive packets from a host. Unlike the data from RIPE Atlas probes, we cannot calculate the exact duration in which an IP address is used by a host. However, we can still extract bounds on the address' duration.

For each sequence of packets with a given node ID we extract two metrics related to lease duration. We use the following notation for a sequence of packets: $p_1, p_2, p_3, \cdots$; and use the functions $\mathsf{IP}(p_i)$ and $\mathsf{TS}(p_i)$ to obtain the source IP address and timestamp of packet p_i respectively. Specifically, we are interested in changes in the IP address associated with a node ID. Let p_i be the first packet associated with IP ad-

dress $\mathsf{IP}(p_i)$, i.e., $\mathsf{IP}(p_{i-1}) \neq \mathsf{IP}(p_i)$. Additionally, let j be the maximum value satisfying $\mathsf{IP}(p_j) = \mathsf{IP}(p_i) \forall j \geq i$. We calculate the *minimum address duration* as $\mathsf{TS}(p_j) - \mathsf{TS}(p_i)$. The *maximum address duration* is $\mathsf{TS}(p_{j+1}) - \mathsf{TS}(p_{i-1})$. The minimum address duration a lower bound on the period of time that an IP address was associated with the node ID; the maximum address duration is an upper bound on the period of time that an IP address was associated with the node ID.

In our analysis we use data from July and August 2015. Due to a darknet outage and a processing error, we are missing data for two days in July and one day in August.

Length of time hosts use IP addresses

We would like to determine the duration for which DHCP assigns an IP address to a host. However, our data reveals bounds on the length of time an IP address is used by a host. Since it is possible that a host will relinquish its IP address, the observed duration of usage may be less than the duration specified by DHCP. We borrow Padmanabhan *et al.*'s terminology and call a typical duration that an IP address is used by a single host a *characteristic address duration.*

We analyze characteristic address durations for nine ASes, checking that the IBR-based inferences match the data from the RIPE Atlas probes. In this section, we only consider minimum address durations with a corresponding maximum address duration.

For all ASes considered in the prior work, we graph the weighted CDF of the minimum lease duration in Figure 6.11. Specifically, we weight each data point by its duration; this weighting makes it so that the CDF approximates the probability of having a lease less than or equal to a given duration. Since we are not guaranteed to see a BitTorrent packet and the exact beginning and end of the lease, we expect the minimum address durations to approach but not exceed the characteristic duration. Thus

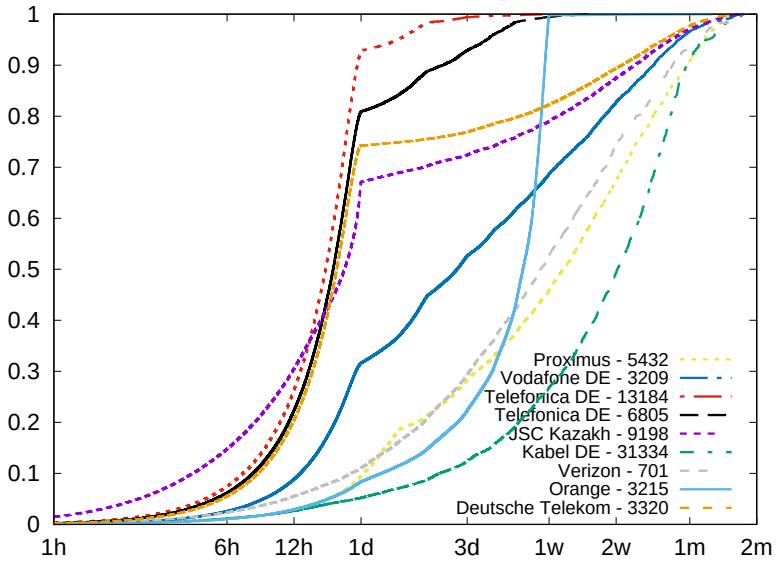


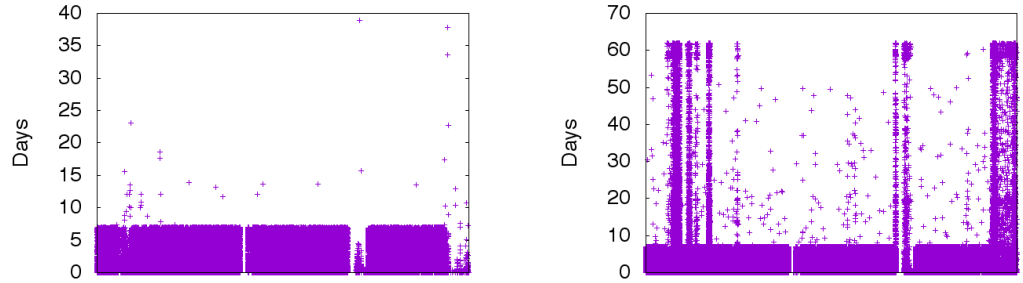
Figure 6.11. Weighted CDF of characteristic address duration. For all validation ASes, we plot a CDF of all minimum address durations with a maximum address duration. Each data point is weighted by its duration, so this graph approaches the probability that a DHCP lease is less than or equal to a given duration.

the characteristic durations appear as "elbows" in Figure 6.11.

Our findings for characteristic lease durations echo Padmanabhan *et al.*'s [155]. For Verizon (ASN 701) and Kable DE (ASN 31334) there are no characteristic lease durations, implying that the AS does not mandate a maximum lease duration (e.g., hosts can renew their lease). The remaining ASes have characteristic lease durations, which are consistent with RIPE Atlas data.

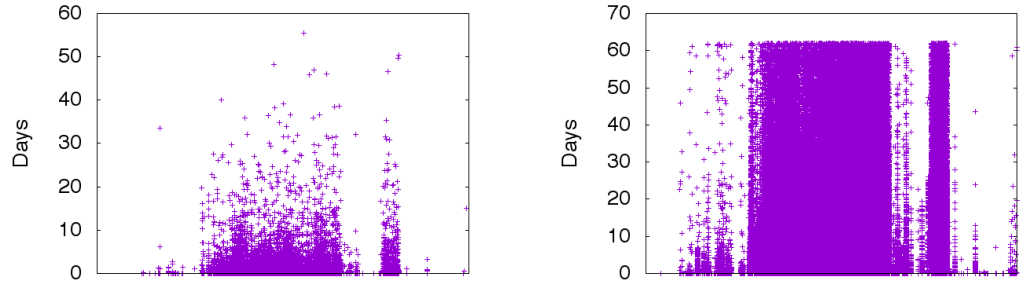
Lease type

Next, we check for regions of the address space where hosts appear to have the same IP address for long periods of time (e.g., they are statically assigned). We look for prefixes in the nine ASes examined by Padmanabhan *et al.* where it is uncommon for hosts to change addresses. We graph the minimum address duration under two scenarios: (1) the lease has ended, which we know because there exists a corresponding maximum



(a) Orange (ASN 3215) — Minimum address duration for expired leases

(b) Orange (ASN 3215) — All minimum address durations



(c) Verizon (ASN 701) — Minimum address duration for expired leases

(d) Verizon (ASN 701) — All minimum address durations

Figure 6.12. Expired lease durations versus all lease durations (Orange and Verizon). Comparing minimum address durations for expired leases and all leases reveals regions of the address space that are likely to be statically assigned, as they only appear Figure 6.12b or 6.12d.

address duration; and (2) for all leases, even when we have yet to observe the node ID in a packet with a different IP address.[16] Scenario (2) includes statically assigned IP addresses because we should never associate the node ID with a different IP address. Scenario (2) may also include hosts that uninstall their BitTorrent client or generate many BitTorrent IDs (such as in a Sybil attack [189]).

In Figure 6.12 we graph all minimum address durations in scenario (1) and (2) for two ASes: Orange (ASN 3215) and Verizon (ASN 701). On the x-axis we have each

[16]To exclude cases of user mobility (e.g., using a laptop at home and school), we only infer maximum address durations when we observe the node ID with another IP address in the same AS.

IP address in announced prefixes of the ASN in sorted numeric order.[17] This ordering places IP addresses in the same announced prefix next to each other.

From Figure 6.12, we can infer three types of leases: absolute, static, and renewable. The absolute leases never exceed the lease duration in both scenarios; most of Orange's address space fits this characterization. Static assignments will only appear in scenario (2), as they never have a maximum address duration; a few subnets within Orange and Verizon use this type of assignment. Renewable regions appear in both scenarios, but still-in-use leases will appear only in scenario (2) as long leases. This describes Verizon's behavior, as the 3.7M leases included in Figure 6.12d are typically longer than the 31k included in Figure 6.12c. Since many Verizon leases approach the two month mark (the duration of our study), it is probable that Verizon hosts can repeatedly renew their leases. Interestingly, these ASes both dynamically and statically assign IP addresses.

Discussion

BitTorrent IBR reveals interesting information about DHCP lease dynamics: we can identify the characteristic lease duration, and static regions of the address space. Based on a preliminary analysis, we expect that IBR will also be a good source to determine patterns of in when reassignment occurs[18], and pools of addresses used in reassignment[19].

We can examine properties of DHCP in any network where hosts use BitTorrent, including ASes that do not have a RIPE Atlas probe. Compared to the RIPE Atlas

[17] For an IP address A.B.C.D, we use the number $A\times2^{24}+B\times2^{16}+C\times2^{8}+D$.

[18] A preliminary analysis of the number of reassignments in an AS per hour varies across ASes. Often we observe a diurnal pattern (consistent with Internet usage being higher during waking hours), but, in a few cases, we observe spikes (possibly due to outages or provider induced events). This analysis will help infer the causes of reassignments.

[19] We are interested in the set of possible addresses a host will be assigned after relinquishing their lease. Such analysis is useful for determining if blacklisting is effective on the /24 or prefix granularity.

probes, IBR provides less precise insight into DHCP lease durations. There can be a large difference between the minimum address duration and the maximum address duration — which causes uncertainty in when reassignment occurred. However, there are significantly more clients contributing to IBR-based observations than RIPE Atlas-based observations. There are 121, 68, and 40 RIPE Atlas probes in Orange, Deutsche Telekom, and Verizon respectively. In our IBR dataset there are 54k, 73k, and 6.8k unique node IDs with both minimal and maximal address duration values[20] in Orange, Deutsche Telekom, and Verizon respectively.

It is future work to examine DHCP lease dynamics before the large increase in BitTorrent IBR. We expect fewer hosts to send BitTorrent traffic, and to receive traffic from those hosts less frequently. If, with lower volume BitTorrent traffic, there is a large degradation in measurement quality, we could also examine other types of IBR with machine identifiers (e.g., Sality packets, TCP timestamps).

6.4.4 Discussion

The case studies in this section revealed weaknesses in an existing algorithm for detecting NAT (Section 6.4.1), and successfully examined CGN (Section 6.4.2) and DCHP deployments (Section 6.4.3). To the best our knowledge, there have been very few efforts to conduct Internet-wide measurements of NAT/CGN and DHCP deployment. These properties are difficult to examine through active measurements, but visible through passive data that includes machine identifiers. Fortunately, IBR contains multiple types of packets with potential machine identifiers — not just the BitTorrent traffic we leveraged for CGN and DHCP analysis. For example, Sality and ZeroAccess C&C packets contain identifiers, and previous research fingerprinted machines based on TCP

[20] We expect the number of node IDs to be roughly equal to the number of machines running BitTorrent. However, Sybil attacks or clients that change their node IDs often could distort the one-to-one mapping. By considering only the node IDs with both minimal and maximal address durations, we select IDs that are consistently used.

timestamps or the IPID field [64, 23].

Our success in using IBR to study network configurations is due to (1) heuristics that are effective on a traffic sample, and (2) an opportunistic increase in traffic volume. Another work studying Internet-wide configurations used the same two-part formula. Sargent *et al.* examined filtering policy by comparing Conficker IBR to DNS sinkhole traffic [180]; their technique did not capture all TCP/445 packets leaving a network, but took advantage of the large volume of Conficker packets reaching darknets. IBR seems well-suited to investigate other types of network configurations assuming they also meet these criteria. Moreover, even with January 2015 data, which contained less BitTorrent IBR than June 2015, we classified 60% of ASes known to deploy CGN correctly.

6.5 Conclusion

In this chapter we have considered a series of case studies that revealed information about address space utilization, and in particular IPv4 address space exhaustion. Although all but one RIR has exhausted its IPv4 address pool, many IP addresses are still unused, as they are not announced in BGP or they do not appear in a number of datasets (Section 6.1). Administrators appear to deliberately and effectively allocate their used addresses: large contiguous blocks are often used by the same type of machine: clients, servers, infrastructure (Section 6.2.2). Additionally, many ASes actually deploy methods to share IP addresses among subscribers, including CGN (Section 6.4.2) and DHCP (Section 6.4.3). These methods of sharing IP addresses should be effective since many machines frequently reboot (Section 6.3.1).

However, the primary goal of this chapter was to evaluate IBR's utility as an Internet-wide data source. To this end, we have leveraged IBR for three different types of tasks: enumerating resources (used machines, HTTP servers, open resolvers, and clients), testing heuristics (a common uptime calculation, p0f's NAT detection), and clas-

sifying behaviors (uptime, BitTorrent client popularity, time to fix a bug, CGN, DHCP).

In terms of enumerating resources, IBR is effective in two scenarios. First, IBR is useful for identifying servers involved in malicious activity: we identified open DNS servers used in amplification attacks (Section 6.2.1), and previous work identified victims of DoS attacks [141]. However, compared to dedicated probing to detect servers, IBR detects significantly fewer open resolvers, despite a large increase in traffic. Second, IBR is useful for client identification. Clients are difficult to measure through active probing, but often send IBR, including traffic from worms, and client applications including Qihoo 360, BitTorrent. We have combined client /24 blocks identified through IBR with /24 blocks containing HTTP servers and routers to produce a Hilbert curve of /24 block functionality (Section 6.2.2). Overall, other non-IBR passive datasets were more effective in supplementing the traditional, ICMP-ping based study of IPv4 address utilization: each revealed about 100k more new /24 blocks than IBR (Section 6.1).

The volume and availability of IBR makes it a great data source for testing existing tools and heuristics. We found that a common method for determining uptime with TCP timestamps is inaccurate for four operating systems: Linux 2.2.x-Linux3.x, Linux 2.4.x, iOS iPhone/iPad and Mac OSX 10.x (Section 6.3.1). Additionally, we found three quirks in p0f's NAT detection algorithm that caused false positives (Section 6.4.1). Obviously, developers should not exclusively test their tools with IBR; however, it is a reasonable expectation that the presence of IBR does induce many false positives when analyzing one-way traffic.

In terms of classifying behaviors, our case studies suggest one general rule of thumb: only use IBR when a biased sample is acceptable. In our BitTorrent client popularity case study (Section 6.3.2), we obtained conflicting results for two different types of BitTorrent packets. Moreover, one result included a client (associated with 10.7% of IP addresses sending BitTorrent IBR) that was not found in non-IBR studies.

Because we were uncertain of all the factors resulting in BitTorrent IBR we could not pinpoint possible causes for our biases. We were more successful at identifying when Qihoo 360 fixed a byte order bug because we understood the process generating IBR (Section 6.3.3). However, using the same BitTorrent IBR that was unacceptable for the client study, we successfully identified networks deploying CGN (Section 6.4.2), and analyzed DHCP dynamics (Section 6.4.3). For both of these case studies, we inferred network properties that are not specific to BitTorrent.

Acknowledgements

Section 6.1, in part, is adapted from material as it appears in the Journal on Selected Areas in Communications (JSAC). Dainotti, Alberto; Benson, Karyn; King, Alistair; Huffaker, Bradley; Glatz, Eduard; Dimitropoulos, Xenofontas; Richter, Philipp; Finamore, Alessandro; Snoeren, Alex C.; IEEE, 2016. The dissertation author was one of the primary investigators and authors of this paper.

Section 6.1.4, in full, is adapted from material as it appears in SIGCOMM Computer Communication Review. Dainotti, Alberto; Benson, Karyn; King, Alistair; Kallitsis, Michael; Glatz, Eduard; Dimitropoulos, Xenofontas; ACM, 2013. The dissertation author was one of the primary investigators and authors of this paper.

Sections 6.2.1, and 6.3.1, in part, are adapted from material as it appears in the proceedings of the Internet Measurement Conference (IMC 2015). Benson, Karyn; Dainotti, Alberto; claffy, kc; Snoeren, Alex C; Kallitsis, Michael; ACM, 2015. The dissertation author was the primary investigator and author of this paper.

Section 6.4.2, in full, is currently being prepared for submission for publication of the material. Livadariu, Ioana; Benson, Karyn; Elmokashfi, Ahmed; Dhamdhere, Amogh; Dainotti, Alberto. The dissertation author was one of the primary investigators and authors of this material.

Section 6.4.3, in full, is currently being prepared for submission for publication of the material. Padmanabhan, Ramakrishna; Benson, Karyn; Dainotti, Alberto. The dissertation author was one of the primary investigators and authors of this material.

Chapter 7

Inferences with IBR: Using IBR to infer network status

In addition to revealing attributes of Internet hosts and networks, IBR can also provide valuable insight into the current status of these hosts and networks. That is, we can use IBR to expose information about network conditions (or changes in conditions) as experienced by end users. Previously, Dainotti *et al.* investigated macroscopic outages using IBR [56, 55]. In this chapter, we present techniques that use IBR to examine two other aspects of network state: when the path used to transmit packets from the network to the darknet changes (Section 7.1), and when hosts in the network experience packet loss (Section 7.2). Identifying path changes and packet loss situations can supplement outage investigation by providing symptomatic details of the event; but these analyses are also interesting on their own.

Determining network state typically involves a comparison between collected traffic and our expectations of the network. Although IBR is an erratic data source, we can form expectations by extracting predictable attributes of IBR. In this chapter, we consider the following, non-exhaustive list of examples:

- Packet header fields that are stable across many types of traffic, such as the TTL field when analyzing path changes in Section 7.1.

- Individual, reliable components of IBR, such as the retransmission properties of Conficker-infected machines or the default behavior of Windows machines in Section 7.2.

- Large aggregations of traffic, such as when Dainotti *et al.* examined traffic originating from an entire country to examine macroscopic outages [56].

We emphasize that there may be alternate ways to infer network status with IBR. Since IBR is a complex assortment of signals, there are multiple ways to extract predictable attributes and evaluate network quality. Thus, each of our techniques are a lower bound on the total number of IBR-analyzable networks. For example, in Section 7.3.1, we provide an alternative packet loss metric.

As we describe in Section 7.3.2, using only IBR, it is difficult to pinpoint which network or link is responsible for a service degradation. For example, we can determine that packets take a different route to our darknets, but not where the path change occurred (e.g., traffic exits an autonomous system via a different router). As a result, IBR-based analysis seems well suited to supplement specialized active probing, as opposed to replacing existing measurement methodologies for assessing network quality. It is easy to conceive of a system that uses IBR to inform when and where to conduct dedicated probing; compared to purely active techniques, such a system would require less probes sent to networks under distress — situations where outage detection, path changes and packet loss analyses are especially interesting.

7.1 Identifying path changes

Detecting and analyzing path changes provides insight into Internet path stability [158, 50], and outages [25, 223, 107]. Our goals with this case study are to explore an inference that: (1) requires successive measurements; (2) has an element of predictabil-

ity (although IBR composition is erratic, TTL is predictable); and (3) shows how to use IBR to reduce the active probing required to infer changes (similar to PlanetSeer [223] and Hubble [107]).

7.1.1 Method of identifying path changes with IBR

Our technique to identify path changes relies on the insight that the TTL of a received packet reflects the number of hops on the path to the darknet. If the path is unchanged, all packets from a host will have the same TTL. Since most operating systems have a starting TTL that is a power of two [182], we calculate the number of hops by subtracting the TTL from the next highest power of 2 (a technique previously used by Beverly [26]), excluding any packets with a TTL of three or less, since they likely originate from traceroute and are not a predictable measure of hop count.[1] When the number of hops from a source to the darknet increases or decreases, we infer a likely path change (similar to a previous technique for monitoring traffic at a CDN [223]). Note this method will not detect changes that result in the same-length path but through different routers.

We divide our datasets into time bins. For each IP address, we calculate for each time bin, t, $\max_t$ and $\min_t$, the most and least number of hops taken at time t respectively. We consider a path to have changed if $\max_t > \max_{t-1}$ or if $\min_t < \min_{t-1}$. We expect most path changes to occur within a time bin, and not at time-bin boundaries. Our requirements capture changes within a time bin as the time bin includes packets with the old TTL and the new TTL. This method should also account for a change in load-balancing paths (the whole distribution shifts). The method will have some false

[1] The first packet received by the darknet during a traceroute probe will have TTL=1, irrespective of the length of the path to the darknet. To be robust in situations where single routers drop but do not generate ICMP time exceeded messages, traceroute will send probes with higher TTL values when it fails to receive responses. As a result, we receive packets with TTL=1, TTL=2, and TTL=3 for each traceroute to a darknet IP address.

positives due to NAT (when a new host, with a longer/short path starts transmitting) as well as false negatives (when all hosts with the longest/shortest path stop transmitting at the same time as a path change).

To study changes affecting larger source granularities, e.g., a prefix or AS, for each time bin we also calculate the percentage of IP addresses, p, that sent packets in both that time bin and the previous one, and also indicated a path change. Using multiple sources from a prefix or AS increases our confidence that an event occurred. In particular, we will have better insight into the core of the Internet as many hosts send packets that traverse its edges.

7.1.2 Number of analyzable networks

The number of path change-inferable networks is a function of time bin duration. With short time bins, we can determine the precise time of a path change. For example, if a path change occurs between retransmits of a packet, we can potentially pinpoint the time of the path change within a few seconds. However, based on our analysis of repeated contact in Section 5.2.2, only countries and a few ASes send IBR to our darknets every minute, implying that analysis at the minute granularity is not possible for many networks.

In most cases, we can increase the number of path change-inferable networks with longer time bins — at the expense of precision. The intuition is that with longer time bins more networks are likely to have hosts transmitting IBR in consecutive bins. With many analyzable hosts, we become more confident that a substantial path change occurred instead of an event affecting a handful of hosts (or abnormal individual host behavior such as sending packets with varying initial TTL values). For some networks, lengthening the time bins does not help. Individual hosts may send in bursts that are entirely contained in a single long time bin (as opposed to spread out over many shorter

Table 7.1. Number of sources for which we can analyze path changes. We show the number of sources for which we can analyze changes throughout the dataset (*always analyzable*), and for sources where it is possible to examine traffic for changes in at least one time bin (*ever analyzable*). The number of analyzable networks is consistent across datasets, and increases with larger time bins — at the expense of precision.

		UCSD-12		UCSD-13				MERIT-13	
						Partial			
		Ever	Always	Ever	Always	Ever	Always	Ever	Always
1 minute time bins	IP addresses	87M	249	78M	471	35M	214	40M	163
	/24 blocks			2.6M	553	1.9M	314		
	Prefixes	161k	1.1k	171k	1.3k			147k	777
	ASes	20k	695	20k	761	16k	579	17k	595
	Countries	230	119	231	126	226	114	230	119
5 minute time bins	IP addresses	81M	2.5k	78M	2.8k	32M	2.4k	38M	2.2k
	/24 blocks	2.5M	2.3k	2.5M	2.6k	1.9M	2.1k	2.2M	2.0M
	Prefixes	158k	3.3k	167k	3.6k	130k	2.7k	147k	2.9k
	ASes	19k	1.6k	16k	1.7k	15k	1.4k	17k	1.4k
	Countries	230	146	231	155	227	145	231	148
15 minute time bins	IP addresses	74M	4.1k	66M	6.4k	27M	3.9k	32M	3.7k
	/24 blocks	2.5M	3.8k	2.5M	6.0k	1.9M	3.5k	2.1M	3.5k
	Prefixes	158k	5.3k	163k	6.5k	129k	4.4k	142k	4.6k
	ASes	19k	2.1k	19k	2.5k	15k	1.9k	17k	1.9k
	Countries	228	159	230	170	227	160	229	161

time bins). Frequently oscillating paths introduce another possible complication with large time bins: both the shorter and longer path may be present in every time bin.

We expect that, when using IBR to analyze specific events (e.g., an outage), researchers will pick a time bin size appropriate to their IBR collection. For the purposes of our analysis, we consider 1-minute, 5-minute, and 15-minute time bins.

We are interested in paths that we can continually monitor, which we call *always analyzable*. For a network to be always analyzable, in every pair of consecutive time bins, at least one host in the network must send traffic to the darknet in both time bins. That is, in every time bin, there is a source whose TTL values we can compare to the previous time bin. From Figure 5.7 in Section 5.2.2, we know most sources do not send IBR every minute (never mind the stricter criterion involving consecutive time bins). Table 7.1 confirms that few networks are always analyzable.

We investigate which networks meet the always analyzable criteria using UCSD-13 (not shown in Table 7.1). Based on the 5-minute granularity, many large

ASes (announcing a /16 or more) are always analyzable: 29% of IBR-observed large ASes meet our criterion. However, about half of the always analyzable ASes are small (announce less than a /16 block). IBR yields the best insight into path changes for transit/access ASes.

Not shown is the significant overlap of such sources across datasets: with 5-minute time bins 1300 ASes are always analyzable using both `UCSD-13` and `MERIT-13`, and 1000 ASes are always analyzable using both `UCSD-13` and `UCSD-12`. This significant overlap implies that we can use IBR to conduct long-term studies of route stability for these ASes.

Special events may provide additional insight into path dynamics. Table 5.2.2 also reports the number of sources that are analyzable at least once which we label *ever analyzable*. Specifically, we consider a source ever analyzable if at least one host from the source sent traffic in at least one pair of consecutive time bins. With 5-minute time bins, more than a quarter of IBR-observed IP addresses, in three-quarters of IBR-observed ASes are ever analyzable in each dataset. Since our binning may restrain bursts of traffic to a single time bin, lengthening time-bin duration decreases the number of ever analyzable sources.

7.1.3 Validation

Traditionally, researchers analyze path changes through traceroute or BGP updates. Traceroute can find both inter-AS and intra-AS path changes on the forward path (from the measurement infrastructure to the remote network), but a comprehensive view requires frequent probes. Analyzing the reverse path (from the remote network to the measurement infrastructure) is difficult, but possible with a complex reverse traceroute tool [108]. BGP-based inferences do not inject packets into the Internet, but can only reveal inter-AS changes. BGP route collectors gather information about all forward paths

(from the route collectors to all announced prefixes).

IBR is similar to traceroute in that we flag both inter-AS and intra-AS path changes. IBR is also similar to BGP updates in that both are collected passively. However, IBR differs from traditional traceroute and BGP updates in two significant ways: (1) it is difficult, if not impossible, to locate where the path changed, and (2) IBR provides insight on the reverse path (from the senders of IBR to the darknet).

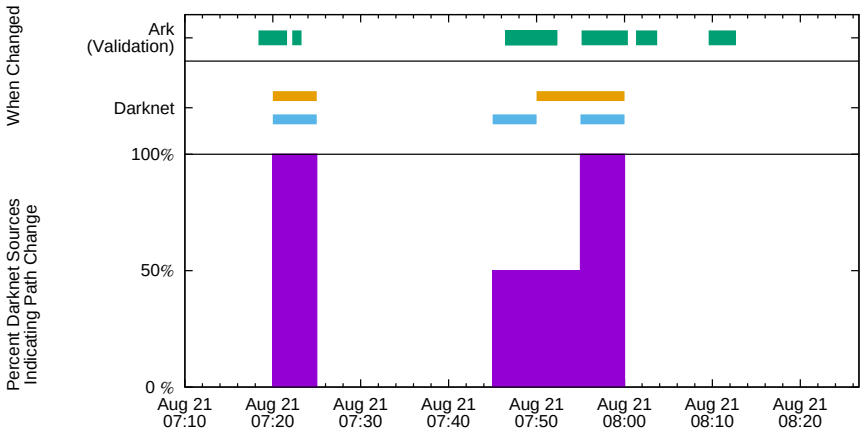
Since IBR captures path changes on the reverse path and traditional methods capture path changes on the forward path, IBR is complementary to existing path-change analysis tools in that it measures a different set of routes. However, the set of paths we can learn about with darknets is limited: there are only a handful of large darknets, each of which are located in a single fixed location.

Additionally, the mismatch in forward/reverse path presents a hurdle for validation: running traceroute or analyzing BGP announcements collected at UCSD provides information about the forward (not reverse) path. Nevertheless, we can use existing measurement infrastructures to partially validate our approach. In particular, Ark [11] monitors send traceroutes to the darknet and BGP monitors collect information about when the path from the monitor to the darknet changes.

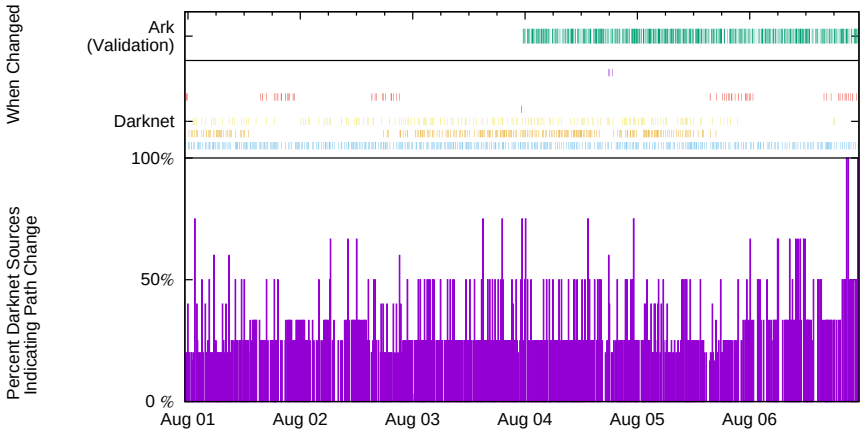
Validation with traceroute data

We validate our method using historical traceroutes from Ark nodes [11] located in always-detectable ASes in `UCSD-13`. The Ark infrastructure uses teams of about 20 nodes to send traceroutes to every routed /24 block over a span of 2-3 days [11]; thus, we can expect about one traceroute per minute from each Ark node to reach the darknet. Nine Ark nodes are in 8 always-detectable ASes, including five educational networks, two large transit providers, and a Regional Internet Registry.

We cannot validate all path changes from the hosts sending IBR, as we do not



(a) KIST



(b) Purdue

Figure 7.1. Example of path changes identified with IBR. The top portion of each figure shows our validation data from Ark. The middle portion of each figure shades, for a sampling of hosts in KIST/Purdue, the periods it inferred a path change. The bottom portion of each figure shows the percentage of darknet IP addresses signaling a path change. We identify the start of path change events at KIST, and route-flapping at Purdue.

know when these hosts start sharing links with Ark monitors to the darknet. However, AS-level path changes should be observable in both Ark data and IBR. We find our analysis of other IBR-transmitting IP addresses frequently corroborates path changes in traceroute data. That is, both IBR and traceroute data indicate the start of the path change event. Figure 7.1 reports, for events in KIST (ASN1237) and Purdue (AS17), the percentage of hosts in the respective AS found in darknet data signaling a path change with 5-minute time bins (in the bottom portion of each graph), and the periods of time that a path change was observed from IPs in both darknet and traceroute data. (in the top portion of each graph, as indicated by a colored time segment).

With both types of data, we infer very few path changes at KIST. Figure 7.1a includes all traceroute-inferred path changes for KIST, and all but one path IBR-inferred change in `UCSD-13`. Most traceroute-inferred path changes occur around the same time as the darknet-inferred changes. Manual inspection of these traceroutes reveals that the path change occurred in the core of the network. Further investigation of the KIST sources suggests that traffic from the darknet sources used multiple paths in the 8:00 to 8:10 time bins (during these time bins the hop count was 16 or 17; outside of the time bins the hop count was 16). For one of the IP addresses, it is possible to look at 1-minute time bins. With this granularity all darknet-inferred changes align with traceroute-inferred changes on August 21, 2013.

Figure 7.1b shows many path changes over a six-day period for Purdue in both Ark (8.9k changes), and darknet data (1.3k 5-minute bins with changes). Several IP addresses produce evidence of frequent path changes. Before August 4, 2013, traceroutes sent by the Ark monitor to UCSD-NT used the same route out of Purdue, but after this date, traffic from the Ark node traversed multiple routes out of Purdue's network. A likely explanation is that some Purdue sources used stable routes, while others used flapping routes; on August 4, 2013 the Ark node switched to using the flapping routes.

At this time, all path change-inferable sources in IBR indicated a change.

The analysis of traceroutes from KIST and Purdue Ark monitors to UCSD-NT validates our IBR-based method of detecting path changes. The traceroute and IBR-based methods characterize path change frequency similarly at KIST (rarely) and Purdue (often). We detected the beginning of path change events in the core of the network (when leaving KIST). For path changes near the edge (Purdue), only a subset of hosts in an AS may actually use a new path to reach a the darknet. In this scenario, using data from multiple hosts is preferable — either from senders of IBR (which is the case for our 6-day period) or from hosts conducting traceroutes (there is only one Ark monitor at Purdue).

Validation with BGP data

Path change announcements for the prefix UCSD.0.0.0/8 observed from Route-views [206] or RIPE RIS [175] peers should be reflected in IBR. There are six Route-views/RIS peers in always-detectable ASes (at the 5-minute time-bin granularity), all are transit/access providers. Since path changes within ASes are not visible through BGP, we only check if we infer a path change in IBR around the same time as a BGP update (not if path changes we observe through IBR are visible in BGP updates).

The time at which we detect a route change may differ in IBR and BGP. First, Routeviews provides updates every 15 minutes, and RIS provides updates every 5 minutes, and our data was timestamped with the first second of the update. Second, it is possible that BGP updates lag behind changes routing within an AS [195]. Furthermore, sometimes we observe multiple changes to the AS-level path within a single time bin (e.g., while BGP converges). For a set of AS-level path changes in a Routeviews/RIS update, we consider our IBR methodology successful if it detects a path change 5 minutes before to 15 minutes after the timestamp of the Routeviews/RIS update. We use

Table 7.2. Time bins with AS-level path changes detected using IBR. For the 6 always analyzable networks with Routeviews/RIS peers we report the number of 5-minute time bins in which an AS-level path changed (total number of AS-level path changes) and the number of corresponding time bins where we detected a path change in IBR, using $p = 25\%$.

AS	Type	BGP	(Number)	IBR	Percent BGP Events Detected with IBR
Telstra (1221)	Transit/Access	2	(3)	2	100%
AOL (1668)	Transit/Access	2	(2)	0	0%
NTT (2914)	Transit/Access	5	(10)	1	20%
Level3 (3549)	Transit/Access	5	(6)	0	0%
Bell Canada (6539)	Transit/Access	2	(2)	2	100%
OBIT LDT (8492)	Transit/Access	34	(58)	30	88%
Total		*50*	*(81)*	*35*	*70%*

5-minute bins and set $p = 25\%$.[2]

Table 7.2 shows that we inferred path changes for 70% of the time periods with BGP updates. However, for two ASes, none of the BGP updates advertised by the AS were considered path changes by our method. This low coverage is most likely due to the fact that there are multiple exit points. In particular, a change in routing may affect only a small portion of hosts within the AS (and Routeview/RIS may not peer with all exit points to determine if the change should affect all host). For the three ASes, we detected over 88% of known AS-level path changes. In these cases, it is likely that a large portion of the AS was affected by the routing change (e.g., the entire AS switches their single upstream provider) or many IBR-sending hosts use the exit points announcing the change. We believe our technique would perform better for ASes at the edge.

7.1.4 Route stability

We use IBR to characterize path change dynamics. Our findings are consistent with previous studies in route stability between PlanetLab nodes in 1994 and 1995 [158] and from traceroutes to over 5k ASes in 2009 and 2010 [50]. Specifically, previous work

[2]We picked $p = 25\%$, since this threshold captures almost all of Ark's path change activity at KIST and Purdue in Figure 7.1. Setting $p = 33\%$ yields the same results.

characterized the connectivity between two end points — called a virtual path. The characterization of a virtual path's stability included prevalence (the fraction of time the most common route is used) and persistence (the time between path changes). In terms of prevalence, Paxson and Cunha *et al.* both found most virtual paths have a route that is active most of the time. In terms of persistence, although there are more short-lived paths, virtual paths spend most of their time in long-lived paths. Paxson noted that a handful of virtual paths oscillate frequently [158], and, Cunha *et al.* found that some virtual paths experienced periods of instability [50].

Without a list of intermediate routers, it is difficult to directly analyze route prevalence and persistence with IBR. However, we can use the time between TTL changes to as a proxy for the duration of time packets traverse the same set of routers. Specifically, we study virtual paths from always analyzable ASes to UCSD-NT using `UCSD-13` with 5-minute granularity. We use $p = 25\%$ to determine when a path changed. Our analysis is an underestimation of path changes, as we miss path changes where the new and old paths are the same length.

Figure 7.2 reports our findings on AS-level route stability using always analyzable ASes in `UCSD-13`. The top portion of Figure 7.2 shows, for each always analyzable AS, the longest time between path changes, or the longest known duration that the AS used a route.[3] This metric is a lower bound on path prevalence: a single route was used for at least the graphed duration, though it is possible that the route was also used in other shorter time periods. Like Paxson [158] and Cunha *et al.* [50], we find that most ASes have a route that is used for multiple days.

The lower portion of Figure 7.2 shows the number of path-changes from each always analyzable AS to UCSD-NT. Most ASes experience very few path changes (like

[3]Since `UCSD-13` lasts 34 days, we do not know the actual duration of the first and last routes. However, we do know the first route was used from at least the start of the *2013 census*, and the last route was used at least until the end of the *2013 census*. We include these routes, using their longest known duration.

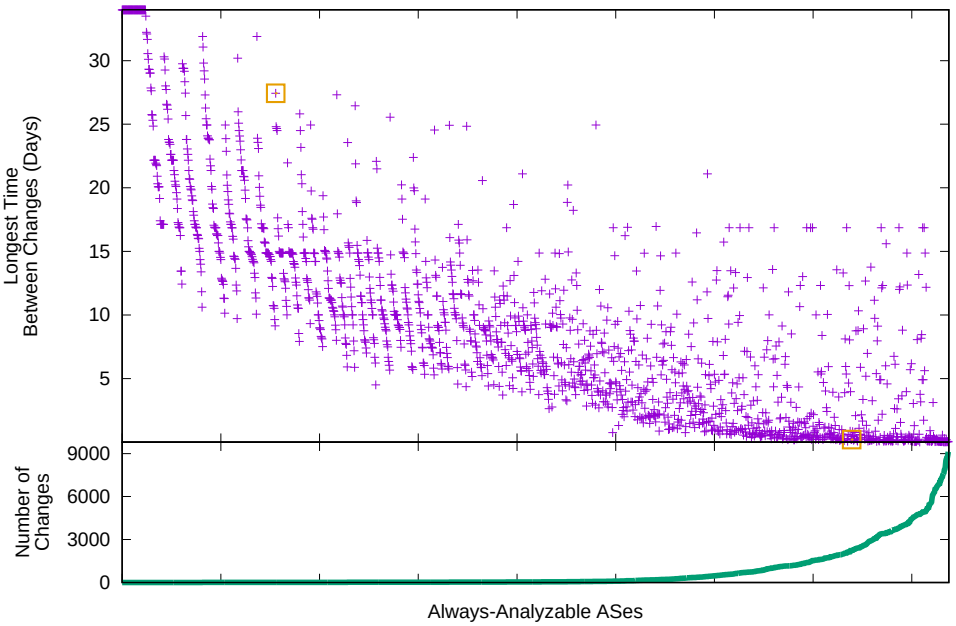


Figure 7.2. Route stability. For each always analyzable AS in `UCSD-13` we show the longest time between detected path changes and the number of path changes (when the percentage of IP addresses signaling a change, p, is greater than 25%). We find that most virtual paths from always analyzable ASes to UCSD-NT are stable: with routes lasting multiple days and few path changes. The highlighted ticks correspond to KIST (few path changes, long time between changes) and Purdue (many path changes, short time between changes).

KIST), and a few ASes in the tail (like Purdue) are responsible for the majority of path changes. This finding is consistent with previous analysis of path persistence [158, 50].

7.1.5 Discussion

Although IBR is an erratic data source, this example shows that it can provide insight into abnormal events and macroscopic dynamics. Our success with this case study is partially due to the aspect of IBR we are evaluating: the expectation that the initial TTL value remains the same is true regardless of the number of sources sending IBR or the volume of IBR, although increases in either would likely improve our coverage and accuracy. This path change detection method would work best in conjunction with other data sources. Like PlanetSeer and Hubble, passive traffic measurements such as IBR can help inform when and where active measurements would be most useful [223, 107]. IBR also provides features that traceroute and BGP data lack, e.g., no injected traffic required, and intra-AS visibility, respectively.

7.2 Recognizing Packet loss

In this section, we investigate new IBR-derived metrics that can provide insights into the causes of macroscopic connectivity disruptions. These metrics can indicate whether an outage involves packet loss, e.g. due to link congestion. A large fraction of IBR traffic is composed of TCP SYNs probing the Internet trying to establish connections to vulnerable (usually Windows) hosts. Because a darknet is completely passive (it does not respond to any packets), sources sending these SYNs must re-transmit them. TCP retransmit behavior (such as how many retransmits per connection attempt, and how much time between them) is typically a function of the host operating system or application, which means it is consistent across large enough populations of hosts to constitute a predictable signal. We derive two metrics from two different dimensions of

this signal: the number of SYN retransmit per TCP flow; and the distribution of inter-packet times (IPT) between them. We show that both metrics can reflect packet loss, providing additional insight compared to metrics that only indicate reachability. We apply this metric to three case studies where either route leaks caused link congestion for an entire AS (and ultimately a complete outage in one case) or packet filtering that almost entirely isolated a country from the rest of the Internet.

Traditional passive approaches to inferring packet loss use attributes such as the sequence number [24, 137] congestion window [102], RTT [102, 103, 71, 137], and TCP acknowledgments [106] – all of which were developed using bidirectional communication. In contrast, we: (i) observe unidirectional traffic from a darknet, and (ii) use retransmitted packets as opportunistic probes that measure large-scale Internet events. With bidirectional communication retransmitted packets are evidence of packet loss [103, 137], while with IBR the lack of retransmissions indicates packet loss. We are not aware of similar studies and we consider this work a first attempt to investigate this approach.

7.2.1 Data source and signal extraction

We analyze IBR traffic captured at UCSD-NT in 2012. A darknet receives but does not respond to traffic, so all flows (defined as the traditional 5-tuple) are unidirectional. When an external host attempts to open a TCP connection, the resulting flow carries only SYN packets, which we call a *SYN flow*.

To derive IBR metrics that correlate with packet loss, we need attributes that are normally consistent yet change during connectivity disruptions. The ideal signal would be strong (statistically significant), stable (low noise), and globally pervasive (seen in most networks). But IBR includes diverse types of traffic [156, 217], so we selected two subsets of IBR that have consistent and predictable enough behavior to use as signals

for packet loss:

- Conficker-like traffic, i.e., SYN flows to TCP port 445, widely publicized during the Conficker episode in 2008 but a target of scanning activity for years; it constitutes a large percentage of the packets observed at the UCSD telescope (more than 40%), is globally pervasive, and consistent [203, 5].

- The default configuration of Windows machines is to send at most 3 SYN packets [194] when attempting to establish a connection, which makes SYN flows from such machines a consistent signal.

To infer packet loss, we selected two attributes of SYN flows – number of retransmissions and IPT – that follow consistent patterns. Since the darknet never responds with an ACK, the number and timing of SYN retransmits is determined by the application or the OS originating such traffic. The consistency of these attributes depends on the conditions of the path traversed by the packets, so substantial drops in SYN retransmits or substantial variation in the IPT may reflect network-induced packet loss.

Conficker-like traffic

Figure 7.3 shows the distribution of SYN flows destined to TCP port 445 as a function of packet size, number of retransmits, and OS (as identified by *p0f* signatures [221]). Most of these SYN flows contain only two SYN packets, consistent with the behavior of Conficker-infected hosts [5]. To obtain a strong and stable signal for a retransmit-based metric, we tried to isolate such behavior (i.e., 2-packet SYN flows) by selecting only flows from Windows XP and Windows NT (about 89% and 9% of the total flows in Table 7.3) with packet sizes of either 48 or 52 bytes. The inter-packet times (IPT) metric is not usable with the Conficker-like traffic since the flows only have two packets; loss of one of them prevents a valid IPT calculation.

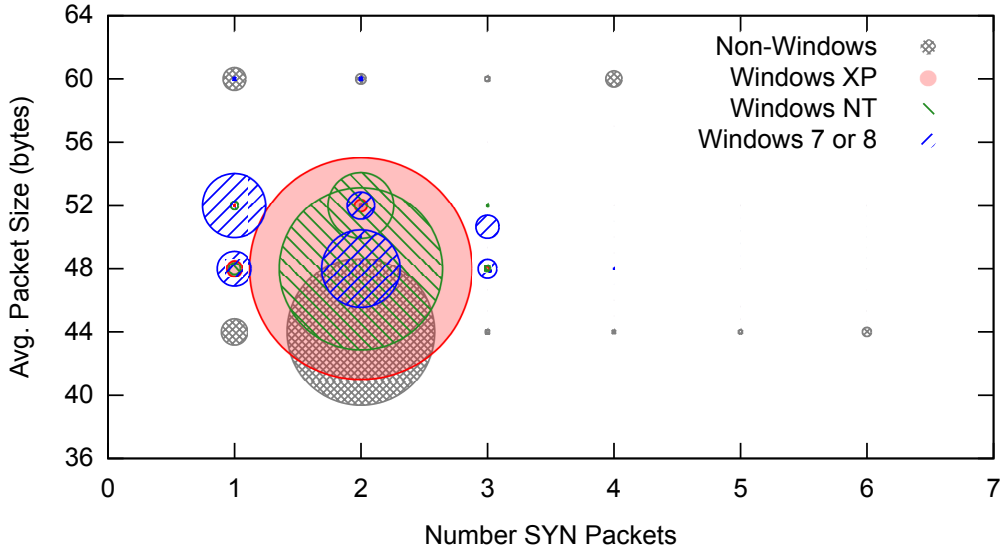


Figure 7.3. Packet size vs. number of packets per flow by OS (Jan. 2012) for Conficker-like traffic. The radius of each circle is proportional to the percentage of SYN flows from the given OS having that packet size and number of packets. Windows 7 or 8 and Non-Windows hosts send a variety of packets per flow and packet sizes - making it hard to extract a stable signal. Conversely, Windows XP and NT consistently send flows with 2 packets of size 48 or 52, creating a stable signal.

Table 7.3. Conficker-like SYN flows observed per OS (Jan. 2012).

Operating System	Number of Flows
Windows XP	2299144254
Windows NT	229961989
Windows 7 or 8	53445230
Other (Linux, BSD, Solaris, . . .)	394731

Table 7.4. Example OS-port combinations used for γ_3. The top four OS-port combination flows of the following categories: at least 25,000 3-packet SYN flows; originating from 64 or more /8 networks; 3-packet SYN flows comprise more than 75% of the SYN flows from the specified OS and port. There are 9 listed in the table because of overlap in the top-four lists. (Jan. 2012)

Port	OS	%3-Flows	Num 3-Flows	Num /8
80	Windows 7 or 8	0.850	6107763	184
443	Windows 7 or 8	0.775	2656821	170
443	Windows NT	0.828	2602825	169
1433	Windows XP	0.814	39476702	114
3260	Windows 7 or 8	0.987	293572	85
4661	Windows NT	0.984	183551	97
4899	Windows 7 or 8	0.993	16108965	73
28931	Windows 7 or 8	0.984	25961	71
22292	Windows XP	0.804	11433398	174

Default Windows behavior

To build a second signal usable for packet loss inference, from IBR observed at the UCSD Network Telescope in January 2012 we selected the port-OS combinations satisfying all of the following criteria:

- more than 75% of their SYN flows carry 3 packets (aiming at a stable signal);
- more than 25000 3-packet SYN flows (aiming at a strong signal);
- their 3-packet SYN flows originate from more than 25% of the total number of /8 IPv4 networks (aiming at a globally pervasive signal).

We selected 100 port-OS pairs that met these criteria, including 56 “Windows 7 or 8" ports, 29 “Windows XP", and 15 “Windows NT". Table 7.4 lists the top four port-OS pairs for each separate criterion.

Although the total number of 3-SYN flows that we select is two orders of magnitude smaller than Conficker-like flows (about 156M vs. 13B in January 2012) and the number of sources generating 3-pkt SYN flows is smaller by a factor of 7 (an average of 14K hosts/hour compared to 100K Conficker hosts/hour in January), having a second traffic signal is still useful, especially to validate findings. Also, the 3-SYN flows met-

rics are not malware-specific, which is especially important as machines are upgraded and patched, limiting the spread of the Conficker-like traffic. The 3-SYN flows are also amenable to IPT calculations when one of the packets is lost, unlike the Conficker-like (mostly two-packet) flows.

7.2.2 Definition of metrics

We define two metrics that we extract from the IBR signals.

Number of packets per SYN flow: γ

We first considered simply the average number of packets per SYN flow from the selected traffic (either Conficker-like or 3-pkt SYN flows). When considering flows sent by only a subset of source IPs, such as the AS-level interpretation (that is, computing such metric only for IBR originating from a specific AS), this value could be significantly skewed as a result of the increased influence of a single host or flow. For example, when a single host conducts a horizontal scan by sending one packet to every IP address in the darknet, the AS-level average is approximately 1 packet per flow regardless of other host activity from that AS. Similarly, a single flow consisting of a large number of SYN packets significantly increases the overall average. The following improvements reduce the impact of such anomalies:

- we exclude all the SYN flows with more than a given number of packets since we definitively know Conficker-infected or Windows machines did not generate them: we set a threshold of *three* for Conficker-like SYN flows (97% of SYN flows had three or fewer packets in our reference dataset of January 2012); *four* for the Default Windows SYN flows;

- we calculate the average number of packets per SYN flow for each distinct source IP, and then take the average (mean) of this distribution, thus limiting the influence

of a single source IP sending packets to the darknet.

If the set of all source IPs is S, F_s denotes the set of flows matching our criteria with source IP s, and the function $\mathsf{packets}(f)$ returns the number of packets in a flow f then our metric is

$$\gamma = \frac{1}{|S|} \sum_{s \in S} \frac{\sum_{f \in F_s} \mathsf{packets}(f)}{|F_s|} \tag{7.1}$$

If there are no sources matching our criteria, then γ is undefined. We call the metric γ_C for the Conficker-like traffic and γ_3 for the flows that are expected to have three packets per SYN flow. We do not combine the two metrics γ_C and γ_3, as the ratio of hosts contributing to each metric is not constant.

Figure 7.4 shows this metric across all ASes for January 2012, calculated in hourly bins. The number of source IPs and γ approximately follow a sinusoidal pattern with a phase of one day. The value of γ_C is always between 1.98 and 2.02. The value of γ_3 is always between 2.59 and 2.78. The large drop in γ_3 seems to be related to traffic on BitTorrent and HTTPS ports.

Outages are likely to affect only a subset of the Internet hosts. Grouping by AS provides a natural way to divide the IP address space. We used CAIDA's Prefix to AS Mapping Dataset and RouteViews BGP data [206]. Figure 7.5 shows γ_C calculated for three ASes of different size. As expected, when calculating γ_C for a single AS, there is higher variance for ASes with fewer infected hosts, typically proportional to their size. Increasing the size of the time bins would reduce such measurement variance, but at the expense of precision in when inferring a connectivity disruption occurred.

Inter-packet times: η

Hosts following RFC 6298 [157] should wait at least one second before retransmitting the initial SYN packet; subsequent retransmission timeouts (RTOs) should back

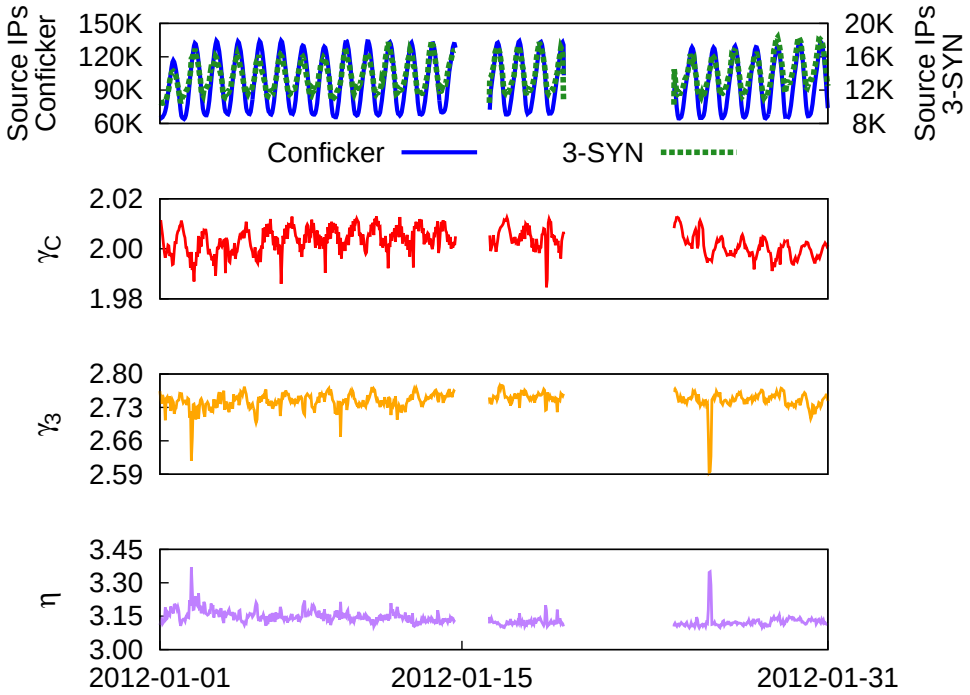


Figure 7.4. Number of source IPs and new metrics for all ASes (Jan. 2012) with time bins of 1 hour. The number of source IPs sending Conficker-like and Default-Windows traffic to UCSD-NT is significant and follows a sinusoidal pattern. γ_C is always between 1.98 and 2.02; γ_3 is always between 2.59 and 2.78; η is always between 3.09 and 3.37. Under normal circumstances, each metric has a small range - which is necessary to identify deviations associated with outages. The telescope was down for about 40 hours starting on 2012-01-14 and for about 120 hours starting on 2012-01-19.

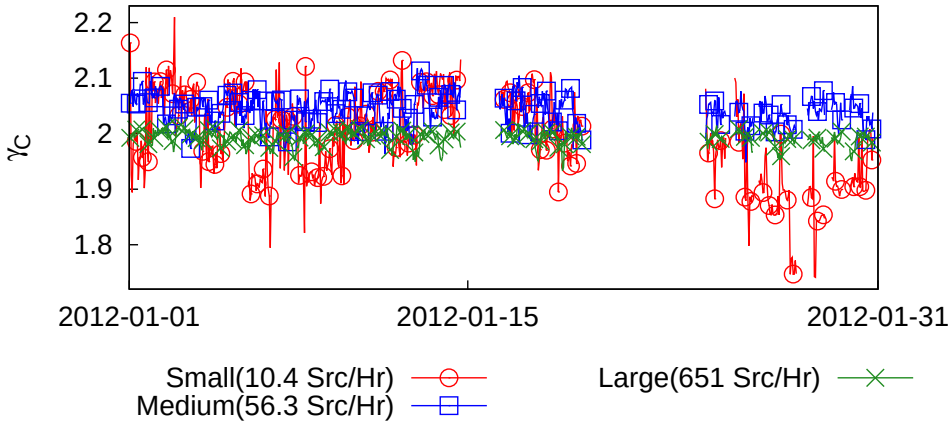


Figure 7.5. AS-level γ_C and number of source IPs for three ASes with different number of infected hosts (Jan. 2012, hourly bins). When there are more infected hosts γ_C is more consistent – meaning it is more useful for discerning abnormalities.

off exponentially by a factor of two. The convention is to use 3 seconds as the initial RTO (i.e., the RTOs are normally 3, 6, 12, 24, ... seconds).[4] The average of the first IPT can be used to verify the findings of the γ metric. In flows with three packets, if a single packet is lost, the first IPT is either (approximately) 6, 9, or 3 seconds corresponding to loss of the first, second, or third packet, respectively. We can only calculate η on expected 3-pkt SYN flows.

As in the calculation of γ, we consider the possibility of skew from a few deviant hosts. Thus we do not take the average of all first IPT, but the average first IPT value over all sources with analyzable traffic. Specifically, if F_s denotes the set of flows with source IP s that are expected to have 3 packets and have at least 2 packets, $S = \{s \in \text{seen source IPs} | F_s \neq \emptyset\}$, and the function $\mathsf{IPT}(f)$ returns the first IPT of a flow f, then our metric is

$$\eta = \frac{1}{|S|} \sum_{s \in S} \frac{\sum_{f \in F_s} \mathsf{IPT}(f)}{|F_s|} \tag{7.2}$$

If $|S| = 0$, then η is undefined.

η is a less precise metric than γ, since it uses fewer flows during connectivity disruptions, thus being more susceptible to skew. However, the combination of η and γ allows for strong inference. A decrease in γ may also mean that fewer packets than expected are actually being sent for a traffic class instead of being lost along the path, but η can help us distinguish between the two cases (i.e., assuming RFC-compliant behavior, η can distinguish between sending only two packets and a random loss of one of three packets). Figure 7.4 shows η calculated across all ASes for January 2012: the metric is slightly higher than the expected value of 3 for the entire period, with slight deviations corresponding to deviations in γ_3.

[4] Although RFC 6298, states that the RTO should be 1 second, we observe in the darknet that the RTO is still ~3 seconds for more than 99% of flows from Windows hosts. If an RTO of 1 second is more widely adopted, we can identify the RTO typically used by each source and normalize the metric.

7.2.3 Packet loss case studies

In this section, we evaluate our metric using three different service-disruption case studies. The first two outages – the "Dodo-Telstra" and the "Bell-Dery" case – had network-induced packet loss as a result of BGP route leaks [89, 200]. The third one – the Libyan Internet blackout – was the result of packet filtering. If effective, our metrics will reflect packet loss in the first two case studies but not in the last.

For each of the case studies, we only use metrics which were stable throughout the entire month preceding the outage.

"Dodo-Telstra" routing leakage

On February 23, 2012, around 2:40 UTC, the multi-homed network operator *Dodo* announced internal BGP routes to its provider *Telstra*, a major ISP in Australia, which erroneously accepted them. As a result, Telstra sent all of its traffic to the small network, Dodo, instead of a large transit provider, inducing a bottleneck leading to a complete outage [89]. The effect was massive: most Australians were left without Internet connectivity for about half an hour [91].

Figure 7.6 plots our metrics for IBR traffic originating from AS1221 (Telstra) calculated in 5-minute bins. The figure shows significant drops of both γ_C and γ_3 during the first phase (20 minutes) of the episode, meaning that far fewer packets per flow were reaching the darknet than normal. However, when γ_C and γ_3 first drop, η increases from about 3 to 5 seconds, which corroborates packet loss (assuming individual hosts did not change their retransmission patterns). This spike was calculated using the 7 distinct source IPs observed from this region at the darknet. In the following three 5-minute time bins the number of sources (0, 1, 2 respectively) contributing to the calculation of η was not statistically significant. Such a significant drop in γ_C and γ_3 and the increase in η are a direct consequence of congestion on the affected links. Routers started dropping

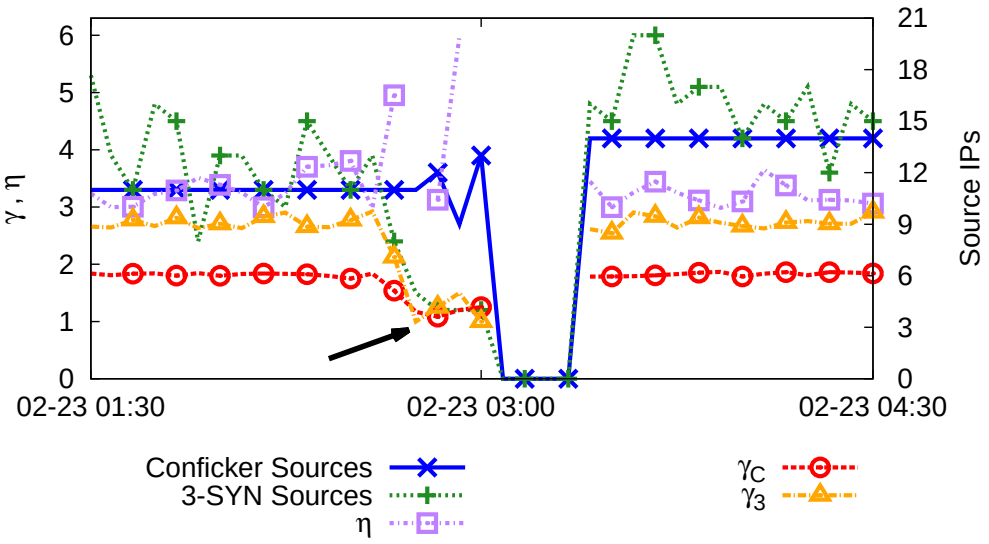


Figure 7.6. Our packet-loss metrics plotted in 5-minute bins for traffic originating from AS1221 during the Dodo-Telstra routing leak in February 2012. The arrow points at the first phase (20 minutes) of the outage, where the metric values indicate a bottleneck, i.e., packet loss: γ_C and γ_3 decreased, and η increased. The number of IPs sending Conficker traffic remained the same, while the number of IPs sending 3-SYN flows decreased – an artifact of the frequency at which each type of host contacts the darknet. In the second phase, no flows were observed in the darknet traffic, implying a complete outage.

packets, including some of the SYN packets destined to the darknet. Eventually, this congestion deteriorated to a complete outage (lasting another 20 minutes), during which the telescope did not observe any sources sending SYN packets from Telstra (so our metrics cannot be calculated).

"Bell-Dery" routing leakage

On August 8, 2012, at 17:27 UTC, dual-homed provider *Dery Telecom* started to leak a full BGP table to the major Canadian ISP *Bell*. These routes were accepted and propagated to Bell's peers [200]. Our analysis shows that the biggest disruption lasted about half an hour.

Figure 7.7 plots our metrics calculated for traffic coming from AS577 (Bell)

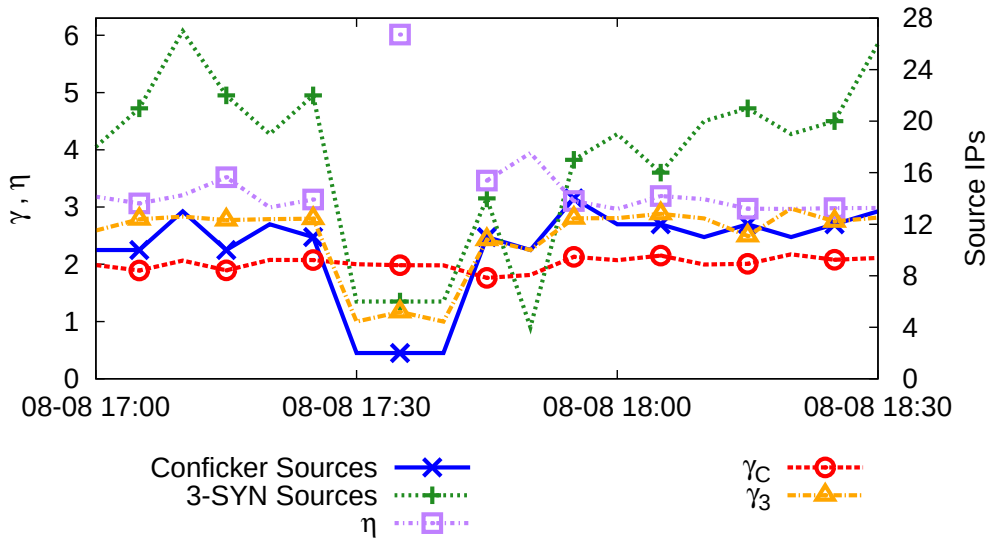


Figure 7.7. Metrics during uring the Bell-Dery routing leak of August 2012. we observed traffic from AS577 during every 5-minute bin. The number of Conficker and 3-SYN source IPs dropped drastically. Two of our metrics, γ_3 and η indicated packet loss, but the γ_C metric did not, which we later discovered was because one network was unaffected by the BGP leak.

surrounding the outage. The Bell network never was completely offline, but the plot indicates a severe disruption (∼17:30-17:45) followed by slight improvement (∼17:45-17:55) before restoration. During this time period, the total number of Conficker and 3-SYN source IPs dropped from about 12 and 20 to 2 and 6, respectively. Both γ_3 and η indicate significant packet loss during this time period. Strangely, γ_C stayed close to 2 during the worst part of the disruption, decreasing slightly when conditions appeared to improve (number of Conficker sources rose from 2 to 11).

To determine the reason behind the differences in γ_C and the other two metrics, we broke down the traffic from AS577 by network prefix and inspected the TTL header fields in the collected packets. In the 90 minutes surrounding the outage, packets from AS577 originated from 63 distinct /16 prefixes, of which 38 sent traffic in at least 9 of 18 5-minute time bins, and all but one experienced a considerable volume drop. Upon further inspection, two IP addresses in this prefix continued to transmit Conficker-like

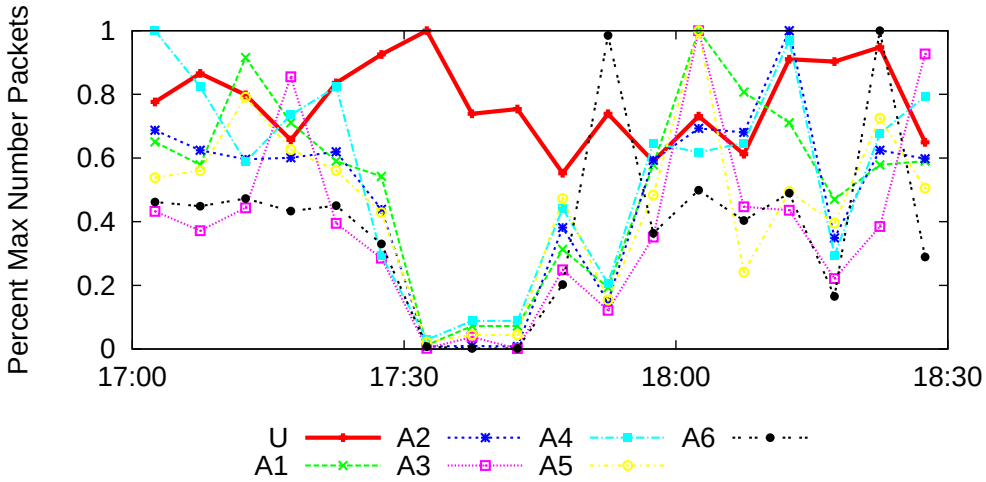


Figure 7.8. SYN traffic volume by prefix during Bell-Dery routing leak. One prefix (U) did not experience loss during the outage. Each point represents, for the given prefix, the fraction of packets sent during a 5-minute bin normalized to the time bin with the most packets. We show only prefixes observably active during all 18 time bins.

traffic at their pre-outage rate, depicted in Figure 7.8.

Since the Bell-Dery event was caused by a route leak, it is possible to observe changes in the way packets were routed by looking at the TTL value, reflecting a different number of hops in the path to the telescope. We discovered that the only two IP addresses whose packet rate at the telescope was not affected by the disruption were also the only two IP addresses whose packets carried a constant TTL both outside and during the disruption (one such IP address depicted in top graph of Figure 7.9). We suspect that traffic from this prefix was re-routed through a different path that was unaffected by the route leak.

Libyan Internet blackout

Our third case study applies our metrics to the Libyan Internet blackout happened in February and March 2011, when the Libyan government used BGP disconnection and later packet filtering to implement nationwide censorship [56]. There were three outages, lasting approximately 7 hours (the first two) and 3.7 days (the last one).

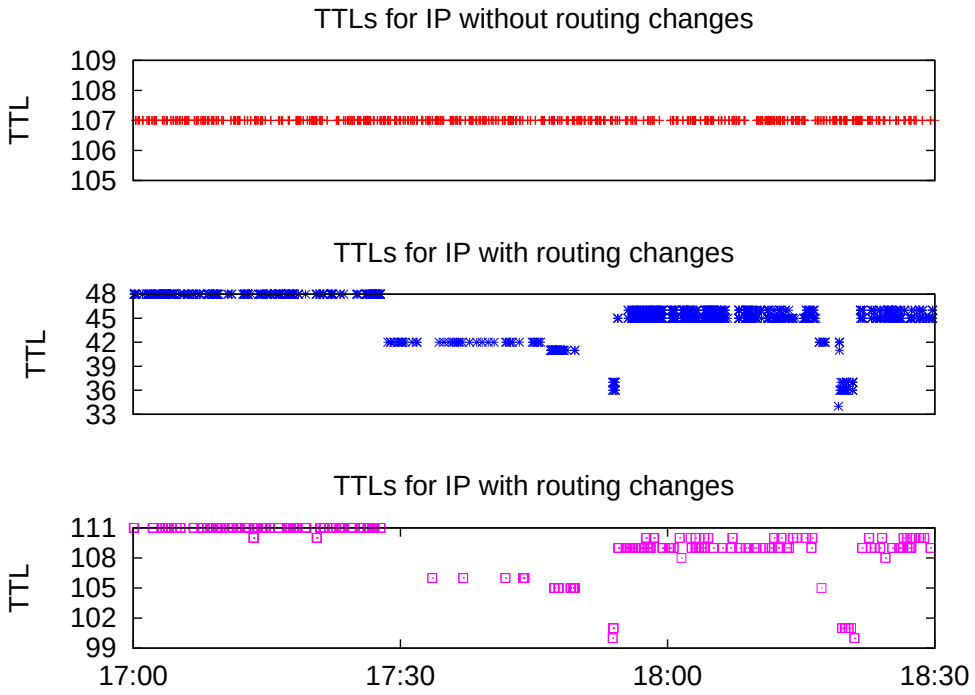


Figure 7.9. TTLs of an IP address with and without routing changes. During congestion, most packets reaching the darknet have a lower TTL than before the outage, indicating that they took a longer path. The top graph plots TTLs of packets from an IP address in the unaffected network, whose path to the telescope presumably does not change. Not until approximately 2012-08-08 21:20 (not shown) does the TTL of packets sent by the second and third IP address return to their pre-outage value.

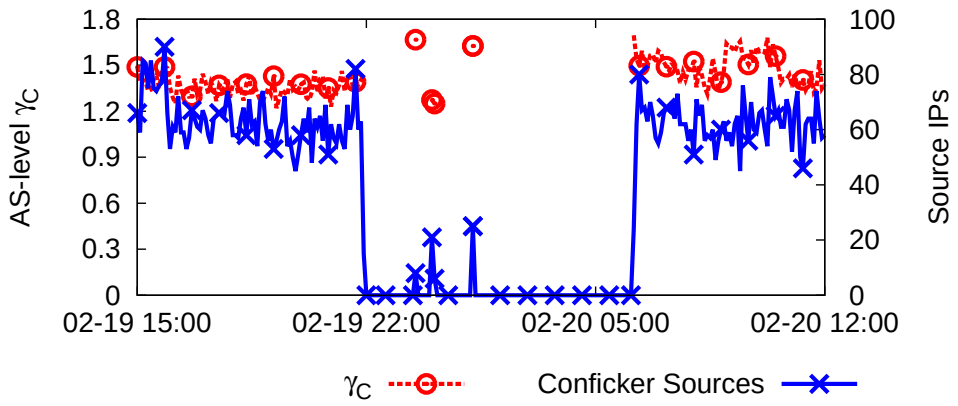


Figure 7.10. γ_C during a censorship event in Libya, which was induced by packet filtering. Libya's second 2011 outage used packet filtering as a method of censorship. Although there were fewer source IPs during the censorship, the hosts that did send Conficker-like flows sent approximately the same number of packets per flow as prior to the outage, indicated by the similar values of γ_C (calculated in 5 minute time bins). γ_3 and η are excluded as there were not enough hosts (2 or less) to accurately make inferences.

We examine the second one, when the state telecom (AS21003) isolated most of the country through packet filtering [56]. This case study illustrates that our metrics effectively distinguish large-scale outages that are characterized by some packet loss from those that are not.

Figure 7.10 shows that when a subset of hosts can communicate through the filtering system, γ_C remains near pre-censorship values, despite fewer sources sending traffic. Thus we can infer that the outage was not caused by an event inducing network packet loss. We excluded γ_3 and η from this measurement, since there were not enough hosts sending 3-packet SYN flows to accurately infer anything from these metrics.

Utility of metrics

In all three case studies, the metrics γ and η provided insight into the nature of the outages. In the "Dodo-Telstra" case study, network congestion preceded the complete outage. In response to congestion, the network dropped packets, decreasing the

number of packets per flow, which reduced the values of γ_C and γ_3 and increased η. In the "Bell-Dery" case study, the metrics extracted from the Conficker signal implied network-induced packet loss (e.g congestion). However, γ_C initially painted a different picture of packet loss: sources able to send Conficker-like traffic were unaffected. A deeper exploration of traffic volume by prefix and TTLs revealed that the connectivity disruption was more severe for some subnets than others. This result demonstrated that multiple data classes and metrics can strengthen the quality of inferences and provide a starting point for further investigation. In the Libyan Internet Blackout example, although the traffic volume was smaller, γ_C remained at pre-censorship levels whenever Conficker-like traffic was observed. This behavior is consistent with filtering packets by IP address or subnet: the number of traffic sources decreases but per-flow characteristics will not change.

7.2.4 Discussion

To augment the binary signal of presence or absence of traffic flows from a particular network, we explored IBR-derived metrics that help characterize connectivity disruptions that induce packet loss, e.g., link congestion. Our metrics are based on SYN retransmits in unsolicited Internet background radiation, visible from passive darknet instrumentation. Because these retransmits typically follow consistent patterns that are a function of operating system or application implementation, we can infer packet loss if some retransmits are not observed by the darknet.

We used three case studies to demonstrate that our γ and η metrics can distinguish a transit bottleneck-induced outage from an intentional nation-wide disconnection caused by packet filtering. One unexpected finding was that in the Bell-Dery route leak incident, different parts of the affected AS reacted differently to the route leak, confirmed by examination of TTL values on a per-prefix level. This analysis provided hints

on how to group parts of a specific AS into finer-grained units that may be affected differently by a disruption.

Our method has several limitations: it only measures packet loss between a given source and our darknet. It also relies on the presence of Conficker-like or IBR TCP traffic in general. But our simple metrics applied to a large darknet traffic segment enable us to continually monitor one aspect of network connectivity (i.e., reachability to our darknet) from all over the world.

Our method is complementary to techniques using active probes to discover outages. For example, research have detected outage by sending probes to highly responsive /24 blocks [165]; but, from our IPv4 address space utilization study, IBR includes some /24 blocks missed through active probing. Alternatively, a combination of ping and BGP data covers 89% of the Internet's edge address space but the focus is on failures lasting longer than 15 minutes [107].

Although we did not conduct a coverage analysis, this metric seems suitable for other connectivity scenarios and other darknet traffic.

7.3 Limitations of using IBR to analyze Internet status

In our path change and packet loss case studies, we discussed a number of limitations of IBR. In this section, we discuss two inherent limitations of using IBR to assess the status of hosts and networks on an Internet-wide scale. First, one of the main goals of this dissertation is to determine our ability to make *Internet-wide* inferences with IBR. While we can determine the coverage (number of analyzable networks) of our techniques, we cannot assess the coverage provided by IBR as a whole. Our coverage analyses are technique dependent, and, as we describe in Section 7.3.1, lower bounds on the insight IBR can provide. Second, we typically associate inferred information with a source address (or its corresponding network) originating the traffic. However,

the inferred information may reflect the status of a transit network routing the traffic (as opposed to the originating network). We discuss difficulties in pinpointing where an event occurred in Section 7.3.2.

7.3.1 Results are lower bounds

We show that our path change methodology permits continual analysis for about 1.5k ASes; although we did not perform a coverage analysis of our technique for analyzing packet loss, our technique intuitively applies to any network sending Conficker-like or 3-SYN packets. However, it is important to note that these coverage findings are for *our techniques* and are a *lower bound* on the number of IBR-analyzable networks. Our reasoning is two-fold. First, researchers are ingenious. We expect other researchers to develop better IBR-based techniques to assess network status. Second, when analyzing a particular event, we should pick parameters appropriate for the associated time period and network. For example, when identifying path changes, if few sources in a network send IBR, it is reasonable to increase the size of the time bin from 5 minutes to 10 minutes.

To further illustrate that a technique's coverage is a lower bound, we propose alternative method for detecting packet loss with IBR using Carna Botnet traffic. As we describe in Section 4.1, the Carna botnet [98] used an incremental scanning strategy. With this knowledge, we can infer for each machine in the botnet (1) the next IP address that will receive a packet (the previously scanned IP address + 70465) and (2) the time at which the next IP will receive a packet (based on the observed scanning rate). When these expected packets fail to reach our darknets, we can infer packet loss along the path.[5]

In `USCD-12`, we receive Carna botnet traffic from host in 5.1k ASes. Of these

[5] We could apply a similar method to any predictable scan. For example, if we know a host scans entire /24 blocks, we can check that we receive packets destined to every IP address in the block.

ASes, 1.9k did not have a Conficker host. Conversely, 3.5k ASes were visible with Conficker traffic but not Carna botnet traffic. Thus, there are cases where will are unable to apply our γ_C metric, but the alternative Carna-based method may reveal packet loss, and vice versa.[6]

This example illustrates that there are multiple ways to extract a predictable signal from IBR, and these signals may provide insight into different sets of networks. It may make sense to analyze both signals (e.g., increase our confidence that packet loss occurred) or select a single signal (e.g., there is no Carna botnet traffic in `UCSD-13`, so Conficker traffic is preferable for longterm analysis). This finding applies generally to IBR-based inferences of network status: each technique provides a lower bound on the potential insight IBR can provide.

7.3.2 Difficulty pinpointing location of change

As discussed in Section 7.1, our TTL-based inference indicates that a path change occurred but not where the on the route the change happened. This inability to pinpoint where a change occurred applies to other IBR-based inferences about network conditions.

Supplemented with a topology map, we may be able to hypothesize where a change occurred. If only one AS exhibits signs of a change then it is likely that the change occurred within the AS. If two ASes whose traffic eventually traverses the same link to reach the darknet both experience changes, it is likely that the common link is at fault. However, it is also possible that two independent changes occurred. For example, macroscopic outages caused by earthquakes affected all ASes with a geographic area [55].

One example where we cannot attribute packet loss to an individual link is during

[6] In addition to observing Conficker or Carna botnet traffic from an AS, to accurately apply our metrics, we need sufficient traffic volume and diversity of hosts.

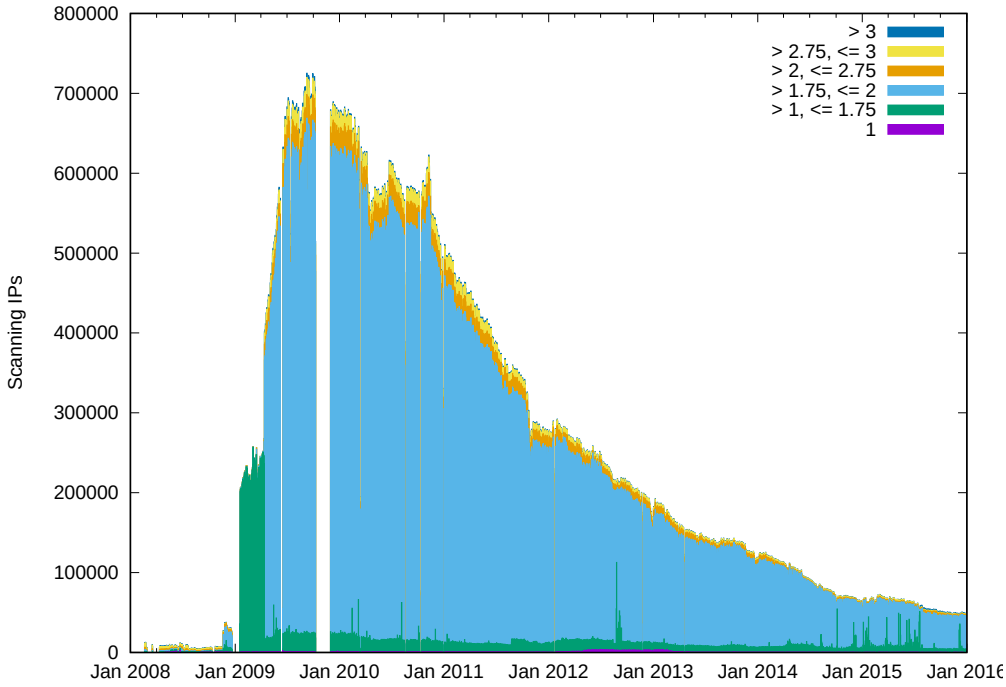


Figure 7.11. Average packets per flow for each source sending TCP port 445 traffic. Most sources sending TCP/445 packets are Conficker-infected hosts, so we expect an average of 2 packets per flow. When we observe sources sending fewer packets per flow, like at the start of the Conficker outbreak in early 2009, indicates packet loss.

the Conficker outbreak. Figure 7.11 shows the average packets per flow per day for each source IP address sending traffic on TCP port 445, which is related to our metric γ_C in Section 7.2. Since Conficker causes most sources to send TCP port 445 traffic, we expect the average packets per flow to be close to (but not exceed) 2. However, at the start of the Conficker outbreak in early 2009, this metric is between 1 and 1.75 for most sources, indicating packet loss. It is unclear, from this data, if this loss is due to an inability to capture all the traffic at UCSD-NT, or if many networks throughout the Internet experienced packet loss. Because of the sudden shift to approximately 2 packets per flow, and a corresponding increase in total sources, a link near UCSD-NT is a likely culprit.[7]

7.4 Conclusion

In this chapter we have applied IBR to two problems: detecting path changes and packet loss. These are valuable tools for assessing the status of networks throughout the Internet. In particular, IBR may provide historical insight into macroscopic events.

Many researchers already study Internet-wide outages [86, 95, 107], topology (including path changes) [11, 206, 175] and, to a lesser extent, packet loss. Our current IBR techniques do not outperform these efforts. However, combining IBR with other data sources should yield improved visibility. For path changes, we gain insight on the reverse path — which is difficult to measure with traditional techniques. For packet loss, existing metrics use bidirectional traffic; IBR can isolate the reverse path. Additionally, some sources sending IBR are in networks that do not host measurement devices (e.g., an Ark node or Routeviews BGP peer).

IBR alone is unlikely to provide insight into the location of an event. But we

[7] A CAIDA network administrator informed us that they imposed rate-limits on IBR until April 14, 2009.

could use dedicated probing to further investigate abnormalities detected with IBR. Reactive probing to passively detected events has yielded better analysis of outages [107]. In particular, combining active and passive measurements could reduce the number of active probes required to analyze macroscopic events, which generally involve distressed networks that are unlikely to welcome excessive probing by the measurement community. CAIDA is working on a system to combine signals from IBR, BGP updates, and Ark probes to detect large-scale outages [37].

Acknowledgements

Section 7.1, in part, is adapted from material as it appears in the proceedings of the Internet Measurement Conference (IMC 2015). Benson, Karyn; Dainotti, Alberto; claffy, kc; Snoeren, Alex C; Kallitsis, Michael; ACM, 2015. The dissertation author was the primary investigator and author of this paper.

Section 7.2, in full, is adapted from material as it appears in the proceedings of 2013 Traffic Monitoring and Analysis Workshop (TMA 2013). Benson, Karyn; Dainotti, Alberto; claffy, kc; Aben, Emile. IEEE, 2013. The dissertation author was the primary investigator and author of this paper.

Chapter 8

Conclusion

This dissertation rigorously evaluated the potential for IBR to improve our understanding of network utilization and conditions on an Internet-wide scale. First, since spoofed packets can cause IBR-based approaches to yield incorrect inferences, we developed and validated a method to remove these packets from IBR datasets. We analyzed IBR to discover phenomena suitable for Internet-wide measurement, that is phenomena generated by a large number of sources; often, bugs and misconfigurations produce these phenomena, instead of malicious traffic. Next, we investigated factors that influence our visibility into networks Internet-wide, including properties of IBR itself (e.g., who sends IBR? how often do they send IBR?) and the collection infrastructure (e.g., size of darknet, time of collection). With this knowledge, we finally inferred properties of networks and hosts generating IBR. Through our analysis of IBR and 11 case studies, we provide the following intuition on when researchers should (or should not) consider using IBR for opportunistic network analysis:

- *To improve findings.* In the field of Internet measurement, it is always advisable to use multiple data sources to improve findings. By using multiple data sources, including IBR, we increased the number of known used /24 blocks from 4.59M to 5.30M. Due to the assortment of traffic in IBR, it has the potential to contribute

to many different types of inferences. In general, additional data sources should be picked based on the diversity they provide. IBR may not be the best choice: other passively collected datasets outperformed IBR, in terms of the quantity of additional used /24 blocks, in our IPv4 address space utilization study. However, improvements in quality provided by IBR may trump any deficiencies in quantity (e.g., although we observe fewer open DNS resolvers with IBR than weekly scans by the Open Resolver Project, we frequently capture traffic from the DNS servers that send IBR).

- *To investigate differences.* Even in enumeration tasks where other data sources greatly outperform IBR's coverage, the differences in findings may reveal interesting insights. We were surprised to find router interfaces in our IBR dataset that were not in a traceroute dataset; this finding suggests that the traceroute data, while discovering router interfaces in more than 10 times the number of /24 blocks than IBR, is also incomplete. Moreover, a source's presence in IBR can provide additional context. For example, the open resolvers observed through IBR were part of a new type of attack on authoritative name servers.

- *To study networks with end users.* Darknets capture traffic from clients, servers and machines supporting the Internet's infrastructure. Dedicated probing for servers and infrastructure yields much better coverage, but is unlikely to identify IP addresses associated with end users. With IBR, we captured a similar number of client /24 blocks — as indicated by sending a BitTorrent payload — as passively monitoring traffic at a major European ISP. This large number of client blocks allowed us to infer uptime, patch effectiveness, DHCP dynamics, and CGN deployments.

- *To test Internet tools.* IBR can act as a diverse traffic sample for testing Internet

tools. We expected to be able to apply existing uptime and NAT heuristics directly to IBR. However, for both case studies we found inaccuracies in the tools' output. This information could be used to improve the tools, or alert users of instances where the tools are inaccurate. Such enhancements will increase the accuracy of the tool when applied to IBR, as well as, productive Internet traffic.

- *To measure properties unrelated to traffic payload.* IBR has many underlying biases (e.g., a large fraction of hosts sending IBR are infected with malware). Without a full understanding of these biases, it is difficult to accurately comment on the processes generating the traffic. For example, we are not able to assess BitTorrent client popularity because we cannot distinguish between the organic installations of each client, client-specific implementation quirks that produce additional IBR, and the victims of targeted attacks.

  We produce more accurate Internet-wide assessments when we infer properties unrelated to the traffic payload. For example, network configurations are unlikely to be correlated with BitTorrent usage. As a result, the same BitTorrent traffic that was unacceptable for analyzing client popularity yields significant insight into DHCP dynamics and CGN deployment. In general, using the IP or transport layer of the packet means correlation with IBR's underlying biases is less likely than analysis using application-layer information: we are able to infer uptime via TCP timestamps for two orders of magnitude more hosts than previous work leveraging the Witty worm's payload.

  We make a similar conclusion for inferences that require a degree of predictability in the observations. Our efforts are widely applicable when the predictable attributes are unrelated to the phenomena generating IBR. Since all IP packets have a TTL, we can apply our path change heuristic to any IBR traffic. In contrast,

as an alternative packet loss metric, we proposed a method that exploits patterns in Carna botnet scans. Since the Carna botnet is no longer scanning the Internet, this precise technique is no longer applicable. We could develop similar methods, that also infer scanning strategy, but those too will need to differentiate between service disruptions and other external factors (e.g., a botnet halts its Internet-wide scan). Our original packet-loss heuristic in Section 7.2 falls in between these two examples, as it is applicable to any traffic expected to send a fixed number of packets. However, when analyzing connections with two packets per connection attempt, we only consider Conficker, which has been decreasing in volume since 2009.

- *To reduce measurement overhead.* IBR should be able to reduce the number of active probes required to conduct enumeration and monitoring tasks. If we can infer the existence of a resource, or the state of a remote network with IBR, probing becomes redundant. Eliminating redundant probes will lighten machine and network loads, will is especially important when a remote network is under distress (e.g., during an outage).

  Moreover, there are inherent limitations to using IBR to infer network state: we only observe packets traversing paths to the darknet, and it is difficult to pinpoint where changes occur. For these reasons, it is unlikely that researchers would conduct in-depth analysis of network conditions exclusively through IBR. It seems more plausible to use IBR to continually monitor networks, but trigger additional active measurements based on IBR observations. This scenario would allow for equivalent coverage with less frequent active probing.

- *When benefits of using IBR outweigh the effort required to exclude spoofed traffic.* Excluding spoofed traffic is necessary for accurate analysis with IBR. Often, this

is as simple as extracting a specific type of traffic. However, for inferences that leverage as many packets as possible it is a non-trivial amount of work to remove spoofed packets using our sanitization technique.

8.1 Future directions

A natural extension of this work is to make additional inferences with IBR. We see potential in three main areas: leveraging an increased understanding of IBR phenomena, inferring network configurations, and combining active measurements with IBR. First, a better understanding of IBR phenomena will produce more specific inferences: identification of scanning techniques will produce better metrics for inferences requiring predictability; the ability to recognize packets originating from mobile users will produce a more detailed characterization of IPv4 utilization. Second, for network configurations, we can improve our understanding of IP address sharing and firewalls. For IP address sharing we can extend our DHCP and CGN work to: extract DHCP pools; differentiate between arbitrary and paired CGN deployments[1]; track users as they move between networks; and use IBR components other than BitTorrent to corroborate our findings and increase coverage. We may also be able to use IBR to infer firewall policies by: inspecting ICMP destination unreachable messages to deduce the hosts behind a firewall, and correlating spikes in spoofed traffic with networks experiencing extreme packet loss to identify networks that do not implement egress filtering. Finally, we are hopeful that IBR will be used in measurement pipelines: previous studies successfully triggered active probing from passive observations, resulting in better insight into service disruptions [107, 223].

[1]There are two ways in which internal addresses are mapped to external addresses: arbitrary and paired [14]. In the arbitrary case, an internal IP address may map to multiple external addresses at the same time. In the paired case, the same external address is used for all sessions associated with the internal address. For proper UDP functionality, RFC 4787 requires paired address pooling.

We expect that the ingenuity of other researchers will yield applications of IBR beyond the ones presented in this dissertation. The changing composition of IBR can serve as an inspiration. For example, we looked for open DNS resolvers in IBR after an increase in DNS traffic reaching our darknets. In particular, some future studies may not be possible with the current composition of IBR.

Beyond additional applications of IBR, this dissertation provides a framework for analyzing the utility of a data source for Internet-wide network analysis. It is future work to apply these metrics to other types of data (e.g., ping datasets, data passively collected from live networks). Such an analysis would permit researchers to compare multiple types of data, and make informed decisions based on their measurement needs.

8.2 Final thoughts

Often when speaking about this dissertation, people comment on the extravagance of using such a large portion of the address space for academic research. The main argument is that relinquishing our darknets could ease some of the difficulties in obtaining publicly routable IP addresses: ARIN's waiting list of organizations with an unmet need of IPv4 addresses could be fulfilled with a single /12 block [12] — $\frac{1}{64}$ of UCSD-NT. This sentiment is worth consideration: does the research stemming from IBR justify difficulties experienced in the productive Internet? In our opinion, continued operation of darknets, including the infrastructure to capture, process, store, and visualize "pollution" is a worthwhile endeavor.

First, switching to IPv6 — not relinquishing our darknet — is the long-term answer to the growing number of Internet devices. In the meantime, operators have deployed technologies to extend IPv4's lifetime: our analysis has shown wide adoption CGN and DHCP. Additionally, Internet registries should attempt to reacquire prefixes unannounced in BGP before darknets. Though, any solution that returns IPv4 addresses

to the Region Internet Registries pools, is only delaying the inevitable exhaustion.

Furthermore, darknets provide an opportunity to study new attacks and their victims: botnets still scan darknets [54], and some DDoS attacks still use TCP-SYN packets with spoofed darknet IP addresses [162]. However, compared to worm outbreaks, popular a decade ago, many popular attacks do not produce IBR, and the attacks that do produce IBR may be difficult to identify. In our experience, the attacks appearing in IBR are complex and novel (e.g., attacks on authoritative DNS servers, and BitTorrent index poisoning attacks).

However, the biggest benefit of collecting IBR is our ability to use the traffic for a variety of measurement tasks. This dissertation has shown that we gain considerable insight into address space utilization and network conditions with IBR. Our case studies are extremely varied: ranging from locating open resolvers to determining if a network deploys CGN to detecting packet-loss. We are unaware of a publicly available data source that provides more versatile, Internet-wide insight. We hope our work encourages others to leverage IBR in their measurement studies, and we are excited to see IBR's continued contribution to Internet-wide studies.

Appendix A

Attributing IBR to responsible Internet phenomena

Throughout this dissertation we refer to a number of IBR phenomena, including a breakdown of the top components in Section 4. However, isolating and attributing traffic to an individual IBR phenomenon can be nontrivial. Unlike Pang *et al.* who responded to unsolicited traffic [156], we passively collect IBR, which limits the amount of information we have to classify this traffic. Moreover, we show in Section A.1 that port-based analysis, used in previous characterizations of IBR [217], is insufficient to attribute traffic to a particular phenomenon.

In this appendix, we outline our effort at classifying IBR into components (that is, classes of phenomena responsible for different traffic) based on observations of initial communication attempts. We summarize our approach of isolating traffic and attributing it to a phenomenon in Section A.2.

A.1 Evidence that port-based techniques are insufficient

We examine the top TCP and UDP destination ports from the `UCSD-13` dataset (in terms of number of source IP addresses). The top TCP ports corresponding to many

Table A.1. Top TCP destination ports in UCSD-13. TCP/445, used by Conficker, is the top TCP port. Most of the other top TCP ports are associated with popular services, botnet, and P2P activity. One port, TCP/7111, is the result of traffic to a single UCSD-NT address.

TCP Destination Port	UCSD-13			
	IPs	/24s	ASes	Countries
445	6,438,158	633,410	7,513	202
80	1,984,799	530,324	11,396	217
3389	761,305	289,006	5,677	191
6881	413,169	119,915	5,449	161
443	204,433	98,779	4,043	183
7111	77,155	48,673	1,647	128
4662	75,577	63,461	1,839	128
6882	74,041	28,383	2,557	129
23	72,891	47,222	3,018	160
22292	71,996	45,125	2,080	163

common services, but we need to perform additional analysis to determine *why* traffic from this port reaches a darknet. For UDP, It is difficult to make sense of the top ports as none correspond to well-known services. However, examining the payload reveals that one application, Qihoo 360, is responsible for the entire top-10 list. As a result, with port-based analysis alone we cannot provide significant insight into the phenomena responsible for IBR.

A.1.1 Top TCP ports

Analyzing TCP-based IBR inherently requires a flow-based technique since the packets do not contain application-layer payloads. However, just using TCP port information is not enough to attribute flows to the process that generated them. Destination ports do not provide information as to whether or not traffic is the result of a scan or a widespread misconfiguration. Similarly, source ports do not provide information as to whether or not traffic is the result of a DoS attack or low-volume backscatter.

Even when we know which process generates the majority of the traffic associated with the port, analyzing ports alone does not immediately reveal the process'

growth or decline. Port 445 tops the list of TCP destination ports (Table A.1) with 6.4M associated IP addresses. This port, used by Conficker, was an influential component in previous characterizations of IBR [217, 156]; though the volume has decreased significantly from 72.5% of total packets in 2010 [217] to 27.8% of non-spoofed total in UCSD-13. This decease is not sufficient to conclude that Conficker declined substantially during this time period.[1] In particular, in UCSD-13, 258k IP addresses send TCP/445 packets to ranges not targeted by Conficker. Similar analysis from 2010 is necessary to comment on Conficker's decline.

Unfortunately, a TCP port's popularity may not reflect the amount of general interest in the port from the live Internet (e.g., the likelihood of a port being scanned, attacked, or hosting a service). Popular services (TCP/445, TCP/80, TCP/443, TCP/3389, TCP/23), P2P activity (TCP/6881, TCP/4662, TCP/6882), and botnets (TCP/22292 [147]) do account for most of the top TCP destination ports in Table A.1. However, a one-off behavior can influence the ordering of the top ports. A single darknet address, UCSD.202.190.88, receives traffic from over 70k source IP addresses on TCP/7111 — a port that is not associated with common applications.

Collectively, the IP addresses in Table A.1 account for 63% of the 15.6M IP addresses sending TCP IBR in UCSD-13. TCP backscatter contributes another 366k IP addresses. As a result, this port-based analysis covers less than two-thirds of sources sending TCP traffic.

A.1.2 Top UDP ports

For each of the top-10 UDP destination ports in UCSD-13, most traffic results from a bug in Qihoo 360 software, which we describe in Chapter 4.3.1. In UCSD-13, we observe over 46M IP address as a result of the top UDP port, UDP/39455. Other work

[1] Additional analysis, in Section 4.1, reveals that Conficker declined significantly from 2010 to 2013.

Table A.2. Top UDP destination ports in UCSD-13. All of the top 10 UDP ports are the result of Qihoo 360 traffic.

UDP Destination Port	UCSD-13			
	IPs	/24s	ASes	Countries
39455	46,723,695	650,429	3,568	205
29991	23,546,302	636,511	4,066	217
29735	22,829,491	640,798	4,255	215
30247	21,990,060	623,568	4,017	214
15399	19,966,580	568,302	2,985	202
30503	14,048,207	570,823	3,383	210
4647	13,931,780	513,987	2,443	190
4903	11,890,457	496,654	2,238	186
30759	10,073,382	539,718	3,150	207
5159	9,015,693	488,792	2,162	188

included this port as a top contributor of IBR traffic [51, 177]. As a result, the analysis of top UDP ports only reveals the scale of the Qihoo 360 bug.

A.2 Our approach to traffic attribution

Fortunately, our darknets collect packets with application-layer payloads. These packets contain a variety of information — beyond ports — that is useful for the classification of IBR. Still, attributing traffic to a phenomenon responsible for the abnormally high number of sources can be challenging. We do not have control over the hosts that send IBR; in particular, we cannot check which software is installed. The analysis is also difficult since IBR researchers do not publicly share signatures for known phenomena.

In some cases, the packets use common protocols. For example, from Section 5.3.1, two IP hotspots in UCSD-13 are due to traffic on UDP port 53; further inspection revealed that they were DNS queries, as expected. Similarly, many TCP SYN packets sent to a single darknet IP indicate that a darknet IP address is mistaken for a server. In some cases, we can conjecture about the type of service (e.g., TCP/80 is likely a webserver), but we are uncertain regarding some ports like TCP/3906.

Responding to the traffic may reveal more information about an abnormality. For example, sending TCP SYN-ACK packets in response to traffic on to port 80 hotspots could expose which web sites the sources contacting the darknet are attempting to access. But this technique only works when we know the application-layer protocol used (and we often cannot identify the application-layer protocol). Additionally, we analyze historical data. Sources sending traffic may no longer generate IBR at the time of analysis.

The remainder of the section outlines additional tools and techniques we use when we are uncertain of the payload associated with IBR.

A.2.1 Existing protocol identification tools

Libprotoident [8], from the University of Waikto, analyzes header information and the first four bytes of a packet (for traffic in live networks, it analyzes both directions). They currently have signatures for over 250 applications. Similarly, Wireshark [45] has built in support for dissecting many protocols. With Wireshark, we manually decode packets [115] and check that there are no errors for the suspected protocol. For example, we identify traffic reaching more than 5 IP hotspots as eMule by running Libprotoident. We examine the eMule specification [111] to confirm the packets' origin. Additionally, we create filters to capture all eMule traffic — not just traffic to the IP hotspots.

A.2.2 Literature on well-studied protocols

Security companies often release white papers on major Internet threats. We leverage these existing analyses to attribute IBR to the malware that generated it. In Section 4.1, we determine which packets are associated with Conficker from its scanning patterns [41]. In Section A.2.7, we determine which packets are associated with

the ZeroAccess and Sality botnets from analysis of the botnets' command and control channels [136, 65] — despite the packets appearing to have an encrypted payload.

One interesting piece of literature is about the Carna Botnet [98], where the bot master released details of their own botnet and the data it discovered from scans. We determine which packets originate from the Carna Botnet in Appendix B.

A.2.3 Web search for common text

Often traffic contains common strings, which we can query in a search engine. For example, traffic whose payload includes "Tsource Engine Query" is due to Steam [183].

Searching is often an iterative process. A number of packets to X.0.0.0 start with the string "SRNT." A Google search for "SRNT udp packet" returns a larger packet trace [173]. From this packet trace we extract additional starting bytes such as "ANNO" and "NANC." A Google search for "' SRNT' 'ANNO' 'NANC'" returns a Chinese message board where students are trying to figure out how to bypass by monitoring software [225]. The forum mentions StudentMain.exe, an executable associated with Classroom Management by Mythware [69].

A.2.4 Analysis of other traffic to a hotspot

The majority of packets to BitTorrent hotspots use the DHT or uTP protocols over UDP. However, there is also considerable TCP SYN traffic and encrypted UDP traffic to the same addresses. These TCP SYN and encrypted UDP packets often use the same ports as the DHT or uTP traffic. This traffic is likely due to older clients that do not support uTP (and still use TCP for downloading torrents) or clients that use BitTorrent's encryption protocol. Looking at all traffic reaching a hotspot allowed us to attribute flows that would be unclassified if analyzed in isolation.

A.2.5 Running software

We can run software suspected of sending IBR for additional analysis. For example, to check if Mythware products send the "SRNT" packets, we installed and ran the trial version of the product in a virtual machine. Running suspected software can also help determine why we receive traffic — not just what type of software sends the packets. In particular, we may be able to determine if we receive traffic due to software bugs or attacks.

A.2.6 Analysis of hosts sending traffic

One payload signature originates from over 1M /24 blocks in `UCSD-13`, `UCSD-12`, and `MERIT-13`. It is a non-encrypted 30 bytes of payload. We could not identify the protocol associated with the packets using the previously mentioned methods. However, through the analysis of the sources sending the traffic we attribute packets to Qihoo 360. We give a detailed explanation in Section 4.3.1. At a high-level, we identified IP addresses sending the traffic in the address space monitored by other researchers; we then analyzed bidirectional traffic from these hosts.

A.2.7 Differentiating between encryption and obfuscation

We can test if the traffic appears to be random bytes — a characteristic of encrypted messages. A simple test is to calculate the entropy of the message payload and check that it is close to $\log_2(|\text{payload}|)$. While this test is not precise for a single small packet [82], we can still get a sense of whether the traffic appears random with many samples.

Often, a seemingly random payload is obfuscated, not encrypted. For example, we found that one hotspot receives seemingly random payloads on port UDP/16464, a port associated with ZeroAccess command and control [147]. Analysis of the botnet

```
16:00:33.000064 IP 189.191.46.255.63057 > X.238.254.254.16464: UDP, length 16
        0x0000:  4500 002c 0b28 0000 7111 XXXX bdbf 2eff  E..,.(..q.%.....
        0x0010:  XXee fefe f651 4050 0018 1270 3b30 1e00  ,....Q@P...p;0..
        0x0020:  2894 8dab c9c0 d199 7eee 7447            (.......~.tG
```

Figure A.1. ZeroAccess command and control packet. The payload of this packet is obfuscated. We can check that the boxed bytes, with swapped byte order, satisfy: 0xAB8D9428 $\wedge$ (0x66747032 $\ll$ 1) = 0x6765744c ("getL" in ASCII). .

```
16:00:06.000065 IP 111.248.55.49.51956 > X.16.56.246.7605: UDP, length 19
        0x0000:  4500 002f 6c48 0000 7011 XXXX 6ff8 3731  E../lH..p..Fo.71
        0x0010:  XX10 38f6 caf4 1db5 001b 8298 7133 0f00  ,.8.........q3..
        0x0020:  643e c2d4 2cf5 42b5 810f 7f01 5344 1e    d>..,.B.....SD.
```

Figure A.2. Sality command and control packet. The boxed bytes act as an RC4 key to obfuscate the remainder of the payload. After using the RC4 key, the first 6 bytes are "0x038200000003," which correspond to: Version 0x03, URL Pack Sequence ID 0x82000000, and Command 0x03 (Pack Exchange).

reveals that the payload is obfuscated using an XOR scheme [136]. Using this scheme, we check that the fourth to eighth bytes of the deobfuscated are the bytes "getL." We show an example in Figure A.1. We compute the XOR and the "getL" test on all UDP packets and find additional traffic — not on port 16464 — that also appear to be ZeroAccess command and control packets.

We use a similar methodology to identify Sality command and control packets. For one hotspot, the third byte of the UDP payloads is twelve less than the UDP length. The format of a Sality command and control message is: [2 bytes hash] [2 bytes length] [RC4 encrypted data] [65]. The hash and length of the data also double as the 4-byte RC4 key for the encrypted data. When deobfuscated the packets have a deterministic payload. We show a sample Sality packet in Figure A.2.

This method requires manual processing and investigation. Even when we find a pattern that describes most of the traffic, it is difficult to find the responsible protocol. For example, there are two IP hotspots that receive UDP packets that have 13 bytes of seemingly random payload, and the first byte is normally 0x02. Figure A.3 shows a sample of packet of unknown origin. However, we do not know the process that

```
16:00:00.007271 IP 209.13.97.34.6294 > X.255.66.92.47890: UDP, length 13
        0x0000:  4500 0029 0676 0000 6611 XXXX d10d 6122  E..).v..f.....a''
        0x0010:  XXff 425c 1896 bb12 0015 03aa 0262 7037  ,.B:.......bp7
        0x0020:  b62b 5ec9 fd16 e340 1f                   .+^....@...
```

Figure A.3. Encrypted packet of unknown origin.

generates these packets.

Closely related is work on generating IDS signatures for C&C encryption [178]. In this work, the authors come up with probabilistic vector signatures for encrypted traffic. We could apply the technique to darknet IP hotspots to characterize random payloads. Then, we could look for similar payloads destined to other darknet addresses. However, since probabilistic vector signatures are not publicly avaialble, we still cannot attribute the traffic to a protocol.

A.3 Discussion

One of the contributions of this dissertation is a modern classification of IBR. Attribution of IBR to the generating phenomena enriches this classification. Attribution provides insight into why we observe the traffic. Automated methods, such as the one used by Brownlee [35], could isolate and characterize new classes of IBR. However, determining which software generates IBR, remains a challenge in IBR research.

Appendix B

Scanning strategy heuristics

In Section 4.1 we commented on Internet-wide scanning campaigns. We assumed that all hosts participating in the campagin use the same technique. To identify such hosts, we developed flow-level heuristics for scanning strategies and applied the heuristics to eight years of IBR (Figure 4.3). In this appendix, we specificy these heuristics.

Our heuristics work as follows. For each IP address we tag as a scanner, we call *IPs* the set of all darknet IP addresses scanned, i.e., *IPs* ={UCSD.B.C.D | UCSD.B.C.D is scanned}. Based on this set, we then report:

- The size of the range of addresses scanned: δ =max({B$\times 2^{16}$+ C$\times 2^{8}$+ D | UCSD.B.C.D $\in$ *IPs*}) - min({B$\times 2^{16}$+ C$\times 2^{8}$+ D | UCSD.B.C.D $\in$ *IPs*})
- The number of /16 network scanned: $\mathscr{B} = \big|\{\text{B}|\ \text{UCSD.B.C.D} \in \mathit{IPs}\}\big|$
- The number of unique "C" values: $\mathscr{C} = \big|\{\text{C}|\ \text{UCSD.B.C.D} \in \mathit{IPs}\}\big|$
- The number of unique "D" values: $\mathscr{D} = \big|\{\text{D}|\ \text{UCSD.B.C.D} \in \mathit{IPs}\}\big|$
- The number of Conficker destinations: $\mathfrak{C} = \big|\{\text{UCSD.B.C.D}|\ \text{UCSD.B.C.D} \in \mathit{IPs} \wedge \text{B} < 128 \wedge \text{D} < 128\}\ \big|$

With these statistics we identify the following classes of scanning strategies: sequential, reverse-byte order, Conficker, and Random.

B.1 Sequential strategies

For the "Complete" and "Incremental" scanning strategies we use δ, the difference between the maximum and minimum address scanned. We infer "Complete" when δ is about the same as the $|IPs|$. We infer "Incremental" when δ modulo $|IPs| = 0$.

An Incremental scanner may cycle through the address space more than once. For popular increments, we develop additional increment-specific heuristics. For example, most of the scanners to both TCP/23 and TCP/210 use a stepwidth of 134218, 137574 or 140929. When a host scanned TCP/23 or TCP/210 we check that $|IPs|$ and δ are consistent with cycling through the address space five or fewer times.

Another special use case is the Carna botnet [98] that used an increment of 70465 to scan many ports. For each hour, if we determine a port is being scanned with a stepwidth of 70465, we mark all other unclassified scans of the same port as "Carna."

B.2 Reverse-byte order strategy

We did not find very many reverse-byte order scans in our longitudial analysis in Section 4.1; however, a stealthy /0 scan used this strategy [54]. Our heuristic for this scan is $\mathscr{D} \approx \frac{|IPs|}{256 \times 256}$, $\mathscr{C} \approx \min(\frac{|IPs|}{256}, 256)$ and $\mathscr{B} \approx \min(|IPs|, 256)$

B.3 Conficker

In the past, researchers exploited the fact that most TCP/445 traffic captured in a darknet originated for Conficker-infected machines. For example, Dainotti *et al.* consider all TCP/445 traffic with certain packet lengths to be Conficker-like [56]; in

Section 7.2.1 we use all TCP/445 traffic with certain packet lengths and from certain operating systems. As Conficker declines, we expect these methods to become less reliable. There is nothing inherent about 48-byte packets and Conficker: we can receive length 48 packets from arbitrary applications and operating systems.

Fortunately, Conficker has a quirk, which we can leverage to identify the traffic it originates. There is a "bug" in Conficker's pseudorandom number generator.[1] When scanning non-locally (the mode used to scan /8 darknets), Conficker only sends packets to IP addresses A.B.C.D where B < 128 and D < 128 [41].

Sargent *et al.* consider packets to be "Likely Conficker" if it is a TCP/445 SYN to an address Conficker targets [180]. However, this method overcounts the Conficker population: one quarter of the packets from a scan of the entire darknet are to Conficker ranges. A basic improvement to this method is to also exclude packets that come from hosts that also target non-Conficker ranges. This still overcounts, because small complete (e.g., to UCSD.0.0.0/25) or incremental scans (e.g., of the .1 addresses in UCSD.0.0.0/9) never target non-Conficker ranges.

Instead, we check if scanning strategy is consistent with randomly scanning the Conficker ranges. If we want high confidence that traffic is sent by Conficker we choose traffic where (1) only Conficker destinations are targets ($\mathfrak{C} = |IPs|$), (2) $|IPs|$ is large enough and (3) $\mathscr{B}$, $\mathscr{C}$, $\mathscr{D}$ are appropriate values given $|IPs|$. With a 95% probability, Conficker hosts sending packets to at least 34 darknet IP addresses will target at least one value B (and by the same argument D) twice, i.e, $\mathscr{B} < |IPs|$. We use this criteria in Figure 4.3 to capture Conficker's decline over recent years.

Interestingly, with this heuristics, we find a host scanning with a Conficker strategy starting on August 9, 2008 — two and a half months before the discovery of Con-

[1] Conficker has other quirks that we could also leverage. For example, Conficker sends only one retransmission packet (most Windows machines send two), probably due to an abnormally short timeout [5].

ficker. The first two hosts geolocate to the Guangdong province in China. It is possible the creators of Conficker used these addresses to test the worm before releasing it.

However, with the high confidence heursitics, decribed above, we miss a large portion of Conficker traffic. First, some Conficker infected machines are behind NAT devices. In this scenario, there may be a mix of Conficker and non-Conficker traffic on TCP port 445.[2] Second, many Conficker infected machines send less than 34 packets per hour (the time granularity we analyze). For Table 4.1, we relax the criteria: we consider all sources where $|IPs| \geq 4$, at least 95% of scans are to Conficker-targets, and $\mathscr{B} \geq 3$. This relaxed criteria results in almost no difference in observed /24 blocks on TCP port 445 between UCSD.0.0.0/9 and UCSD.128.0.0/9 while excluding small sequential and "Random" scans.

B.4 Random strategies

For strategies involving (seemingly) random selections, we extend the intution of the Conficker heuristic. By choosing appropriate values of $\mathscr{B}$, $\mathscr{C}$, and $\mathscr{D}$, checking for the various types of random scanning is a generalized form of the birthday problem. Specifically, when determining if a scan is "Random", we ask, given $|IPs|$ scans that target 256 possible values (i.e., birthdays) for each $\mathscr{B}$, $\mathscr{C}$, and $\mathscr{D}$, how many unique values can we expect?

[2]Note that our technique to identify Conficker traffic does not estimate the number of infected machines. Due NAT and DHCP, the number of IP addresses sending malicious traffic is not equivalent to the number of infected machines [167, 105]. However, Weaver used Lévy's form of the Central Limit Theorem to estimate the size of the Conficker population [212].

References

[1] URL: http://seclists.org/nanog/2009/Feb/2.

[2] Internet Addresses Census dataset, PREDICT ID: USC-LANDER/internet_address_census_it55w-20130723/rev3638. Traces taken 2013-07-23 to 2013-08-25. Provided by the USC/LANDER project (http : //www.isi.edu/ant/lander).

[3] Internet Addresses Census dataset, PREDICT ID: USC-LANDER/internet_address_census_it49c-20120731/rev3167. Traces taken 2012-07-31 to 2012-09-02. Provided by the USC/LANDER project (http : //www.isi.edu/ant/lander).

[4] 360 Total Security Software License and Service Agreement. URL: https://www.360totalsecurity.com/en/license/360-total-security/.

[5] E. Aben. Conficker/Conflicker/Downadup as seen from the UCSD Network Telescope. 2008. URL: http://www.caida.org/research/security/ms08-067/conficker.xml.

[6] Bernhard Ager, Nikolaos Chatzis, Anja Feldmann, Nadi Sarrar, Steve Uhlig, and Walter Willinger. Anatomy of a Large European IXP. In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '12)*, 2012.

[7] Akamai Warns Of UPnP Devices Used In DDoS Attacks. October 2014. URL: https://www.akamai.com/us/en/about/news/press/2014-press/akamai-warns-of-upnp-devices-used-in-ddos-attacks.jsp.

[8] Shane Alcock and Richard Nelson. Libprotoident: Traffic Classification Using Lightweight Packet Inspection. Technical report. WAND Network Research Group, 2012.

[9] Mark Allman, Vern Paxson, and Jeff Terrell. A Brief History of Scanning. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC '07)*, 2007.

[10] David S. Anderson, Chris Fleizach, Stefan Savage, and Geoffrey M. Voelker. Spamscatter: Characterizing Internet Scam Hosting Infrastructure. In *Proceedings of the 16th USENIX Security Symposium (USENIX Security '07)*, 2007.

[11] Archipelago Measurement Infrastructure. 2006. URL: www.caida.org/projects/ark.

[12] ARIN. WAITING LIST FOR UNMET REQUESTS. April 2015. URL: https://www.arin.net/resources/request/waiting_list.html.

[13] Grenville J Armitage. Inferring the Extent of Network Address Port Translation at Public/Private Internet Boundaries. Technical report. Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, 2002.

[14] F. Audet and C. Jennings. Network Address Translation (NAT) Behavioral Requirements for Unicast UDP. RFC 4787 (Best Current Practice). Updated by RFCs 6888, 7857. Internet Engineering Task Force, January 2007. URL: http://www.ietf.org/rfc/rfc4787.txt.

[15] M. Bagnulo, P. Matthews, and I. van Beijnum. Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers. RFC 6146 (Proposed Standard). Internet Engineering Task Force, April 2011. URL: http://www.ietf.org/rfc/rfc6146.txt.

[16] M. Bailey, E. Cooke, F. Jahanian, and D. Watson. The Blaster Worm: Then and Now. *IEEE Security & Privacy*, 3(4):26–31, July 2005.

[17] Michael Bailey, Evan Cooke, Farnam Jahanian, and Jose Nazario. The Internet Motion Sensor - A Distributed Blackhole Monitoring System. In *Proceedings of the Network and Distributed System Security Symposium (NDSS'09)*, 2005.

[18] Michael Bailey, Evan Cooke, Farnam Jahanian, Niels Provos, Karl Rosaen, and David Watson. Data Reduction for the Scalable Automated Analysis of Distributed Darknet Traffic. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement (IMC '05)*, 2005.

[19] F. Baker. Requirements for IP Version 4 Routers. RFC 1812 (Proposed Standard). Updated by RFCs 2644, 6633. Internet Engineering Task Force, June 1995. URL: http://www.ietf.org/rfc/rfc1812.txt.

[20] F. Baker, W. Harrop, and G. Armitage. IPv4 and IPv6 Greynets. RFC 6018 (Informational). Internet Engineering Task Force, September 2010. URL: http://www.ietf.org/rfc/rfc6018.txt.

[21] Paul Barford, Rob Nowak, Rebecca Willett, and Vinod Yegneswaran. Toward a Model for Source Addresses of Internet Background Radiation. In *Proceedings*

of the International Conference on Passive and Active Network Measurement (PAM '06), 2006.

[22] Steve Beattie, Seth Arnold, Crispin Cowan, Perry Wagle, Chris Wright, and Adam Shostack. Timing the Application of Security Patches for Optimal Uptime. In *Proceedings of the Sixteenth Systems Administration Conference (LISA '02)*. Volume 2, 2002, pages 233–242.

[23] Steven M. Bellovin. A Technique for Counting NATted Hosts. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement (IMW '02)*, 2002.

[24] Peter Benko and Andras Veres. A Passive Method for Estimating End-to-End TCP Packet Loss. In *Proceedings of the Global Telecommunications Conference (GLOBECOM'02)*. IEEE, 2002.

[25] K. Benson, A. Dainotti, k. claffy, and E. Aben. Gaining Insight into AS-level Outages through Analysis of Internet Background Radiation. In *Proceedings of the International Workshop on Traffic Monitoring and Analysis (TMA '13)*, 2013.

[26] Robert Beverly. A Robust Classifier for Passive TCP/IP Fingerprinting. In *Proceedings of the International Workshop on Passive and Active Network Measurement (PAM '04)*, 2004.

[27] Robert Edward Beverly IV. Statistical Learning in Network Architecture. AAI0820515. PhD thesis. MIT, 2008.

[28] Robert Beverly and Steven Bauer. The Spoofer Project: Inferring the Extent of Source Address Filtering on the Internet. In *Proceedings of the USENIX Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI '05)*, 2005.

[29] Robert Beverly, Matthew Luckie, Lorenza Mosley, and kc claffy. Measuring and Characterizing IPv6 Router Availability. In *Proceedings of the International Conference on Passive and Active Network Measurement (PAM '15)*, 2015.

[30] BIND remote denial of service. July 2013. URL: https://www.freebsd.org/security/advisories/FreeBSD-SA-13:07.bind.asc.

[31] Zachary S Bischof, John S Otto, and Fabián E Bustamante. Distributed Systems and Natural Disasters. *Proceedings of the Special Workshop on Internet Disasters (SWID)*, 2011.

[32] Zachary S Bischof, John S Otto, Mario A Sánchez, John P Rula, David R Choffnes, and Fabián E Bustamante. Crowdsourcing ISP Characterization to the Network Edge. In *Proceedings of the 1st ACM SIGCOMM Workshop on Measurements up the Stack (W-MUST)*, 2011.

[33] Elias Bou-Harb, Mourad Debbabi, and Chadi Assi. Behavioral Analytics for Inferring Large-Scale Orchestrated Probing Events. In *Proceedindgs of 2014 IEEE INFOCOM Workshops (INFOCOM WKSHPS)*, 2014.

[34] R. Braden. Requirements for Internet Hosts - Communication Layers. RFC 1122 (INTERNET STANDARD). Updated by RFCs 1349, 4379, 5884, 6093, 6298, 6633, 6864. Internet Engineering Task Force, October 1989. URL: http://www.ietf.org/rfc/rfc1122.txt.

[35] N. Brownlee. One-way Traffic Monitoring with iatmon. In *Proceedings of the International Conference on Passive and Active Network Measurement (PAM '12)*, 2012.

[36] Fabián E. Bustamante and Yi Qiao. Friendships that Last: Peer Lifespan and its Role in P2P Protocols. In *Proceedings of the 8th International Workshop on Web Content Caching and Distribution (WCW '03)*, 2003.

[37] CAIDA. Detection and analysis of large-scale Internet infrastructure outages (IODA). 2012. URL: https://www.caida.org/funding/ioda/.

[38] Martin Casado and Michael J Freedman. Peering Through the Shroud: The Effect of Edge Opacity on IP-Based Client Identification. In *Proceedings of the 4th USENIX Conference on Networked Systems Design & Implementation (NSDI '07)*, 2007.

[39] Martin Casado, Tal Garfinkel, Weidong Cui, Vern Paxson, and Stefan Savage. Opportunistic Measurement: Extracting Insight from Spurious Traffic. In *Proceedings of the 4th ACM Workshop on Hot Topics in Networking (HOTNETS-IV)*, 2005.

[40] Weifeng Chen, Yong Huang, Bruno F. Ribeiro, Kyoungwon Suh, Honggang Zhang, Edmundo de Souza e Silva, James F. Kurose, and Donald F. Towsley. Exploiting the IPID Field to Infer Network Path and End-System Characteristics. In *Proceedings of the International Workshop on Passive and Active Network Measurement (PAM '05)*, 2005.

[41] Eric Chien. Downadup: Attempts at Smart Network Scanning. January 2009. URL: http://www.symantec.com/connect/blogs/downadup-attempts-smart-network-scanning.

[42] David R Choffnes and Fabián E Bustamante. Taming the Torrent: A Practical Approach to Reducing Cross-ISP Traffic in Peer-to-Peer Systems. In *Proceedings of the ACM SIGCOMM 2008 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '08)*, 2008.

[43] T. Chown. IPv6 Implications for Network Scanning. RFC 5157 (Informational). Obsoleted by RFC 7707. Internet Engineering Task Force, March 2008. URL: http://www.ietf.org/rfc/rfc5157.txt.

[44] CIA. The World Factbook: Population. 2013. URL: https://www.cia.gov/library/publications/the-world-factbook/rankorder/2119rank.html.

[45] Gerald Combs. Wireshark: Go Deep. 2015. URL: https://www.wireshark.org/.

[46] Evan Cooke, Michael Bailey, Farnam Jahanian, and Richard Mortier. The Dark Oracle: Perspective-Aware Unused and Unreachable Address Discovery. In *Proceedings of the 3rd USENIX Conference on Networked Systems Design & Implementation (NSDI '06)*, 2006.

[47] Evan Cooke, Michael Bailey, Z. Morley Mao, David Watson, Farnam Jahanian, and Danny McPherson. Toward Understanding Distributed Blackhole Placement. In *Proceedings of the 2004 ACM Workshop on Rapid Malcode (WORM '04)*, 2004.

[48] Evan Cooke, Z. Morley Mao, and Farnam Jahanian. Hotspots: The Root Causes of Non-Uniformity in Self-Propagating Malware. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN '06)*, 2006.

[49] M. Cotton and L. Vegoda. Special Use IPv4 Addresses. RFC 5735 (Best Current Practice). Obsoleted by RFC 6890, updated by RFC 6598. Internet Engineering Task Force, January 2010. URL: http://www.ietf.org/rfc/rfc5735.txt.

[50] Ítalo Cunha, Renata Teixeira, and Christophe Diot. Measuring and Characterizing End-to-End Route Dynamics in the Presence of Load Balancing. In *Proceedings of the International Conference on Passive and Active Network Measurement (PAM '11)*, 2011.

[51] Jakub Czyz, Kyle Lady, Sam Miller, Michael Bailey, Michael Kallitsis, and Manish Karir. Understanding IPv6 Internet Background Radiation. In *Proceedings of the 13th ACM SIGCOMM Conference on Internet Measurement (IMC '13)*, 2013.

[52] A. Dainotti, K. Benson, A. King, k. claffy, M. Kallitsis, E. Glatz, and X. Dimitropoulos. Estimating Internet Address Space Usage through Passive Measurements. *ACM SIGCOMM Computer Communication Review (CCR)*, 44(1), December 2013.

[53] A. Dainotti, K. Benson, A. King, B. Huffaker, E. Glatz, X. Dimitropoulos, P. Richter, A. Finamore, and A. Snoeren. Lost in Space: Improving Inference of IPv4 Address Space Utilization. *IEEE Journal on Selected Areas in Communications (JSAC)*, April 2016.

[54] A. Dainotti, A. King, K. Claffy, F. Papale, and A. Pescapè. Analysis of a /0 Stealth Scan from a Botnet. In *Proceedings of the 12th ACM SIGCOMM Conference on Internet Measurement (IMC '12)*, 2012.

[55] Alberto Dainotti, Roman Amman, Emile Aben, and Kimberly C. Claffy. Extracting Benefit from Harm: Using Malware Pollution to Analyze the Impact of Political and Geophysical Events on the Internet. *ACM SIGCOMM Computer Communication Review (CCR)*, 42(1):31–39, January 2012.

[56] Alberto Dainotti, Claudio Squarcella, Emile Aben, Kimberly C. Claffy, Marco Chiesa, Michele Russo, and Antonio Pescapé. Analysis of Country-wide Internet Outages Caused by Censorship. In *Proceedings of the 11th ACM SIGCOMM Conference on Internet Measurement (IMC '11)*, 2011.

[57] Sam Dean. It' s 2015 —You' d Think We' d Have Figured Out How To Measure Web Traffic By Now. July 2015. URL: http://fivethirtyeight.com/features/why-we-still-cant-agree-on-web-metrics/.

[58] Casey Deccio. Personal Correspondence. January 2016.

[59] A. Dhamdhere and C. Dovrolis. Twelve Years in the Evolution of the Internet Ecosystem. *IEEE/ACM Transactions on Networking*, 19(5):1420–1433, September 2011.

[60] R. Droms. Dynamic Host Configuration Protocol. RFC 1541 (Proposed Standard). Obsoleted by RFC 2131. Internet Engineering Task Force, October 1993. URL: http://www.ietf.org/rfc/rfc1541.txt.

[61] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. A Search Engine Backed by Internet-Wide Scanning. In *Proceedings of the 22nd ACM Conference on Computer and Communications Security (CCS '15)*, 2015.

[62] Zakir Durumeric, Michael Bailey, and J. Alex Halderman. An Internet-Wide View of Internet-Wide Scanning. In *Proccedings of the 23rd USENIX Security Symposium (USENIX Security '14)*, 2014.

[63] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. ZMap: Fast Internet-Wide Scanning and its Security Applications. In *Proceedings of the 22nd USENIX Security Symposium (USENIX Security '13)*, 2013.

[64] Roya Ensafi, David Fifield, Philipp Winter, Nick Feamster, Nicholas Weaver, and Vern Paxson. Examining How the Great Firewall Discovers Hidden Circumvention Servers. In *Proceedings of the 15th ACM SIGCOMM Conference on Internet Measurement (IMC '15)*, 2015.

[65] Nicolas Falliere. Sality: Story of a Peer-to-Peer Viral Network. 2011. URL: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/sality_peer_to_peer_viral_network.pdf.

[66] Xun Fan and John Heidemann. Selecting Representative IP Addresses for Internet Topology Studies. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC '10)*, 2010.

[67] FAQ: What is the coefficient of variation? UCLA Institute for Digital Research and Education. 2016. URL: http://www.ats.ucla.edu/stat/mult_pkg/faq/general/coefficient_of_variation.htm.

[68] P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827 (Best Current Practice). Updated by RFC 3704. Internet Engineering Task Force, May 2000. URL: http://www.ietf.org/rfc/rfc2827.txt.

[69] file.net. StudentMain.exe Windows process - What is it? 2015. URL: http://www.file.net/process/studentmain.exe.html.

[70] Alessandro Finamore, Marco Mellia, Michela Meo, Maurizio M. Munafø', and Dario Rossi. Experiences of Internet Traffic Monitoring with Tstat. *IEEE Network*, 25(3), 2011.

[71] N. Fonseca and M. Crovella. Bayesian Packet Loss Detection for TCP. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communication Societies (INFOCOM '05)*, 2005.

[72] V. Fuller and T. Li. Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan. RFC 4632 (Best Current Practice). Internet Engineering Task Force, August 2006. URL: http://www.ietf.org/rfc/rfc4632.txt.

[73] V. Fuller, T. Li, J. Yu, and K. Varadhan. Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy. RFC 1519 (Proposed Standard). Obsoleted by RFC 4632. Internet Engineering Task Force, September 1993. URL: http://www.ietf.org/rfc/rfc1519.txt.

[74] C Galamhos, Jose Matas, and Josef Kittler. Progressive Probabilistic Hough Transform for line detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.

[75] Carrie Gates. Coordinated Scan Detection. In *Proceedings of the Network and Distributed System Security Symposium (NDSS '09)*, 2009.

[76] Vinicius Gehlen, Alessandro Finamore, Marco Mellia, and Maurizio M. Munafò. Uncovering the Big Players of the Web. In *Proceedings of the International Workshop on Traffic Monitoring and Analysis (TMA '12)*, 2012.

[77] Geoff Huston. IPv4 Address Report. URL: http://www.potaroo.net/tools/ipv4/.

[78] Steve Gibson. NAT Router Security Solutions: Tips & Tricks You Haven't Seen Before. August 2006.

[79] Eduard Glatz and Xenofontas Dimitropoulos. Classifying Internet One-way Traffic. In *Proceedings of the 12th ACM Conference on Internet Measurement (PAM '12)*, 2012.

[80] Global RIPE Atlas Network Coverage. 2016. URL: https://atlas.ripe.net/results/maps/network-coverage/.

[81] F. Gont. Security Assessment of the Internet Protocol Version 4. RFC 6274 (Informational). Internet Engineering Task Force, July 2011. URL: http://www.ietf.org/rfc/rfc6274.txt.

[82] Jean Goubault-Larrecq and Julien Olivain. Detecting Subverted Cryptographic Protocols by Entropy Checking. Technical report (LSV-06-13). Laboratoire Spécification et Vérification, ENS Cachan.

[83] Robert David Graham. MASSCAN: Mass IP port scanner. 2014. URL: https://github.com/robertdavidgraham/masscan.

[84] T. Hain. Architectural Implications of NAT. RFC 2993 (Informational). Internet Engineering Task Force, November 2000. URL: http://www.ietf.org/rfc/rfc2993.txt.

[85] Warren Harrop and Grenville Armitage. Defining and Evaluating Greynets (Sparse Darknets). In *Proceedings of the the IEEE Conference on Local Computer Networks (LCN '05) 30th Anniversary*, 2005.

[86] John Heidemann, Yuri Pradkin, Ramesh Govindan, Christos Papadopoulos, Genevieve Bartlett, and Joseph Bannister. Census and Survey of the Visible Internet. In *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement (IMC '08)*, 2008.

[87] Thorsten Holz, Markus Engelberth, and Felix Freiling. *Learning more about the underground economy: a case-study of keyloggers and dropzones*. Springer, 2009.

[88] Jeffeny Hoogervorst. Is your home router being used for a DDoS attack? February 2014.

[89] How the Internet in Australia went down under. February 27, 2012. URL: http://bgpmon.net/blog/?p=554.

[90] Geoff Huston. IPv4: How long do we have? *The Internet Protocol Journal*, 6(4):2008–2010, 2003.

[91] Geoff Huston. Leaking Routes. March 2012. URL: http://www.potaroo.net/ispcol/2012-03/leaks.html.

[92] Geoff Huston. The Changing Foundation of the Internet: Confronting IPv4 Address Exhaustion. *The Internet Protocol Journal*, 11(3):19–36, 2008.

[93] Young Hyun, Bradley Huffaker, Dan Andersen, Matthew Luckie, and Kimberly C. Claffy. The IPv4 Routed /24 Topology Dataset. 2014. URL: http://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml.

[94] Important: kernel security and bug fix update. July 2012. URL: http://www.redhat.com/archives/rhsa-announce/2012-July/msg00030.html.

[95] Information of Sciences Institute, University of Southern California. ANT Censuses of the Internet Address Space. 2003–2016. URL: https://ant.isi.edu/address/.

[96] Information of Sciences Institute, USC. Internet Address Survey Binary Format. 2012. URL: http://www.isi.edu/ant/traces/topology/address_surveys/binformat_description.html.

[97] Insecure.Com LLC. Nmap Security Scanner. URL: http://nmap.org.

[98] Internet Census 2012: Port scanning /0 using insecure embedded devices. 2012. URL: http://internetcensus2012.bitbucket.org/paper.html.

[99] Internet World Stats. 2014. URL: http://www.internetworldstats.com.

[100] Tomas Isdal, Michael Piatek, Arvind Krishnamurthy, and Thomas Anderson. Leveraging bittorrent for end host measurements. In, *Passive and active network measurement*, pages 32–41. Springer, 2007.

[101] V. Jacobson, R. Braden, and D. Borman. TCP Extensions for High Performance. RFC 1323 (Proposed Standard). Obsoleted by RFC 7323. Internet Engineering Task Force, May 1992. URL: http://www.ietf.org/rfc/rfc1323.txt.

[102] Sharad Jaiswal, Gianluca Iannaccone, Christophe Diot, Jim Kurose, and Don Towsley. Inferring TCP Connection Characteristics Through Passive Measurements. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communication Societies (INFOCOM '04)*, 2004.

[103] Sharad Jaiswal, Gianluca Iannaccone, Christophe Diot, Jim Kurose, and Don Towsley. Measurement and Classification of Out-of-Sequence Packets in a Tier-1 IP Backbone. *IEEE/ACM Transactions on Networking (ToN)*, 15(1):54–66, 2007.

[104] Cheng Jin, Haining Wang, and Kang G Shin. Hop-Count Filtering: An Effective Defense Against Spoofed DDoS Traffic. In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, 2003.

[105] Chris Kanich, Kirill Levchenko, Brandon Enright, Geoffrey M Voelker, and Stefan Savage. The Heisenbot Uncertainty Problem: Challenges in Separating Bots from Chaff. In, 2008.

[106] Sachin Katti, Dina Katabi, Eddie Kohler, and Jacob Strauss. M&M: A Passive Toolkit for Measuring, Correlating, and Tracking Path Characteristics. Technical report. MIT CS and AI Lab, 2004.

[107] E. Katz-Bassett, H. V. Madhyastha, J. P. John, A. Krishnamurthy, D. Wetherall, and T. Anderson. Studying Black Holes in the Internet with Hubble. In *Proceedings of the 5th USENIX Conference on Networked Systems Design & Implementation (NSDI '08)*, 2008.

[108] Ethan Katz-Bassett, Harsha V Madhyastha, Vijay Kumar Adhikari, Colin Scott, Justine Sherry, Peter Van Wesep, Thomas E Anderson, and Arvind Krishnamurthy. Reverse Traceroute. In *Proceedings of the 7th USENIX Conference on Networked Systems Design & Implementation (NSDI '10)*, 2010.

[109] John Kristoff. Ephemeral Source Port Selection Strategies. Team Cymru. August 2015. URL: https://www.cymru.com/jtk/misc/ephemeralports.html.

[110] Marc Kührer, Thomas Hupperich, Christian Rossow, and Thorsten Holz. Exit from Hell? Reducing the Impact of Amplification DDoS Attacks. In *Proceedings of the 23rd USENIX Security Symposium (USENIX Security '14)*, 2014.

[111] Yoram Kulbak and Danny Bickson. The eMule Protocol Specification. 2005. URL: http://note.tc.edu.tw/upload/2009Oct/20091027143657.pdf.

[112] Abhishek Kumar, Vern Paxson, and Nicholas Weaver. Exploiting Underlying Structure for Detailed Reconstruction of an Internet-scale Event. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement (IMC '05)*, 2005.

[113] Craig Labovitz, Abha Ahuja, and Farnam Jahanian. Experimental Study of Internet Stability and Backbone Failures. In *Proceedings of the 29th Annual International Symposium on Fault-Tolerant Computing*, 1999.

[114] Craig Labovitz, Scott Iekel-Johnson, Danny McPherson, Jon Oberheide, and Farnam Jahanian. Internet Inter-Domain Traffic. In *Proceedings of the ACM SIGCOMM 2010 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '10)*, 2010.

[115] Ulf Lamping, Richard Sharpe, and Ed Warnicke. Customizing Wireshark. In, *Wireshark User Guide*, part 10, 2014. URL: https://www.wireshark.org/docs/wsug_html_chunked/ChCustProtocolDissectionSection.html.

[116] M. Larsen and F. Gont. Recommendations for Transport-Protocol Port Randomization. RFC 6056 (Best Current Practice). Internet Engineering Task Force, January 2011. URL: http://www.ietf.org/rfc/rfc6056.txt.

[117] Stevens Le Blond, Chao Zhang, Arnaud Legout, Keith Ross, and Walid Dabbous. I Know Where You Are and What You Are Sharing: Exploiting P2P Communications to Invade Users' Privacy. In *Proceedings of the 11th ACM SIGCOMM Conference on Internet Measurement (IMC '11)*, 2011.

[118] Robert Lemos. Heartbleed is the gift that keeps on giving as servers remain unpatched. August 2014. URL: http://arstechnica.com/security/2014/08/heartbleed-is-the-gift-that-keeps-on-giving-as-servers-remain-unpatched/.

[119] Kirill Levchenko, Andreas Pitsillidis, Neha Chachra, Brandon Enright, Márk Félegyházi, Chris Grier, Tristan Halvorson, Chris Kanich, Christian Kreibich, He Liu, D. McCoy, N. Weaver, V. Paxson, G.M. Voelker, and S. Savage. Click Trajectories: End-to-End Analysis of the Spam Value Chain. In *2011 IEEE Symposium on Security and Privacy (S&P)*, 2011.

[120] Zhichun Li, Anup Goyal, Yan Chen, and Vern Paxson. Automating Analysis of Large-scale Botnet Probing Events. In *Acm symposium on information, computer, and communications security (asiaccs)*, 2009.

[121] Jian Liang, Naoum Naoumov, and Keith W. Ross. The Index Poisoning Attack in P2P File Sharing Systems. In *Proceedings of the 25th Annual Joint Conference of the IEEE Computer and Communication Societies (INFOCOM '06)*, 2006.

[122] Yan Liu and Yulong Yang. Analysis of P2P Traffic Identification Methods. *Journal of Emerging Trends in Computing and Information Sciences*, 4(5), 2013.

[123] Ioana Livadariu, Ahmed Elmokashfi, Amogh Dhamdhere, et al. A First Look at IPv4 Transfer Markets. In *Proceedings of the 9th ACM Conference on Emerging Networking Experiments and Technologies*, 2013.

[124] Andrew Loewenstern and Arvid Norberg. DHT Protocol. January 31, 2008. URL: http://www.bittorrent.org/beps/bep_0005.html.

[125] Matthew Luckie, Young Hyun, and Bradley Huffaker. Traceroute Probe Method and Forward IP Path Inference. In *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement (IMC '08)*, 2008.

[126] A. Lutu, M. Bagnulo, A. Dhamdhere, and k. claffy. NAT Revelio: Detecting NAT444 in the ISP. In *Proceedings of the International Conference on Passive and Active Network Measurement (PAM '16)*, 2016.

[127] Gordon "Fyodor" Lyon. Firewall/IDS Evasion and Spoofing. In, *Nmap Network Scanning*, part 15. Nmap Project, 2009. URL: https://nmap.org/book/man-bypass-firewalls-ids.html.

[128] Gordon "Fyodor" Lyon. Port Scanning Techniques. In, *Nmap Network Scanning*, part 15. Nmap Project, 2009. URL: https://nmap.org/book/man-port-scanning-techniques.html.

[129] Gordon "Fyodor" Lyon. Port Scanning Techniques and Algorithms: TCP Idle Scan. In, *Nmap Network Scanning*, part 5. Nmap Project, 2009. URL: https://nmap.org/book/idlescan.html.

[130] Gordon "Fyodor" Lyon. Remote OS Detection: Usage and Examples. In, *Nmap Network Scanning*, part 8. Nmap Project, 2009. URL: https://nmap.org/book/osdetect-methods.html%5C#osdetect-ts.

[131] Harsha V Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iPlane: An Information Plane for Distributed Services. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI '06)*, 2006.

[132] Gregor Maier, Fabian Schneider, and Anja Feldmann. NAT Usage in Residential Broadband Networks. In *Proceedings of the International Conference on Passive and Active Network Measurement (PAM '11)*, 2011.

[133] Damon McCoy, Andreas Pitsillidis, Grant Jordan, Nicholas Weaver, Christian Kreibich, Brian Krebs, Geoffrey M Voelker, Stefan Savage, and Kirill Levchenko. PharmaLeaks: Understanding the Business of Online Pharmaceutical Affiliate Programs. In *Proceedings of the 21st USENIX Security Symposium (USENIX Security '12)*, 2012.

[134] Bret McDanel. TCP Timestamping - Obtaining System Uptime Remotely. 2001. URL: http://seclists.org/bugtraq/2001/Mar/182.

[135] Robert McMillan and Cade Metz. The Inside Story of the Extra Second That Crashed the Web. July 2012. URL: http://www.wired.com/2012/07/leap-second-glitch-explained/.

[136] Kevin McNamee. Malware Analysis Report: New C&C Protocol for ZeroAccess/Sirefef. 2012. URL: http://botnetlegalnotice.com/zeroaccess/files/Ex_14_Decl_Anselmi.pdf.

[137] Marco Mellia, Michela Meo, Luca Muscariello, and Dario Rossi. Passive analysis of TCP anomalies. *Computer Networks*, 52(14):2663–2676, 2008.

[138] Merit Network, Inc. Merit Darknet IPv4. URL: http://software.merit.edu/darknet/.

[139] Bradley Mitchell. TCP and UDP Port Numbers for Xbox Live. 2008. URL: http://compnetworking.about.com/b/2008/11/15/tcp-and-udp-port-numbers-for-xbox-live.htm.

[140] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the Slammer Worm. *IEEE Security & Privacy*, 1(4):33–39, August 2003.

[141] D. Moore, C. Shannon, D. Brown, G. Voelker, and S. Savage. Inferring Internet Denial-of-Service Activity. *ACM Transactions on Computer Systems*, 24(2):115–139, May 2006.

[142] D. Moore, C. Shannon, and J. Brown. Code-Red: a case study on the spread and victims of an Internet worm. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement (IMW '02)*, 2002.

[143] D. Moore, C. Shannon, G. Voelker, and S. Savage. Network Telescopes: Technical Report. Technical report. Cooperative Association for Internet Data Analysis (CAIDA), July 2004.

[144] David Moore. Network Telescopes: Observing Small or Distant Security Events. Presented at USENIX. August 2002. URL: http://www.caida.org/publications/presentations/2002/usenix_sec/.

[145] H.D. Moore. Project Sonar. 2008. URL: https://community.rapid7.com/community/infosec/sonar/blog.

[146] Nemo. Can BitTorrent clients be fingerprinted? Message posted to http://security.stackexchange.com/questions/37167/can-bittorrent-clients-be-fingerprinted. December 2014.

[147] Alan Neville and Ross Gibb. ZeroAccess Indepth. October 2013. URL: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/zeroaccess_indepth.pdf.

[148] Arvid Norberg. BitTorrent DHT security extension. URL: http://www.libtorrent.org/dht_sec.html.

[149] Arvid Norberg. Mainline DHT extensions. URL: http://www.libtorrent.org/dht_extensions.html.

[150] Arvid Norberg. uTorrent transport protocol. June 22, 2009. URL: http://www.bittorrent.org/beps/bep_0029.html.

[151] Masayuki Ohta, Yoshiki Kanda, Kensuke Fukuda, and Toshiharu Sugawara. Analysis of Spoofed IP Traffic Using Time-to-Live and Identification Fields in IP Headers. In *Proceedings of the 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications (WAINA)*, 2011.

[152] Open Resolver Project. 2014. URL: http://openresolverproject.org.

[153] OpenResolverProject trends. URL: http://openresolverproject.org/breakdown-graph1.cgi.

[154] John S. Otto, Mario A. Sánchez, David R. Choffnes, Fabián E. Bustamante, and Georgos Siganos. On Blind Mice and the Elephant: Understanding the Network Impact of a Large Distributed System. In *Proceedings of the ACM SIGCOMM 2011 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '11)*, 2011.

[155] Ramakrishna Padmanabhan, Emile Aben, Amogh Dhamdhere, kc claffy, and Neil Spring. Dynamic address durations in RIPE Atlas probes. Presentation at CAIDA AIMS. 2016. URL: http://www.cs.umd.edu/~ramapad/docs/aims16_dhcp_full.pdf.

[156] Ruoming Pang, Vinod Yegneswaran, Paul Barford, Vern Paxson, and Larry Peterson. Characteristics of Internet Background Radiation. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC '04)*, 2004.

[157] V. Paxson, M. Allman, J. Chu, and M. Sargent. Computing TCP's Retransmission Timer. RFC 6298 (Proposed Standard). Internet Engineering Task Force, June 2011. URL: http://www.ietf.org/rfc/rfc6298.txt.

[158] Vern Paxson. End-to-end Routing Behavior in the Internet. In *Proceedings of the ACM SIGCOMM 1996 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '96)*, 1996.

[159] S. Perreault, I. Yamagata, S. Miyakawa, A. Nakagawa, and H. Ashida. Common Requirements for Carrier-Grade NATs (CGNs). RFC 6888 (Best Current Practice). Internet Engineering Task Force, April 2013. URL: http://www.ietf.org/rfc/rfc6888.txt.

[160] David Plonka and Arthur Berger. Temporal and Spatial Classification of Active IPv6 Addresses. In *Proceedings of the 15th ACM SIGCOMM Conference on Internet Measurement (IMC '15)*, 2015.

[161] J. Postel. Internet Control Message Protocol. RFC 792 (INTERNET STANDARD). Updated by RFCs 950, 4884, 6633, 6918. Internet Engineering Task Force, September 1981. URL: http://www.ietf.org/rfc/rfc792.txt.

[162] Matthew Prince. The DDoS That Almost Broke the Internet. March 2013. URL: https://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet.

[163] Niels Provos. A Virtual Honeypot Framework. In *Proceedings of the 13th USENIX Security Symposium (USENIX Security '04)*, 2004.

[164] Qihoo 360 Technology Company Limited Investor Relations Investor FAQs. January 2016. URL: http://ir.360.cn/phoenix.zhtml?c=243376&p=irol-faq.

[165] Lin Quan, John Heidemann, and Yuri Pradkin. Detecting Internet Outages with Precise Active Probing (extended). Technical report (ISI-TR-2012-678b). USC/Information Sciences Institute, 2012. URL: http://www.isi.edu/~johnh/PAPERS/Quan12a.html.

[166] Lin Quan, John Heidemann, and Yuri Pradkin. When the Internet Sleeps: Correlating Diurnal Networks With External Factors. In *Proceedings of the 14th ACM SIGCOMM Conference on Internet Measurement (PAM '14)*, 2014.

[167] Moheeb Abu Rajab, Jay Zarfoss, Fabian Monrose, and Andreas Terzis. My Botnet is Bigger than Yours (Maybe, Better than Yours): why size estimates remain challenging. In *Proceedings of the First Workshop on Hot Topics in Understanding Botnets (HotBots '07)*, 2007.

[168] Anirudh Ramachandran and Nick Feamster. Understanding the network-level behavior of spammers. In *Proceedings of the ACM SIGCOMM 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '06)*, 2006.

[169] Charles Reis, Steven D Gribble, Tadayoshi Kohno, and Nicholas C Weaver. Detecting In-Flight Page Changes with Web Tripwires. In *Proceedings of the 5th USENIX Conference on Networked Systems Design & Implementation (NSDI '08)*, 2008.

[170] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address Allocation for Private Internets. RFC 1918 (Best Current Practice). Updated by RFC 6761. Internet Engineering Task Force, February 1996. URL: http://www.ietf.org/rfc/rfc1918.txt.

[171] P. Richter, M. Allman, R. Bush, and V. Paxson. A Primer on IPv4 Scarcity. *ACM SIGCOMM Computer Communication Review (CCR)*, 45(2):21–31, April 2015.

[172] Philipp Richter, Florian Wohlfart, Narseo Vallina-Rodriguez, Mark Allman, Randy Bush, Anja Feldmann, Christian Kreibich, Nicholas Weaver, and Vern Paxson. A Multi-perspective Analysis of Carrier-Grade NAT Deployment. *Arxiv preprint arxiv:1605.05606*, 2016.

[173] rico2922. Problème de postes isolés derrière un switch. Message posted to http://www.developpez.net/forums/d1448594/systemes/reseaux/architecture/probleme-postes-isoles-derriere-switch/. December 2006.

[174] RIPE Atlas. URL: https://atlas.ripe.net/.

[175] RIS Raw Data. URL: http://www.ripe.net/data-tools/stats/ris/ris-raw-data.

[176] Martin Roesch. Snort: lightweight intrusion detection for networks. In *Lisa*. Volume 99. (1), 1999, pages 229–238.

[177] Christian Rossow. Amplification Hell: Revisiting Network Protocols for DDoS Abuse. In *Proceedings of the 21st Annual Network and Distributed System Security Symposium (NDSS '14)*, 2014.

[178] Christian Rossow and Christian J Dietrich. Provex: detecting botnets with encrypted command and control channels. In, *Detection of intrusions and malware, and vulnerability assessment*, pages 21–40. Springer, 2013.

[179] Routeviews Prefix to AS mappings Dataset for IPv4 and IPv6. 2015. URL: http://www.caida.org/data/routing/routeviews-prefix2as.xml.

[180] Matthew Sargent, Jakub Czyz, Mark Allman, and Michael Bailey. On The Power and Limitations of Detecting Network Filtering via Passive Observation. In *Proceedings of the International Conference on Passive and Active Network Measurement (PAM '15)*, 2015.

[181] Kyle Schomp, Tom Callahan, Michael Rabinovich, and Mark Allman. On Measuring the Client-side DNS Infrastructure. In *Proceedings of the 13th ACM SIGCOMM Conference on Internet Measurement (IMC '13)*, 2013.

[182] Albin Sebastian. Default Time To Live (TTL) values. 2009. URL: http://www.binbert.com/blog/2009/12/default-time-to-live-ttl-values/.

[183] Server queries. URL: https://developer.valvesoftware.com/wiki/Server_queries.

[184] Yoichi Shinoda, Ko Ikai, and Motomu Itoh. Vulnerabilities of Passive Internet Threat Monitors. In *Proceedings of the 14th USENIX Security Symposium (USENIX Security '05)*, 2005.

[185] Smith, Roy. Running out of Internet addresses? Archive/Hypermail of Early TCp-IP Mail List. 1988. URL: http://www-mice.cs.ucl.ac.uk/multimedia/misc/tcp_ip/8813.mm.www/0121.html.

[186] Richard L. Smith. Determining the sample size. Powerpoint Slides. 2010. URL: https://www.unc.edu/~rls/s151-2010/class23.pdf.

[187] Ganesh Srinivasan. Microsoft Azure' s use of non-US IPv4 address space in US regions. Microsoft Azure. 2014. URL: https://azure.microsoft.com/en-us/blog/windows-azures-use-of-non-us-ipv4-address-space-in-us-regions/.

[188] P. Srisuresh and M. Holdrege. IP Network Address Translator (NAT) Terminology and Considerations. RFC 2663 (Informational). Internet Engineering Task Force, August 1999. URL: http://www.ietf.org/rfc/rfc2663.txt.

[189] Moritz Steiner, Taoufik En-Najjary, and Ernst W Biersack. Exploiting KAD: Possible Uses and Misuses. *ACM SIGCOMM Computer Communication Review (CCR)*, 37(5):65–70, 2007.

[190] Moritz Steiner, Taoufik En-Najjary, and Ernst W Biersack. Long term study of peer behavior in the kad dht. *IEEE/ACM Transactions on Networking (ToN)*, 17(5):1371–1384, 2009.

[191] Brett Stone-Gross, Marco Cova, Lorenzo Cavallaro, Bob Gilbert, Martin Szydlowski, Richard Kemmerer, Christopher Kruegel, and Giovanni Vigna. Your Botnet is My Botnet: Analysis of a Botnet Takeover. In *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*, 2009.

[192] Linda Summers. Survey Finds Nearly Half of Consumers Fail To Upgrade Software Regularly And One Quarter of Consumers Don't Know Why To Update Software. July 2012. URL: http://blogs.skype.com/2012/07/23/intl-tech-upgrade-week/.

[193] SWITCH. Swiss Tele Communication System for Higher Education. URL: http://www.switch.ch/.

[194] TcpMaxConnectRetransmissions. URL: https://technet.microsoft.com/en-us/library/cc938209.aspx.

[195] Renata Teixeira, Aman Shaikh, Tim Griffin, and Jennifer Rexford. Dynamics of Hot-Potato Routing in IP Networks. In *Proceedings of the International Conference on Measurements and Modeling of Computer Systems (SIGMETRICS '04/Performance '04)*, 2004.

[196] S.J. Templeton and K.E. Levitt. Detecting Spoofed Packets. In *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX '03)*, 2003.

[197] Teredo Overview. 2003. URL: https://technet.microsoft.com/en-us/library/bb457011.aspx.

[198] The Bro Project. TCP Scan detection. 2014. URL: https://www.bro.org/sphinx/scripts/policy/misc/scan.bro.html.

[199] The Heartbleed Bug. http://heartbleed.com/. April 2014.

[200] Andree Toonk. A BGP Leak Made in Canada. August 8, 2012. URL: http://www.bgpmon.net/a-bgp-leak-made-in-canada/.

[201] J. Touch. Updated Specification of the IPv4 ID Field. RFC 6864 (Proposed Standard). Internet Engineering Task Force, February 2013. URL: http://www.ietf.org/rfc/rfc6864.txt.

[202] Daniel Turner, Kirill Levchenko, Alex C Snoeren, and Stefan Savage. California Fault Lines: Understanding the Causes and Impact of Network Failures. In *Proceedings of the ACM SIGCOMM 2010 Conference on Applications, Technolo-*

gies, Architectures, and Protocols for Computer Communication (SIGCOMM '10), 2010.

[203] UCSD Network Telescope Global Attack Traffic (current). URL: http://www.caida.org/data/realtime/telescope/.

[204] Johanna Ullrich, Peter Kieseberg, Katharina Krombholz, and Edgar Weippl. On Reconnaissance with IPv6: A Pattern-Based Scanning Approach. In *Proceedings of the 2015 10th International Conference on Availability, Reliability and Security (ARES)*, 2015.

[205] University of California, San Diego. The UCSD Network Telescope. URL: http://www.caida.org/projects/network_telescope/.

[206] University of Oregon Route Views Project. URL: http://www.routeviews.org.

[207] Ernesto Van der Sar. uTorrent Keeps BitTorrent Lead, BitComent Fades Away. September 16, 2011. URL: https://torrentfreak.com/utorrent-keeps-bittorrent-lead-bitcomet-fades-away-110916/.

[208] Bruce Van Nice. Drilling Down into DNS DDoS. NANOG 63. February 2015. URL: http://www.nanog.org/sites/default/files/nanog63-dnstrack-vannice-ddos.pdf.

[209] Version History for 360 Total Security. URL: https://www.360totalsecurity.com/en/version/360-total-security/.

[210] M. Vrable, J. Ma, J. Chen, D. Moore, E. Vandekieft, A. Snoeren, G. Voelker, and S. Savage. Scalability, Fidelity and Containment in the Potemkin Virtual Honeyfarm. *SIGOPS Operating Systems Review*, 39(5):148–162, October 2005.

[211] Simon Waite. Interesting clients found in the BitTorrent DHT. URL: http://www.simonwaite.com/projects/bittorrentdht.

[212] Rhiannon Weaver. A Probabilistic Population Study of the Conficker-C Botnet. In *Proceedings of the International Conference on Passive and Active Network Measurement (PAM '10)*, 2010.

[213] Songjie Wei and Jelena Mirkovic. Correcting Congestion-based Error in Network Telescope's Observations of Worm Dynamics. In *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement (IMC '08)*, 2008.

[214] M. West and S. McCann. TCP/IP Field Behavior. RFC 4413 (Informational). Internet Engineering Task Force, March 2006. URL: http://www.ietf.org/rfc/rfc4413.txt.

[215] Rene Wilhelm. RIPE Labs Blog: How to Define Address Space as 'routed'? https://labs.ripe.net/Members/wilhelm/content-how-define-address-space-routed. October 2009.

[216] World IPv6 Launch. URL: http://www.worldipv6launch.org/.

[217] Eric Wustrow, Manish Karir, Michael Bailey, Farnam Jahanian, and Geoff Huston. Internet Background Radiation Revisited. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC '10)*, 2010.

[218] Vinod Yegneswaran, Paul Barford, and Dave Plonka. On the Design and Use of Internet Sinks for Network Abuse Monitoring. In *Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID '04)*, 2004.

[219] Vinod Yegneswaran, Paul Barford, and Johannes Ullrich. Internet Intrusions: Global Characteristics and Prevalence. In *Proceedings of the International Conference on Measurements and Modeling of Computer Systems (SIGMETRICS '03)*, 2003.

[220] Tim Yeh. Netis Routers Leave Wide Open Backdoor. Trend Micro. August 2014. URL: http://blog.trendmicro.com/trendlabs-security-intelligence/netis-routers-leave-wide-open-backdoor/.

[221] Michal Zalewski. p0f v3: passive fingerprinter. 2012. URL: http://lcamtuf.coredump.cx/p0f3/README.

[222] Sebastian Zander, Lachlan L. H. Andrew, and Grenville Armitage. Capturing Ghosts: Predicting the Used IPv4 Space by Inferring Unobserved Addresses. In *Proceedings of the 14th ACM SIGCOMM Conference on Internet Measurement (IMC '14)*, 2014.

[223] Ming Zhang, Chi Zhang, Vivek Pai, Larry Peterson, and Randy Wang. PlanetSeer: Internet Path Failure Monitoring and Characterization in Wide-Area Services. In *Proceedings of the 6th USENIX Symposium on Operating Systems Design and Implementation (OSDI '04)*, 2004.

[224] John Zorabedian. SophosLabs at VB2014: How cunning malware fights analysis by security researchers. August 29, 2014. URL: https://blogs.sophos.com/2014/08/29/sophoslabs-at-vb2014-how-cunning-malware-fights-analysis-by-security-researchers/.

[225] 一点即过不留影. 关于学校信息教室的多媒体教学软件. Transanslated Danny Huang. Message posted to http://tieba.baidu.com/p/2952998221. April 2014.