On BGP Inter-domain Routing: an Investigation of Scalability with Respect to Churn

Ahmed Elmokashfi

Doctoral Dissertation

Submitted to the
Faculty of Mathematics and Natural Sciences
University of Oslo
in partial fulfillment of the requirements for the degree
Philosophiae Doctor

March 2011

Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo
No. 1074

ISSN 1501-7710

Cover: Inger Sandved Anfinsen.
Printed in Norway: AIT Oslo AS.

Produced in co-operation with Unipub.
The thesis is produced by Unipub merely in connection with the
thesis defence. Kindly direct all inquiries regarding the thesis to the copyright
holder or the unit which grants the doctorate.

To my family

Acknowledgements

When I joined Simula as a PhD student I had a very vague idea about doing research, albeit I was sure that I am thirsty for knowledge. Now that I am close to the end, I find myself quite grateful and amazed when reflecting on the past four years. The journey has been extremely rewarding with many memorable moments. Several people around me have been very supportive and helpful, without them this work would have not been accomplished.

I am greatly indebted to my supervisors, Dr. Amund Kvalbein, Dr. Tarik Cicic, and Professor Olav Lysne for their great support and outstanding mentoring. I owe a great deal of my gratitude to Amund, he has been guiding me through this journey and teaching me how to do research, well beyond the call of duty. I am grateful for his high standards for work, kindness, friendliness, and sincerity. Tarik has been extremely helpful and I really appreciate his care and thoughtful guidance. The reassurance and valuable comments from Olav have been very inspiring.

During the past four years I have had a great mentor, Professor. Constantine Dovrolis. He has taken a key role in this research since its early days. Working and collaborating with Constantine have been extremely insightful, inspiring, and fun. I would also like to thank him for inviting me to spend a few months at Georgia Tech, in late 2007.

The Rate-Limiting measurement work would not have been possible without the support of Tarik Cicic and his staff at MNS and especially Jon Marius Evang. I would like also to thank Samantha (Sau Man) Lo for her inputs and help during the early stages of the churn evolution work.

I am also grateful to my friend and former office mate Ole Kristoffer Apeland. The time we spent sharing the office and doing courses at the university was a lot of fun. We had great discussions and a great deal of cooperation.

The greatest thanks go to my colleagues for making Simula such an attractive and dynamic work place. I am grateful to the IT support department at Simula and in particular to Bjørn Dyresen and Margareth Adaa for their kindness and help. During my PhD period I had pleasure of the company of many people at Simula. Many thanks to Audun Hansen, Carsten Griwodz, Ernst Gunnar, Jie Xiang, Knut-Helge Vik, Omar Al-khayat, Stuart Clark, Tor Skeie, Wenche Angel, and Åshild Solheim.

My stay at Georgia Tech could have been very lonely without the many lab-mates and friends I had there. Special thanks to Amund and Siv, Constantine and Marie, Ahmed Mansy, Amogh Dhamdhere, Partha Kanuparthy, and Mazin Makki.

And last but not least, I want to thank my friends and family for their unconditional love and support throughout these years. Thank you my parents for always believing in me, and thank you my brother and friend Elsiddig for your unlimited support.

Preface

This is a doctoral thesis submitted to the Department of Informatics, Faculty of Mathematics and Natural Sciences, University of Oslo in partial fulfillment for the degree Philosophiae Doctor in Computer Science for the year 2011.

The work in this thesis has been carried out at Simula Research Laboratory from January 2007 to March 2011, under the supervision of Dr. Amund Kvalbein, Dr. Tarik Cicic, and Professor Olav Lysne. Financial support for my thesis has been provided by the Resilient Networks project at Simula Research Laboratory (Project Number 175629).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Internet started as an experimental academic network interconnecting universities and research institutes. It was later considered, after its commercialization in mid-1990s, as an alternative infrastructure for migrating and re-inventing traditional services such as: newspapers, telephony, and documents exchange. Nevertheless, the past decade witnessed a paradigm shift in terms of recognizing the Internet as an independent platform for service creation. The introduction of many innovative applications and services (e.g. Blogging, Skype, Facebook, Youtube) has contributed to a dramatic increase in our reliance on the Internet as a medium for socializing, communication, and entertainment. About one third of today's world population has access to the Internet, this number is likely to rise in a fast pace given the high penetration of networked devices and services. This tremendous growth was not expected at the early days of the Internet. Hence, today's Internet is facing an array of challenges and limitations that includes scalability, security, and robustness. Addressing these limitations and designing tomorrow's Internet are high on the agendas of several public and private agencies.

1.1 Problem statement

The Internet can be described as a large collection of interconnected networks. Each such network is administrated by a single entity and called an Autonomous System (AS). The Border Gateway Protocol (BGP), the only deployed inter-domain routing protocol, is used for exchanging reachability information between ASes.

The current version of inter-domain routing protocol, BGPv4, was introduced in 1994. Since then, both the size and importance of the Internet have increased significantly. Today's Internet consists of about 36000 ASes in comparison to less than 3000 ASes in 1997. BGP has handled this in an impressive way; the flexibility of BGP is arguably one of the main factors behind its success. But recently, there have been concerns about BGP's ability to continue to cope with the growing network size in an efficient, reliable, and a secure way. This thesis focuses on issues related to inter-domain routing scalability.

Inter-domain routing scalability is a major concern for the Internet community. Scalability is an issue in two different aspects: increasing routing table size, and increasing rate of BGP updates (churn). The growing size of the routing table requires increasingly larger fast memory, but it does not necessarily slow down packet forwarding as long as address lookups are performed using Ternary Content Addressable Memory (TCAM) or constant-time longest-prefix matching algorithms [Var04]. Churn, however, is a more serious concern because processing BGP updates can be computationally intensive (updating routing state, generating more updates, checking import/export filters), and it can trigger a wide-scale instability. If the current best route to a destination is modified, the global Routing Information Base (RIB) and the Forwarding Information Bases (FIBs) on the line cards need to be updated. To make things worse, routing updates are known to be very bursty, with peak rates several orders of magnitude higher than daily averages. When the rate of updates becomes too high, the fear is that there will be (or there are already) periods when routers will be unable to maintain a consistent routing table. *There is a need to improve our understanding about the impact of different factors (e.g. topology, routing events) on churn; and to characterize the severity of the problem. This understanding is vital for improving the current architecture and an important input to designers of future architectures. This thesis focuses on addressing some of these questions.*

1.2 Approach

Several different factors can influence BGP churn. First, it is plausible to assume that inter-domain routing activity is dependent on the network size, i.e., number of ASes and prefixes. It is expected that the rate of BGP updates a router receives will increase with the number of routable destination

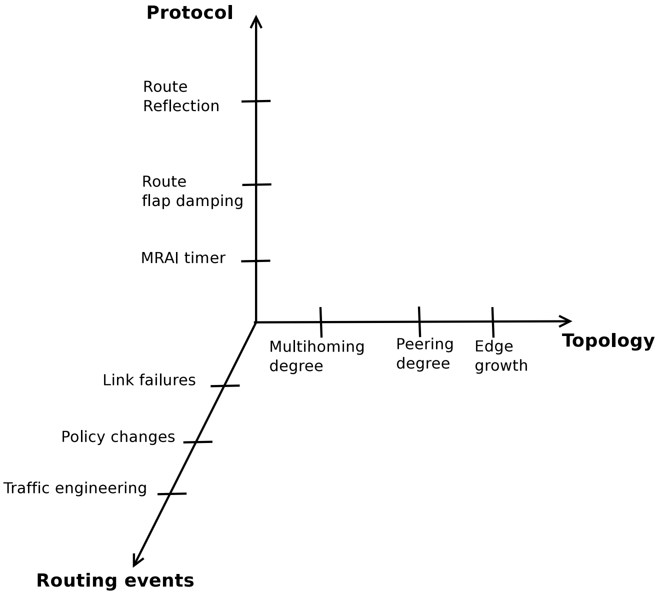


Figure 1.1: Churn factors

prefixes. Other prominent factors, beside the network size, can be grouped along three different axis as illustrated in Fig. 1.1 [1], more specifically:

1. The characteristics of the Internet *topology*

2. The *routing protocol*, including policy annotations and various BGP mechanisms and implementations choices

3. *Routing events* like prefix announcements, link failures, session resets, traffic engineering

There is a need to understand the impact of each contributor independently and then in association with other related factors. In this thesis, we

[1] Note that, the factors mentioned on the axis in Fig. 1.1 are examples, and not a complete list.

look at routing dynamics and its evolution from several angles and that necessitates employing different approaches. Hence, we adopt two approaches, simulations and measurements. First, we use simulations to understand the roles of topology, protocol configurations, and routing events; but with less focus on the later. We start by isolating a certain factor (e.g. multihoming) and constructing several what-if scenarios (e.g. what will be the impact of stub failures if multihoming is growing faster at the core of the Internet?). By answering such questions, we will be able to characterize the impact of different factors. Then, we can relate our findings to what we know so far about the evolution of the global routing system. Second, we measure different properties of the routing system and their evolution. More specifically, we measure the evolution of BGP dynamics at the core of the Internet over a long time period, as well as the impact of specific protocol configurations.

1.3 Contributions

In this work, we shed light on the impact of topology growth, BGP configurations, and routing events on churn evolution. We also present the most comprehensive study of churn evolution at the core of the Internet. The rest of this section elaborates on our contributions.

A framework for simulating BGP. We propose a flexible model for simulating BGP. It consists of a topology generator that produces AS-level graphs which are annotated with business relationships; and a light-weight BGP simulator that is capable of capturing routing dynamics and scaling to network sizes of thousands of nodes.

The role of topology growth. Using our flexible topology model, we explore scalability in several plausible or educational "what-if" scenarios for the growth of the AS-level topology. We show that the most important topological factor in deciding the number of generated updates is the connectivity in the core of the network. We also demonstrate that peering links play a very different role than transit links with respect to scalability. Furthermore, we also show that the depth of the hierarchical structure in the Internet plays a significant role; flatter topologies are more scalable. Finally, we demonstrate how densification through increased multihoming degree can impact churn.

The role of update rate-limiting. We also explore how different BGP rate-limiting implementations and configurations affect the level of churn. In addition, by looking at update traces from a large number of route monitors in the RouteViews project [rou], we discover that the arrival pattern of BGP updates for single prefixes is remarkably stable across BGP sessions from a diverse set of ASes across the Internet. This allows us to derive a formulation that quantifies the expected reduction in churn for different rate-limiting implementations and configured timer values.

Churn growth at the core of the Internet This thesis investigates the evolution of churn at four monitors located in the core of the Internet over a period up to seven years and eight months. The corresponding time series are very bursty, with large churn spikes and level-shifts. We perform an in-depth analysis of the time series in order to identify and explain the main sources of churn. After filtering all pathological routing updates and those related to the monitored networks, we reach at a version of churn time series, the baseline churn, that represents the background churn in the Internet. Analyzing the baseline churn surprisingly shows that the churn rate increases more slowly than the number of prefixes in the routing table.

1.4 Thesis organization

This thesis is organized into four parts.

Part I. This part consists of four chapters. In Chapter 2, we present a general overview of the Internet AS topology and BGP basic operation. Chapter 3 discusses open issues and limitations in the current inter-domain routing architecture, i.e. scalability, reliability, correctness. We highlight different factors that constitute BGP churn and we explore possible methods for investigating them in Chapter 4. This part concludes with Chapter 5, where we propose SIMROT as a comprehensive toolbox for simulating BGP.

Part II. There are two chapters in this part. Chapter 6 examines the role of topology growth on the scalability of BGP. Besides, Chapter 7 that investigates how different BGP rate-limiting implementations and configurations affect the level of churn.

Part III. This part consists of Chapter 8 where we investigate the evolution of churn at four monitors located in the core of the Internet over a period up to seven years and eight months.

Part IV. This part concludes the thesis. We draw our conclusions in Chapter 9 and sketch possible future work directions in Chapter 10.

1.5 Publications

Most of the work in this thesis has been published in conference proceedings and journals. In the following we list all related publications:

1. Ahmed Elmokashfi, Amund Kvalbein, and Constantine Dovrolis. *On the Scalability of BGP: the Roles of Topology Growth and Update Rate-Limiting. In ACM CoNEXT'08, Madrid, Spain, December 2008*

2. Ahmed Elmokashfi, Amund Kvalbein, and Constantine Dovrolis. *BGP Churn Evolution: a Perspective from the Core. In IEEE INFOCOM'10, San Diego, USA, March 2010*

3. Ahmed Elmokashfi, Amund Kvalbein, and Constantine Dovrolis. *On the Scalability of BGP: The Role of Topology Growth. IIn EEE Journal on Selected Areas in Communications, 2010*

4. Ahmed Elmokashfi, Amund Kvalbein, and Tarik Cicic. *On Update Rate-Limiting in BGP. To appear In IEEE ICC'11*

5. Ahmed Elmokashfi, Amund Kvalbein, and Constantine Dovrolis. *BGP Churn Evolution: a Perspective from the Core. Submitted to a journal*

6. Ahmed Elmokashfi, Amund Kvalbein, and Constantine Dovrolis. *SIMROT: A Scalable Inter-domain Routing Toolbox. In submission.*

Part I

Background and Approach

Chapter 2

Internet Architecture and Routing

The Internet is to an increasing degree taking a central role in our daily life. Almost 30% [int10] of the world population today has access to the Internet. This high penetration rate is consequential to the large pool of offered services. Usage patterns vary across societies and countries. At one end of the spectrum lie developed societies that employ the Internet for a wide range of purposes that virtually touch upon all aspects of life. Users in less privileged areas who are situated at the other end of the spectrum mainly use the Internet for browsing the World Wide Web.

For most of the users the Internet is a black box which they interact with through web browsers and specialized client side applications. But, in reality the Internet is a large collection of networks, that use a wide variety of protocols and applications to communicate. In the rest of this chapter we briefly explain the structure of the Internet topology and operation.

2.1 The Internet Architecture

The Internet can be described as a large collection of networks that are interconnected together to facilitate content reachability, creation, and consumption. These networks are administered independently and called Autonomous Systems (ASes) [1]. An AS is identified by a unique identifier (AS-number) which comes in two formats either 16-bit or 32-bit integers. The later was

[1] A single administrative entity can control multiple ASes.

recently introduced [VC07] as a response to the fast growth in the number of ASes, currently there are about 36,000 ASes [CID10]. The way ASes are connected is of a paramount importance for the performance of different applications and services. Therefore, understanding the structure of the Internet topology is central to improving and developing the current architecture.

2.1.1 The AS-Level Topology

Inter-AS connectivity is decided by several factors that represent ASes' business and strategic drives. Thus, inter-AS connectivity is better understood when considered as a union of multiple contractual agreements between different entities, that are mainly regulated by their business focus and geographical presence. To this end, understanding business motives of ASes and types of inter-AS contractual agreements is an important starting point.

An AS can be broadly classified as a transit provider or a stub network [DFJG01]. Transit providers business model is based on selling transit service to their customers (i.e. connecting them to the rest of the Internet). Customers of transit providers can be both transit and stub ASes. Stub networks regard the Internet as a communication medium and service platform that they pay to be part of. The majority of ASes are stubs, as of December 2010 there are about 36,000 ASes, 86% of them are stubs [CID10]. Contractual relationships between ASes, in general, are either transit (Provider-Customer or Customer-Provider) or settlement-free agreements (Peer-to-Peer) [Gao01]. In transit relationships customers pay their providers as explained above. Settlement-Free peering, on the other hand, means that involved ASes mutually agree to transit traffic originated at their networks and from their customers for free. In addition, when two ASes are controlled by the same administrative entity, they form a third type of relationships that is called Siblings. In Siblings relationships, involved ASes export their routes and customer routes, as well as routes learned from providers and peers. ASes can connect using dedicated physical links. However, they usually interconnect at Internet eXchange Points (IXPs), which are infrastructures that facilitate interconnecting ASes directly, since it is cheaper besides the possibility of connecting to multiple networks. Currently, there are over 350 IXPs worldwide [PCC10].

Types of ASes. A classification of ASes beyond transits and stubs can be achieved by considering their business focus and the types of agree-

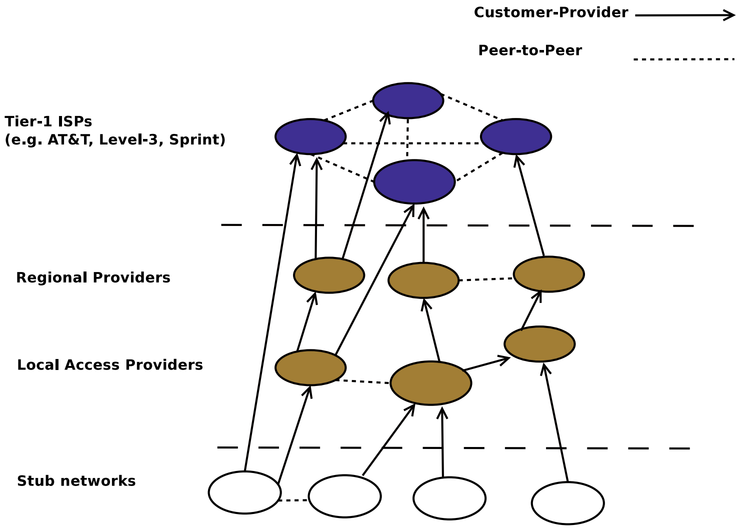


Figure 2.1: Internet hierarchy

ments they enter. Dhamdhere and Dovrolis in [DD08] taxonomized ASes into four categories: Large Transit Providers (LTP), Small Transit Providers (STP), Content/Access/ Hosting Providers (CAHP), and Enterprise Customers (EC). Generally speaking, LTP and STP form the core of the Internet; and CAHP and EC represent its periphery. LTP are large ISPs with global presence that focus on selling transit services and have a relatively large number of customers. STP ASes are regional ISPs that also focus on transit business, however, they have fewer customers. ASes fall into the other categories buy transit service from the former two types and have different business focuses. CAHP usually have incentives to improve their connectivity and to cut down their transit cost, and therefore they enter settlement-free peering agreements.

ASes, in today's Internet, are organized in a hierarchical structure as depicted in Fig. 2.1. About 20 large transit providers, tier-1 ASes, constitute the top of the hierarchy. These tier-1s do not have upstream providers. Instead, they approximately form a full mesh of settlement-free peering links among themselves. Tier-1 ASes have global presence and form the core of the Internet. Their business model mainly focuses on selling transit to other ASes. Regional and country level providers, that are not tier-1s, buy transit service from other regional/country ISPs and tier-1s. Further they usually peer with each other to reduce transit costs and improve quality of service. They also focus on selling transit, but they are restricted by their geographical presence. The remaining networks (85%) are located at the edge of the network. Stub networks pay for transit, furthermore, some of them (e.g. content and access providers) get involved in peering relationships with other stubs and transit providers. These peering relationships are motivated by content and access providers business models which are usually related to content access and delivery. In general, peering relationships at the edge are sought to reduce transit costs and to improve reachability to various networks.

Recent measurement studies [GALM08, LIJM$^+$10], however, showed that the AS-level topology is departing slowly from being strictly hierarchical and becoming flatter. This effect is consequential to the emergence of large content and access providers (e.g. Google and Comcast) that originate or receive a substantial fraction of the Internet traffic, and therefore, it is becoming sensible for transit providers to peer with them. In addition, the increase in the number of IXPs makes it easier and cheaper for such content/access providers to have a wide geographical presence and that in turn encourages transit providers to peer with them.

The global AS topology. The aforementioned discussion of the AS-level topology only describes different types of networks and their hierarchical organization. Nevertheless; the task of describing, understanding, and modeling the properties of the AS-level graph and it's evolution is crucial for assessing and improving the current architecture. This task has been the subject of much research (and heated debate) in the last decade. Early models [Wax88] considered the AS-level topology to be a random graph [ER59], where ASes as nodes connect randomly with each other. However, as described above ASes are not identical, and inter-AS links are characterized by different types business relationships. This stems from the fact that different networks have different operational and economical motives, and that inter-AS links reflect contractual and strategical agreements between networks. Hence, regarding the AS-level topology as a random graph is too simplistic and counterintuitive. But, these inaccurate views were largely consequential to the unavailability of databases that describe inter-AS connectivity. In fact, ISPs and networks are reluctant to share such information since it would potentially conceal their business goals and market positioning which could in turn help their competitors. Some databases exist, for example RIPE WHOIS [RWI10] which describes peering between ASes, however, they are maintained manually by ISPs in a volunteer fashion. Hence, WHOIS information is incomplete and sometimes outdated.

In a response to the lack of information, the networking community commissioned several routing monitoring projects. These projects can be grouped in two categories. The first group includes projects (e.g. Route-Views [rou] and RIPE RIS [RRIS]) that passively monitor the global routing system through peering with operational networks. Projects of the second group (e.g. Skitter [CAI10b], DIMES [DIM10], ARK [CAI10a]) use active measurements by executing periodic Traceroutes between a set of monitors and a large number of destinations that are spread globally. These projects became important sources for constructing and inferring AS-level graphs. However, as noted before this graph is not just a collection of nodes and links. Hence, it is important to annotate inter-AS links with the type of the corresponding contractual agreements. The work of Gao [Gao01] was a pioneering step towards annotating inferred topologies. Several works followed (e.g. [DPP03, DKF$^+$07]), that focused mainly on improving the inference quality.

Measurement data availability motivated a large body of research that aim at revealing properties of the AS-level connectivity. The seminal work of Faloutsos et al. [FFF99] was the first to illustrate that the ASes' degree distribution follows a power law where a very few nodes have many links. This observation made several researchers describe the AS-level topology as a scale-free graph [BA99]. Following research efforts, (e.g. [SFF02, GMZ03, MKF$^+$06, LKF07, LCMF08, DD08]), pointed out a set of structural properties of the AS-level graph, for example the presence of strong clustering and negative assortativity. However, some of these findings, such as the origins of the observed power law degree distribution, have been a subject to heated debate and controversy [MMB00, WGJ$^+$02, WADL04]. The debate is fueled by the fact that currently inferred topologies are incomplete; they miss a large fraction of peering links. Much research followed and focused on quantifying the amount of missing links and improving the collection of AS-level topologies [ZLMZ05, CR06, HSFK07, OPW$^+$10]. All these efforts contributed significantly to our knowledge about the inter-AS connectivity.

2.2 Routing and Addressing in the Internet

In principle the underlying topology of a network facilitates connectivity between its members. However, knowledge about existing network paths is not readily available at nodes' disposal. Therefore, there is a need to communicate such information. Routing is the process of finding a suitable network path between involved nodes, which is handled by specialized networked devices (i.e. routers). Performing routing requires signaling reachability information, comparing different potential routes, and maintaining a state that describes how to reach different destinations. Routers also need a naming convention to refer to each other and to destination networks; and protocols that regulate their communication and operation. Networks operators configure routing to achieve certain goals (e.g. minimize network delay, minimize transit costs) that are motivated by their business strategy.

2.2.1 Addressing

Naming in Internet Protocol (IP) networks is realized through IP addresses. There are two version of IP in use; IPv4 and IPv6, the former is older and more prevalent. IPv4 addresses are 32-bits numerical labels, while IPv6's are 128-bits. We refer to IPv4 addresses as IP addresses in the rest of this thesis because of its dominance in today's IP networks. An IP address is both an identifier and a locater. It identifies networked services and applications that run by a node. This is done by associating the node's IP address with another numerical label that refers to the respective application or service (i.e. port number). Further it is used by routing protocols to locate network nodes.

Routing protocols maintain a routing table that tracks how to reach different destinations. A typical entry in a routing table contains a destination address, the next hop router, and other attributes needed to describe the network path. A destination address is not a single IP address but rather a label that refers to a group of IP addresses (IP prefix), in order to reduce the state each router keeps. IP prefixes are expressed in Classless Inter-Domain Routing (CIDR) notation [FLYV93], which consists of two parts. The leading part is the first address in the group of addresses covered by an IP prefix, while the second part is the prefix length in bits. The two parts are separated by a slash. For example, the prefix 192.168.2.0/24 reserves 24-bits for the network address and 8-bits for hosts, a total of 256 addresses, two of them are reserved as network and broadcast addresses and 254 usable host addresses.

IP addresses are managed by The Internet Assigned Numbers Authority (IANA) [IAN10], which delegates them to Regional Internet Registries (RIRs), for example ARIN in North America [RIR10a], and RIPE in Europe [RIR10b]. RIRs delegates them further to local Internet registries and ISPs.

2.2.2 Routing

There are two levels of routing in today's Internet. Locally inside each AS (intra-domain routing), and globally between different networks (inter-domain routing). Therefore, we have two types of routing protocols:

Intra-domain routing protocols or Interior Gateway Protocols (IGPs) such as OSPF [Moy98] and IS-IS [O$^+$90]; are protocols used internally within a network that is usually administrated by a single entity. IGPs can be classified into two groups; distance vector (DV) routing protocols, and link state (LS) routing protocols. DV protocols use Bellman-Ford's algorithm for calculating paths, while LS protocols employ Dijkstra's algorithm for that. These algorithms are used by each router to find the shortest path (i.e. in terms of network distance) to any advertised destination. Distance between routers is expressed as hop counts or sum of link weights, the later are set to reflect metrics such as delay, bandwidth, or simply administrative decisions.

Inter-domain routing protocols or Exterior Gateway protocols (EGPs), are protocols used to relay routing information between autonomous systems. Each router keeps information about reachable network prefixes and how to reach them, further it informs its neighbors of routing changes. EGPs are very important since they are central to realizing contractual agreements between networks, and therefore, impact directly network operators' business. Border Gateway Protocol (BGP) [RL95, RLH06] is the current de-facto standard and only inter-domain routing protocol in use, which replaced older EGPs, i.e. GGP and EGPv3 [Mil84]. BGP is used for communicating routing information between neighboring routers of different ASes as well as between border routers within a single AS. When BGP runs between neighbors in different ASes, it is called external BGP (eBGP). On the other hand, when BGP runs between routers of the same AS, it is referred to as internal BGP (iBGP). In the rest of this chapter we explain the basics of BGP and its operation.

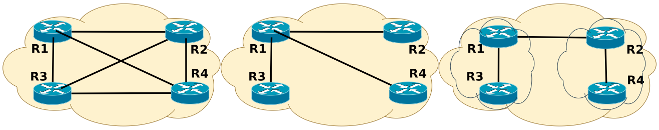
2.3 BGP Basic Operation

BGP, the only deployed inter-domain routing protocol, is a simple path vector protocol that is used for exchanging reachability information between ASes. It gives operators a large degree of freedom in defining the policies that govern the best-path selection process. Operators are free to define their own specialized rules and filters, supporting complex relationships between ASes. This flexibility is arguably one of the main factors behind BGP's success.

BGP neighbors start by establishing a TCP connection. Next, one of the involved routers sends an OPEN message as a request for establishing a BGP session. The OPEN message includes information about the sender, supported, and negotiated session's options. After establishing the session, they exchange routing information by transferring their current routing tables, note that routes exported by each neighbor are controlled by the configured routing filters and rules. After the initial bulky routes exchange, a BGP router behaves incrementally, it informs its neighbors whenever the best route for a prefix changes, by sending UPDATE messages. Note that a BGP router announces to its neighbors only one route per each announced prefix.

A BGP router sends periodic KEEPALIVE messages to maintain active sessions; every 60 seconds by default. In addition, it keeps a HOLDDOWN timer for each active session and if it does not receive an update or a KEEPALIVE message before the elapse of this timer, it considers the session to be down, signals this error to the respective neighbor by sending the so-called NOTIFICATION message, finally it removes all the routes reached via the affected neighbor. The HOLDDOWN timer's default value is 180 seconds. Besides, a BGP speaker can bring down an operational session and informs its neighbor by sending a NOTIFICATION message in case it receives a malformed UPDATE message or fails to negotiate the initial session establishment.

As pointed above there are two variants of BGP; eBGP and iBGP. In an eBGP session, a router can potentially announce all known routes to a neighbor except routes that are reachable through that neighbor (i.e. to avoid loops). However, export filters are usually used to to control route announcements. While, in an iBGP session there are restrictions depending on the organization of the iBGP topology. Next, we describe possible ways to organize iBGP topologies.



(a) Full Mesh (b) Route Reflection (c) AS Confederations

Figure 2.2: iBGP organization

2.3.1 iBGP Topologies

An iBGP topology may consist of multiple border routers. It is important to connect them in a scalable and efficient way to ensure proper routes propagation and availability across the AS. There are three widely used and recommended ways to organize iBGP topologies:

Full Mesh. In this organization iBGP routers are connected in a full mesh as illustrated in Fig. 2.2a. If we have n participants, each router will maintain $n-1$ sessions. When a participant selects a route through an internal peer, it can only be advertised externally and vice versa.

This way of connecting iBGP routers results in a large extent of path diversity within the AS. However, it also scales poorly as the number of participants increases.

Route Reflection. This organization was introduced to overcome scalability limitations of full mesh iBGP [BCC00]. Basically one or more routers will be selected as route reflectors and remaining routers are regarded as route reflector clients. A client establishes iBGP sessions only with route reflectors, which minimizes the number of iBGP sessions significantly. Figure. 2.2b depicts such topology with $R1$ as a route reflector and the other routers as clients.

A route reflector learns routes from its clients and external peers, chooses one route locally if multiple routes are available, and communicates the best selected routes to its clients. While clients advertise externally learned routes to their route reflectors. Furthermore, when there are multiple route reflectors, they can be organized in a full mesh of iBGP sessions or in a hierarchy.

Route Reflection improves significantly the scalability of iBGP topologies. However, it limits the extent of path diversity available at the disposal of the clients; a route reflector passes along the best selected route to its clients, and therefore, it hides less preferred routes. This incomplete view of

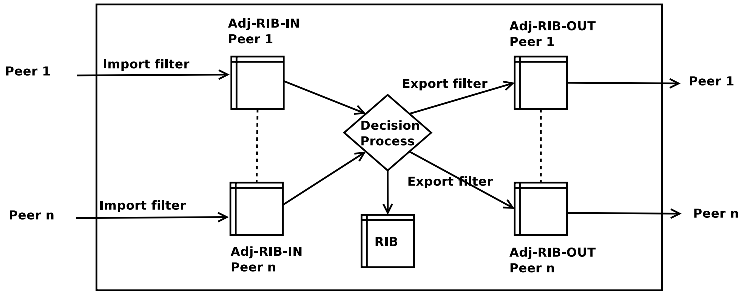


Figure 2.3: Typical BGP router control plane functions

available routes can lead to the propagation of unnecessary BGP messages between external peers when a route fails [SFPB09].

AS Confederations. This organization was also introduced to overcome scalability limitations of full mesh iBGP [TMS01, TMS07]. An AS will be divided into sub-ASes which are grouped in one confederation. The confederation is a group of routers that use the same AS number when communicating with the rest of the Internet. Routers in a sub-AS are connected in a full mesh of iBGP sessions, and each sub-AS maintains small number of sessions with the other members of the confederation. Figure. 2.2c shows an AS confederation of two sub-ASes. The first sub AS consists of R1 and R3; while R2 and R4 are members of the second sub AS.

The sub-ASes are assigned private AS numbers[2] to be used only inside the AS. Route Reflection and AS Confederations serve the same purpose; favoring one of them is a design choice [Dub99].

So far we describe the basic operation of BGP sessions; differences between eBGP and iBGP; and recommended ways to organize iBGP topologies. In the subsequent subsections we explain how BGP learns about routing changes and chooses the best route to a destination. In addition we describe the inter-domain routing policies that are expressed and enforced by BGP.

[2] IANA has reserved AS64512 through to AS65535 to be used as private AS numbers.

Path attribute	Type
AS_Path	WELL-KNOWN MANDATORY
Origin	WELL-KNOWN MANDATORY
Next_Hop	WELL-KNOWN MANDATORY
Local_Preference	WELL-KNOWN DISCRETIONARY
Community	OPTIONAL TRANSITIVE
Multi_Exit_Discirminator (MED)	OPTIONAL NON-TRANSITIVE

Table 2.1: A subset of BGP Path attributes

2.3.2 Updating BGP Routes

BGP is an incremental protocol, where a BGP router informs its neighbors of routing changes using UPDATE messages. There are two types of UPDATE messages: announcements and withdrawals. Announcements advertise a new network prefix or modify reachability information of a previously advertised prefix. Withdrawals state that a previously reachable prefix is no longer reachable.

Withdrawals are simpler than announcements since they only contain besides the message header a list of prefixes that are withdrawn. However, announcements can contain multiple fields that describe attributes of announced routes (i.e. path attributes). Path attributes are classified into WELL-KNOWN MANDATORY,WELL-KNOWN DISCRETIONARY, and OPTIONAL. WELL-KNOWN MANDATORY attributes must be included in every announcement. WELL-KNOWN DISCRETIONARY attributes on the other hand, must be supported by all implementations but need not to be included in every announcement. Furthermore, OPTIONAL path attributes can be grouped into transitive and non-transitive. Transitive attributes will be sent to neighboring ASes by intermediaries. Table. 2.1 presents the main path attributes. We elaborate on this list in the following:

AS_Path: the AS_Path is a list of all ASes that an inter-domain route crosses to reach the prefix originator. It appears as a space separated series of AS numbers, with the prefix originator at the end and the next hop AS at the beginning.

Origin: this attribute specifies the source a router receives a route from. It can take one of three values. Internal, External, or Incomplete. Internal means that the route is received via an IGP. Routes received via an EGP are labeled

External. Incomplete indicates that the source of the route is unknown, these routes are most likely inserted as static routes.
Next_Hop: for a route learned over an eBGP session this attribute is the IP address of the external peer that announced it. While for a route reachable over an iBGP session, the Next_Hop is the address of the border router that announced it.
Local_Preference: is a numeric value set within an AS to indicate the preference of a route. The higher the value, the more preferred the route. Local_preference has a relevance only inside a single AS, therefore it is not relayed to neighboring ASes.
Community: is a 32-bit optional transitive attribute [CTL96, STR06]. Communities can be used to provide extra information about a route. For example, geographic communities indicate the geographic location where the route was received. Furthermore, they can be used by a route originator to influence the way its route is handled by neighbors. One such example is the default No-Export community that marks prefixes as not to be advertised by neighboring ASes to their external BGP peers.
MED: is an integer optional non-transitive attribute. When two ASes interconnect at more than one point, one of them may advertise the same route to a prefix at multiple egresses. By setting different MED values when announcing a prefix at different egress points, the advertising AS can indicate an entry point at which it prefers to receive the traffic destined to the respective prefix. The lower the MED value the more preferred the respective ingress point is.

When a BGP router receives an update message from one of its neighbors, it first applies the import filter corresponding to the that neighbor. Import filters are used to accept routes, and tag them for example by setting local preferences. As illustrated in Fig. 2.3 a typical BGP router keeps a routing table for each neighbor, Adjacency RIB IN (Adj-RIB-IN), where it stores all routes learned from that neighbor. After applying import filters, the router updates the corresponding Adj-RIB-IN by adding, modifying, or removing the updated route(s).

Whenever a route changes, the BGP router must run a decision process where it compares existing alternatives and picks the best available route. Routing and forwarding tables will be updated subsequently to reflect the new routing decision. Further this new routing decision needs to be announced to peers.

1- Highest Local_preference

2- Shortest AS-Path length

3- Lowest Origin type

4- Lowest MED

5- eBGP over iBGP routes

6- Lowest IGP cost to the egress

7- Lowest Router ID

Figure 2.4: BGP decision process

BGP routers set export filters for each peer to decide whether to export a route or not, and to modify path attributes before updating peers. A typical BGP router keeps a routing table for each neighbor, Adjacency RIB OUT (Adj-RIB-OUT), where it stores all routes exported to that neighbor. After applying export filters, the corresponding Adj-RIB-OUT is updated if applicable. Finally, an update message will be sent out to the respective peer.

As explained above, BGP decision process is responsible of comparing routing alternatives and choosing the best route. Therefore, it is central to the operation of BGP. Next, we explain this decision process and different involved metrics.

2.3.3 Decision Process

BGP decision process is a multi-step process that compares routing possibilities until it identifies the best existing route. Figure 2.4 displays the standard version of the decision process.

The decision process starts by comparing the local_preferences and picking the route(s) with the highest value. If multiple routes have the same local_preference, the decision process proceeds by comparing the length of their AS-Paths and prefers the route(s) with the shortest AS-PATH. A route with a shorter path is considered to be closer in terms of network distance and thus better.

In cases when more than one route remain after the second step, the route(s) with the lowest origin type will be favored. IGP is lower than EGP, and EGP is lower than Incomplete. Next, the routes(s) with lowest MED are chosen, by doing so a BGP router follows routing preferences indicated by the next-hop AS.

If several routes make it to step five or six, the decision process will consider routes that have shorter distance to the egress. This type of decision is referred to as hot-potato routing. By performing hot-potato routing, the AS reduces the volume of traffic redirected in its internal network.

As a last resort, router IDs (i.e. addresses) are compared to tie-break routes that are not differentiated by any of the aforementioned metrics.

2.3.4 Policies

As explained in Sec. 2.1.1, there are currently two dominant types of business relationships between ASes; Provider-Customer and Peer-to-Peer. The interaction of these relationships and ASes' aim of optimizing transit cost resulted in a form of routing that prefers economically-efficient routes over shorter network paths. This is often referred to as policy routing. Further, it imposes stringent requirements on routing information dissemination.

In today's Internet, ASes use the so-called "no-valley and prefer-customer" policies [Gao01]. Routes learned from customers are announced to all neighbors, while routes learned from peers or providers are only announced to customers. In addition, an AS prefers a route learned from a customer over a route learned from a peer, over a route learned from a provider. Strictly speaking, an AS path conforms with "no-valley" policies if the following conditions hold true. First, a Provider-Customer edge can only be followed by a Provider-Customer or Sibling edges. Second, a Peer-to-Peer edge can only be followed by a Provider-Customer or Sibling edges. Simply put, "no-valley" routing means that an AS does not provide transit service between its providers, or between its peers; and it does not transit traffic between a peer and a provider and vice versa.

The fact that BGP can order routes using metrics other than their length is crucial for implementing policy routing. For enforcing "prefer-customer" policies, an AS needs to configure import filters to set the local_preferences of routes based on their sources (i.e. providers $<$ peers $<$ customers). Furthermore, export filters should be set to enforce "no-valley" policies.

2.4 Summary

This chapter has presented an overview of today's Internet architecture. It has started by describing how different networks and ISPs interconnect together and the typically involved contractual agreements. Further it has highlighted the concepts of addressing and routing. Then it has elaborated on explaining the current de-facto standard inter-AS routing protocol; illustrating its simplicity, flexibility, and its central role in the current architecture. However, given the complexity of the inter-AS connectivity and the growing Internet size, the emergence of new routing challenges is inevitable. The next chapter discusses emerging and foundational issues with the global routing system in details.

Chapter 3

BGP: Open Issues and Limitations

The current version of inter-domain routing protocol, BGPv4, was introduced in 1994. Since then, both the size and importance of the Internet have increased significantly. Today's Internet consists of about 36000 ASes in comparison to less than 3000 ASes in 1997. Further the Internet has increasingly become a mission critical platform and an essential part of our daily life. BGP has handled this in an impressive way; the flexibility of BGP is arguably one of the main factors behind its success. But recently, there have been concerns about BGP ability to continue coping with the growing network size in an efficient, reliable, and a secure way. This chapter discusses open issues and limitations in the current inter-domain routing architecture i.e. scalability, reliability, correctness; with a focus on issues related to this thesis work.

3.1 Origin of Challenges

Some of the challenges that face today's global routing system are fundamental to the design choices behind BGP. Others are related to practices and configuration errors. Identifying the origins of challenges is crucial for fixing and mitigating them.

To this end, we start by thinking the principal goals that influenced the design of today's global routing architecture. Basically, any efficient inter-domain routing protocol should be scalable and must support business policies. There are two limiting factors for BGP scalability; the routing table size and the rate of routing updates, which dictate BGP routers' memory and CPU requirements. To mitigate the impact of these factors, the amount of information and level of details communicated by BGP should be kept minimal. This is achieved by aggregating topology and routing information. The aggregation is done at different levels, examples include: prefix aggregation; restricting BGP updates to include few details about underlying topology and failures; and using route reflection to limit number of routes exchanged over iBGP sessions. On the other hand as explained in Sec. 2.3.4, BGP allows ASes to flexibly set policies about importing, exporting, and ordering routes.

In general, open issues that face the current inter-domain routing architecture can be classified under two categories. The first category includes issues that are caused by BGP falling short of supporting scalable policy routing. Limitations in the second class are by-products of choices made to ensure BGP scalability and flexibility.

First, we consider limitations that fall into the first category. Regarding policy based routing, BGP is currently able to support common business policies in a flexible way. However, the scalability of BGP is questionable with a dramatic growth in the number of ASes and network prefixes. Recently, this has raised a significant concern among both Internet operators and researchers; A workshop organized by the Internet Architecture Board (IAB) concluded that "routing scalability is the most important problem facing the Internet today" [MZF07]. The concern is that we are soon reaching the point where the global routing system, and the core routers in particular, will no longer be able to keep up with routing dynamics.

We now turn to issues that are by-products of BGP design. In order to support scalability, the amount of information and level of details communicated by BGP should be kept minimal. This is achieved by aggregating topology and routing information. The aggregation is done at different levels, examples include: prefix aggregation; restricting BGP updates to include few details about underlying topology and failures; and using route reflection to limit number of routes exchanged over iBGP sessions. This thorough aggregation, however, does not come without negative side effects. Routers are left with an incomplete view of the topology and no information about root causes behind routing updates. This results in challenges such as: prolonged convergence process, exchange of unnecessary amount of update messages, and transient failures.

Policy routing, on the other hand, can potentially lead to oscillations that result in prolonging or even preventing routing convergence. This happens when a group of ASes has a set of conflicting preferences. Moreover, common routing policies implies a monotonic preference of routes based on their sources (i.e. providers < peers < customers). This makes BGP vulnerable to bogus advertisements from sources with higher preference.

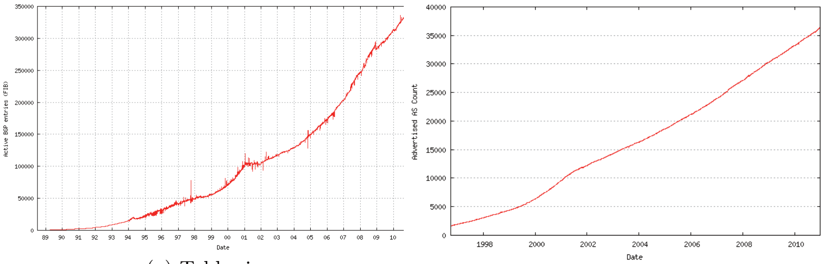
So far, we have presented a broad overview of challenges and issues confronting today's inter-domain routing architecture. Following subsections discuss the most relevant challenges.

3.2 Scalability

The scalability of BGP routing is a major concern for the Internet community. Scalability is an issue in two different aspects: increasing routing table size, and increasing rate of BGP updates. The growing size of the routing table requires increasingly larger fast memory, but it does not necessarily slow down packet forwarding as long as address lookups are performed using TCAMs or constant-time longest-prefix matching algorithms [Var04]. Churn, however, is a more serious concern because processing BGP updates can be computationally intensive (updating routing state, generating more updates, checking import/export filters), and it can trigger a wide-scale instability. If the current best route to a destination is modified, the global RIB and the FIBs on the line cards need to be updated. To make things worse, routing updates are known to be very bursty, with peak rates several orders of magnitude higher than daily averages. When the rate of updates becomes too high, the fear is that there will be (or there are already) periods when routers will be unable to maintain a consistent routing table. On the other hand, churn is far more complex with many influencing factors that drive its amount. The rest of this section elaborates on these two aspects.

3.2.1 Routing Table Size

The size of the global routing system has increased tremendously during the past 15 years. The number of advertised ASes, as depicted in Fig. 3.1b has grown from 2,473 in mid-1997 to 36,383 in the end of 2010 (a factor of 14.7). While the number of network prefixes, as shown in Fig 3.1a, has increased from 46,948 in mid-1997 to 341,173 in the end of 2010 (a factor of 7.3). The reported number of prefixes, is a measure of the default free zone



(a) Table size

(b) Number of ASes

Figure 3.1: The global routing system over time. Source:http://bgp.potaroo.net

(DFZ) routing table size. A router belongs to the DFZ if it does not use a default route to reach any destination network. Handling such large growth in the routing table size imposes certain requirements on routers hardware. An increasingly large memory is needed to cope with the increasing number of table entries. In addition, faster CPUs are required to perform fast address lookups and to handle frequent table re-computations.

In fact, the growing size of the routing table was recognized as a challenge in early 1990s. This recognition along with other factors, motivated the introduction of CIDR [RL93, FLYV93] which, by departing from the original classful addressing, resulted in a flexible address allocation. CIDR enables a more aggressive aggregation of network routes, i.e. one routing table entry summarizes the route to multiple prefixes. Such aggregation is done topologically, an upstream provider delegates a subset of its address space to its customers, further it announces one prefix that summarizes the whole address space. For example, an upstream provider owning a /19 address block can split it to four /21 blocks that may be assigned to customer ASes. A shorter prefix that describes a number of destinations is said to be less specific. While, a longer prefix that describes a subset of the destinations covered by a less specific route is said to be more specific. A necessary condition for aggregation is that a customer AS must not announce a more specific prefix to upstream providers other than the delegator. CIDR slowed down table size increase between 1994 and 1998 changing it from superlinear to linear [Hus02], however, the fast increase returned again after 1998.

The table size growth motivated several research efforts to measure the increase and investigate its root causes. Notably, continuous efforts by Huston [Husb] and Bates et al. [BSH] are central sources of information in this respect. An early assessment [Hus02] illustrated that the table size at least doubled every year between late 1998 and early 2001. Later, Bu et al. [BGT04] investigated the contribution of factors such as load balancing and failure to aggregate to table size growth. Meng et al. [MXZ$^+$05] studied BGP routing table evolution in the light of allocated IPv4 address blocks in a six year period (1998-2004). Conclusions of aforementioned efforts and the IAB report [MZF07] confirmed that part of the observed table size increase is inevitable because of the Internet growth. However, they attributed a large fraction of the observed increase to aggressive de-aggregation of network prefixes. Several factors and operational practices are cited as causes for de-aggregation of prefixes:

Provider Independent Address Space. Pre-CIDR, assignments of IP addresses were handled by IANA on a first come first served basis. Later this responsibility was transferred to RIRs, which introduced a new approach for assigning IP addresses by classifying prefixes into Provider Independent (PI) and Provider Aggregatable (PA) [BCK$^+$10]. PI prefixes are given by the respective RIR to requesting sites and can be kept indefinitely. PA prefixes, on the other hand, are handed by ISPs from their own allocated address space to their customer ASes. Customers are expected to return PA prefixes when changing providers. Returning assigned prefixes implies a need for renumbering, which can be cumbersome since IP addresses are usually hardcoded in configuration files of routers and servers. This inconvenience combined with ASes desire to avoid of provider lock-in led to a high demand for PI prefixes. Prefixes assigned this way are clearly not aggregatable by upstream providers which consequently contributes to an increasing table size.

Multihoming. By multihoming, we refer to the case where an AS has more than one upstream provider. ASes connect to several upstreams to increase their resilience to failures and ability to load-balance their traffic. There are no restrictions on multihomed sites in using either PI or PA addresses. When a multihomed AS uses PA prefixes, it receives a set of prefixes from each provider. However, more specific prefixes are usually not aggregated by upstream owners to ensure the reachability of these prefixes via all upstreams. For example, if an AS is multihomed to two upstreams A and B and it announces a prefix p to both providers, p was assigned by A from its address space. In case A decides to aggregate p, other networks will send their traffic to p through B(i.e. not the principal owner of respective address space) because of the longest-prefix matching rule. This conflict motivates upstream owners not to aggregate more specific prefixes. Multihoming aggravates routing table size problem by preventing the aggregation of PA prefixes [ALD+05].

Traffic Engineering. Operators usually want to control the flow of their incoming and outgoing traffic, in order to improve the performance of their networks. However, BGP lacks support for common traffic engineering tasks such as distributing incoming traffic over several downstream links, and sending traffic over lower-cost links. One way operators follow to work around these limitations is deaggregating their address space and announcing different prefixes to different upstreams

Address Space Fragmentation. By address space fragmentation we mean that an AS announces a number prefixes that can not be aggregated. Potential causes are: Pre-CIDR's assignments of non-overlapped prefixes, and RIRs handing non-contiguous address blocks i.e. forced by the exhaustion of IPv4 addresses.

Security. Networks divide their address space to smaller chunks in order to reduce impact of prefix hijacking, i.e. their prefixes being announced by other networks [BFZ07]. By doing this, they ensure that their traffic will not be hijacked given that the hijacking prefix is shorter (the longest-prefix matching rule).

However, a recent measurement study that investigated extent of prefix de-aggregation and its impact on BGP scalability [CMU+10], while asserting existing problems caused by de-aggregation, concluded that: "there is no trend towards more aggressive prefix de-aggregation or traffic engineering over time". Instead, the authors attributed the increase in the DFZ table size to the growth at the periphery of the AS-level topology [DD08]. These findings are in well accordance with another recent study by Carpenter [Car09], that observed a consistent linear relationship between BGP table size and number of ASes.

To sum up, the past decade or so witnessed a fast increase in the DFZ table size; that is caused by prefix de-aggregation, and the growth at the periphery of the AS-level topology. The problem may be exacerbated in the future given the projected IPv4 address space exhaustion by late 2011 [ipa]. The depletion of IPv4 addresses can result in a transition to IPv6, which is projected to increase RIB and FIB sizes by four times [MZF07]. Another possibility is that operators will resort to advertising longer prefixes, currently prefixes longer than /24 are filtered by most networks, which in turn can potentially lead to more aggressive de-aggregation.

3.2.2 Churn

An earlier study by Huston and Armitage reported an alarming growth in churn [HA06]. During 2005, the daily rate of BGP updates observed by a router in AS1221 (Telstra) almost doubled, while the number of prefixes grew by only 18%. Based on these measurements, the authors projected future churn levels and concluded that current router hardware will need significant upgrades in order to cope with churn in a 3-5 years perspective.

In general, an increase in the routing table size (number of routable prefixes) also increases churn. However, churn is a result of a complex interplay of 1) the *routing protocol*, including policy annotations and various BGP mechanisms and implementations choices, 2) *events* like prefix announcements, link failures, session resets, traffic engineering operations that generate routing updates, and 3) the characteristics of the Internet *topology.* While some of these factors are organic due to the growth of the Internet; others are results of BGP's limitations, implementation decisions, and configurations. Next, we explore several factors that contribute unnecessary churn.

Path Exploration: Following a route withdrawal a BGP router may try several transient routes before converging to a new stable path or declaring the affected destination prefix unreachable. For example, in Fig. 3.2, AS 1 initially announces a prefix P to its three upstreams; hence AS 5 has three routes to P, but it prefers the shortest among them. If AS 1 withdraws P; AS 5 will first loose the current best path, choose instead the second shortest path (i.e. {2,1}), and send an update message informing its neighbors of the change. Later, it will receive a withdrawal from AS 2 that invalidates the current best path. This will force it to choose the last available route(i.e. {4,3,1}), and inform its neighbors. Then, AS 4 will eventually withdraw P from AS 5. Consequentially, AS 5 will not have a route to P and send a withdrawal to its neighbors. The first two update messages sent out by AS 5 are clearly superficial, since these intermediate paths were not operational. Hence, path exploration results in unnecessary churn.

The phenomenon of *path exploration* was first discussed by Labovitz et al. in [LABJ00], and upper and lower bounds for the number of updates

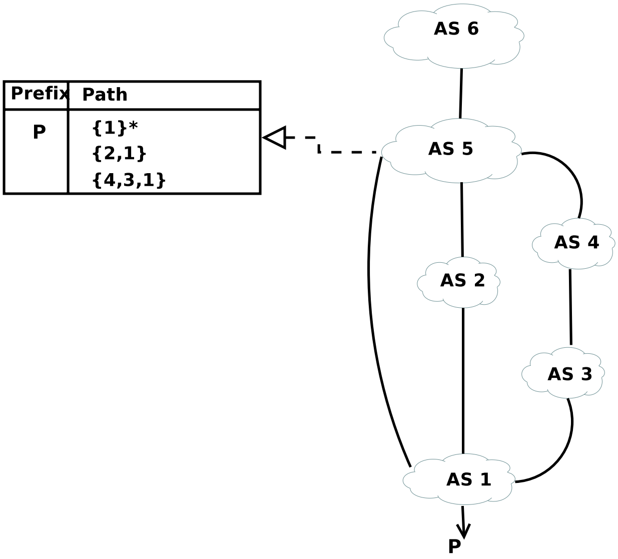


Figure 3.2: Path Exploration

exchanged during convergence were given. In a follow-up work, it was shown that the duration of path exploration depends on the length of the longest possible backup path to the affected destination [LAWV01]. In a more recent measurement study, it was shown that path exploration is less severe at the core of the network than at the edges [OZP$^+$06].

Main reasons behind the path exploration phenomenon are: the path vector nature of BGP that implies trying other routes when a route is unavailable; BGP updates do not carry information about root causes of changes; and discrepancy in propagation speed of BGP updates over different network paths. Hence, reducing the impact of path exploration can be achieved by disclosing information about root causes behind routing changes, and delaying the propagation of intermediate updates. Holding back an intermediate update increases the chance that it will be invalidated by a subsequent update, resulting in less churn.

To delay the propagation of intermediate updates, the BGP standard [RLH06] recommends the use of a MinRouteAdvertisementIntervalTimer, which specifies the minimum time interval between two consecutive updates for a destination prefix (MRAI timer). The recommended value of this timer on eBGP sessions is 30 seconds. To avoid peaks in the number of received updates, the standard recommends jittering the MRAI timer by multiplying its value with a random number between 0.75 an 1. While, MRAI can substantially reduce churn, it prolongs BGP convergence and that affects data forwarding negatively [WMW+06]. We investigate this in Chapter 7.

Griffin and Premore [GP01] showed through simulation that the optimal MRAI value that minimizes convergence delay and network-wide number of updates is different from one network to another depending on the size and structure of the topology. In addition, the convergence process will depend on the nature and location of the underlying routing events; routing changes that result in a complete withdrawal of a network prefix involve more path exploration than those ending with an alternative path [OZP+06]. Hence, it is difficult to find generic timer settings that work well in the Internet based on these results. Later work [QHL05] followed in the same direction and demonstrated through formal analysis, simulation, and PlanetLab experiment that the optimal MRAI value can be 5-10 times lower than the current recommended value. In addition, several other papers [DS04, BBAS04, LT06, LBU09] proposed modifying the MRAI timer to reduce the convergence delay.

Other methods (e.g. RCN [PAMZ05] and EPIC [CDZK05]) suggest adding information about root causes of routing changes to BGP updates. This way BGP routers can avoid exploring paths affected by same event. However, such solutions are difficult to implement; ISPs often are not willing to reveal details about their internal network structure.

Flapping and Highly Active Prefixes: Several studies focused on analyzing the contribution of different ASes and prefixes to the observed churn. Broido et al. [BNkc02] showed that a small fraction of ASes is responsible for most of the churn seen in the Internet. Similarly, the work in [RWXZ02], [VIRZZ05], and [Husa] reported that only a small subset of highly active prefixes are responsible for a large percentage of churn. This high instability is caused by misconfigurations, persistent flapping, topological failures, and path exploration. In particular, a persistent flapping behavior can result in excessive churn depending on the flapping link location. Unfortunately, unstable links and flapping prefixes are reported to be common in today's inter-domain routing system [WMRW05, BFF05].

Route Flap Damping (RFD) [VCG98] was proposed to suppress continuously flapping prefixes. It is a penalty based system, where routes are not advertised if the penalty attached with them has crossed a certain threshold. This penalty decays over time and a suppressed route can be re-used if its penalty becomes lower than the re-use threshold.

Mao et al. [MGVK02] were the first to investigate the effect of RFD on BGP convergence time. They showed that RFD can significantly increase the convergence delay of a well-behaved routes by misinterpreting normal path exploration and treating it as route flapping. Furthermore, they proposed augmenting BGP updates with extra information to signal if the announced route is less or more preferred than its predecessor in order to avoid false-positives. The authors in [ZPMZ05] showed that misinterpreted path exploration accounts only for 30% of false positive route suppressions. The interaction between RFD reuse timers at different nodes accounts for the rest. In order to eliminate these effects they proposed employing root cause notification [PAMZ05]. A later work [DCK+04] presented a new algorithm to identify path exploration through adding a new community attribute specifying announced routes relative preference. The negative effects of RFD motivated RIPE to recommend turning it off [SP06].

Redundant Updates: Labovitz et al. [LMJ97] were the first to show that BGP suffers from excessive churn caused by pathological protocol behavior and suggested practical ways to fix broken BGP implementations; a main concern was that these redundant updates are pointless. By “duplicate announcement”, we mean an announcement that is identical to the last seen announcement for the same prefix, i.e., no change in either the AS-path or in any of the transitive route attributes. In a follow-up work [LMJ99], they found that better router implementations had reduced churn by an order of magnitude, but that duplicate announcements still contributed much unnecessary churn. A later measurement study [LGW+07] concluded that the state of BGP routing is “healthier” than it was a decade ago with duplicates accounting for 15% of BGP updates. Other studies, including this thesis, put this number higher. A recent study [PJL+10] attributed duplicates in BGP churn to interactions between iBGP and eBGP.

Avoiding redundant updates requires deploying improved BGP implementations that avoid sending redundant updates to the global routing system. Such improvements would need, however, per-neighbor state at BGP routers to keep track of what was sent to each peer earlier, so that duplicate updates can be detected before they are transmitted. Arguably, these changes are not worth doing, given the lightweight handling of duplicates.

Lack of Path Diversity: The positive impact of Route Reflection and AS Confederation on iBGP scalability, comes at a price of reduced path diversity as explained in Sec. 2.3. When a prefix fails, this lack of diversity may result in sending of withdrawals to external peers [SFPB09] despite the availability of other routes inside the AS. Such withdrawals trigger a convergence sequence at affected neighbors and thus the sending of unnecessary updates.

The authors in [SFPB09] proposed using communities to inform routers of the existence of other unannounced routes. Another solution is the newly proposed Add-Paths [WRCS10] extension that allows BGP routers to advertise multiple path.

3.2.3 Solving Scalability Challenges

Scalability limitations of the current inter-domain routing architecture have motivated several research efforts during the past few years. In addition, there are currently several research initiatives and projects that work on addressing challenges of the current architecture and designing future Internet, e.g., [fir], [gen], [RRG]. Some of these efforts suggested evolving the system in place; others proposed radically new architectures. In the following, we highlight some of these efforts.

Subramanian et al. proposed HLP [SCE+05], a hybrid link-state and path-vector protocol that leverages existing inter-domain business policies model to prevent local routing information and changes from spreading globally. HLP is reported to reduce churn by a factor of 400, but there are several open issues, like mapping from ASNs to IP prefixes, and routes propagation between its link-state and path-vector parts. As a response to the increased routing table sizes, a radically different routing strategy called compact routing has been proposed [KkcFB07]. This approach can give routing table sizes that scale logarithmically with the number of routable addresses, but performs poorly under dynamic conditions.

Recently several clean-slate architectures, centered around separating the core of the Internet from its edge [JMY+08], have been suggested (e.g. HAIR [FCM+09] and LISP [Mey08]). LISP, for example, is a router-based solution that proposes separating IP addresses to EndPoint Identifiers (EIDs) and Routing Locators (RLOCs). EIDs identify end hosts and are not globally routed, while RLOCs are globally routed addresses of routers. In addition, there is a mapping service that maps EIDs to RLOCs. This way, the number of routable prefixes will drop dramatically.

To summarize, inter-domain routing scalability is an issue in two different aspects: increasing routing table size, and increasing rate of BGP updates. The later is a more serious concern and to make things worse it is far from being well-understood. *We believe that there is a need to improve our understanding about the impact of different factors (e.g. topology, routing events) on churn; and to characterize the severity of the problem. This understanding is vital for improving the current architecture and an important input to designers of future architectures.*

3.3 Slow Convergence

Fast and guaranteed convergence after failures is an important performance requirement of routing protocols. This metric dictates the time needed to re-route affected traffic. BGP, however, is known to suffer from prolonged convergence and even persistent oscillations in some cases. Many studies (e.g. [WMW$^+$06] and [KKK07]) reported the negative impact of delayed BGP convergence on data forwarding.

Early measurement work [LABJ00] showed that BGP convergence can take several minutes because of path exploration. To reduce this, they proposed implementing sender side and receiver side loop detection; and not applying MRAI to explicit withdrawals. In a follow-up work [LAWV01], they investigated the impact of topology and routing policies on BGP convergence, and demonstrated that the convergence delay for a multihomed destination is mainly dependent on the length of the longest possible backup path. Moreover, they suggested re-thinking the MRAI timer. A later study [OZP$^+$06] confirmed the prevalence of path exploration and slow convergence in the Internet.

BGP convergence delay does not matter, when there is no alternative route in the network for a failing destination. However, most failures are transient; the authors in [WGWQ05], using one month of BGP data from several tier-1 and large ISPs, reported that over half of observed failures were transient with a failure duration up to 100 seconds. In the majority of these cases operational alternative routes existed. However, the path vector nature of BGP triggered a lengthy path exploration process. Several reactive and proactive recovery solutions were proposed (e.g. [BFF05], [KKKM07],

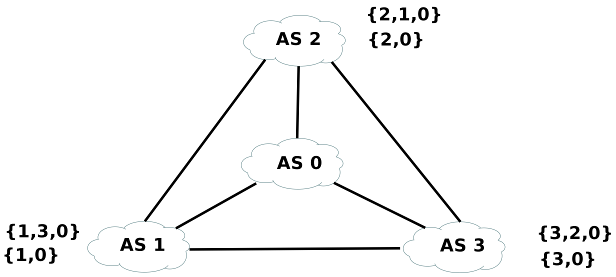


Figure 3.3: Policy dispute

and [LGGZ08]) to speed up the convergence process. However, the problem is still affecting operational networks, since none of these solutions is implemented at a large scale.

In addition to delayed convergence, it was shown [GW99, VGE00, GSW02] that the interaction between independent policy configurations in some cases may lead to persistent route oscillations. These oscillations occur when, for a group of ASes, there is no possible routing solution that leaves at least one of them with no better route to select than the currently chosen. Figure 3.3 illustrates a famous example of such disputes known as the bad gadget [GW99]. In this configuration, each AS prefers to reach AS 0 through the anti-clockwise path rather than the direct path, which consequently leads to BGP divergence. Determining whether a group of ASes would experience route oscillations induced by policy disputes is an NP-complete problem [GSW02]. Gao and Rexford [GR00] gave a set of guidelines for ISPs that guarantees BGP convergence. The guidelines assume that every AS considers each neighboring AS as either a provider, a peer, or a customer. Further, it requires that ASes apply the "no-valley" policies to avoid cycles. Fortunately, in today's Internet ASes generally conform with these guidelines, however, ASes may deviate from them and cycles in AS-level topology are known to exist [DSK08].

Another common type of policy induced instability is MED oscillation [GW02a, GW02b]. This anomaly is internal to ASes and results in routers persistently changing their egress selection. There are two reasons behind MED oscillations. The lack of path diversity caused by Route Reflection, and the constraint that MED is considered in the BGP decision process only if tied routes share the next hop AS. Therefore, existing solutions suggest modifying iBGP to advertise multiple paths [BOR+02, WRCS10]

3.4 Misconfiguration

BGP configuration dictates the way prefixes are imported from neighbors, treated by the decision process, and exported. The task of configuring BGP can be quite complex depending on the number of routers and policies an AS wants to deploy. In fact, a large AS usually has several hundreds of routers and thousands different policies, this may result in configuration files that are over 10,000 lines [FB05]. To make things worse, configuration is usually done manually with commands that differ across vendors and routers operating systems versions. This makes human operator error one of the main factors behind routing faults.

The impact and extent of misconfiguration was subject of several studies. Labovitz et al. [LAJ99] studied the origins of failures that occur within provider backbones and concluded that 12% of instabilities were probably caused by human error and misconfiguration. The work in [ZPW+01] analyzed prefixes with multiple origin ASes and identified misconfiguration as one of the potential causes. However, Mahajan et al. [MWA02] were the first to investigate misconfigurations systematically. They studied two types of faults, prefix hijacking and export misconfiguration, and showed that up to 1200 prefixes per day suffered from such faults. A later measurement study [XGF07] showed that misconfiguration is a principal cause of persistent forwarding loops in the Internet. Aforementioned studies concluded that misconfiguration leads among others to an increase in churn, policy violation, prefix hijacking, and forwarding loops.

Some misconfiguration incidents caused severe global Internet outages. One famous such incident took place on the 25th of April 1997, when a small ISP in Florida announced a bogus reachability information to the majority of the Internet causing an Internet-wide instability that lasted for two hours [700]. Several other major prefix hijacking events happened during the past decade, examples include AS9121 leak of 100,000 prefixes [PPU05], and Youtube's prefix hijacking by Pakistan Telecom [RIP08]. Recently, on the 8th of April, 2010 a customer AS (AS 23724) of China Telecom hijacked 50,000 networks [Cow10]. Figure 3.4 shows the number of prefixes with multiple ori-

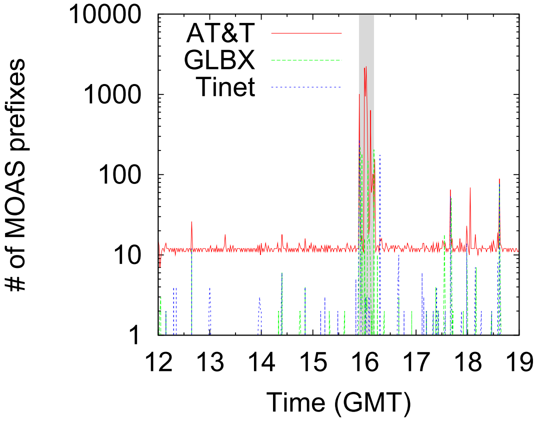


Figure 3.4: Example of misconfiguration

gin AS (MOAS) [1], as seen by monitors that belong to three different tier-1 ISPs, on the 8th of April, 2010. The plot is based on BGP dumps obtained from the RouteViews project [rou]. We observe an increase in the number of MOAS prefixes at the time of the hijacking incident (i.e. the grey shaded area). In addition, we note that the impact varies across monitors; possible reasons could be topological effects and the deployment of defensive methods. However, it seems that even large tier-1 ISPs are vulnerable to such attacks; AT&T was the most affected with traffic disruption to 2,000 prefixes, i.e., 4% of all hijacked prefixes.

Several tools and systems were proposed to help ISPs managing and verifying their configurations. RCC [FB05] suggested performing these task by statically examining configuration files. Applying RCC on configuration files from 17 networks revealed 10,000 configuration mistakes. Other tools focused on reducing configuration complexity through automation and providing better ways to represent policies. Gottlieb et al. [GGRW03] proposed a system for automatically provisioning BGP customers. Further [BFM$^+$05, VQB09] presented frameworks for formulating, verifying, and configuring routing policies.

[1] The presence of a prefix with MOAS indicates a case of misconfiguration or deliberate hijacking.

3.5 Security

BGP is designed to ensure operational simplicity with no emphasis on routing security or providing performance guarantees. In general, the current interdomain routing architecture is a trust-based system. An AS usually accepts routes announced by its neighbors unconditionally, and selects paths that best fit configured local routing policies. Improving global routing security has lately, following a series of high profile Internet outages, been recognized as a strategical and vital task [U.S03, sid].

The lack of built-in support for routes authentication and validation makes the Internet routing infrastructure highly susceptible to several threats. Next, we describe a set of well known issues.

Prefix hijacking. This vulnerability comes in one of two forms either a hijacking of all addresses covered by prefix or a subset of them through announcing a more specific route. A bogus route will be selected by an AS if it is more prefered policy-wise over legitimate alternatives, or as a result of longest prefix matching. The impact of hijacking can be anything from affecting few destinations to causing a global outage as demonstrated in the previous section. Most of the known hijacking incidents are unintentional due to misconfiguration, however, purposeful incidents aiming to impersonate networks, launch DDOS attacks, or gain a legitimate address space for spammers; are not unheard of. For example, Ramachandran and Feamster showed that up to 10% of all spam is sent from addresses belonging to short-lived hijacked prefixes [RF06]. In addition, prefix hijacking can be leveraged to perform a man in the middle attack [PK08]. In this case, an attacker announces a more specific of a prefix belongs to the victim. This result in attracting traffic destined to the hijacked prefix; the traffic will further be relayed in a transparent way to the victim. The last part is done by keeping one legitimate path to the victim.

***Bogon* prefixes.** Another issue is announcing prefixes that are unallocated or part of the private address space. ASes avoid these announcements by maintaining lists of all such prefixes. However, this is a tedious task; the list should be updated continuously not to block newly allocated prefixes. A study by Feamster et al. [FJB05] observed several *bogon* routes every few days at eight vantage points. *bogon* routes can be used for malicious activities and are difficult to trace.

Several comprehensive architectures to secure BGP have been proposed;

notable examples are S-BGP [KLMS00], soBGP [Ng04], and IRV [GAG+03]. These approaches defines a Public Key Infrastructure (PKI) for authenticating and authorizing BGP updates and routes. However, architectural complexities and required BGP modifications hinder their deployment. Other approaches, on the other hand, focus on building alert systems that detect and report anomalies, next we highlight a few of these methods. PHAS [LMP+06] uses BGP updates from RouteViews and RIPE to detect hijacked routes and notifies their owners. Hu and Mao provided a solution for detecting hijacks in real time by combining passive measurements as in PHAS with active probing [HM07]. Pretty Good BGP (PGBGP) [KFR06], another detection system, is based on keeping historical data that tracks origins of prefixes. Furthermore, this state is used to judge the legitimacy of new routes. Suspicious routes are kept unused for 24 hours given the availability of other alternatives.

The IETF is currently working on standardizing an infrastructure for origin authentication, based on a Resource Public Key Infrastructure (RPKI). This infrastructure will provide public keys to ASes and routers; and Route Origin Authorizations (ROAs) for mapping network prefixes to origin ASes [sid].

3.6 Summary

In this chapter, we have discussed the open issues and limitations in today's global routing architecture. We illustrate that BGP suffers from scalability problems, prolonged convergence, misconfiguration, and security vulnerabilities. Among these issues the scalability stands out as an important limitation. Scalability is an issue in two different aspects: increasing routing table size, and increasing rate of BGP updates. While the first aspect is well explored with several suggested solutions, the second aspect is far from being well understood. In fact, churn is a more serious concern since it can trigger wide-scale instabilities. We devote the rest of this thesis to investigating factors that govern churn and characterizing the extent of the problem. The next chapter explores possible approaches and methods for studying BGP churn.

Chapter 4

Studying BGP Churn

The observed BGP churn is a product of a complex interaction between topology; routing events; and protocol implementations and configurations. Understanding this interplay is crucial for maintaining and evolving the global routing system. Unfortunately, the complexity of the topology and the large number of involved factors make BGP churn far from being well-understood. BGP churn can be studied through measurements, modeling, and simulations. The decision of adopting one of them largely depends on the problem under investigation. In this chapter, we elaborate on the factors that constitute churn, and explore different methods for studying these factors.

4.1 Dissecting Churn

Several different factors can influence BGP churn. First, it is plausible to assume that inter-domain routing activity is dependent on the network size, i.e., number of ASes and prefixes. It is expected that the rate of BGP updates a router receives will increase with the number of routable destination prefixes. Roughly speaking, each prefix corresponds to a destination network. If these destination networks fail and recover independently and with the same probability, we would expect a linear relation between the size of the routing table and churn. Of course this is a very simplistic model, but still we can expect that there is a positive correlation between the number of routable prefixes and churn.

Other prominent factors, beside the network size, can be grouped along three different axis as illustrated in Fig. 4.1, more specifically:

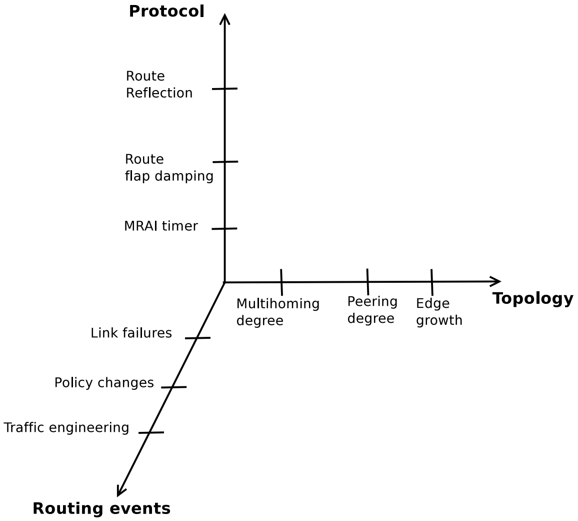


Figure 4.1: Churn factors

1. The characteristics of the Internet *topology*
2. The *routing protocol*, including policy annotations and various BGP mechanisms and implementation choices
3. *Routing events* like prefix announcements, link failures, session resets, traffic engineering

There is a need to understand the impact of each contributor independently and then in association with other related factors. As a first step to dissect the components of BGP dynamics, we present a general discussion around possible effects of different factors. Furthermore, the following chapters will examine critically the most influential contributors and characterize their impact on churn.

Event types. The observed churn will depend on the routing activity of individual prefixes at their origin AS. Over the past few years, it has become increasingly common for stub ASes to be multihomed to several providers [DD08]. Multihoming enables load-balancing by selectively announcing different prefixes to different providers. As this practice gradually becomes more common, we can expect that it contributes to increasing churn when a network destination becomes unreachable. Another source of churn is routing events taking place in or between transit ASes. Such events include link failures (physical failures, router reboots, etc), policy changes that result in new preferred routes, or changes in the IGP or iBGP configuration of a transit AS. Importantly, these operations often affect a large number of prefixes at the same time. The amount of churn observed at a router after such events will also depend on the the topology and policies.

The structure of the network topology. Topological properties of the AS-level Internet graph will also affect the churn rate. Increased multihoming in the Internet increases the churn generated when a destination prefix is announced or withdrawn from the source AS. On the other hand, increased connectivity can reduce the impact of failures, if a local alternative is available.

Policies and protocol configuration. There are BGP mechanisms and parameter settings that can reduce the observed churn. Two important mechanisms are the MRAI timer and RFD. In addition, the use of ingress/egress filtering and route reflectors in iBGP can limit or increase churn [SFPB09]. The interactions between different protocol implementations and configurations, or their impact on BGP churn, is far from well understood.

In a nutshell, the observed BGP churn is not totally decided by one of the above mentioned factors but rather by their interactions. Following any routing event, a BGP router enters a path finding process where it tries available routes and eventually converges to a stable route or completely withdraws an already installed route. The length of this convergence process and the number of explored routes determine the resulting churn. Strictly speaking, the convergence process will depend on the nature and location of the underlying routing events; routing changes that result in a complete withdrawal of a network prefix involve more path exploration than those ends with an alternative path [OZP+06].

The aforementioned discussion highlights to some extent the complexity of BGP churn. In this thesis, we look at routing dynamics from several angles and we use in that two approaches. In the first approach, we isolate a certain factor (e.g. multihoming) and construct several what-if scenarios (e.g. what will be the impact of stub failures if multihoming is growing faster at the core of the Internet?). By answering such questions, we will be able to characterize the impact of different factors. Then, we can relate our findings to what we know so far about the evolution of the global routing system.

The second approach involves measuring different properties of the routing system and their evolution.

4.2 Methodologies

BGP churn can be studied through measurements, modeling, and simulations. The decision of adopting one of them largely depends on the problem under investigation. Simulations are more viable for investigating questions that intend to explore different what-if scenarios related to the impact of topology growth on routing, new routing enhancements, and radical architectural changes. Mathematical modeling can also be used for studying and characterizing BGP. However, the complexity of BGP and large Internet-like topologies make it difficult to create a tractable and useful mathematical model. Measurement, on the other hand, could be employed in assessing the impact of incremental protocol changes ; and the current health and status of the routing system

4.2.1 Measuring BGP Churn

Inter-domain routing dynamics and performance can be studied by conducting active and passive measurements. In the following, we discuss both approaches and enumerate a list of cases where each can be applied.

Active Measurements can be done through injecting changes (e.g. announcing and withdrawing network prefixes) at a certain location in the network and observing their impact at a set of vantage points. Several studies used this technique, e.g., [LABJ00, LAWV01, WMW$^+$06], for characterizing churn; measuring convergence times; and quantifying the interaction between routing and data forwarding. Setting up an active measurement infrastructure can be difficult since it involves a considerable coordination with operational networks; there is a need to have globally visible network prefixes and operators that agree to host this activity. Currently, there are two public active measurement infrastructures. The BGP beacons project [MBGR03], and RIPE RIS beacons [rip]. A BGP beacon is a BGP speaker that announces and withdraws a network prefix following a predetermined periodic pattern. The activity of these prefixes represents a set of well defined routing events. Unlike other routing changes in the Internet we know exactly the location, time, and cause of the observed routing activity. Therefore, BGP beacons provide useful insights about convergence time, propagation of update messages. Further, they can be combined with active probing to study the impact of routing changes on data forwarding. However, conclusions reached using beacon prefixes should be approached carefully since they only represent a handful of networks. For example, if we want to use beacons to characterize the duration of a routing event, we need to observe them at a large set of diverse vantage points to assure the impartiality and significance of gained insights. Active measurements approaches have some limitations. The number of beacons and their locations influence to large extent observed results. Another issue is the properties of vantage points used to monitor a beacon prefix; for example, their topological connectivity, and configurations. Finally, the fact that the Internet is a mission critical infrastructure limits the flexibility and extent of active measurements.

Passive Measurements can be performed by logging BGP routing tables and updates through peering with operational routers, RouteViews [rou] and RIPE RIS [RRIS] projects are pioneering efforts in this direction. Data collectors usually use multi-hop eBGP session to peer with the monitored routers. In other words, a collector and the respective monitor are not directly connected. Many studies, e.g. [LGW+07, VIRZZ05, WZP+02, FMM+04], employed passive measurements for investigating different questions, examples include characterizing activity of prefixes, quantifying path exploration, inferring root cause of routing changes, and understanding BGP behavior during attacks and major failures. Passive measurements best fit research questions that are related to the current status of the global routing system and its evolution over the past few years. In addition, it can be used for quantifying the impact of different implemented BGP knobs and extensions.

An important challenge for passive measurements is monitor session failures. If the multi-hop BGP session between a monitor and the collector is broken and re-established, the monitor will re-announce all its known paths, giving large bursts of updates. This is a local artifact of the measurement infrastructure, and it does not represent genuine routing dynamics. Several measurement studies developed heuristics for identifying table transfers and filtering related updates. For example, Rexford et al. [RWXZ02] discarded all duplicate announcements, however, that also would remove genuine duplicates. Another work [AFBB02] grouped updates into time intervals of 30 seconds and discarded all bins that contained over 90% of the number of prefixes announced by the respective monitored-AS. Zhang et al. [ZKL+05] proposed an algorithm called MCT to accurately identify updates that are part of table transfers. MCT is currently considered as the best available tool for identifying table transfers. A recent study [CZZZ10] employed MCT to assess the extent of session failures of RouteViews and RIPE RIS monitoring sessions. They showed that session resets are rather frequent with a few of them every month.

In addition, the generality and representativeness of passive measurements studies are highly dependent on monitor selection. A Monitor's location and connectivity decide its feasibility for observing diverse routing events and conclusive static topology properties [ZZM+07]. Finally, both active and passive measurements cannot be used to ask what-if questions about routing dynamics, protocol modifications, and topology properties.

The above mentioned downsides of measurement approaches do not mean that they are not useful. Nevertheless, it necessitates that experiments should be carefully planned by taking into consideration these inherent limitations. In this thesis, we employ passive measurements to characterize churn evolution and to quantify the impact of different implementations of BGP rate-limiting timers

4.2.2 Modeling BGP Churn

Although simulation results can help us to investigate and understand certain aspects related to BGP churn, it is hard to enumerate all possible scenarios in simulation. On the contrary, when a closed-form solution is available through formal analysis, we can easily identify the role of various influential factors in routing dynamics. Unfortunately as pointed above, the complexity of the global routing system makes it hard to create a tractable mathematical model. Such modeling has been attempted before [ZZM+05], but only for regular topologies and using a simplified BGP operation model.

4.2.3 Simulating BGP Churn

Simulating inter-domain routing is a viable option to circumvent the limitations of measurements and mathematical modeling. However, there are many factors that control and influence the realism and generality of simulation results.

First, a representative topology model that captures the properties of the

Internet AS-level graphs is needed. The model should be able to capture reasonably well the observed properties of the AS-level graph (e.g. power law degree distribution, strong clustering, constant average path length). In addition, it should be able to annotate inter-AS links with business relationships.

Second, a reasonable implementation of BGP and a clear description of routing changes and events are crucial for the correctness of the results. The choices one makes regarding these factors will influence the suitability and correctness of the simulation model. Essentially, there is a trade-off between the computational complexity of the simulation model and the level of details that is captured. The computational complexity is a function in the size of the simulated network, the level of topological details, and protocol implementation. Ideally the simulation model should be able to simulate topologies that are comparable to the current internet in terms of size in a scalable way.

Topology Generators. Generating graphs that capture the observed properties of the AS-level topology has been a subject of much research in the past decade. In general, the proposed topology generators focused on capturing the abstract properties of the AS-level graph.

Early topology generators such as BRITE [MLMB01], Inet [WJ02], and PLRG [ACL00] tried to reproduce the node degree distribution of the AS-level graph. These generators managed to reproduce the node degree distribution reasonably; however, they did not succeed in capturing other abstract properties such as the clustering coefficient and the joint node degree distribution. These limitations are expected since this class of generators assumes that the node degree distribution is a one dimensional independent variable. Thus, the degree of a node in such topologies is unrelated to the properties of its neighboring nodes. Later, Mahdevan et al. [MKFV06] proposed another approach to overcome these limitations by using a group of distributions that capture the correlations of degrees among a set of connected nodes (i.e. a subgraph of the AS-level topology). They further employed this approach to generate re-scaled topologies of different sizes [MHK$^+$07].

The above mentioned topology generators fulfilled reasonably their design purposes. But, they share a common limitation, and that they considered the AS-level topology as a generic collection of links and nodes. In fact, nodes and links in the Internet are far from being generic. ASes geographical presence, sizes, and connectivity differ depending on their roles and business models (e.g. transit providers, content providers). More importantly, inter-AS links are different based on business relationships between the involved

ASes. These relationships control and regulate the inter-domain routing, and therefore, generating topologies that are annotated with them is crucial.

Several other efforts focused on generating topologies that are annotated with business relationships. The GHITLE topology generator [dL] used a set of simple design heuristics to produce such topologies. However, it did not account for the subtle differences between different node types and did not model the number of settlement-free peering (p2p) links in a realistic way. Furthermore, the impact of geographic presence on both peer and providers selection was not captured. The work by Dimitropoulos et al. [DKVR08] proposed generating re-scaled annotated topologies by generalizing the work in [MHK$^+$07]. He et al. proposed HBR [HFKC08] as a method for generating annotated graphs of various sizes through sampling them from larger inferred AS-level topologies. The last two approaches generate topologies in a top-down fashion by starting from an inferred AS-level topology and work to reproduce the measured abstract graph properties. Therefore, they do not provide enough flexibility for controlling different topological characteristics.

Most of the existing topology generators focused on generating AS-level topologies rather than router-level topologies. An important reason behind this is the fact that there is a better understanding for the Internet topology at the AS level. In fact, routers connectivity varies across networks and is dependent on choices taken locally at each network. In addition, it is constrained by many factors such as maximum possible degree per router (i.e. maximum number of interfaces), number of the points of presence a network has, and different engineering decisions taken in order to optimize the topology. An extensive discussion about this can be found in [LAWD04]. Several measurement studies aimed at inferring router-level topologies in different networks using traceroutes. A notable example of such approaches is Rocketfuel [SMWA04]. However, inference techniques suffer from limitations that are caused by their sampling nature [LBCX02] and traceroutes failure in resolving router aliases [TMSV03].

Aiming at addressing the lack for router-level topology generators, Quoitin et al. [QdSFB09] proposed IGen as a generator that builds topologies through considering network design heuristics. IGen is a promising step in the direction of building realistic router-level topologies, however, since it depends on many heuristics, it is not clear how well the generated topologies are able to match known properties of the AS-level graph.

Simulators. Existing inter-domain routing simulators fall into two broad categories. Either they only calculate steady state routes, and do not capture routing dynamics [QU05], or they include a detailed model of each BGP session(e.g. underlying TCP connections). Simulators that belong to the second category [ssf, DR06] are suitable if one wants to study questions that involve both routing and data forwarding. However, this large level of details limits their scalability; they do not scale to network sizes in the order of today's AS-level Internet topology.

4.3 Summary

In this chapter, we have presented a high level overview of different factors that constitute BGP churn. We have explored possible methods for investigating the impact of these factors. The work in this thesis employs passive measurements and simulations for studying BGP scalability with respect to churn. Our discussion has demonstrated that current AS-level topology generators and inter-domain routing simulators suffer from an array of limitations. In the next chapter, we propose SIMROT as a new toolbox for accurately simulating BGP dynamics in a scalable way.

Chapter 5

SIMROT: A Scalable Interdomain Routing Toolbox

In this chapter, we propose SIMROT as a toolbox for simulating BGP. Consisting of a topology generator and a scalable event-driven simulator. The knobs of our topology generator are parameters with operational relevance in practice, such as the multihoming degree (MHD) of stubs versus transit providers, instead of abstract measures such as degree distributions or assortativity. Therefore, it allows the user to explore a wide range of "what-if" possibilities that none of the existing models captures in a parsimonious and intuitive manner. Our simulator, on the other hand, make several simplifying assumptions by focusing only on capturing the operations of the control plane and leaving out some of the details (e.g. the operation of the underlying TCP sessions). It is capable of capturing the exchange of routing updates, and scales to network sizes of thousands of nodes. In the subsequent sections we describe and evaluate SIMROT.

5.1 Topology Model

In this section, we first describe some key properties that characterize the AS-level Internet topology. We believe that these properties will remain valid in the foreseeable future. We then describe a model that allows us to construct topologies with different configurable properties while still capturing these key properties.

The current version of our topology generation model is limited to AS-level graphs. Nevertheless, it can be extended to produce router-level graphs if there exists reasonable models. A possible scenario would be combining approaches such as IGen [QdSFB09] and our model for achieving this goal.

5.1.1 Stable Topological Properties

The AS-level Internet topology is far from a random graph. Over the past decade it has experienced tremendous growth, but the following key characteristics have remained constant:

1. *Hierarchical structure.* On a large scale, the nodes in the Internet graph form a hierarchical structure. By hierarchical we mean that customer-provider relationships are formed so that there are normally no provider loops, where A is the provider of B who is the provider of C who again is the provider of A.

2. *Power-law degree distribution.* The degree distribution in the Internet topology has been shown to follow a truncated power-law, with few very well-connected nodes, while the majority of nodes have only few connections [FFF99]. The well connected nodes typically reside at the top of the hierarchy.

3. *Strong clustering.* The nodes in the Internet are grouped together in clusters, with nodes in the same cluster more likely to be connected to each other. One reason for this clustering is that networks operate in different geographical areas.

4. *Constant average path length.* Measurements show that in spite of a tremendous growth in the number of nodes, the AS-level path length has stayed virtually constant at about 4 hops for the last 10 years [DD08].

5.1.2 SIMROT-top

Next, we describe a flexible model for generating topologies that captures the above properties about the AS-level graph. Several design choices and parameters in our topology generator were guided by a recent measurement study [DD08].

We use four types of nodes in our model. At the top of the hierarchy are the tier-1 (T) nodes. T nodes do not have providers, and all T nodes are

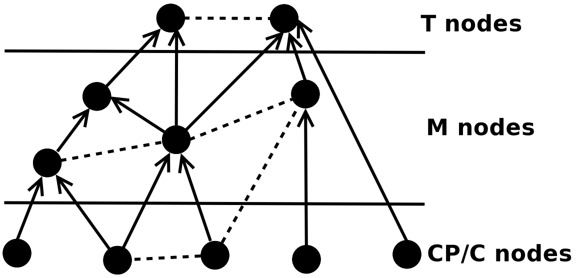


Figure 5.1: Illustration of network based on our topology model.

connected in a clique using peering links. Below the T nodes, we have the mid-level (M) nodes. All M nodes have one or more providers, which can be either T nodes or other M nodes. In addition, M nodes can have peering links with other M nodes. At the bottom of the hierarchy, we have two different types of stub nodes. We distinguish between customer networks (C) and content providers (CP). In this context, CP nodes would include content provider networks, but also networks providing Internet access or hosting services to non-BGP speaking customers. In our model, the difference between C and CP nodes is that only CP nodes can enter peering agreements with M nodes or CP nodes, while C nodes do not have peering links. Figure 5.1 shows a generic network of the type described above. Transit links are represented as solid lines with arrowheads pointing towards providers, while peer-to-peer links are dotted.

To capture clustering in our model, we introduce the notion of *regions*. The purpose of regions is to model geographical constraints; networks that are only present in one region are not allowed to connect with networks that are not present in the same region. In our model T nodes are present in all regions. 20% of M nodes and 5% of CP nodes are present in two regions, the rest are present in only one region. C nodes are only present in one region.

We generate topologies top-down in two steps. First we add nodes and transit links, then we add peering links. The input parameters n_T, n_M, n_{CP} and n_C decide how many of the n nodes belong to each node type, respectively. First, we create the clique of T nodes. Next, we add M nodes one at a time. Each M node connects to an average of d_M providers, uniformly distributed between one and twice the specified average. M nodes can have providers among both T and M nodes, and we use a parameter t_M to decide the fraction of providers that are T node. M nodes can only select providers that are present in the same region. M nodes select their providers using preferential attachment, which gives a power-law degree distribution [BA99].

We then add the CP and C nodes, which have an average number of providers d_{CP} or d_C, respectively. CP and C nodes can select T nodes as providers with a probability t_{CP} and t_C, respectively. Just like the M nodes, C and CP nodes select their providers using preferential attachment.

When all nodes have been added to the topology, we add peering links. We start by adding p_M peering links to each M node. As for the provider links, p_M is uniformly distributed between zero and twice the specified average. M nodes select their peers using preferential attachment, considering only the peering degree of each potential peer. Each CP node adds p_{CP-M} peering links terminating at M nodes, and p_{CP-CP} peering links terminating at other CP nodes. CP nodes select their peers among nodes in the same region with uniform probability. Importantly, we enforce the invariant that a node not peer with another node in its customer tree. Such peering would prey on the revenue the node gets from its customer traffic, and hence such peering agreements are not likely in practice.

5.1.3 Internet-Like Topologies

Next, we illustrate how to configure our topology generator for producing graphs that resemble the growth of the Internet over the last decade. The sample configuration parameters are inspired by recent measurements of the evolution of the Internet topology over the last decade [DD08]. The growth is characterized by a slow increase in the MHD of stub nodes, and a faster growth in the MHD and the number of peering links at middle nodes. In this sample configuration we use 5 regions, containing one fifth of all nodes each. Table 5.1 gives the parameter values for sample configuration. Note that n in Tab. 5.1 is the total number of nodes in the graph.

We validate that the generated topologies capture the four stable properties of the Internet topology discussed in Sec. 5.1.1, and compare some properties of the generated graphs to inferred Internet topologies. We generate topologies of sizes 5000 and 10000 nodes respectively, and compare against two inferred AS-level topologies of sizes 3247 and 17446 nodes. The smaller topology is provided by Dhamdhere and Dovrolis [DD08] and it is based on RouteViews [rou] and RIPE [RRIS] BGP routing tables from Jan-

	Meaning	**Value**
n_T	Number of T nodes	$4-6$
n_M	Number of M nodes	$0.15n$
n_{CP}	Number of CP nodes	$0.05n$
n_C	Number of C nodes	$0.80n$
d_M	Avg M node MHD	$2+2.5n/10000$
d_{CP}	Avg CP node MHD	$2+1.5n/10000$
d_C	Avg C node MHD	$1+5n/100000$
p_M	Avg M-M peering degree	$1+2n/10000$
p_{CP-M}	Avg CP-M peering degree	$0.2+2n/10000$
p_{CP-CP}	Avg CP-CP peering degree	$0.05+5n/100000$
t_M	Prob. that M's provider is T	0.375
t_{CP}	Prob. that CP's provider is T	0.375
t_C	Prob. that C's provider is T	0.125

Table 5.1: Topology parameters

uary to March 1998. The second inferred topology is provided by Mahadevan et al. [MKF$^+$06] and based on RouteViews BGP routing tables from March 2004. Note that the inferred topologies miss a large fraction of peering links, which distorts their characteristics quantitatively [DD08]. Therefore, our aim is that our topologies match the major topological properties of the Internet qualitatively rather than quantitatively.

Hierarchical structure. This is trivially fulfilled through the way we construct the topologies.

Power-law degree distribution. Figure. 5.2a shows the CCDF of the node degree on a log-log scale. We observe that our model captures the power-law scaling of the node degrees reasonably well, and is comparable to that of the inferred Internet topologies. The use of preferential attachment when selecting which nodes to connect to gives the observed power-law degree distribution [AB02].

Strong clustering. We measure the local clustering (or clustering coefficient) of each node in a topology. The local clustering of a node is defined as the ratio of the number of links between that node's neighbors to the maximum possible number of such links (i.e. a full clique). Hence, the local clustering measures how well connected a node's neighborhood is. Figure. 5.2b reports the average local clustering, across all nodes of the same degree, as a function of node degree. To keep the figure readable, we plot results for only two topologies (the other pair of topologies show similar results). Our model matches qualitatively the trends seen in the inferred topologies: first, local clustering decreases with the node's degree, and second, the clustering versus degree relation follows a power-law. It should be noted however that our model produces lower clustering than the inferred Internet topologies.

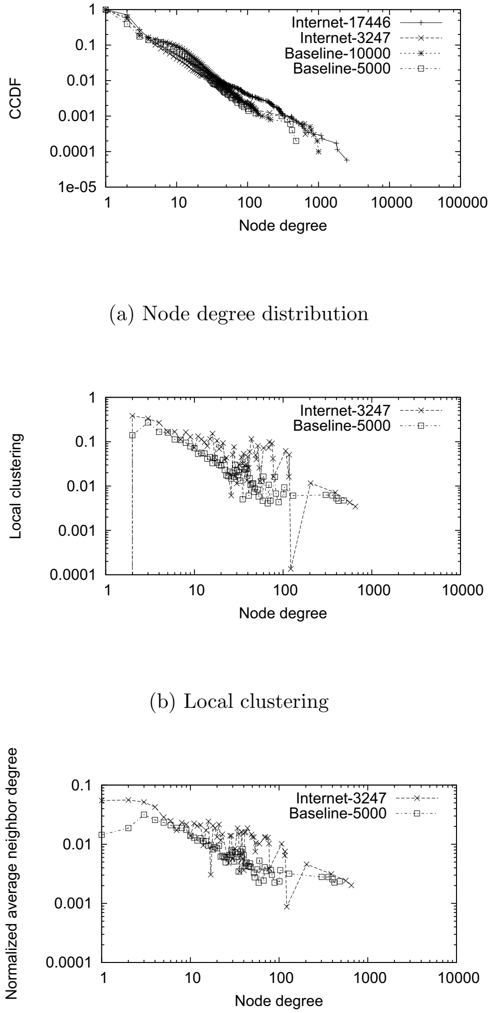
Constant average path length. The average path length in topologies produced by our model is constant at around four hops as the network grows from 1000 nodes to 10000 nodes. This matches closely the average path length in the inferred Internet topology at least since 1998 [DD08].

In addition to confirming that our generated topologies capture the previous four stable properties, we also investigate the *average neighbor connectivity* [MKF$^+$06], which has been difficult to capture by existing topology generators [HFJ$^+$09]. The average neighbor connectivity of a node is simply the average degree of its neighbors. This metric relates to the assortativity of a graph. It measures whether a node of a certain degree prefers to connect with higher or lower degree nodes. Figure. 5.2c shows the average neighbor connectivity as a function of the node degree. We normalize the average neighbor connectivity by the maximum possible value which is (the total number of nodes in the graph - 1), in order to compare topologies of different sizes. Our model gives an average neighbor connectivity that matches well the inferred Internet topologies, with smaller degree nodes having a higher average local connectivity than the higher degree nodes (referred to as negative assortativity).

The aforementioned validations illustrate that our topology generation model can reasonably produce graphs that match several important properties of the measured AS-level topology. In the next section we present our BGP simulation model.

5.2 SIMROT-sim

Simulations of any system of the size and complexity of inter-domain routing require to make several simplifying assumptions based on the goals of the



(a) Node degree distribution

(b) Local clustering

(c) Normalized average neighbor connectivity

Figure 5.2: Topologies validation

simulations. In this section we present our simulator ”SIMROT-sim” and describe the choices and assumptions we make in its development.

SIMROT-sim is a discrete event simulator that is capable of capturing the exchange of routing updates and hence, simulate BGP dynamics. Furthermore, it is able to scale to network sizes of several thousands of ASes.

In order to realize the aforementioned scalability we make two key simplifying assumptions. Firstly, we model a BGP session between two nodes as a logical variable that is either established or not, and thus ignore the underlying TCP nature of the session. This choice enhances the scalability of our simulator since we abstract the TCP details and all involved overhead and signaling (e.g. sessions KEEPALIVE messages). We argue that this simplification does not have an impact since we only simulate the operation of BGP. The details of BGP sessions are important only when studying the interaction between data plane and control plane (e.g. the impact of data traffic on the stability of BGP sessions).

The second assumption is that we model each AS as a single node, and connections between two neighboring ASes as a single logical link. This implies that we do not capture routing effects within an AS, introduced by iBGP or interactions with IGP routing protocols (e.g., hot-potato routing). However, while such effects have an impact on the absolute amount of churn, they do not influence the general trends of BGP churn.

SIMROT-sim simulates policy-based routing, with the use of MRAI timers to limit the frequency with which a node sends updates to a neighbor. By “policies”, we refer to a configuration where relationships between neighboring ASes are either peer-to-peer or customer-provider. We use normal “no-valley” and “prefer-customer” policies. Routes learned from customers are announced to all neighbors, while routes learned from peers or providers are only announced to customers. A node prefers a route learned from a customer over a route learned from a peer, over a route learned from a provider. Ties among routes with the same local preference are broken by selecting the route with the shortest AS path, then based on a hashed value of the node IDs.

By “MRAI” or “rate-limiting”, we refer to a configuration where two route announcements from an AS to the same neighbor must be separated in time by at least one MRAI timer interval. We use a default MRAI timer value of 30 seconds. To avoid synchronization, we jitter the timer as specified in the BGPv4 standard. According to the BGP-4 standard [RLH06], the MRAI timer should be implemented on a per-prefix basis. However, for efficiency

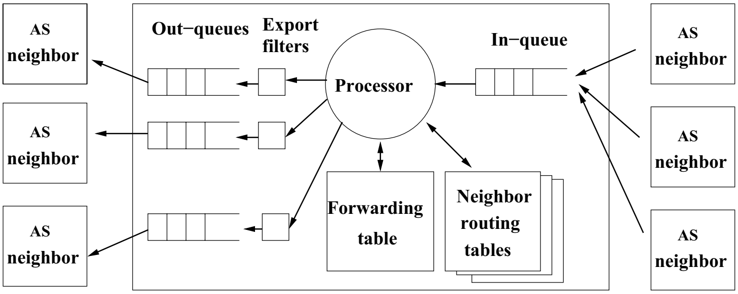


Figure 5.3: Model for a node representing an AS.

reasons, router vendors typically implement it on a per-interface basis. We adopt this approach in our model. We follow the MRAI implementation recommended in the most recent RFC (RFC4271), which specifies that both announcements and explicit withdrawals should be rate-limited. Note that the value of the MRAI is configurable in SIMROT-sim.

Figure 5.3 shows the structure of a node in our simulater. A node exchanges routing messages with its neighbors. Incoming messages are placed in a FIFO queue and processed sequentially by a single processor. The time it takes to process an update message is uniformly distributed in a user defined range. Each node maintains a table with the routes learned from each neighbor, we call these tables Adjacency-RIB Ins (Adj-RIB-Ins). Upon receiving an update from a neighbor, a node will update this table, and re-run its decision process to select a new best route. The new preferred route is then installed in the forwarding table and announced to its neighbors, the forwarding table is called the Local-RIB (Loc-RIB). For each neighbor, we maintain an export filter that blocks the propagation of some updates according to the policies installed in the network. Outgoing messages are stored in an output queue until the MRAI timer for that queue expires. If a queued update becomes invalid by a new update, the former is removed from the output queue. We further introduce a set of interrupt messages to signal events such as failures and restorations of various components (e.g. links, nodes, sessions) to the affected nodes, which consequently trigger BGP updates as a response to the signaled change.

There are two factors that determine the scalability of SIMROT-sim.

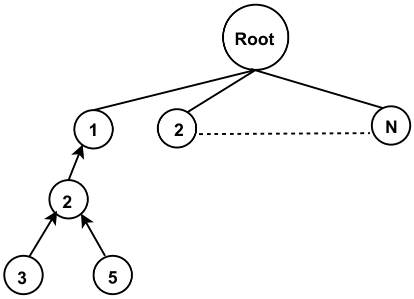


Figure 5.4: Global RIB data structure

The first one is the state that each node maintains (i.e. routing table), which decides memory requirements. We design a scalable data structure for storing routing tables in SIMROT-sim that minimizes memory requirements by removing the redundancy in the RIBs entries shared by many nodes. For example if a node X and a node Y share the path $\{A, D, F\}$ to a certain destination prefix p, It will be more efficient to store a single entry for this path in the memory and keep two pointers at X and Y to it.

This approach is implemented by maintaining a global tree data structure that represents a global routing information base shared by all nodes in the network (Global-RIB). The tree has a single root which is used to maintain the structure of the tree. The root has n children (i.e. n is the number of the ASes in the network) each one represents a node in the network and is labeled with the corresponding AS number. When an AS X announces a prefix, it sends the AS-PATH information as a pointer to the tree node that labeled as X at level-1 of the Global-RIB. A neighbor of X performs two tasks when receiving this pointer. First it determines the actual AS-PATH by backtracking from the tree node that the pointer refers to in an upward direction until it reaches the root of the Global-RIB. Second it creates a new node that is labeled with its AS number, and will be added as a child to the tree node that the pointer refers to. A pointer to the newly added node will then be sent further as the corresponding AS-PATH information.

For example assume that AS 1 is announcing a destination prefix p and it has AS2 as an immediate neighbor. When AS 1 announces p to AS 2 it sends the AS_PATH information as a reference to node 1 in the tree. AS 2 keeps this reference in its routing table and adds a new child for node 1 in the tree labeled with its AS number (i.e. 2). When AS 2 announces p to its neighbors 3 and 5, It just sends the reachability information as a reference to the newly added tree node (i.e. tree node 2). Furthermore, 3 and 5 will back track from the tree node 2 upwards until they reach the root in order to extract the corresponding AS_PATH(i.e. {2,1}). The corresponding GRIB data-structure is illustrated Figure 5.4.

The second factor that can potentially limit the scalability of our simulator is the number of enqueued BGP updates for processing. The impact of this factor is more evident during the initial convergence phase of the simulation. In this phase all prefixes that are part of the simulation are announced by their owners, which results in exchanging a large number of updates, and performing many decision process operations.

Instead of simulating the exchange of BGP updates during the initial convergence phase, we implement a routing solver that computes for each node its steady-state reachability information and installs the computed entires in the respective routing tables. Our routing solver working principle is similar to that of C-BGP [QU05]. This optimization allows us to reduce the required resources for the initial convergence phase. After performing the initial convergence one can choose to proceed with simulating various routing events (e.g. a link failure).

In the rest of this section we validate and evaluate the performance of SIMROT-sim and compare it to that of SSFNET. SSFNET is a large scale discrete-event simulator that is often regarded as the state-of-the-art tool for simulating BGP.

5.2.1 Performance Evaluation

For validating the operation of SIMROT-sim and comparing its performance with SSFNET, we generate six topologies in the range between 1000 and 6000 nodes using the example configuration described in Sec. 5.1.3. Then in each topology we simulate the withdrawal of a prefix from a C-type node. The experiment is repeated for 100 different C nodes, and the number of received updates is measured at every node in the network. We record the execution time and memory requirements of each simulation run. We have performed these experiments on a Dell machine (quad core Intel Xeon CPU 3.00 GHZ, 4GB RAM).

Simulation results. The goal of this comparison is to determine whether

SIMROT-sim is able to simulate BGP dynamics reasonably. Our main metric in the average number of updates received at a T node after withdrawing a prefix from a C-type node. Figure. 5.5 shows the results of SIMROT-sim and SSFNET. The vertical bars the width of the confidence interval for SIMROT-sim results at a 99% confidence level. We observe that the results of the two simulators match well. The slight differences can be explained by the fact that each simulator uses a different random number generator. The deviations, however, are still within the calculated confidence intervals. Further the results of both simulators do not show a monotonic increase in the average number of updates due to differences between topologies; the topologies are generated as snapshots of certain sizes rather than in an evolving manner.

Execution time. We record the time each simulator takes per each run. We then average over the 100 runs. Figure. 5.6a shows the average execution time. The measurement reflects clearly that SIMROT-sim execution time is significantly lower than that of SSFNET. The difference can reach up to two orders of magnitude. For example SSFNET takes about 1000 seconds to simulate the above described event in a topology of 6000 nodes; however, SIMROT-sim takes around 35 seconds for that. The large difference can be attributed to the simplifying choices we describe above.

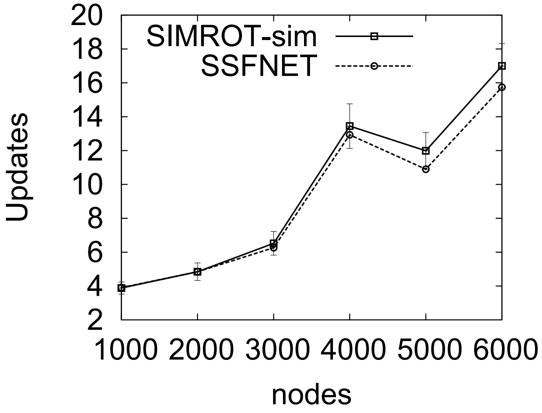
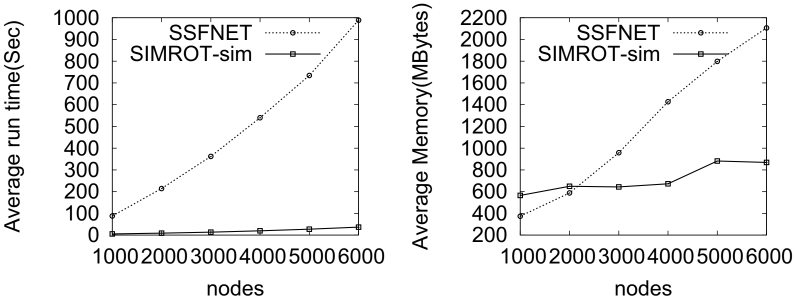


Figure 5.5: Simulation results

Memory requirements. We also measure the memory requirements of



(a) Average time per simulation run (b) Memory requirements

Figure 5.6: SIMROT-sim performance evaluation

each simulator per each run. We then average over the 100 runs. The average memory requirements is illustrated in Fig. 5.6b. The memory requirements in SIMROT-sim is characterized by a slow increase (600 to 800 MBytes). On the contrary, the memory requirements of SSFNET has increased significantly between 400 MBytes and 2.1 GBytes (i.e. an increase of 400%). The slow increase in SIMROT-sim can be attributed to the simple BGP model that it uses, and the Global-RIB data structure explained above. This data structure minimizes memory requirements by removing the redundancy in the RIBs entries shared by many nodes.

The above presented validation and performance evaluation show that our simulation model can accurately simulate BGP dynamics in a scalable way.

5.3 Summary

This chapter proposes SIMROT as a comprehensive framework for simulating BGP. SIMROT-top, the first component of SIMROT, is a flexible topology generator that produces AS-level graphs which are annotated with business relationships. The second component of the framework is SIMROT-sim, a light-weight BGP simulator that is capable of capturing routing dynamics and scaling to network sizes of thousands of nodes. We have validated the topology generator and illustrate its ability in reproducing various known properties of AS-level topology. Besides, we have compared the performance and correctness of SIMROT-sim when simulating BGP dynamics, with that of the widely used SSFNET simulator. This benchmarking confirms that our simulator significantly outperforms the SSFNET simulator in terms of processing time and memory requirements, while producing similar results.

Part II

BGP Scalability: the Roles of Topology Growth and Update Rate-Limiting

Chapter 6

The Role of Topology Growth

The previous part of the thesis discussed several limitations in today's global routing system and identified BGP scalability as one of the most important challenge we are currently facing. In particular, the rate of routing updates (churn) that BGP routers must process is a major concern. As explained in Chapter 4, BGP churn is a product of a complex interplay between topology properties; routing events; and BGP implementations and configurations. Our objective in this chapter is to investigate how topological characteristics of the AS-level graph influence the scalability of BGP churn as the network grows. We look at several "what-if" growth scenarios, and investigate their scalability implications and interaction with different failure types; for example "What if the MHD of stub ASes increases with the network size instead of staying constant?" and "What if the Internet becomes denser mostly due to peering links?". These scenarios are either plausible directions in the evolution of the Internet or educational corner case. Our findings explain the dramatically different impact of multihoming and peering on BGP scalability, highlight negative and positive effects of multihoming on churn and reachability, and identify which topological growth scenarios will lead to faster churn increase for different failure types.

6.1 Approach

We can only study the problems described above using simulations. Since our goal is to look at scalability under different hypothetical topology growth models, our investigation cannot be performed by doing measurements in the current Internet. Also, the complexity of BGP and large Internet-like topologies make it difficult to create a tractable and useful mathematical model. To this end, we use our BGP simulation toolbox (SIMROT) for generating required topologies and performing simulations.

Essentially our simulations aim to understand how topological characteristics of the AS-level graph influence the scalability of BGP churn that is caused by well-defined routing events. Using our topology generator, we establish the factors that determine churn at different locations in the Internet hierarchy, and investigate the importance of each factor in a growth model that resembles the evolution of the Internet over the last decade. We then examine several deviations from this growth model, and investigate how the number of routing updates generated by different routing events grows with the size of the topology in each case. In the following, we describe the parameterization of our simulations.

Topologies. We start by defining a Baseline topology growth model that will later be used as a reference scenario for looking at how different topological factors influence BGP churn. The Baseline growth model resembles the evolution of the Internet over the last decade. Note that our aim is to look at the scalability of different hypothetical growth models, and it is not our goal that the Baseline model should be an exact copy of the historical Internet. Still, the parameters used are inspired by recent measurements of the evolution of the Internet topology over the last decade [DD08]. The baseline growth model is characterized by a slow increase in the MHD of stub nodes, and a faster growth in the MHD and the number of peering links at middle nodes. Table 5.1 gives the parameter values for the Baseline growth model. We further generate topologies that are single-dimensional deviations from the Baseline model presented above. These topologies represent different "what-if" scenarios about Internet growth, such as: “What if the MHD of stub ASes increases with the network size instead of staying constant?” “What if the Internet becomes denser mostly due to peering links?” “What if tier-1 providers dominate the transit market, reducing the number of tier-2 providers?”

Routing events. We look at three different events that generate BGP updates. First, we focus on events where individual destination prefixes are withdrawn and then re-announced by stub ASes. This is the most basic routing event that can take place in the Internet, and at the same time the most radical. In the absence of aggregation these changes must be communicated all over the network. Second, we study events where a single link connecting a stub AS to one of its providers fails and is restored. For these two event types, *we measure the number of routing updates received by nodes at different locations in the network.* Third, we investigate the impact of a link failure that happens between transit networks by measuring the number of affected nodes and paths.

Protocol and Policies. We apply normal "no-valley" and "prefer-customer" policies, with the use of MRAI timers to limit the frequency with which a node sends updates to a neighbor. A recent change in the BGP specification RFC (RFC4271) [RLH06] required that explicit withdrawals are subject to MRAI rate-limiting, just as any other update. This implementation of the MRAI timer is referred to as WRATE. We assume WRATE in our simulations. However, we also investigate the dramatic impact of this specification change on BGP churn when individual destination prefixes are withdrawn and then re-announced by stub ASes.

6.2 Explaining Churn in a Growing Network

In this section, we first present an analytical model for describing the number of updates received at a node. Then we use the Baseline growth model to show how this model can be simplified for the different node types, and to determine the most important factors driving the churn growth.

Our main metric in this section is the number of updates received at a node after withdrawing a prefix from a C-type node, letting the network converge, and then re-announcing the prefix again. The experiment is repeated for 100 different C nodes (increasing this number does not change the results), and the number of received updates is measured at every node in the network. We then average over all nodes of a given type, and report this average. In the following, we refer to this procedure as a “C-event”. Note that due to the heavy-tailed node degree distribution, we expect a significant variation in the churn experienced across nodes of the same type. We return to the other event type in Sec. 6.4.

6.2.1 A Framework for Update Analysis

We give a formulation for the number of updates received at a node after a C-event, and discuss how churn increase depends on the use of policies, the topological properties of the network, and the convergence properties of the

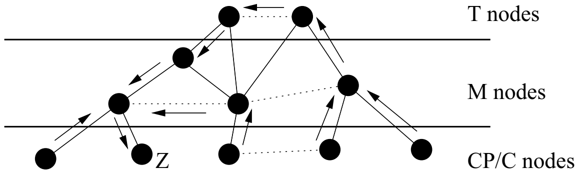


Figure 6.1: Illustration of network based on our topology model.

routing protocol used.

Figure 6.1 shows a generic network of the type described in Sec. 5.1. Transit links are represented as solid lines, while peer-to-peer links are dotted. For each node, we have indicated the preferred path to the *event originator* Z, which is the node announcing the active prefix. The routing updates that establish these paths flow in the opposite direction. We observe that due to the use of policies, updates (and the resulting paths) will follow a particular pattern: a node N will only announce a route to its providers and peers after an event at node Z if N has Z in its customer tree. On the other hand, N will always send an update to its customers, unless its preferred path to Z goes through the customer itself.

Let $U(X)$ denote the number of updates a node of type X receives after a C-event. X can be either of the four node types in our model; T, M, CP or C. We distinguish between the number of updates received from customers $U_c(X)$, peers $U_p(X)$ and providers $U_d(X)$ respectively. The total number of updates will be the sum of these: $U(X) = U_c(X) + U_p(X) + U_d(X)$. Each of these values will depend on three factors - the number $\mathbf{m}_{y,X}$ of direct neighbors of a given business relation y, the fraction $\mathbf{q}_{y,X}$ of these neighbors that sends updates during convergence, and the number of updates $\mathbf{e}_{y,X}$ each of these neighbors contribute. The expected number of updates from a certain class of neighbors will be the product of these three factors, and we can write

$$U(X) = \mathbf{m}_{c,X}\mathbf{q}_{c,X}\mathbf{e}_{c,X} + \mathbf{m}_{p,X}\mathbf{q}_{p,X}\mathbf{e}_{p,X} + \mathbf{m}_{d,X}\mathbf{q}_{d,X}\mathbf{e}_{d,X} \tag{6.1}$$

Note that for some node types, some of these terms will be 0, e.g., T nodes have no providers, and stub nodes have no customers. In the sequel, we will discuss how each of these factors depend on various topological characteristics and their interactions with properties of the routing protocol.

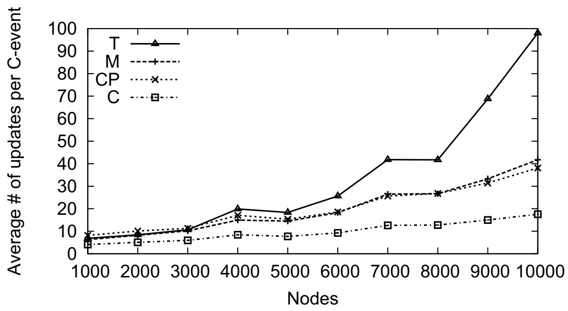


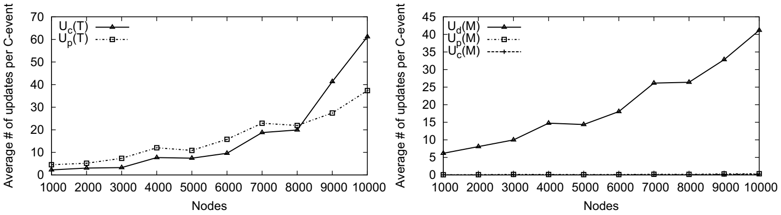
Figure 6.2: Number of updates received at T, M, CP and C nodes.

6.2.2 Churn at Different Node Types

Figure. 6.2 shows the growth in the average number of updates received by different node types after a C-event. We have calculated 95% confidence intervals for the values shown in Fig. 6.2, and they are too narrow to be shown in the graph. This tells us that increasing the number of event originators beyond the 100 used in this experiment will not reduce the observed variance. This variance is a result of the often significant differences between topology instances of different size, caused by the heavy-tailed node degree distribution. We observe that T nodes experience stronger growth in received updates. Followed by M and CP nodes. C nodes are the least affected.

We focus our discussion on the churn experienced by transit providers (T and M nodes), and content providers (CP nodes). These are the AS types that are most likely to be affected by increasing churn rates, since they must maintain larger routing tables with few or no default routes. Also, as discussed above, these are the nodes that experience the stronger growth in the number of updates received after a C-event.

Churn at T nodes. T nodes have no providers, so we have $U(T) = U_p(T) + U_c(T) = \mathbf{m}_{p,T}\mathbf{q}_{p,T}\mathbf{e}_{p,T} + \mathbf{m}_{c,T}\mathbf{q}_{c,T}\mathbf{e}_{c,T}$. Figure. 6.3a shows $U_c(T)$ and $U_p(T)$, for topologies of increasing size created with our Baseline topology model. We observe that both $U_c(T)$ and $U_p(T)$ increase with network size, and that both these factors contribute significantly to the total number of updates. As the network grows, the increased multihoming increases the number of routes that a T node learns from both its customers and peers. $U_p(T)$ is the



(a) # of updates received from peers and customers for T nodes.

(b) # of updates received from providers, peers, and customers for M nodes.

Figure 6.3: Analyzing number of updates received at T and M nodes.

larger factor for small network sizes, and it grows approximately linearly with network size, with a coefficient of determination $R^2 = 0.93$. The strongest growth is seen in $U_c(T)$, which dominates for larger network sizes. Regression analysis shows that the growth of $U_c(T)$ is *quadratic*, with a coefficient of determination $R^2 = 0.95$.

Churn at M nodes. While routes are only exported to peers and providers if they are received from a customer, routes are always exported to customers. As we can see in Fig. 6.3b, M nodes receive the large majority of their updates from their providers. Hence, a good estimate for the number of updates at M nodes is $U(M) = U_d(M) = \mathbf{m}_{d,M}\mathbf{q}_{d,M}\mathbf{e}_{d,M}$. The intuition behind this is that M nodes reach the "main part of the Internet" through their providers, and hence also receive the majority of routing updates from them. This is a major simplification, that makes our analysis more tractable. The same is true for CP nodes, so we limit our discussion to M nodes in the following.

Figure 6.4 shows the increase ratio in $U_c(T)$, $U_p(T)$ and $U_d(M)$. Each term is normalized so that the number of updates is 1 for $n = 1000$. To explain the observed trends for these terms, we look at the different factors described in Eq. 6.1 to find out how much of the growth is caused by each of them.

First, we look at the increase in the number of neighbors of different types. Figure 6.5 (top) shows the relative increase in the $\mathbf{m}_{c,T}$, $\mathbf{m}_{p,T}$ and $\mathbf{m}_{d,M}$ factors as the network grows. $\mathbf{m}_{c,T}$ grows much faster than the other factors. With our Baseline topology growth model, $\mathbf{m}_{c,T}$ grows approximately

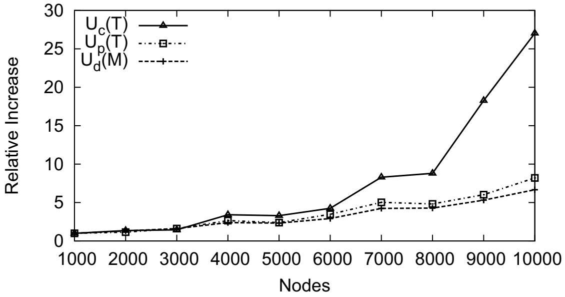


Figure 6.4: Relative increase in $U_c(T)$, $U_p(T)$ and $U_d(M)$.

linearly with n in the range of network sizes we consider. The number of peers $\mathbf{m}_{p,T}$ is given directly by $n_T - 1$, which grows very slowly with n. Similarly, $\mathbf{m}_{d,M}$ is determined by the MHD of M nodes $d_M = 2 + 2.5n/10000$, which also grows linearly with n.

The middle panel in Fig. 6.5 shows the relative increase in $\mathbf{e}_{c,T}$, $\mathbf{e}_{p,T}$ and $\mathbf{e}_{d,M}$, representing the average number of updates received from each neighbor of a given type that exports a route.

The increase in the $\mathbf{e}$ factors we see here is caused by path exploration. The increase is stronger for $\mathbf{e}_{p,T}$ and $\mathbf{e}_{d,M}$, since they represent links that are further away from the event originator, giving more chances for more paths to be explored during convergence.

We also see how the increase in the number of received updates is stronger from neighbors that have a larger number of policy-compliant paths that can be explored. The number of valid paths from a T node to an event originator increases superlinearly, which also causes a superlinear growth in $\mathbf{e}_{p,T}$, while the slower growth in the number of paths exported by customers gives a slower growth in $\mathbf{e}_{c,T}$. This is compliant with discussion of path exploration in [LABJ00].

The bottom panel in Fig. 6.5 shows the fraction of neighbors of a given type that announces a route after a C-event, represented by the $\mathbf{q}_{c,T}$, $\mathbf{q}_{p,T}$ and $\mathbf{q}_{d,M}$ factors. A provider will always announce a route to its customer, unless it prefers the path through the customer itself. Hence $\mathbf{q}_{d,M}$ is almost constant, and always larger than 0.99. $\mathbf{q}_{c,T}$ and $\mathbf{q}_{p,T}$ are both generally increasing with network size. This illustrates how *increased multihoming makes it increasingly likely that the event originator is in the customer tree*

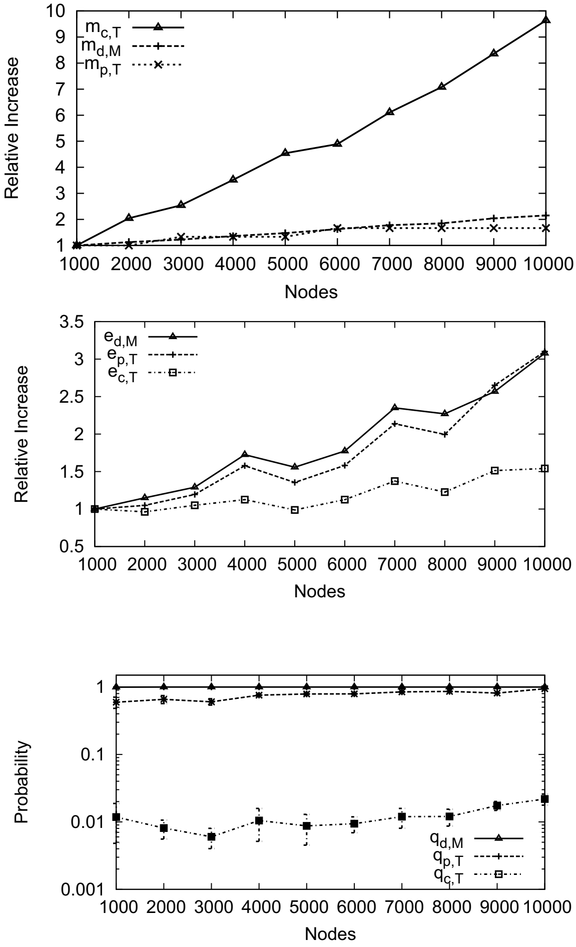


Figure 6.5: Relative increase in the factors that determine the update growth rate.

of a given customer or peer of a T node. This probability is much higher for peers than for customers of T nodes, since the peers, which are T nodes themselves, have a much larger number of nodes in their customer tree.

To sum up our discussion, we have shown that the churn at M nodes is dominated by the updates received from providers. The number of updates $U_d(M)$ grows with network size, since both the number of providers $\mathbf{m}_{d,M}$ and the average number of updates $\mathbf{e}_{d,M}$ received from each active provider grows, while the probability that a provider will announce a path is constant. The growth in $U_d(M)$ (a factor 6.7 in our range of $n = 1000$ to $n = 10000$, as seen in Fig. 6.4) is dominated by the growth in $\mathbf{e}_{d,M}$ (factor 3.1) and the (linear) growth in the MHD (a factor 2.2), which makes the total growth seem slightly superlinear.

For T nodes, both the updates $U_c(T)$ received from customers and the updates $U_p(T)$ received from peers are important, and both grow with network size. The strongest growth is contributed by $U_c(T)$, with a factor 27. Much of this growth can be attributed to a strong linear growth in the number of customers (a factor 9.5). Combined with the generally increasing trend for $\mathbf{q}_{c,T}$ (a factor 1.85) and an increase in $\mathbf{e}_{c,T}$ (a factor 1.5), this gives a clearly superlinear growth in $U_c(T)$.

The number of updates $U_p(T)$ received from peers also grows, but at a slower rate (a factor 8.2). This is mainly because of the much slower growth in the number of peers - while the number of customers $\mathbf{m}_{c,T}$ increases with a factor of 9.5 over our range of topology sizes, the number of peers $\mathbf{m}_{p,T}$ grows only by a factor 1.7. Furthermore, factor $\mathbf{q}_{p,T}$ also contributes to the growth in $U_p(T)$ by a factor of 1.6. However, most of the growth can be attributed to the increase in $\mathbf{e}_{p,T}$ (a factor of 3.1).

This section has shown how the T nodes experience the highest growth in churn as the network grows with our Baseline growth model. This increase is driven mainly by an increased number of updates from customers. M and CP nodes also see increased churn, driven mainly by their increased MHD. In the next section, we will see how changes in the topology growth model affect the various churn factors.

6.3 Topology Growth Scenarios

In this section, we look at several single-dimensional deviations from the Baseline model presented above. By looking at how BGP churn increases at various hypothetical growth models, we are able to answer different "what-if" questions about Internet growth. For example, what if multihoming to several providers becomes much more common than today for stub networks? Or what if buying transit services from tier-1 nodes becomes so cheap that they drive regional providers out of business? Our goal is not always to create realistic growth scenarios, but also to highlight the effect of altering different topological properties. Hence, we sometimes look at the effect of large changes to a single property at a time.

As seen in Sec. 6.2, T nodes experience both the strongest churn in absolute terms, and the strongest increase as the network grows. Hence, we focus mainly on the number of updates received at T nodes.

6.3.1 The Effect of the AS Population Mix

First, we look at how the mix of different node types affects churn, by considering four different deviations from the Baseline model with respect to the mix of T, M, CP and C nodes. These deviations illustrate how economic factors can create a very different fauna of networks than what we see today. To implement these scenarios in our model, we change the parameters n_T, n_M, n_{CP} and n_C, while keeping all other parameters fixed.

NO-MIDDLE In the first deviation, we look at a network without M nodes, by setting $n_M = 0$. This illustrates a scenario where the price for transit services from the globally present tier-1 nodes is so low that they have driven regional transit providers out of business.

RICH-MIDDLE In the second deviation, we focus on the opposite scenario, where the ISP market is booming and there is room for a plethora of M nodes. We implement this by multiplying n_M by 3 ($n_M = 0.45n$), and reducing n_{CP} and n_C accordingly (while keeping their ratio constant).

STATIC-MIDDLE In the third deviation, we look at a situation where all network growth happens at the edges of the network. The number of transit providers (T and M nodes) is kept fixed, and the network grows only by adding CP and C nodes. This could be a plausible scenario for the future, if the ISP population becomes stable.

TRANSIT-CLIQUE In the fourth and final deviation, we let all transit nodes be part of the top-level clique. This scenario may seem far-fetched, but it is important because it shows what would happen if the transit provider hierarchy collapses to a clique of "equals" connected by peering links. We implement this by setting $n_T = 0.15n$ and $n_M = 0$.

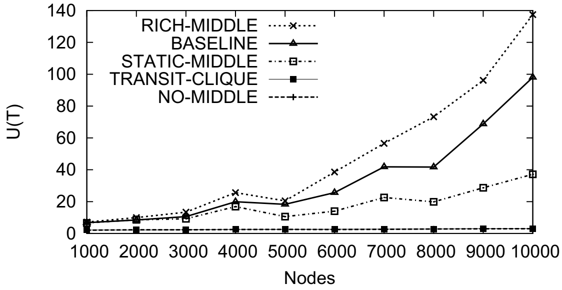


Figure 6.6: The effect of the AS population mix on T nodes.

Figure 6.6 shows the average number of updates seen after a C-event at a T node for each deviation as the network grows.

A first observation from the graphs is that the node mix has a substantial influence on churn. In particular, the comparison of RICH-MIDDLE, Baseline, and STATIC-MIDDLE shows that the number of M nodes is crucial. There are two ways in which M nodes increase churn at T nodes. First, an increasing number of M nodes increases the customers $m_{c,T}$ of T nodes. For instance, in the RICH-MIDDLE deviation $m_{c,T}$ increases by a factor of 10.2 when n increases from 1000 to 10000. On the other hand, $m_{c,T}$ increases only by a factor of 5.3 in the STATIC-MIDDLE deviation. Second, an increasing number of M nodes, when they are multihomed to other providers (M or T nodes), tends to also increase the factor $q_{c,T}$. The reason is that M nodes create additional valid paths from the source of a C-event (at stub networks) to T nodes, and so it becomes more likely that a T node will receive updates from its peers and customers after a C-event. Regression analysis shows that the growth of $U(T)$ in the RICH-MIDDLE, Baseline and STATIC-MIDDLE deviations can be modeled as *quadratic*, with different scaling factors.

We also observe that the number of T nodes in the network does not have any impact on the number of updates by itself. The only difference between deviations NO-MIDDLE and TRANSIT-CLIQUE is in the number of T nodes, and we see that the number of updates is the same in these two scenarios. In the absence of M nodes, T nodes will receive one update for each provider the event originator has - either directly from the event originator, or from a peer. This number only increases as a function of the multihoming degree of the event originator, and is not influenced by the network size per se.

An important conclusion from the above observations is that *the increased number of updates does not primarily come from an increased number of transit nodes, but from the hierarchical structure in which they are organized.* An Internet with several tiers of providers buying transit services from other providers gives a much higher update rate than a more flat topology where most stub networks connect directly to tier-1 providers. Whether the Internet will move towards a more hierarchical or flat topology in the future is hard to tell. We do know however, that the average path length, measured in AS-level hops, has remained roughly constant, at around 4 hops, during the last ten years [DD08]. This may suggest that the Internet has retained some hierarchical structure, and that the depth of that structure does not seem to vary with the size of the network.

6.3.2 The Effect of the Multihoming Degree

Next, we look at the effect of varying the number of transit links each node brings to the network. Both stub and mid-tier nodes have an incentive to connect to several providers to increase their reliability and load balancing capability. We implement these scenarios by varying the d_M, d_{CP} and d_C parameters, while keeping all other parameters fixed.

DENSE-CORE We look at the effect of much stronger multihoming in the core of the network (M nodes). We implement this deviation by multiplying d_M by 3.

DENSE-EDGE We look at the effect of densification at the edges of the network. In this deviation, stub nodes increase their multihoming degree. We implement this by multiplying d_C and d_{CP} by 3.

TREE We look at a tree-like graph, where all nodes have only a single provider. Here, d_M, d_{CP} and d_C are all set to 1. This is clearly not a realistic scenario, but helps us explore the extreme version of a trend.

CONSTANT-MHD Finally, we look at a scenario where the multihoming degree of all nodes stays constant. We implement this by removing the component of d_M, d_{CP} and d_C that depends on n.

Figure 6.7 shows the number of received updates (top) and the number of customers m_c (bottom) for T nodes in the different scenarios. First, note that there is a clear connection between the MHD and the number of updates seen at a T node - for the same network size, a higher MHD causes larger churn. Second, even though the number of customers $m_{c,T}$ is about the same

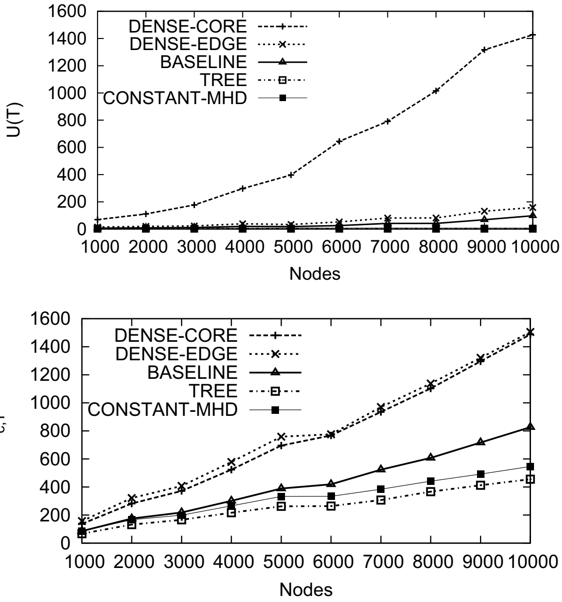


Figure 6.7: The effect of the multihoming degree at T nodes.

in DENSE-CORE and DENSE-EDGE, the churn is significantly higher in the former. This fact illustrates how the meshed connectivity of multihomed M nodes increases the likelihood that a T node will receive updates from a peer or customer. In other words, increased multihoming at the core of the network causes a larger growth in the factor $q_{c,T}$ than increased multihoming at the edges of the network. Specifically, we measured that $q_{c,T}$ increased by a factor of 1.6 in DENSE-CORE, while it increased by a factor of 1.3 in DENSE-EDGE.

When the MHD degree stays constant (in TREE and CONSTANT-MHD), the churn at T nodes is much less. In the extreme case of the TREE model, churn at T nodes remains constant at two updates per C-event, because the T node learns about the event from exactly one peer or customer (once for the DOWN event and once for the UP event). In the CONSTANT-MHD model, the number of updates is also roughly constant because the increase in the number of customers $m_{c,T}$ as the network grows is offset by a corresponding decrease in the probability $q_{c,T}$ that any given customer of the T node will have the source of that C-event in its customer tree.

According to a recent measurement study [DD08], the average MHD of both stub nodes and providers has been increasing during the last decade (from 1.4 to 1.8 for stub nodes and from 1.8 to 3.2 for providers). The fact that the MHD has been increasing more rapidly in the core of the network implies that the Internet is closer to the DENSE-CORE model than to the DENSE-EDGE or the CONSTANT-MHD deviations. This can be viewed as bad news, at least in terms of BGP churn.

6.3.3 The Effect of Peering Relations

In this subsection, we look at the impact of varying the peering degree between different types of nodes. The fraction of peering links in the Internet has increased over the last decade [DD08]. However, various difficulties in detecting such links do not allow us to know which peering model is most realistic.

NO-PEERING There are no peering interconnections, except in the clique of T nodes. This is clearly not realistic, but it serves as a reference point.

STRONG-CORE-PEERING We look at densification through more peering links in the core of the network. We model this deviation by doubling p_M.

STRONG-EDGE-PEERING Another variation is densification by adding more peering links at the network edges. We model this deviation by multiplying p_{P-M} and p_{P-C} by 3.

Since the peering degree is only changed at M and CP nodes, we show the number of updates received at M nodes rather than T nodes. Figure 6.8 shows the number of updates received at M nodes as a function of network size for the Baseline and each deviation. The main conclusion is that the peering degree does *not* cause a significant change in the generated churn. Adding or removing a significant number of peering links at the edge or at the core of the network does not give major differences in the number of updates. This conclusion also holds for other node types. To explain this observation, recall that updates are propagated over peering links only for customer routes. Hence, the fraction of peering links that are active during a C-event is low. Moreover, such updates have limited export-scope (only to customers), compared to routes received from customers.

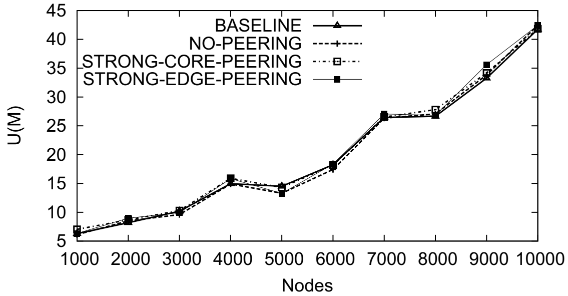


Figure 6.8: The effect of peering relations.

6.3.4 The Effect of Provider Preference

Next, we look at the effect of provider preferences, i.e., the probability that a node chooses to buy transit services from a T or an M node. This choice has implications for how the network will grow; a higher preference for T nodes gives a more "flat" structure, while a higher preference for M nodes diverts more paths through several layers of hierarchy. We define two deviations of the Baseline model:

PREFER-MIDDLE In the first deviation, nodes prefer to buy transit services from M nodes rather than T nodes. We implement this by setting $t_{CP} = t_C = 0$, and limiting the number of T providers for M nodes to one at most.

PREFER-TOP In this deviation, nodes prefer to buy transit services directly from T nodes. We implement this by limiting the number of M providers for M, CP and C nodes to be at most one.

The top panel in Fig. 6.9 shows that a scenario where most nodes buy transit from M nodes results in a higher churn at T nodes, while more direct connections to T nodes decreases churn. *If we are moving towards an Internet in which customers and content providers at the edges prefer to connect to mid-tier ISPs, the number of BGP updates at T nodes will be much higher than if they prefer to connect to tier-1 ISPs.* Looking at the different factors that determine $U(T)$, we observe that the PREFER-TOP deviation gives a much higher $\mathbf{m}_{c,T}$ than PREFER-MIDDLE, but that this is more than offset by a strong decrease in $\mathbf{q}_{c,T}$, as shown in the middle, and bottom panels in Fig. 6.9. An M node is more likely to notify its provider about a C-event than a stub node, because an M node has several potential event sources in its customer tree.

A recent study [DD08] observed that content providers and regional transit providers tend to buy transit service from either tier-1 or tier-2 providers with almost equal probability. The equivalent of C nodes ("Enterprise Customers") however, show a preference for tier-2 providers during the last 3-4 years, justifying the selection of the corresponding probabilities in the Baseline model.

6.4 Edge Link Failure

In this section, we focus on a different type of event than the C-event considered so far, namely the failure and subsequent restoration of a link connecting a stub node to one of its providers. We will refer to this event as an "L-event". We argue that this is a relevant event type to study, even if a single link in our topology model can sometimes represent several physical connections between two ASes. At the edge of the network, these links will often not be replicated, and hence such events are not unlikely to occur in practice.

Unlike a C-event, an L-event does not have to be communicated to all nodes in the network. Stub nodes may have more than one provider, and some nodes will in this case prefer a path that is not affected by the failure. Hence, we expect that an L-event will result in a lower number of updates being propagated in the network compared to a C-event.

We employ the framework introduced in Sec.6.2 to examine the impact of an L-event in our Baseline growth model. The experiment is repeated for 100 different links (increasing this number does not change the results), and the average number of updates received by each type of node over all experiments is reported. Figure 6.10 shows the number of updates received at different node types after an L-event. The general trends remain similar to those presented for C-events in Fig. 6.2. T nodes experience the highest number of updates, and the strongest increase as the network grows. However, the number of received updates $U(T)$ is significantly lower than after a C-event, and the increase rate is lower, with a growth factor of 4.4 in our range of $n = 1000$ to $n = 10000$ compared to a factor of 14.4 after a C-event.

We break these numbers down according to Eq. 6.1. Since the **m** factors depend on the topology and hence are the same for C-events and L-events, the lower number of updates is caused by the reduced growth and absolute numbers for the **e** and **q** factors. We find that both the number of updates

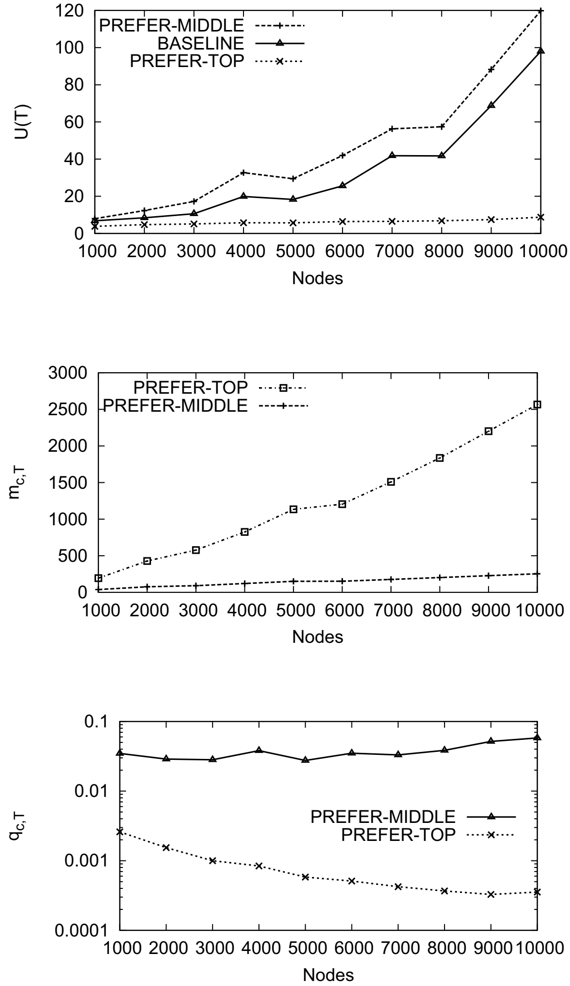


Figure 6.9: The effect of provider preference

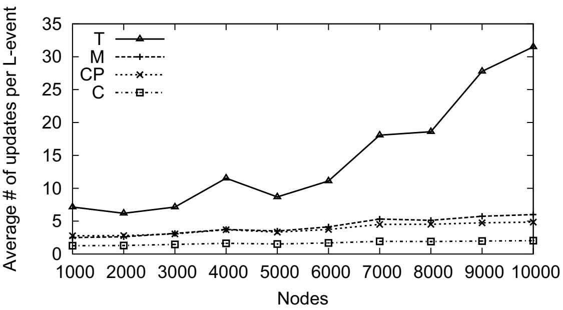


Figure 6.10: Updates received at T, M, CP and C nodes.

received from peers $U_p(T)$ and the number received from customers $U_c(T)$ increases with topology size. $U_c(T)$ shows the stronger growth with a factor of 6.6 for our range of topology sizes, compared to a factor 27 for C-events. This is caused by a strong linear growth in $\mathbf{m}_{c,T}$ with a factor 9.5, a slow increase in $\mathbf{e}_{c,T}$ by a factor 1.3, combined with an overall decrease in $\mathbf{q}_{c,T}$. As seen in Fig. 6.11, $\mathbf{q}_{c,T}$ shows significant variations across different topology instances. $U_p(T)$ increases with a factor 2.0 for L-events, compared to a growth of 8.2 for C-events. This can be attributed to a growth factor of 1.6 in $\mathbf{m}_{p,T}$, a growth of 1.2 in $\mathbf{q}_{p,T}$, and a constant $\mathbf{e}_{p,T}$.

The $\mathbf{q}$ factors reflect how the number of neighbors that send updates to a T node is reduced for an L-event compared to a C-event. In other words, many nodes prefer other paths to the stub node, and are not affected by the link failure. The $\mathbf{e}$ factors reflect how the affected nodes explore much fewer paths before the network converges after an L-event.

Next, we investigate the interaction between a set of topology growth scenario and L-events.

6.4.1 Topology Growth Scenarios

The upper panel in Fig. 6.12 shows how the number of updates received at a T node increases as the network grows, for different scenarios with respect to the multihoming degree.

First, note that the number of updates received is significantly lower after an L-event than after a C-event in all growth scenarios, as expected. Further, we observe that the DENSE-CORE scenario gives a significantly

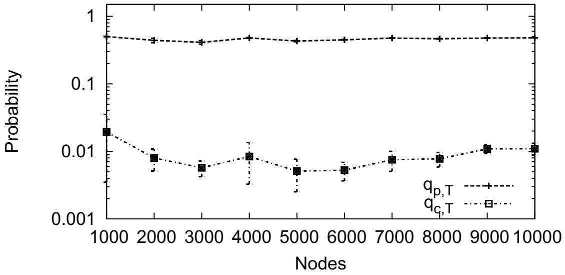


Figure 6.11: q factors.

higher churn growth rate than the other scenarios, while CONSTANT-MHD gives the lowest churn rate. The strong growth in the DENSE-CORE case is caused by the same effect as in the C-event discussed in Sec. 6.3.2; the provider connected to the failed link has 3x more providers, and will hence send an update to many more nodes. Correspondingly, the low number of providers in the CONSTANT-MHD scenario gives a very limited churn.

The DENSE-EDGE scenario, interestingly, has a lower growth rate than the Baseline scenario. This is different from the situation in a C-event, as shown in the middle panel of Fig. 6.12. This reduction in churn can be explained by observing that the provider connected to the failed link will send updates to the same number of (transit) neighbors in the DENSE-EDGE and Baseline scenarios, since the multihoming degree in the core is the same in the two scenarios. However, the probability that nodes receiving these updates will change their preferred path is lower in the DENSE-EDGE case, since the stub node can also be reached through several other providers.

The differences between the different growth scenarios is also visible in the bottom panel in Fig. 6.12, which shows the fraction of transit nodes in the network that select a new preferred path after the L-event. While this fraction is constant or slightly increasing with network size in the CONSTANT-MHD, Baseline and DENSE-CORE scenarios, it decreases from an already lower level in the DENSE-EDGE scenario.

These results show that the effect of densification through increased multihoming degree on the experienced churn level is different depending on the type of event that triggers the re-convergence. *While densification at the edge increases churn after a prefix failure, it gives reduced churn after a link*

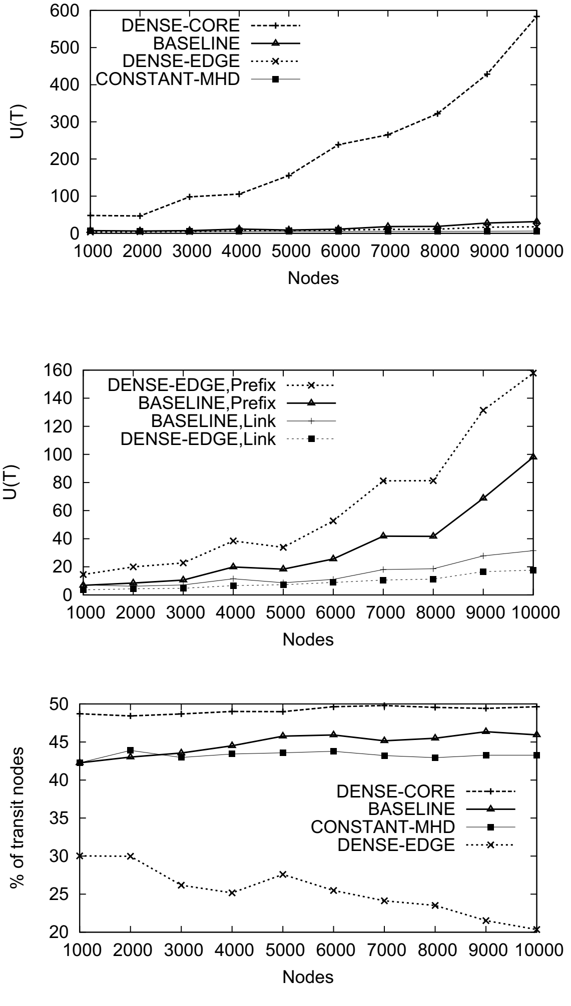


Figure 6.12: The effect of multihoming degree on churn after an L-event

failure.

We have also investigated churn after an L-event in the other topology growth scenarios discussed in Sec. 6.3. Generally, we find that the number of updates received by a T node is lower for L-events than C-events. However, the general trends observed in churn after an L-event remain similar to what is observed after a C-event.

A recent study [OZP+06] showed that the number of routing events that ends with completely withdrawing a prefix, is lower than the number of events that ends with a new preferred path. Further, the extent of path exploration for the later type of events is lower than the former one. These observations fit well with our results.

6.5 Core Link Failure

So far, our focus has been on events taking place at the periphery of the network, i.e., on the stubs and their connected links. In this section, we look at events in the core of the network. Such events are important when studying BGP churn; they can potentially affect a large number of destination prefixes, and trigger a large number of routing updates.

Several types of events in the core of the network, such as link failures, policy changes or other configuration changes, will trigger a re-convergence and hence generate churn. In several of these events, a link that was previously available (unavailable) for routing now becomes unavailable (available). In our AS-level topology model, we investigate these core events by removing a logical link connecting two core (M or T) nodes. In the real Internet, transit ASes are often connected by more than a single link, and a single event may not result in a complete disconnection between two ASes. Some prefixes may be announced over a different link that is not affected by the failure. Hence the consequences of a link removal in our AS-level topology model represent the worst-case scenario where the two ASes are completely disconnected. We will speak of this event as a "core link failure", but note that this is also representative of other event types in the core.

The number of routing updates generated by the failure of a core link depends on the number of destination prefixes announced over that link, which again depends on the distribution of destination prefixes across ASes. Using the number of updates as a metric for these failures would then require a good model for this distribution, which is challenging. In addition, running simulations with a large number of dynamically routed prefixes will severely limit scalability, since all nodes must maintain state for all destinations throughout the convergence process.

Instead, we use two other metrics to describe the impact of core link failures. First, we investigate the fraction of nodes that experience a change in their routing table. This metric describes churn by characterizing how widely updates are propagated. Second, we measure the fraction of (source, destination) AS pairs affected by the failure. This metric is related to the number of prefixes that need to be updated after a core link failure. The two metrics are calculated by first computing the preferred policy compliant paths from each source to each destination, and then taking down a transit link. We repeat for all transit links, and report the average value. Both of these metrics are related to the number of updates; the first describes the scope of the re-convergence, while the second is related to the number of prefixes affected.

Calculating the best paths between all sources and destinations is an expensive process with our dynamic simulator, which simulates the entire convergence process. Hence we employ the algorithm presented in [WZMS07] for this purpose.

6.5.1 The Effect of the AS Population Mix

We look at the baseline scenario and different deviations with respect to the mix of different types of ASes. We focus on the failure of a link between an M node and a T node. The same experiment has been performed with links between M and M nodes, with similar results.

The top panel in Fig. 6.13 shows the fraction of nodes that change the preferred path to at least one destination when the link is removed. First, we note that the *mix in the node population does not impact the number of affected nodes* after a core link failure in any significant way. This is explained by observing that even when the fraction of M nodes in the network increases, some destinations are still reached over the failed link. Consider the case where the failed link connects the T node t to the M node m. The fraction of nodes in the network whose best path to m goes through t is little dependent on the number of M nodes in the topology, since the MHD of all nodes is the same in all the considered deviations. Second, we observe that as the network grows, the fraction of nodes affected by the failure decreases for all deviations. This is because the MHD increases (at a similar rate in all

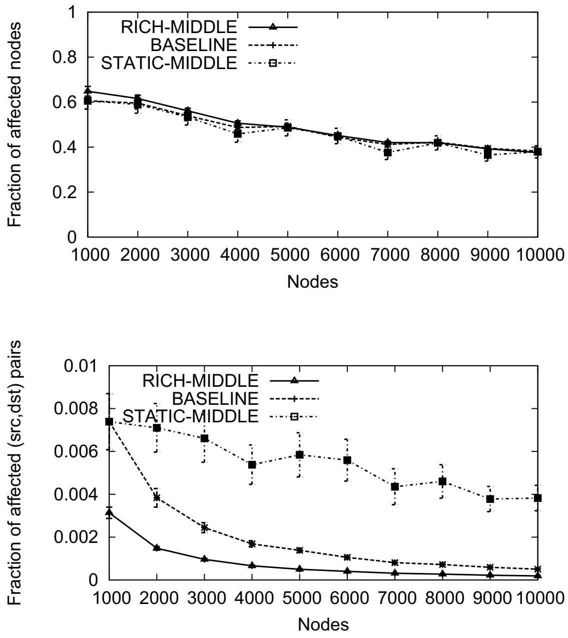


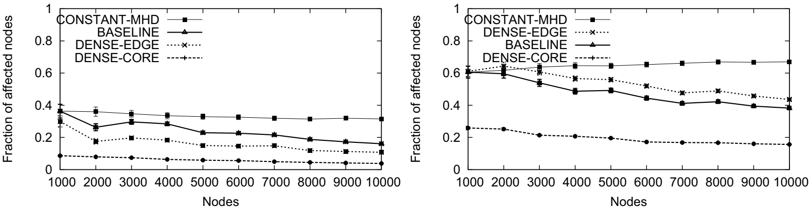
Figure 6.13: The effect of the AS population mix

deviations) with network size.

The bottom panel in Fig. 6.13 shows that while the fraction of nodes affected by the link failure is similar for all growth scenarios, the fraction of affected source-destination pairs varies significantly. This is natural, since when the number of possible paths between a source and a destination increases, the probability that the selected best path will cross a particular link will decrease. The STATIC-MIDDLE scenario has a low number of transit nodes in the middle of the network, and the number of available paths between a source and destination only increases because of increased MHD of the edge nodes. In the RICH-MIDDLE scenario, the path diversity increases at a much faster pace with network size, due to the increasing MHD of the high number of M nodes. The high variability and wide confidence intervals in the STATIC-MIDDLE scenario can be explained by the much smaller sample size; there are relatively few T-M links in this scenario, since no new M nodes are added as the network grows.

6.5.2 The Effect of the Multihoming Degree

Figures 6.14a and 6.14b respectively, show the fraction of affected nodes after failing an M-M transit link connecting two M nodes and an M-T transit link between an M node and a T node, for different growth scenarios with respect to multihoming degree.



(a) M-M link failure

(b) M-T link failure

Figure 6.14: The effect of multi-homing degree.

We first observe that since links connecting two M nodes are less central than links connecting an M node and a T node, the number of affected nodes is lower for all growth scenarios.

For both types of core link failures, the CONSTANT-MHD shows the largest fraction of affected nodes, since the number of possible paths is low (and constant with a growing topology) in this scenario. For all other scenarios, the fraction of affected nodes slowly decreases with topology size, since the MHD and hence the number of available paths increases.

Increasing the MHD in the core of the network (DENSE-CORE) decreases the fraction of affected nodes after both types of link failure. Since the customer m_c connected to the failed link has a much higher number of providers than in the Baseline scenario, the probability that the failed link was the preferred route to m_c for some nodes decreases.

Interestingly, the effect of densification at the periphery of the network (DENSE-EDGE) is different for the two failure types. For an M-M link failure, the number of affected nodes is lower than in the Baseline, while it is higher for an M-T failure. To explain this, note first that if an M node m_c has one M provider m_p and one T provider t, then nodes in the customer tree of t will normally prefer the path through t because it is shorter. Other nodes may prefer the path through m_p (if they are in the customer tree of m_p or any of m_p's providers). With DENSE-EDGE, the probability that a stub node is in the customer tree of t increases, since each stub node is connected to more providers. Hence, M-T links will be used by more edge nodes to reach M nodes. Correspondingly, fewer stub nodes will rely on M-M links. *Increased multihoming at the periphery of the network increases the importance of links close to the core of the network, and reduces the importance of links connecting mid-tier nodes.*

We have also measured the number of affected source-destination paths under the different growth scenarios for the two core link failure types (not shown). The trends are the same as for the fraction of affected nodes; when more nodes are affected, more paths are also affected. As observed in Fig. 6.13, the fraction of affected paths decreases with network size, for the reason explained above.

6.6 The Effect of WRATE

In this section, we look at a particular aspect of updates rate-limiting in the light of our topology growth model; rate-limiting will be the main focus of the next chapter. We investigate the effect of implementing the MRAI timer with the WRATE option. According to the outdated BGP specification (RFC1771) [RL95], explicit withdrawals are *not* subject to the MRAI timer. This approach is still used by some router implementations, including the open-source software router project Quagga [qua]. In the most recent RFC (RFC4271) [RLH06] however, it is stated that explicit withdrawals *should* be rate-limited just like other updates. The reasons for making this important change are not clear to us. Discussions on the IDR mailing list (see [Scu07] and the related thread) indicate that some people opposed treating withdrawals differently than other updates, since there is no general way to distinguish "good news" from "bad news" based on the update type. While this might be true for some corner cases, we do not think this argument is a

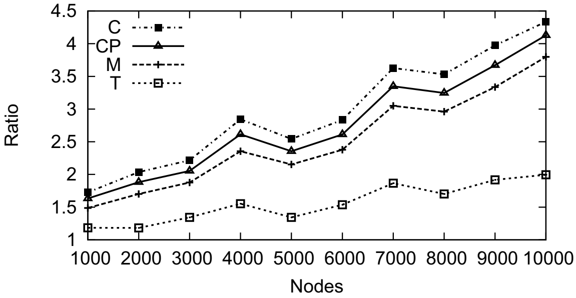
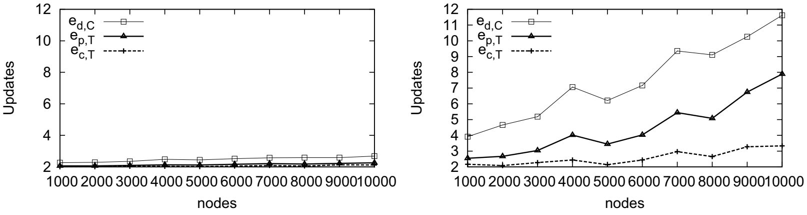


Figure 6.15: Number of updates for the WRATE case.

sufficient reason for a change that we here show has a major impact on routing scalability. With WRATE, it takes more time for the route withdrawals to reach all nodes. During that time a node will announce other alternate paths, and hence we expect increased churn. To clearly capture the effect of WRATE, we study WRATE impact on churn generated after a C-event. This type of events results in sending a large number of explicit withdrawals network-wide.



(a) NO-WRATE

(b) WRATE

Figure 6.16: Comparing e-factors.

Figure. 6.15 shows the increase in updates received at T, M, CP and C nodes when WRATE is applied. We use the Baseline topology growth model. The plot shows the number of updates $U(X)$ received with WRATE, divided by $U(X)$ in the NO-WRATE case. We observe that WRATE causes a significant increase relative to NO-WRATE for all node types. Further, this increase factor grows with network size. For T nodes, the churn doubles when $n = 10000$. The relative increase is larger for nodes at the periphery of the network, since they generally have longer paths to the event originator, and hence a larger potential for path exploration. However, the absolute number of updates is much smaller at those nodes (see Fig. 6.2).[1]

The plots in Figure. 6.16 show $\mathbf{e}_{d,C}$, $\mathbf{e}_{p,T}$ and $\mathbf{e}_{c,T}$ the case of NO-WRATE and WRATE respectively. These plots confirm that the difference between the two MRAI implementations is caused by the increase in the e factors. We also see that the increase in the number of received updates is stronger when the neighbor node has a larger number of policy-compliant paths that can be explored. In the Baseline growth model, the number of valid paths from a T node to an event originator increases superlinearly, which causes a superlinear growth in $\mathbf{e}_{p,T}$. On the other hand, the slower growth in the number of paths exported by customers also gives a slower growth in $\mathbf{e}_{c,T}$.

We have also explored the effect of WRATE in the topological deviations described in Sec. 6.3. Those results show that the churn increase with WRATE is stronger in a more well-connected network, especially at its core, where there are more potential paths that can be explored after a prefix withdrawal. In the DENSE-CORE deviation, the number of updates received at T nodes when $n = 10000$ increases by a factor 3.6 with WRATE, compared to 2.0 in the Baseline model.

The results presented here show how the use of WRATE exacerbates path exploration, and hence increases churn. This effect gets stronger as the network grows, and even more so in a network that is densely connected at its core. In summary, the previous observations make us question the wisdom of requiring WRATE in the most recent specification of the BGP protocol [RLH06].

6.7 Summary

We have examined the role of topology growth on the scalability of BGP. We started by looking at the number of updates received at nodes at different

[1] Again, the large variance in this graph is due to the natural heterogeneity in the underlying topologies. The confidence interval at each network size is too narrow to show.

locations in the AS hierarchy after a C-event. For different node types, we have identified the most significant sources of churn, and described how different factors contribute to increased churn as the network grows. We have shown that nodes at the top of the AS hierarchy experience both the highest churn in absolute terms, and the strongest increase as the network grows. We further looked into the impact of L-events on routing scalability, and demonstrated that certain topology growth scenarios scale differently depending on failure types. Besides, we investigated the impact and visibility of transit link failures between core nodes, by calculating the number of affected nodes and source-destination pairs.

Using our flexible topology model, we have explored scalability in several plausible and educational “what-if” scenarios for the growth of the AS-level topology. We have shown that *the most important topological factor deciding the number of updates generated is connectivity in the core of the network.* In particular, the number mid-tier transit providers and the multihoming degree of these nodes plays a crucial role, since *transit nodes in the mid-level of the Internet hierarchy have a special role in multiplying update messages.* Another important finding from this study is that peering links play a very different role than transit links with respect to scalability. *The peering degree in the Internet does not influence churn.* We have also shown that the depth of the hierarchical structure in the Internet plays a significant role. *A relatively flat Internet core is much more scalable than a vertically deep core.* Finally, we have demonstrated that densification through increased multihoming degree will have a different impact on routing scalability depending on its location and the failure type. *While densification at the edge increases churn after a prefix failure, it gives reduced churn after a link failure. On the other hand, a denser core reduces the scope and impact of a transit link failure but it increases churn because of both edge prefix and edge link failures.*

In the last section of this chapter, we have investigated the role of a particular aspect of the rate-limiting mechanism - whether explicit route withdrawals are subject to the MRAI timer. We have shown that rate-limiting explicit withdrawals leads to a significant increase in churn under our event model, because of the effect of path exploration. This difference grows with network size, and is stronger in well-connected networks. The next chapter investigates the impact of different update rate-limiting implementations and configurations on churn.

Chapter 7

The Role of Update Rate-Limiting

To limit the rate of updates that a router must process, it is common to perform some type of rate-limiting in BGP sessions. When an event affects the best route to a destination prefix, this will often trigger a sequence of update messages before the network stabilizes on the new preferred route. By delaying the transmission of an update message for a configured amount of time, that message will often be invalidated by the subsequent update for the same destination prefix. This way, it is often possible to mask out intermediate states, and thus reduce the number of updates sent over the BGP session. A larger configured delay gives a stronger reduction in churn, but also increases the time used to converge to the new steady state.

Earlier work by Griffin and Premore [GP01] analyzed the impact of different rate-limiting settings on churn and convergence time by simulating single prefix announcements and withdrawals in small generic topologies. The main insight from their work is that it is possible to find a timer setting that minimizes convergence time while keeping the number of updates low, but these settings vary depending on the size and structure of the topology. In addition, the convergence process will depend on the nature and location of the underlying routing events; routing changes that result in a complete withdrawal of a network prefix involve more path exploration than those ends with an alternative path [OZP$^+$06]. Hence, it is difficult to find generic timer settings that work well in the Internet based on these results. To give practical guidelines for the use of rate-limiting timers, the update arrival pattern that is observed in the Internet must be characterized and taken into account. Analyzing the impact of different rate-limiting implementations and timer settings on real measured BGP sessions is the main goal of this chapter.

Our starting point is the time series of BGP updates produced by border routers in a well-connected stub AS. This data is a product of the Internet topology (as seen from the monitored AS) and the mix of routing events that takes place during our measurement period. Starting from this data, we investigate the different ways in which major router vendors implement BGP rate-limiting. We explain how the two main approaches (MRAI timers and OutDelay) have very different effects on the churn rate and convergence time. Based on measurements on parallel monitoring sessions with and without rate-limiting, we quantify the churn reduction achieved with a given configuration setup. Further, we use the measured data to emulate different rate-limiting implementations and timer values and quantify the corresponding churn levels.

By looking at update traces from a large number of route monitors in the RouteViews project [rou], we discover that the arrival pattern of BGP updates for single prefixes is remarkably stable across BGP sessions from a diverse set of ASes across the Internet. This allows us to derive a formulation that quantifies the expected reduction in churn for different rate-limiting implementations and configured timer values.

Our findings present a general framework for helping network operators that want to use rate-limiting in deciding which implementation to choose. Furthermore, it helps in finding the right balance between churn reduction and increased convergence times when setting the timer value. These findings can be particularly interesting for networks wishing to receive full Internet routing information while using edge routers with limited processing power.

7.1 Rate-Limiting Implementations

To limit the rate of BGP updates, the BGP standard [RLH06] recommends the use of a MinRouteAdvertisementIntervalTimer (MRAI timer) which specifies the minimum time interval between two consecutive updates for a destination prefix. The recommended value of this timer on eBGP sessions is 30 seconds. To avoid peaks in the number of sent updates, the standard recommends jittering the MRAI timer by multiplying its value with a random number between 0.75 an 1.

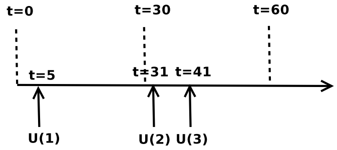


Figure 7.1: Example of a convergence sequence

Some implementations (e.g. Cisco IOS and the Quagga software router), implement per-session timers rather than per-prefix timers in order to reduce overhead. When the MRAI timer is enabled on a certain session, the router queues updates for all prefixes and sends them out in a burst when the timer expires[1].

Another common BGP implementation (Juniper's JunOS) implements rate-limiting using the *OutDelay* parameter. Unlike the MRAI timer implementation described above, this delay is added to each update for each prefix individually. When a router changes its best path to a destination prefix, it will not inform its peer about the change unless the route has been present in its routing table for the specified out delay. The outDelay parameter value is 0 seconds by default in Juniper routers (i.e., no rate-limiting).

The different implementation choices give different reductions in the number of updates and different increase in convergence time. Figure. 7.1 illustrates a sequence of updates for a prefix p arriving at a router. First, let us assume the router uses an MRAI timer with a value of 30 seconds, and that the timer expires at $t = 0, 30, 60$. Then, the router sends out an update informing about the change caused by U_1 at $t = 30$. The router queues an update that reflects the change caused by U_2 to be sent out at time $t = 60$. However, the queued update is invalidated by the arrival of U_3, and only a single update sent at $t = 60$. In this example, two updates are sent, and the convergence process takes 55 seconds, measured from the arrival of the first update.

Second, let us consider a session with an OutDelay of 30 seconds. When the router receives U_1, it schedules an update to be sent at time $t = 35$. However, when receiving U_2 at $t = 31$, the scheduled update transmission is cancelled. Instead, a new update that reports the change caused by U_2 is scheduled to be sent at $t = 61$, which in turn is cancelled after receiving U_3.

[1] This is the most common form of rate-limiting used in the Internet today, since it is turned on by default in Cisco routers.

The receiving of U_3 finally results in scheduling a new update that is sent at $t = 71$. In this case, the router sends out only one update and converges in 66 seconds. This example shows that the OutDelay implementation gives a stronger reduction in churn than MRAI timers, but that it also converges slower, since updates are always delayed by a full timer interval. In this chapter, we use a combination of measurements and emulations to compare the two different implementations, and to evaluate the churn reduction achieved with different parameter settings.

7.2 Measurement Setup and Data

Our measurement setup consists of two collectors, each connected to three different routers (referred to as monitors) in a stub AS. The collectors that receive BGP updates from the monitors are implemented on computers that run the Quagga routing suite [qua] using private AS numbers. The monitored stub AS is well connected to the Internet through multiple transit providers and direct peering, and has a geographical presence in several cities in both North America and Europe. The collectors use multi-hop eBGP sessions to peer with the monitors. A monitor sends a BGP update to the collector every time there is a change in the preferred path from the monitor to a destination prefix. In addition, we dump a snapshot of the routing table of each monitor every two hours.

The three monitors belong to the DFZ, meaning that their routing tables contain an entry for practically all destination networks in the Internet. They are located in different POPs, two of them in Europe and the third in North America. All three monitors are Juniper routers that run JunOS, and they are connected in a full mesh of iBGP sessions. Each monitor runs a separate BGP session with each of the two collectors. In one of the sessions, the default OutDelay value of 0 seconds is used. We refer to the measured time series of updates for this session as M_0. The other session uses an OutDelay of 30 seconds. We refer to this time series as M_{OD30}. The two time series contain all BGP updates received in the period from March 7 2009 to July 6 2009[2].

If the BGP session between a monitored router and the collector is broken and re-established, the monitor will re-announce its full routing table. Such table transfers are a local artifact of the measurement infrastructure, and does

[2] One of the collectors was unavailable for a few days in the beginning of May.

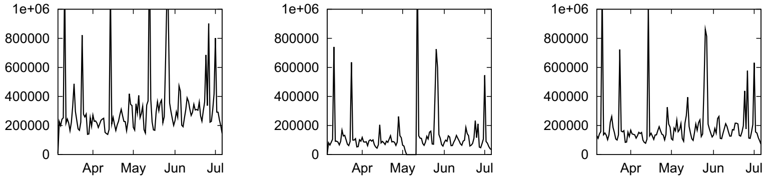
not represent genuine routing dynamics. In order to remove these updates, we use the algorithm described in [ZKL$^+$05]. We verify the inferred table transfers against those identified using the BGP session logs, and find that the algorithm is able to identify all table transfers but with a mismatch in the starting time of the transfer in some cases, up to one minute in length. We therefore use the collectors' logs to identify the start of each table transfer and use the reported length of the transfer by the algorithm plus one minute to decide the transfer period. This extra minute is added to the transfer duration to assure we do not include updates that belong to a table transfer to our filtered time series.

After filtering the updates caused by session resets, we record more than 161 million updates in M_0 and more than 56 million updates in M_{OD30} from the respective three monitors collectively. The total number of updates in M_{OD30} is about one third of those in M_0. However, both time series for each monitor contain spikes in the total number of daily updates, which exceed one million updates.

We further examine the measured daily churn across the three monitors in both configurations for similarity. Since the churn time series include outliers and that can affect the accuracy of a parametric correlation test such as Pearson's, we use instead the Spearman's rank correlation. This is a non-parametric test that does not assume a conformance between the underlying data and any probability distribution, and hence is less affected by outliers. The pairwise rank correlation coefficients between different monitors vary between 0.75 and 0.85 for both M_0 and M_{OD30}. The close association between our monitors is not surprising because they belong to the same AS and are connected in a full mesh of iBGP sessions. Because of the close relation between our three monitors, we mostly present results from only one monitor in the rest of this chapter.

Figures 7.2a and 7.2b show the number of updates per day for M_0 and M_{OD30} respectively. A main observation from these plots is that the use of OutDelay for rate-limiting significantly reduces the number of BGP updates. Across the three monitors, churn levels were reduced by 64% over the measured period. This can also be observed in Tab. 7.1, which shows that the reduction in the median daily churn rate is consistent across all three monitors.

There are significant spikes in the daily update rate in both M_0 and M_{OD30}. On closer inspection of the data, we find that these are normally



(a) M_0 (b) M_{OD30} (c) E_{MRAI30}

Figure 7.2: Daily BGP churn

Monitor	M_0	M_{OD30}	M_{OD30}/M_0
A	302415	105084	0.35
B	286234	104553	0.37
C	257550	91550	0.36

Table 7.1: Measured median daily churn

caused by underlying events in the transit paths that affect a large number of destination prefixes simultaneously. Rate-limiting will reduce the number of updates for each individual prefix after such events, but at least one update still has to be sent for each affected prefix.

In the following section we investigate the impact of different rate-limiting implementations and timer values on churn reduction.

7.3 Emulating Rate-Limiting

Our next goal is to quantify the different reduction in churn when performing rate-limiting with OutDelay and MRAI timers and the impact of the timer value.

7.3.1 OutDelay vs MRAI

Our measurement set-up captures only OutDelay of 30 seconds. Therefore, we use emulations for evaluating other configurations . The emulation script takes M_0 as an input. This series is collected from a BGP session that is configured with no rate-limiting. Furthermore, for emulating the MRAI timer, it identifies the timestamps at which the timer is supposed to expire. Then, it loops through the input workload and groups all updates for the same prefix that arrive between two consecutive timer instances. All grouped updates are invalidated except the last one. This results in a new time series that reflects the effect of the timer. We also develop another script that emulates OutDelay.

For our purpose, we run the M_0 churn time series through the emulation scripts to emulate both MRAI and OutDelay timers of 30 seconds. The output from the scripts is two new time series that reflect the impact of the chosen timer implementation. We denote these emulated time series as E_{OD30} and E_{MRAI30} respectively.

To validate the sanity of the emulation scripts, we first apply them on M_0 using an OutDelay of 30 seconds, to obtain E_{OD30}. This time series is directly comparable to the measured time series M_{OD30}. For the two time series M_{OD30} and E_{OD30}, we count the number of updates in every hour in our measurement period. One hour granularity is chosen because it gives a reasonably large sample size, which improves our validation process. Three different statistical measures are then used for comparing M_{OD30} and E_{OD30}. Spearman rank correlation coefficient is used to measure the statistical dependence as it changes temporally. The correlation coefficient gives an idea about the relative dependence between two random variable, but it does not report similarities between absolute values. Therefore, we use Kolmogorov-Smirnov [LK99] and Kullback-Leibler divergence [KL51] tests to examine the similarity in churn distributions. The two-samples Kolomgorov-Smirnov (K-S) test examines the difference between two samples in order to check whether they come from the same population, while the Kullback-Leibler (K-L) divergence test also measures the divergence between two samples but from information theoretic approach.

Table 7.2 shows the statistical test results. M_{OD30} and E_{OD30} are strongly correlated with a Spearman's correlation coefficient ρ of 0.96 in all three monitors. The Kolmogorov-Smirnov (K-S) test reports relatively small divergence in all three monitors. Kullback-Leibler (K-L) divergence on the other hand, shows strong similarity between M_{OD30} and E_{OD30} in monitors A and C, but somewhat lower similarity in monitor B. The observed differences between M_{OD30} and E_{OD30} can be explained by the fact that the emulation is based on a different BGP session that operates independently from the measured timer-enabled session. Each session experiences independent session resets, and the timers in M_{OD30} and E_{OD30} are not synchronized.

Figure 7.2c shows the number of updates per day in one of our monitors for E_{MRAI30}. We observe that the reduction in churn is smaller with MRAI

Monitor	Spearman ρ	K-S	K-L
A	0.96	0.1863	0.06
B	0.96	0.1855	0.26
C	0.96	0.1315	0.03

Table 7.2: Measurements vs Emulation

Monitor	M_0	E_{MRAI30}	E_{MRAI30}/M_0
A	302415	189377	0.63
B	286234	180339	0.63
C	257550	151472	0.59

Table 7.3: Emulated median daily churn using MRAI timer

timers than when using OutDelay with the same timer setting. This can also be seen in Tab. 7.3, which shows the same ratio as presented in Tab. 7.1 for M_{OD30}.

7.3.2 Rate-limiting Timer Value

The extent of churn reduction achieved by rate-limiting is strongly dependent on the arrival pattern of updates for each prefix, and the timer value. A longer timer helps in invalidating more intermediate states, but it is also increases convergence time [GP01]. We use M_0 as input to our emulation scripts in order to determine the churn reduction for different OutDelay and MRAI timer values in the range between 5 and 300 seconds.

Figure 7.4a shows how churn is reduced for increasing timer values. The y-axis shows the total churn in the measurement period with a timer value x, as a fraction of the total churn in M_0. A first observation from the figure is that the OutDelay implementation gives a stronger reduction in churn than the MRAI implementation, as explained in Sec. 7.1. Recalling that rate-limiting timers delay routing convergence by up to one timer interval for each BGP session that an update traverses, this figure also illustrates the tradeoff between churn and the configured timer value.

Using regression analysis, we find a logarithmic decay in churn for increasing timer values. For both the OutDelay and MRAI approaches, the fraction of churn in the rate-limited case scales as $R(T) = \alpha - \beta ln(T)$, where T is the timer value. For our data set, we find that $R(T)_{OutDelay} = 0.86 - 0.11ln(T)$ with coefficient of determination 98.4%, while $R(T)_{MRAI} = 0.98 - 0.17ln(T)$ with coefficient of determination 99.3%.

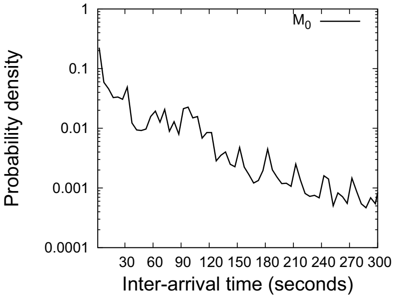
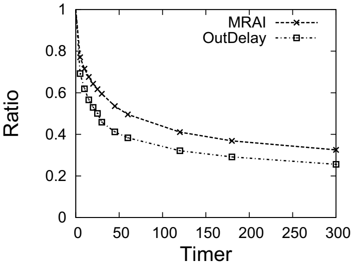
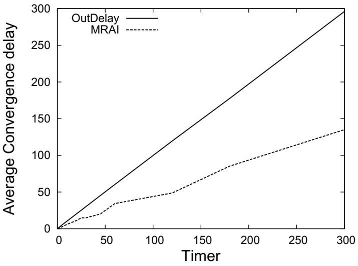


Figure 7.3: Update inter-arrival times for individual prefixes



(a) Churn reduction



(b) Convergence delay

Figure 7.4: The impact of rate-limiting timer's value

We observe that a timer value of 5 seconds results in a 31% reduction in the level of churn in the OutDelay implementation, while a timer value of 10 seconds cuts down the level of churn by 39%. This shows that a significant reduction in churn can be achieved even with a relatively low timer value. This effect can be further understood by looking at the distribution of update inter-arrival times for individual prefixes in the M_0 time series, shown in Fig. 7.3. This figure shows that a significant fraction of updates arrive shortly after the previous update for the same prefix, and will hence be filtered out by the timer even at low values. Note that there are peaks in the inter-arrival time distribution around multiples of 30 seconds; these peaks can be explained by the timers employed on the incoming BGP updates to the monitored router. These observations suggest that the recommended default MRAI timer given in the BGP standard is often too conservative, as pointed out also in [Jak08].

On the other hand, Fig. 7.4b illustrates the convergence delay introduced due to rate-limiting. We group routing updates of the same prefix into events by adopting the definition and threshold given in [WMRW05]. The reported delay is the difference between the convergence time of a routing event in M_0 and the respective implementation time series. OutDelay delays convergence by one timer interval, while MRAI results in a delay of one half the timer value.

7.4 A Model for Churn Reduction Using Rate-Limiting Timers

Figure 7.3 illustrates the importance of the update inter-arrival pattern for individual prefixes for determining the effect of rate-limiting. In this section, we characterize the distribution of update inter-arrival times, and use this information to develop a model for churn reduction using rate-limiting timers.

Let $f(t)$ denote the probability density function of the inter-arrival times for updates concerning a given destination prefix (Fig. 7.3 shows this function for one of the monitors in our dataset), and let $F(t)$ denote the corresponding CDF. With rate-limiting using OutDelay, all updates that arrive less than one timer interval T before the subsequent update for the same prefix will be invalidated. In this case, the remaining churn (as a fraction of the non rate-limited churn) is

$$R(T)_{OutDelay} = 1 - F(T) \quad 1 \leq T \tag{7.1}$$

With rate-limiting using MRAI timers, an update is invalidated if the subsequent update for the same prefix arrives within the same MRAI interval (i.e., before the MRAI timer expires). Hence, on average, an update is invalidated if the subsequent update for the same prefix arrives within $T/2$ seconds. Taking into account that the MRAI timer is jittered by multiplying with a random number in [0.75, 1], the remaining churn with an MRAI timer value of T is given by

$$R(T)_{MRAI} = 1 - F(\frac{0.875T}{2}) \quad 1 \leq T \tag{7.2}$$

To formulate a model for $R(T)$, we need to characterize $F(T)$ using empirical data. A main question is how universal $F(T)$ is across different BGP

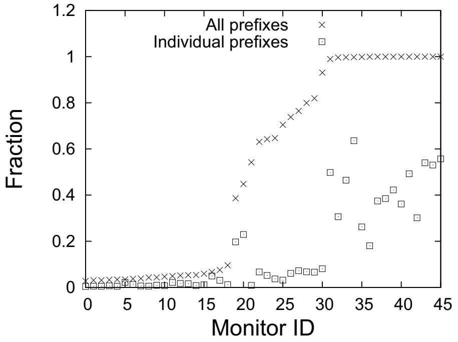


Figure 7.5: Identification of rate-limited monitoring sessions.

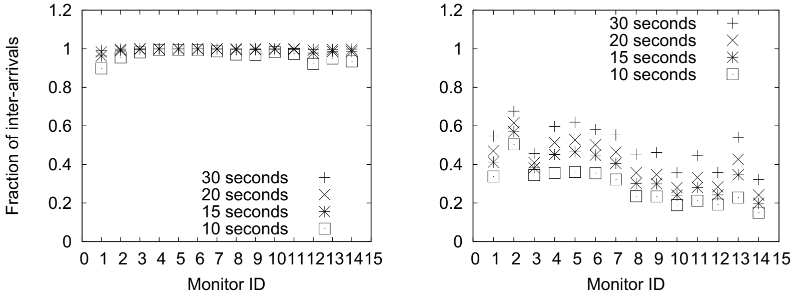
sessions in the Internet[3]. To answer this, we look at update traces from a large number of monitoring sessions operated by the RouteViews project. However, we are only interested in monitoring sessions that are not rate-limited by MRAI or OutDelay, and this information is not available from the RouteViews repository.

To determine whether a monitoring session is rate-limited, we look at the time series of updates for that monitoring session. If a monitor uses MRAI, we expect to see a pattern where updates arrive in bursts every time the timer expires. In other words, we should see very few inter-arrival times in the range [0-22] seconds, assuming a jittered default MRAI timer value. On the other hand, if a monitor uses OutDelay to perform rate-limiting, we do not expect the same bursty pattern of updates. Instead, we should see a pattern where updates arrive in a steady flow, but where two updates for the same prefix are always spaced by at least the OutDelay timer value.

Figure 7.5 shows the fraction of update inter-arrival times for individual prefixes and across all prefixes that is less than 23 seconds, for all 45 monitors that peer with the RouteViews Oregon-IX collector. We have excluded all inter-arrival times of 0 seconds. The figure shows data from the first week of 2008; we have repeated the same exercise for the first week in each year from 2006 to 2009.

We observe a low fraction of inter-arrivals below 23 seconds for both individual prefixes and across all prefixes in first 20 monitors, which indicates that these monitors apply MRAI timers. The next 10 monitors show a low

[3] We only consider eBGP sessions in this work.



(a) MRAI

(b) OutDelay

Figure 7.6: Identification of non-standard timer values

fraction of inter-arrivals below 23 seconds only when looking at each prefix individually, while there is a high fraction when looking at inter-arrivals across all prefixes. This indicates that these monitors perform rate-limiting using OutDelay. The last 15 monitors show a large fraction of inter-arrivals below 23 seconds for both individual prefixes and across all prefixes, indicating no rate-limiting.

So far, our identification of rate-limiting deployment has investigated the use of default timer values. However, some operators may choose to tune down the timer for achieving faster convergence. Such cases result in misidentifying a rate-limited session as non rate-limited. To avoid that, we use our rate-limiting identification method to check, all sessions that are identified as non rate-limited above, for the use of non-standard timer values. Figure 7.6a shows the fraction of update inter-arrivals across all prefixes below four different time thresholds in each monitor that we identify as non rate-limited in the first week of 2008. We observe a high fraction of inter-arrivals below all four time values and across monitors. Figure 7.6b shows the same but for update inter-arrivals of individual prefixes, we also observe a reasonably large fraction of inter-arrivals below all four time values. These observations indicate that none of our monitors configures rate-limiting with non-standard timer values; all these monitors do not deploy any form of rate-limiting. We have not detected a deployment of rate-limiting with non-standard timer values by any monitor in our data set.

The set of non rate-limiting monitors include monitors in tier-1, large re-

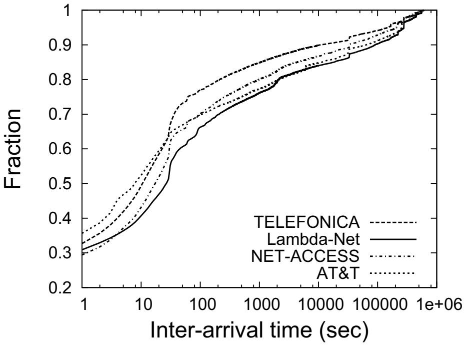


Figure 7.7: Inter-arrival distribution for Routeviews monitors.

gional providers, and stub ASes. Furthermore, we observe that rate-limiting monitors rate-limit both announcements and withdrawals, i.e. WRATE.

For the identified non rate-limiting monitors, we calculate the inter-arrival distribution of updates for each prefix. For each monitor, we look at updates for the first two months in each year from 2006 to 2009. This gives us a set of 48 distributions that is diverse across both time and (topological) space. Figure 7.7 shows the CDF of inter-arrival times for single prefixes[4]. To keep the plot readable, we show results for only a subset of the monitors and only one year. Results of other monitors and years are similar.

The CDF plot shows a clear similarity between the inter-arrival distributions for the different monitors. To confirm this similarity we use the two-samples Kolomgorov-Smirnov (K-S) test, which confirms that all 48 inter-arrival distributions computed from the RouteViews data come from the same population at a confidence level of 95%.

Using non-linear regression on this data, we find that the CDF of the inter-arrival times for individual prefixes is on the form $F(T) = \alpha + \beta ln(T)$ for $1 \leq T \leq 300$. Averaging across the selected RouteViews time series, we find that $\alpha = 0.18$ with a standard deviation $\sigma_\alpha = 0.14$, while $\beta = 0.10$ with a standard deviation $\sigma_\beta = 0.01$. The parameter $\alpha = F(1)$ corresponds to the fraction of inter-arrival times that is less than or equal to 1 second. This parameter shows quite large variation across the time series. Looking closer at the data, we observe that monitors in stub networks show smaller α values, while monitors in large tier-1 ISPs show larger α values. This indicates that

[4] Since we are mainly interested in inter-arrival times in the order of a rate-limiting timer, we have imposed a maximum inter-arrival time of 1 week.

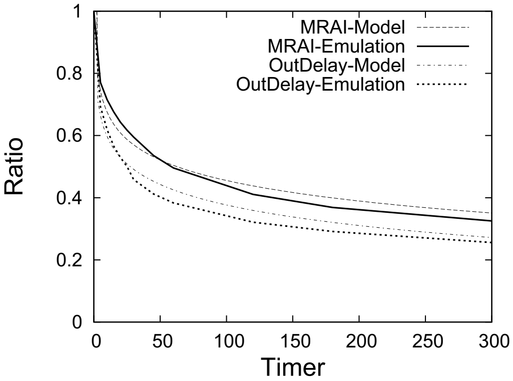


Figure 7.8: Reduction in churn: model vs data

Monitor	M_{OD30}	$Model$
A	22.99	22.71
B	18.54	16.91
C	15.22	16.50

Table 7.4: Total number of updates (million), model vs data

the value of α might be depending on path diversity and the connectivity at the monitor AS. Looking closer at this is left for future work.

Returning to our original goal, we can now give a formulation for the expected churn level as a function of the rate-limiting timer, based on the empirical data from the RouteViews monitors. Substituting $F(T)$ in (7.1) and (7.2), we get

$$R(T)_{OutDelay} = 0.82 - 0.10ln(T) \quad 1 \leq T \leq 300 \tag{7.3}$$

$$R(T)_{MRAI} = 0.90 - 0.10ln(T) \quad 1 \leq T \leq 300 \tag{7.4}$$

Figure 7.8 shows our model for $R(T)$ along with the churn reduction from our emulated rate-limiting in Sec. 7.3. Recall that using regression, we estimated $R(T)_{MRAI} = 0.98 - 0.17ln(T)$ and $R(T)_{OutDelay} = 0.86 - 0.11 * ln(T)$ based on our measurement data in that section. Table 7.4 shows the total number of updates measured in M_{OD30} during our study period along with numbers approximated using our model. The model and the emulated data are in a good accordance; the model is able to predict the reduction in churn within a few percent for our dataset.

7.5 Summary

This chapter has explored how different BGP rate-limiting implementations affect the level of churn. Measurements were performed on two parallel BGP sessions, with and without rate-limiting respectively, to three routers in a stub AS. Our measurements have shown that the sustained level of churn is strongly reduced when enabling the rate-limiting timer, and we have explained how the OutDelay implementation (used in Juniper routers) gives a stronger reduction than MRAI timers (used in Cisco routers). Using emulation on the measured churn time series, we have shown that *the reduction is significant for both implementations already at low timer values* (which keep the convergence delay acceptable). With an OutDelay of 30 seconds, churn is reduced by as much as two thirds.

Using data from a large number of RouteViews monitors, we have investigated the update inter-arrival pattern in BGP sessions that are not rate limited. We have found a strong similarity in the distribution of inter-arrival times across monitoring sessions in different parts of the Internet, and across different years. This observed universality allows us to formulate an expression that quantifies the expected reduction in churn levels for different rate-limiting implementations and timer values. This expression is able to predict the churn reduction observed in our measurements within a few percent.

The expression for churn reduction given in this work will be useful for network operators for deciding a rate-limiting configuration that gives the right balance between churn reduction and convergence times.

In conclusion, this part of the thesis has studied the impact of topology growth and update rate-limiting on BGP scalability. We have identified several topological factors that influence the amount of routing updates due failures at the periphery of the network. In addition, we have quantified the relation between topological properties and the scope of a transit link failure. We have also investigated how different update rate-limiting implementations and configurations influence churn. All aforementioned findings shed light on the complex nature of BGP dynamics and arguably a first systemic step towards improving our understanding of churn. The next part of this thesis, assesses the extent of the scalability challenge that today's global routing system is facing, by studying the evolution of churn at the core of the Internet over the past seven years.

Part III

BGP Scalability: Measuring Churn Evolution

Chapter 8

Churn Evolution: a Perspective from the Core

So far we have investigated the impact of topology growth and update rate-limiting on BGP churn. Besides understanding the impact of different factors on churn, it is also important to assess how churn has evolved in the Internet during the past few years. That will help in assessing the severity of the scalability challenge faced by the global routing system.

An earlier study by Huston and Armitage reported an alarming growth in churn [HA06]. During 2005, the daily rate of BGP updates observed by a router in AS1221 (Telstra) almost doubled, while the number of prefixes grew by only 18%. Based on these measurements, the authors projected future churn levels and concluded that current router hardware will need significant upgrades in order to cope with churn in a 3-5 years perspective. It was this study that largely motivated the work in this chapter.

Specifically, in this chapter we present a longitudinal study of BGP churn spanning a longer time frame (more than 7 years) and more monitors (routers in 4 tier-1 ISPs) than previous studies. Our goal is to understand the different components that constitute churn time series. Furthermore, we want to identify long tern trends after filtering pathological updates and effects that are local to the monitored networks.

Generally, the churn time series is very noisy, dominated by frequent large spikes, and "level shifts" that last for several weeks or even months. There are periods in which churn is slowly increasing, others in which it is decreasing, and major differences between monitors. One option could be to characterize the evolution of churn using "black-box" statistical or time series analysis methods. That approach would answer questions about the correlation structure and the marginal distribution of the underlying time series, attempting to fit the data in a standard time series model (e.g., ARIMA [Cha96]). That descriptive method, however, would not be able to *explain* what causes spikes, level shifts or trends in BGP churn.

We prefer, instead, to take Tukey's exploratory data analysis approach [Tuk77] that focuses on the *causes behind the observed phenomena.* The approach we take is to first analyze what causes some major characteristics of the raw time series (spikes, level shifts, etc) and then, after we remove pathologies or effects that are not related to the long term evolution of churn, to apply statistical trend estimation on the remaining "baseline" churn. Through our analysis we are able to identify and isolate the main reasons behind many of the anomalies in the churn time series. We find that duplicate announcements is a major churn contributor, and responsible for most large spikes in the churn time series. Other intense periods of churn are caused by misconfigurations or other special events in or close to the monitored AS, and hence limiting these is an important mean to limit churn. We then analyze the remaining "baseline" churn, and find that it is increasing with a rate much slower than the increase in the routing table size.

8.1 Dataset

Our analysis is based on BGP update traces collected by the Routeviews project [rou]. Routeviews collectors run BGP sessions with several routers, referred to as *monitors*, in many networks. A monitor sends a BGP update to the collector every time there is a change in the preferred path from the monitor to a destination prefix. In addition, Routeviews dumps every two hours a snapshot of the routing table that contains the best selected paths of advertised prefixes from each monitor. We use those snapshots to observe the growth of the routing table size over the last few years.

We focus on update traces from monitors at large transit networks in the core of the Internet. Specifically, we analyze the churn time series from four monitors at AT&T, Sprint, Level-3 and France Telecom (FT). The corresponding monitors belong to the DFZ, meaning that they do not have a default route to another provider, and so they know a route to practically all destination networks in the Internet.

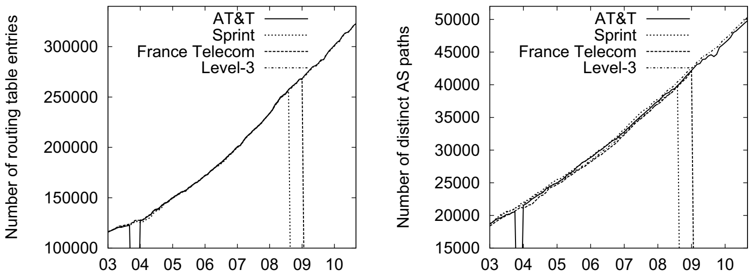
Routeviews provides historical update traces spanning more than seven years for these four monitors. In some cases, the IP address of the monitor had changed during our study period. We identified the corresponding IP addresses and concatenated the update time series after confirming that they correspond to the same actual monitor. Our time series cover the period from January 1 2003 to August 31 2010, giving us more than 7.5 years worth of routing updates from four backbone monitors. However, the Sprint monitor was unavailable during the last two years of our study period, while the FT monitor was unavailable after February 2009. Furthermore, the AT&T monitor was unavailable during 2.5 months in late 2003.

We use the method described in [ZKL+05] to identify and remove updates that are part of table transfers between our four monitors and the Routeviews collector. After such filtering, our dataset consists of more than 1.8 billion updates. Note that the updates received from a monitor is not a direct estimate for the total number of updates a backbone router must process. A router typically has several active BGP sessions, and so the total load on the router is the sum of the churn from all BGP sessions.

Differences within an AS. Due to complex iBGP configurations using confederations or route reflectors, different edge routers of the same AS do not necessarily see the same set of paths to different destinations. In order to to understand the impact of such differences on churn we obtained a set of ten pairs of monitors. Members of each pair are routers of the same AS, which were peering with the RouteViews Oregon-IX collector simultaneously. Then, we measured the hourly churn time series in four different months (Aug'03, Mar'04, May'05, and Feb'07) for all monitors in the selected set. We calculated the cross-correlation between the hourly number of updates in each pair time series. Since churn time series involve spikes and non-stationary periods, we chose two non-parametric measures for estimating the cross-correlation: Kendall's tau [Ken38] and Spearman's rho [Spe04]. Both measures indicated a reasonably high cross-correlation between monitors from the same AS. Table 8.1 shows the minimum, median, and maximum measured cross-correlation across all the ten pairs. Note that a value of 1.0 denotes a perfect correlation, while a value of -1.0 denotes a perfect anticorrelation. Even though we can not claim that these observations are true in general, it is reasonable to expect that two routers of the same AS would produce similar (but not identical) churn.

	Minimum	Median	Maximum
Kendall's tau	0.70	0.80	0.93
Spearman's rho	0.84	0.92	0.98

Table 8.1: Measured cross-correlation.



(a) Routing table size

(b) Observed AS paths

Figure 8.1: Global routing growth

8.2 The "Raw" Churn Time Series

Before we focus on the churn time series, we first show two important aspects of growth in the BGP routing system. Figure 8.1a shows the number of routing table entries in the four monitors, sampled on a monthly basis. The number of entries in the different monitors is very similar, which is expected since these monitors are all DFZ routers. The *number of routable prefixes increased by 168% during our study period*, from about 120K to 322K entries. The increase in the table size fits well with a quadratic function, with a coefficient of determination of 99.9%. Figure. 8.1b shows the number of distinct AS paths (routing paths) in the routing tables (after removing the effects of AS path-prepending) again on a monthly basis. This metric has also increased dramatically (163%) during our measurement period, from 19K to 50K paths.

One may expect that since the size of the routing table and the number of routing paths have more than doubled during the studied seven years, BGP churn should also show a similar consistent and significant increase. This is not the case however. The left column in Fig. 8.2 shows the "raw"

BGP churn time series, measured as the number of BGP updates received daily from each monitor. Some high-level observations are necessary before we proceed with the analysis.

The raw time series is dominated by frequent and large spikes. At all monitors, there are days with dramatically higher churn than usual. We have truncated the y-axis of these plots to make the graphs more readable; in some days, the number of updates reached several millions.

Large spikes are particularly frequent in the Level-3 monitor. Such spikes cannot be ignored as "statistical outliers"; instead, we need to understand what causes them.

There are several "level shifts". In addition to spikes, we see several periods of sustained increased activity that last for weeks or months. For example, we see a period that lasted about 6 months in mid-2006 at the Level-3 monitor. Again, level shifts cannot be viewed just as incidents of statistical non-stationarity; we need to understand what causes them.

There is little correlation between monitors of different ASes. The spikes and level shifts at the four monitors do not follow the same pattern. We measured the cross-correlation between the different monitors using the Kendall's tau coefficient, since it is less affected by spikes and non-stationarity. The estimated cross-correlation coefficient is between 0.17 and 0.3, which illustrates a small correlation between the four monitors. This indicates that churn is highly dependent on the location and configuration of the corresponding router. We cannot understand the evolution of BGP churn by just looking at one monitor.

Churn is highly bursty even at large time scales. As seen in the left column of Fig. 8.2, churn is highly bursty even in the relatively large time scale of a day. We also examined the churn time series in shorter time scales (5 minutes and one hour) and observed that in some cases the majority of the daily churn is produced during short periods that last for few minutes.

It can be misleading to infer long-term trends from the raw churn time series. Because of the previous issues, it is clear that the blind application of statistical trend estimation methods can fail to detect a trend or it can produce misleading results. The approach we take is to first analyze what causes some major characteristics of the raw time series (spikes, level shifts, etc) and then, after we remove pathologies or effects that are not related to the long term evolution of churn, to apply statistical trend estimation on the remaining "baseline" churn.

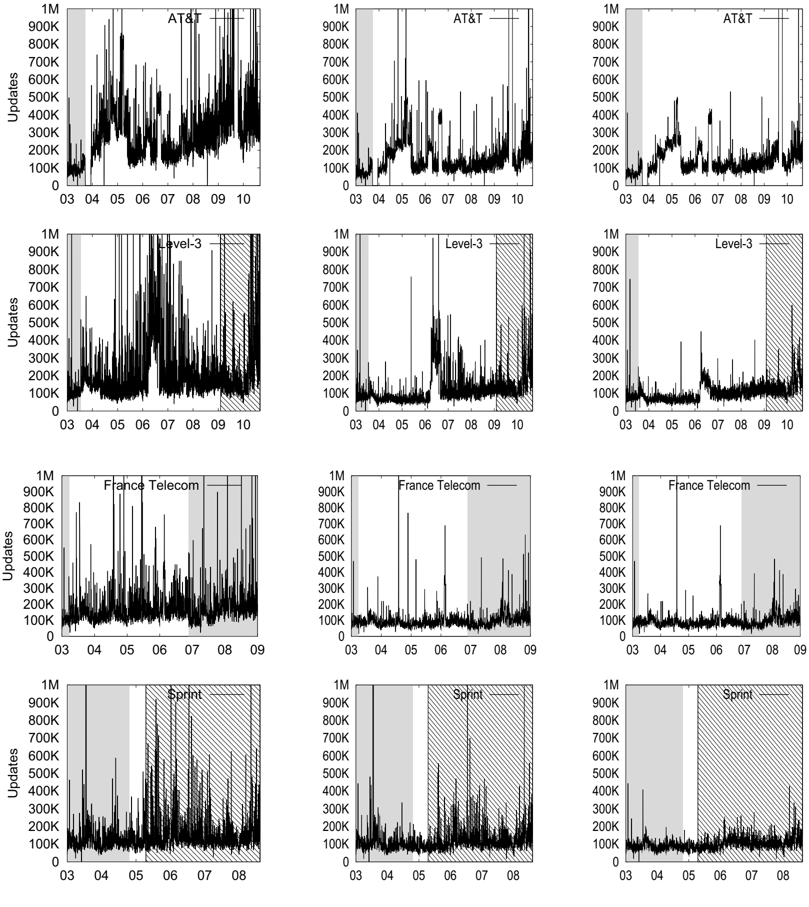


Figure 8.2: Daily BGP churn: raw time series (left), after removing duplicates (middle), after removing duplicates and large events (right) at our four monitors. Grey shaded areas indicate periods in which the MRAI timer was deployed, and diagonally shaded areas to indicate the same for the out-delay timer.

8.3 Rate-Limiting Timer Deployment and Impact

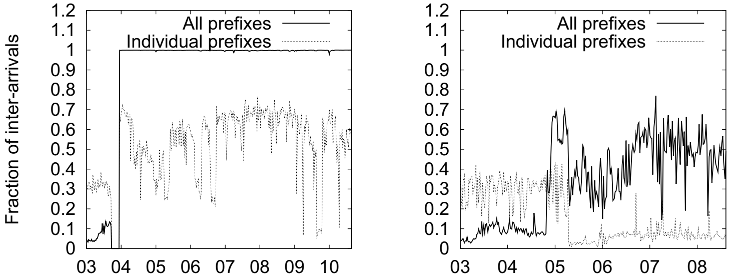
In this section, we analyze the deployment of the rate-limiting timer by the four monitors during our study period. Our goal is to assess the impact of rate-limiting configuration on the observed churn. Further, by doing this we will be able to understand any sudden changes in churn level when the timer is toggled.

As explained in the previous chapter, BGP uses a rate-limiting timer to filter intermediate routing states. The BGP standard [RLH06] recommends the use of the MRAI timer with a default value of 30 seconds on eBGP sessions. It specifies the minimum time interval between two consecutive updates for a destination prefix. However, some implementations (e.g. Cisco IOS) implement a per-session timer rather than per-prefix to reduce overhead. Another common BGP implementation (Juniper's JunOS) implements rate-limiting using the *out-delay* parameter. Unlike the MRAI timer implementation described above, this delay is added to each update for each prefix individually.

To determine whether a monitoring session is rate-limited, we look at the time series of updates for that monitoring session. If a monitor uses MRAI, we expect to see a pattern where updates arrive in bursts every time the timer expires. In other words, we should see very few inter-arrival times in the range [0-22] seconds, assuming a jittered default MRAI timer value. On the other hand, if a monitor uses *out-delay* to perform rate-limiting, we do not expect the same bursty pattern of updates. Instead, we should see a pattern where updates arrive in a steady flow, but where two updates for the same prefix are always spaced by at least the *out-delay* timer value (i.e. 30 seconds). So, to detect whether a rate-limiting timer was deployed at the four monitors during the study period, we used the following two-step approach.

1. For each monitor we selected one day from each week of the study period, which resulted in a sample of 288 days per monitor. We then calculated the distribution of update inter-arrival times for each day in the sample.

2. We calculated the fraction of update inter-arrival times across all prefixes (IAT-update) that is less than 22.5 seconds. If that fraction is significant, MRAI was probably *not* deployed on the corresponding day. Furthermore, we calculated the fraction of update inter-arrival times for individual prefixes (IAT-prefix) that is less than 30 seconds. If that fraction is significant, *out-*



(a) AT&T

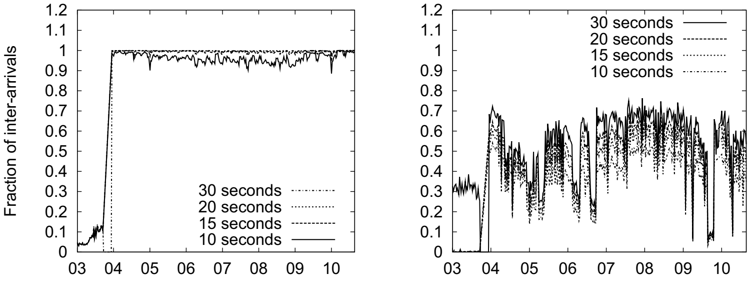
(b) Sprint

Figure 8.3: Identification of rate-limited periods

delay was probably *not* deployed on the corresponding day. We find that setting the threshold for the fraction of both inter-arrivals anywhere between 0.15 and 0.2 results in detecting the same periods in each time series when rate-limiting was set by the respective operator. We further employ our two-step approach to investigate for the deployment of rate-limiting with non-standard timer values. In this case, we check for the fraction of inter-arrivals below values that are smaller than 22.5 seconds.

Figure. 8.3 shows the fraction of IAT-prefix and IAT-update that is less than 30 and 22.5 seconds respectively, for AT&T and Sprint monitors. In AT&T the fraction of IAT-update started at about 5% during the first nine months of 2003, then it increased steeply to about 99% in the rest of the study period. The fraction of IAT-prefix remained relatively large during the study period. In Sprint the fraction of IAT-update was about 10% in the period between Jan'03 to Oct'04, and after Oct'04 it increased to about 40%. The fraction of IAT-prefix was about 30% between Jan'03 and Apr'05, then it decreased steeply to less than 10% in the rest of our study period.

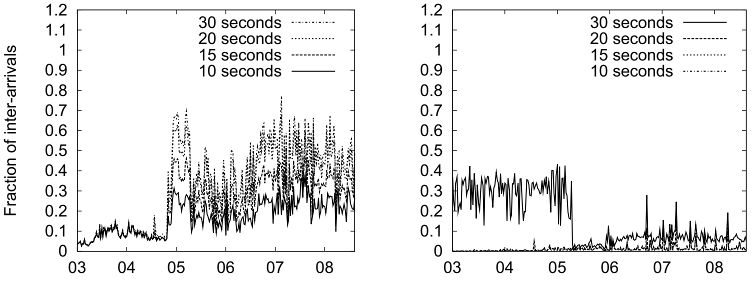
The above observations suggest that the MRAI timer at the AT&T monitor was active initially, it was then turned off. In addition, the observations indicate that the MRAI timer at the Sprint monitor was active between Jan'03 to Oct'04, it was then turned off for six months, and then *out-delay* was used from Apr'05 until the end of our study period. Switching between MRAI and *out-delay* suggests that the router hardware of the Sprint monitor was replaced. To confirm the correctness of the aforementioned rate-limiting



(a) IAT-update

(b) IAT-prefix

Figure 8.4: Non-standard timer values, AT&T



(a) IAT-update

(b) IAT-prefix

Figure 8.5: Non-standard timer values, Sprint

inference we investigated the deployment of rate-limiting with smaller timer values. Figure 8.4a shows the fraction of IAT-update that is less than four different MRAI values (30,20,15, and 10 seconds) for AT&T. We also depict in Fig. 8.4b the fraction of IAT-prefix in AT&T below the same values. Both figures show a large fraction of inter-arrivals below all investigated timer values. Hence, there is no deployment of rate-limiting with non-standard timer value by the AT&T monitor. The plots in Fig. 8.5 show the same for Sprint.

Using a similar analysis we inferred the rate-limiting deployment periods for the two other monitors. At the start of the study period, MRAI was deployed by both monitors. It was then switched off at a different time for each monitor. However, the rate-limiting was turned on again at both monitors. At the France-telecoms monitor, the MRAI timer was enabled again after Nov'06. At the Level-3 monitor, we observed a deployment of *out-delay* after Feb'09. In Figs. 8.2 and 8.13, we use grey shaded areas in the time series to indicate periods in which the MRAI timer was deployed, and diagonally shaded areas to indicate the same for the out-delay timer.

To investigate the impact of the rate-limiting timer, we measured the median daily churn rate in a three month period before and after each identified deployment transition. For each transition, we calculated the ratio (median churn without rate-limiting)/(median churn with rate-limiting). We mostly found that the churn level increases when the rate-limiting timer is turned off, while it decreases when it is turned on. For example at Level-3, churn increased by a factor of 1.8 when the timer was turned off. The reciprocal of this value (0.56) quantifies the extent of churn reduction when using rate-limiting. These observations are in accord with our measurements in chapter 7. Table 7.3 shows the extent of churn reduction we measured when deploying MRAI. The reported values in that table are close to the measured reduction in the Level-3 case above. We believe that this similarity stems from the observed universality, we report in chapter 7, in the distribution of inter-arrival times across monitoring sessions in different parts of the Internet. In the next section, we examine the frequency of duplicate BGP updates in the churn time series.

8.4 Duplicate Updates

The conventional wisdom is that BGP implementations generate a large number of duplicate updates, which imposes an unnecessary processing load on routers. It has been pointed out that one reason for the large number of redundant updates is stateless BGP implementations that do not keep track of the last update sent to a peer [LMJ97].

We identified all duplicate updates (announcements and withdrawals) in our dataset. By "duplicate announcement", we mean an announcement that is identical to the last seen announcement for the same prefix, i.e., no change in either the AS-path or in any of the transitive route attributes. These announcements are redundant and can be viewed as a pathology of the BGP implementation at the corresponding monitor. A recent measurement study [PJL$^+$10] attributed much of the observed duplicates in BGP churn to interactions between iBGP and eBGP.

To our great surprise, we measured that, across all four monitors, duplicate announcements are responsible for about 40% of the churn during the study period! On the other hand, duplicate withdrawals are close to zero (except Level-3, where they account for about 1% of the updates). It is disappointing that almost half of the observed churn is not really necessary for correct protocol behavior. This number is higher than the 16% of the duplicate announcements "AADupType1" reported earlier [LGW$^+$07]; that study looked at monitors located in ASes of different sizes during a 6-month period in 2006. Furthermore, this number is also higher than what is reported in [PJL$^+$10].

The number of duplicate updates per day is highly variable, and shows no correlation across monitors. It is also difficult to identify a consistent long-term trend in the number of duplicates. Table 8.2 shows the fraction of duplicate announcements per year in each monitor. It is worth noting that the Sprint monitor shows a much lower fraction of duplicates than the other three monitors. These results indicate that the specific implementation of BGP and local configuration details can greatly influence the amount of redundant updates.

Our findings indicate that there is still much to be gained by deploying improved BGP implementations that avoid sending redundant updates to the global routing system. Such improvements would require, however, per-neighbor state at BGP routers to keep track of what was sent to each peer earlier, so that duplicate updates can be detected before they are transmitted. This shows that there is a trade-off between allowing the generation of duplicate updates (that will be filtered at the receiving router) versus more heavy weight processing at the sending router that would also eliminate duplicate updates [RRG10]. Arguably, these changes are not worth doing, given the lightweight handling of duplicates.

The second column in Fig. 8.2 shows the four time series after filtering out duplicate updates[1]. Note that removing duplicate updates has the additional benefit that most of the spikes are also removed. This indicates that redundant updates are not only responsible for a large fraction of churn, but they are also responsible for generating large bursts of churn that may put the highest burden on router CPUs.

[1] Raw and filtered datasets are available at http://vefur.simula.no/bgp-churn/.

Monitor	AT&T	Level-3	FT	Sprint
2003	23.7%	40.7%	33.0%	7.2%
2004	47.6%	53.8%	45.2%	23.5%
2005	34.8%	61.7%	52.0%	41.1%
2006	31.8%	46.1%	43.5%	17.7%
2007	52.6%	42.3%	50.0%	14.3%
2008	59.6%	32.4%	43.2%	12.9%
2009	39.7%	22.6%	-	-
2010	20.2%	34.9%	-	-

Table 8.2: Fraction of duplicate updates per monitor.

8.5 Large Events

After removing duplicates, we focus on "large routing events", or simply *large events*, loosely defined as events that affect a large number of prefixes at about the same time. The intuition is that incidents in the core of the Internet, such as link failures between two transit networks or inside a transit network, have the potential to introduce instability to a large number of prefixes simultaneously, causing major churn spikes. In addition, large events can be triggered by internal routing and policy changes. For instance, changes in IGP's link weights in an AS, may result in changing the next hop address of a large number of prefixes (i.e. hot-potato routing). Another example, an AS may tag routes received at different POPs using community attributes, in such settings a large number of updates can potentially be sent if these routes switch their exit POP. Large events can potentially impose a high burden on a router's CPU, because they affect a large number of prefixes simultaneously. Its important to characterize large events in order to understand the extent and evolution of their impact.

When an underlying incident triggers a routing change, it often results in several updates for each affected prefix. Based on this we define a *prefix event* to be a sequence of updates for a given prefix that are likely generated by the same underlying incident. The updates of a prefix event typically have short inter-arrivals. Here, we adopt the definition given in [WMRW05] for identifying prefix events:

Definition 1. *Two consecutive updates for the same prefix belong to the same* prefix event *if they are no more than 70 seconds apart. The maximum* *duration for a prefix event is set to 10 minutes. Events with duration longer than 10 minutes are considered to be flapping.*

The previous thresholds were determined based on measuring the convergence times for the beacon prefixes [MBGR03]. The authors in [WMRW05] showed that over 98% of the updates inter-arrival times are less than 70 seconds across all prefixes; confirming the robustness of the 70 seconds thresholds to erroneously grouping updates that are triggered by different routing events or splitting those belong to the same event.

Some routing incidents affect several prefixes. We group prefix events that occur at about the same time into *events*.

Definition 2. *Starting with a prefix event p, the event that follows p consists of all prefix events that start no later than 5 seconds after the start of p.*

The intuition is that when a routing incident affects multiple prefixes, the first updates for these prefixes should arrive in a burst. With this low threshold of 5 seconds, we attempt to avoid the risk of erroneously grouping prefix events that are caused by different underlying incidents into the same event.

To check the robustness of the event grouping threshold we investigated how events size distribution changes (i.e. the number of grouped prefixes) as we vary the grouping threshold. Figure 8.6 shows the 99th percentile for event size as we change the grouping threshold in the range between 1 second and 30 seconds. We identified and grouped events that occur in three different weeks during our study period (the first week of Feb'03,Feb'05, and Feb'08) in all four monitors. Our objective is not to analyze differences between monitors or across time, but to check the robustness of the event grouping threshold. We observe that changing the grouping threshold between 1 and 20 seconds results in a little or no difference. This confirms the robustness of our threshold.

Next, we define a *large event* as an event that affects many prefixes. To choose an appropriate threshold for classifying an event as a large event, we identified all events that took place during the month of January in each year of our study period, for all four monitors. Fig. 8.7 shows the distribution of the number of affected prefixes per event - each curve represents the events during the period of one month and for one monitor. Our objective is not to analyze the differences between monitors or months, but to observe the

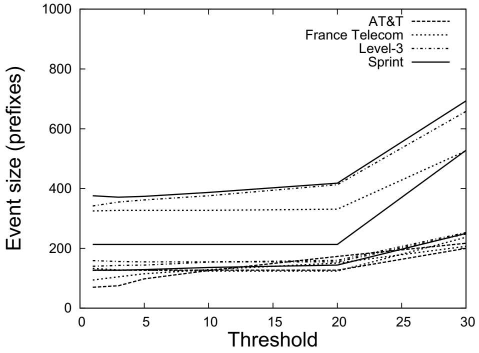


Figure 8.6: Event grouping sensitivity.

"typical" distribution of event sizes. We show only the tail of the distribution - the full CDF shows that half of all events affect less than 10 prefixes, while more than 90% of events affect less than 40 prefixes. Based on this graph, we decided to use a threshold of 2000 prefixes. Note that all CDFs flatten out after the selected threshold. With this definition, (at most) the top 0.2% of all events are considered to be large events

Definition 3. *A large event is an event that includes at least 2000 prefix events.*

The number of large events over our study period varies significantly across monitors between 1554 (France-Telecoms) and 15054 (AT&T). Note that 12265 of the 15054 large events at the AT&T monitor were observed during a period of two days between June-29-2010 and July-01-2010. These events resulted in over 156 million updates. A closer look indicated that this high activity is caused by continuous flapping of about 217K prefixes that were originated from 25000 different ASes. The flapping involved 1937 different next hops. The involvement of a large number of prefixes and the high diversity in terms of next hops and origins show that the cause of these events is likely to be local to the monitor. In the rest of this section, we focus on the remaining 2789 large events in the AT&T time series.

Next, we characterize large events with respect to their type; and the evolution of their size, frequency, and duration. Furthermore, we investigate

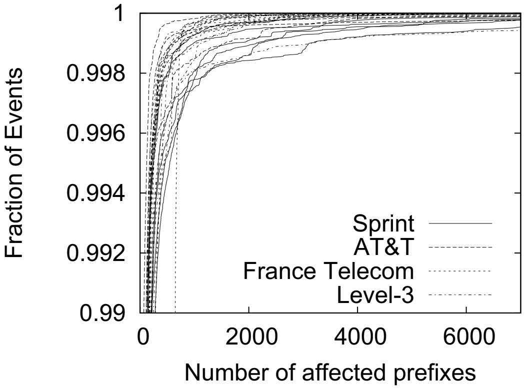


Figure 8.7: Distribution of the number of affected prefixes per event.

large events' correlation across monitors.

8.5.1 Types of large events

Given that the number of large events is significant, it is difficult to examine them individually in order to pin-point their root causes. Instead, we categorize large events based on the most dominant routing change (i.e. path instabilities, path changes, or path withdrawals) in the underlying single prefix events that constitute them.

We classify a single-prefix event into different types, depending on the *best known path before the event*, and the *best known path after the event* [LMJ99, OZP$^+$06]:

WA: Starts with no known path to the prefix, and ends up with a path.

AW: Starts with a path and ends with no path.

AAC: Starts with a path P and ends with a different path P'.

AAD: Starts with a path P and ends with the same path P, but at least one different path P' is seen during the convergence process.

AAS: Starts with a path P and ends with the same path P, and no other path is seen during the convergence process.

If a large event was caused by a certain routing incident, the majority of involved prefixes would likely show a similar change signature. To investigate that we group single-prefix events that constitute a large event based on the change signature they show. Further we identify the largest group and the corresponding change signature (the dominant event signature type). Figure 8.8 shows the CDF of the percentage of the largest single-prefix events group that have an identical event signature in a large event. The plot confirms that our large events grouping mostly groups prefix events that are characterized by a similar change signature. In 99% of the large events at least 50% of the underlying single prefix events have a similar event signature.

To make our analysis easier, we proceed to classify large events based on the dominant signature of the corresponding prefix events. We say that a large event is of type A if at least 70% of the involved single prefix events are of type A. As seen in Tab. 8.3, most large events can be assigned to one of the event classes using this definition. We observe that the dominant classes differ from one monitor to another. AAC, AW, and WA are dominant at AT&T. At Sprint, the majority of large events are of AAS type. AAD, AAC, and AAS dominate Level-3, while the majority are of AAC and AAD types at FT.

Events that involve a change or a disturbance of reachability (i.e. AAC, AAD, AW, and WA) can be triggered by a variety of causes such as failures, restorations, and policy changes. One possible approach to further characterize large events would be to employ root cause analysis techniques [CSK03, FMM^{+}04] to pinpoint the origins of events. There are, however, several challenges involved in this type of analysis, and it would be very difficult to perform with the limited data we have available. Hence, we refrain from such approach. We observe, however, that large events of type AAS, which dominate Level-3 and Sprint, often communicate a change in the COMMUNITY attribute. The communities that changed are geographical communities which describe a route exit point within the monitored network. Likely causes behind such events could be a sort of traffic engineering controller that continuously alternate routes exit points.

Generally speaking, large events reflect genuine routing activity, i.e. failures and restorations of links and policy changes. However, some of them can

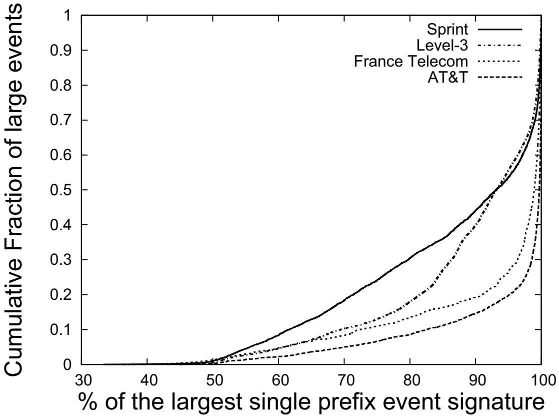


Figure 8.8: Classifying large events.

Monitor	AT&T	Level-3	FT	Sprint
AW	21.1%	1.8%	4.4%	1.6%
WA	21.7%	1.6%	4.1%	1.6%
AAC	46.1%	23.9%	66.9%	6.4%
AAD	8.9	32.8%	15.6%	1.5%
AAS	0.7%	36.6%	0.3%	70.1%
ND	1.2%	3.1%	8.8%	18.7%

Table 8.3: Large events classification.

be limited or avoided. For example, updates that communicate a change in the geographical communities can be limited to neighbors that are interested in such changes.

8.5.2 Correlation of Large Events

In this subsection we investigate how large events are related across monitors from two different angels. First, we examine whether large events tend to happen at the same time at different monitors. Second, we study the propagation of large events; if a large event is observed at a monitor M, what is its impact on other monitors?

Large events across monitors. To calculate the cross correlation of large

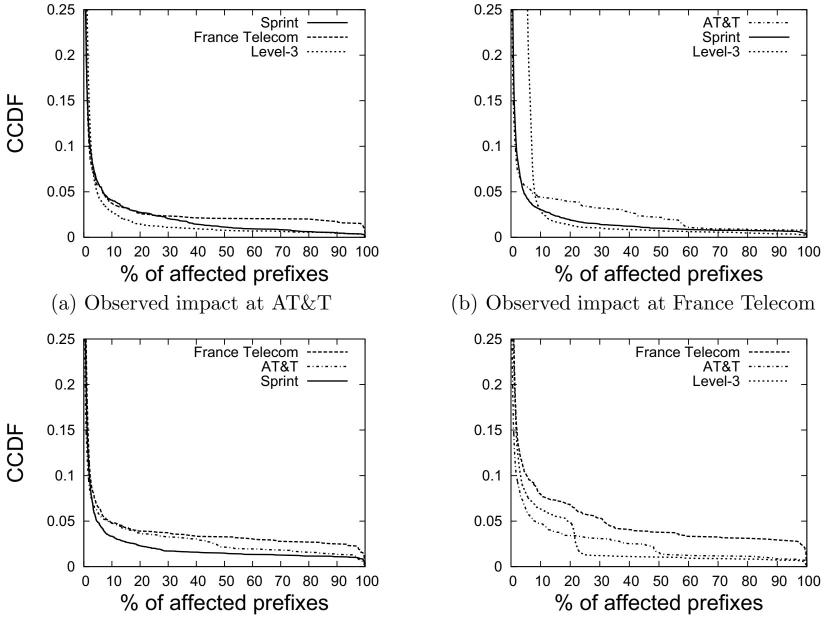
Monitor	AT&T	FT	Level-3	Sprint
AT&T	-	0.05	0.04	0.06
FT	0.05	-	0.03	0.05
Level-3	0.04	0.03	-	0.03
Sprint	0.06	0.05	0.03	-

Table 8.4: Large events cross-correlation

events' time series across monitors, we divide each time series into bins of 10 minutes. Then we construct a new binary time series such that a bin will be assigned a value of one if we record at least one large event during the time covered by that bin, and zero otherwise. Further, we correlate the resulting binary time series between all pairs of monitors at lags 1,0, and -1. Our approach avoids misidentifying correlated large events by using a large bin size and correlating at three different lags. Table 8.4 shows the calculated cross-correlation coefficients between every pair of monitors at lag zero. We observe a negligible correlation between monitors, indicating that large events are mostly independent. All cross-correlation coefficients values at lag 1 and -1 are smaller than at lag zero.

The propagation of large events. Our goal here is to investigate whether a large event at monitor M_1 is visible (perhaps not as large event) at monitor M_2. For doing that, we start by identifying prefixes that are affected by a large event at monitor M_1. For each such prefix we check if it was active within a time window of width W in the update traces of monitor M_2. Then, we calculate the fraction of prefixes in a large event that shows such temporal correlation. This analysis is performed for all observed large events and between all pairs of monitors. We try three different correlation window's sizes (1 minute, 5 minutes, and 10 minutes). Increasing the correlation window size from 5 to 10 does not affect the results significantly. In the following we report results that are based on a window size of 5 minutes.

The plots in Fig. 8.9 depict the CCDF of affected prefixes per large event at our four monitors; we only show the tail of the distribution. For every monitor there is one plot that includes three curves; each curve corresponds to the correlation with events taking place at one of the other three monitors. We observe that large events that happen at a monitor M have little impact on other monitors. Generally, the percentage of correlated prefixes is between 1% and 2% in 90% of the cases. Furthermore, the extent of correlations varies



(a) Observed impact at AT&T

(b) Observed impact at France Telecom

(c) Observed impact at Level-3

(d) Observed impact at Sprint

Figure 8.9: The propagation of large events

between different pairs of monitors; for example, Sprint shows a relatively high number of active prefixes after large events observed in FT.

To summarize, the time series of large events show little or no correlation between different monitors. In addition, large events that are observed at one monitor have mostly negligible impact on other monitors. Therefore, the number and magnitude of observed large events are highly dependent on the monitoring point.

8.5.3 Evolution of Large Events

Next, we turn to exploring the evolution of large events and their characteristics. More specifically, we investigate how the size, intensity, duration, and frequency of large events have changed over time.

Large events' size. The evolution of large events' size in terms of involved prefixes is an important measure. This metric is a reasonable estimate of expected load, imposed by large events, on routers' CPUs. For estimating this measure we group large events into consecutive windows of length three months, and then report the 90th percentile of the large events' size within each window. Figures. 8.10a and 8.10b show the evolution of large events' size at all four monitors (the graphs are split into two figures for readability). Using the Mann-Kendall statistical test for trend detection, we observe no trends at Sprint and AT&T time series and an increasing trend at both FT, and Level-3. The 90th percentile large event's size at FT has increased by 11% during our study period. The increase at Level-3 is much higher at 121%, but starting from a much smaller absolute size. In addition, to the measured trends, we note the presence of periods with a sharp increase in the 90th percentile large event's size. These periods match periods with high level of updates in the measured churn time series.

Contributed churn and frequency. Figures. 8.10c and 8.10d illustrate the evolution of the frequency of large events and their contributed churn at AT&T and Level-3; these figures report the monthly rather than the daily aggregates. We choose one month granularity due to the sparseness of the large events time series [2]. Note that both plots show large variations (up to two orders of magnitude). We have not detected any trend in both measures at AT&T and FT. However, we identify an increasing trend at Level-3. The monthly number of large events and their contributed churn have increased during our study period by 160% and 282% respectively. We also observe a decrease by 96% in the monthly number of large events at Sprint, however, we don't identify any trend in the monthly contributed churn. The absence of a trend in churn contributed by large events at AT&T and FT implies that the number of updates per a prefix in large event has remained stable over time. The median number of updates per prefix in a large event has stayed roughly constant at AT&T, FT, and Level-3 between 1.2 and 1.5 depending on the monitor. However, we record a 45% increase at Sprint from about 1.1 updates in 2003 to 1.6 updates in 2008. This increase at Sprint doesn't affect the observed monthly churn because it is offset by the decrease in the number of large events per month.

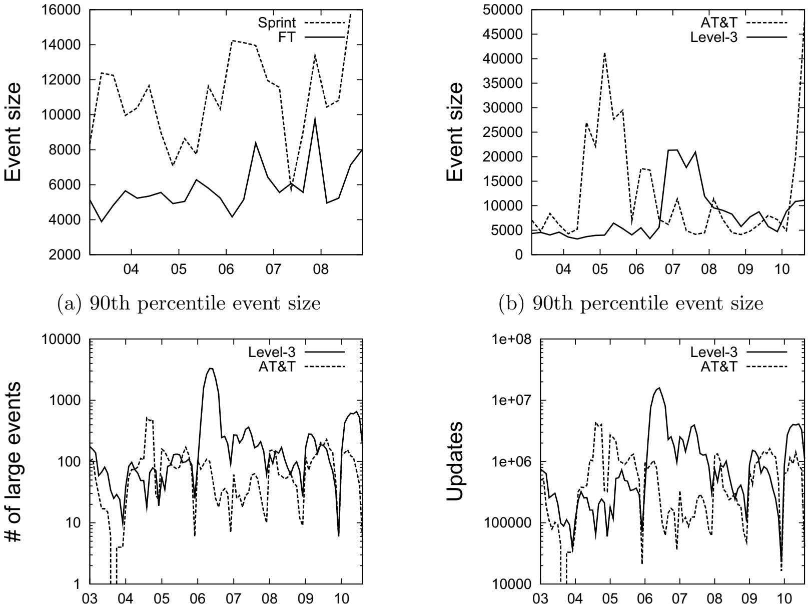
Duration. Another important characteristic of a large event is its duration, we define a large event duration as the time difference between the first and

[2] It is typical not to observe large events on several days in each month.

last update in that large event. The median of this duration has remained stable between 50 and 60 seconds at AT&T and Level-3. We also observe an increasing trend at FT and Sprint. The median event duration has increased from 65 seconds in early 2003 to 85 seconds by the end of 2008 at FT; and from 64 seconds to 76 seconds at Sprint in the same period. The duration of an event is related to the number of updates per affected prefix, and this number has stayed roughly constant as discussed above. Furthermore, the observed increasing trend at FT and Sprint is likely caused by their deployment of rate-limiting timers for a relatively long duration.

Inter-arrival patterns. Another interesting aspect, with regard to the frequency of large events, is the distribution of their inter-arrival times. This is particularly important, since it translates into the spacing in time between instants where we expect a high stress on routers. For estimating this measure we group large events into consecutive windows of length three months, and then calculate the median inter-arrival time within each window. Figure. 8.11a depicts the evolution of the median inter-arrival time of large events at AT&T and Level-3. AT&T demonstrates lower inter-arrival times in orders of seconds and tens of seconds. We also observe that the mean inter-arrival time at Level-3 has remained stable around 1000 seconds. The evolution of inter-arrival times at FT and Sprint is similar to the evolution at Level-3. In general, large events time series are sparse with no large events on several consecutive days. Hence, the relatively short inter-arrival times, observed above, denote that large events tend to be clustered. For example, figure 8.11b shows the number of daily large events at AT&T during the last six months of 2008. We observe that there is often more than one large event per day whenever we observe a large event. The low inter-arrival times at AT&T is probably caused by the fact that almost 50% of the large events at AT&T are of types AW and WA as shown in Tab. 8.3. It is likely that a failure of a large number of routes will shortly be restored if it is caused by a transient loss of reachability.

We observe that the time series of large events show little or no correlation between different monitors. In addition, large events tend to be clustered in time and don't involve a large extent of path exploration. The later observation illustrates that large events are probably triggered by transient events rather than long lasting routing changes. With regard to the evolution of the large events, we record that most of the large events' characteristics have remained stable over time, that include their size, contributed churn, duration,



(a) 90th percentile event size

(b) 90th percentile event size

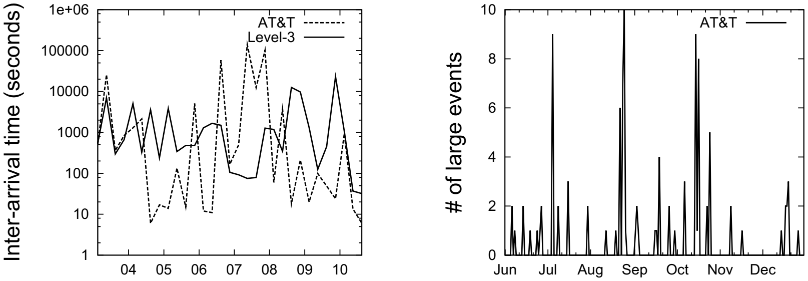
(c) Monthly number of large events

(d) Monthly churn by large events

Figure 8.10: Evolution of large events

and frequency. One important observation in this respect is the relative stability of large events' size despite the increase in the total number of routable prefixes. A possible explanation is the increased path diversity caused by the observed topology densification [DD08], which reduces the impact of single link failures and transient changes.

Unlike duplicates discussed in the previous section, updates caused by large events are necessary for correct routing, and are not unwanted artifacts of the protocol implementation. However, they are less important for understanding the long-term evolution of "background" churn. The third column in Fig. 8.2 shows the churn, after removing updates due to large events. Comparing this time series with the churn after removing duplicates, we see that most remaining large spikes in the duplicate-free churn are related to large events. Even though the remaining time series, after excluding the impact of



(a) Inter-arrival of large events

(b) Daily large events at AT&T, 2008

Figure 8.11: Inter-arrivals of large events

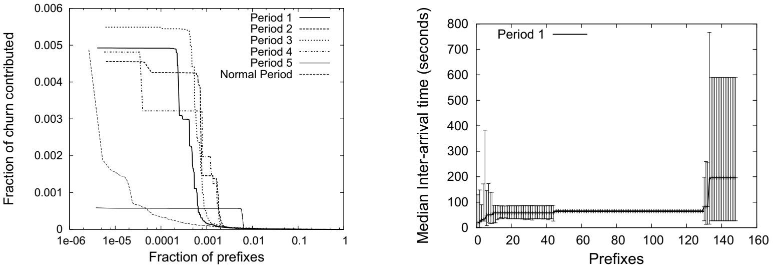
duplicate updates and large events, are much smoother, it still shows several significant level shifts; they are the subject of the next section.

8.6 Analyzing Level Shifts

The time series (for the AT&T and Level-3 monitors in particular) are still dominated by level shifts where the magnitude of churn differs substantially from the periods before and after. The presence of these level shifts makes it difficult to reliably detect any long-term trend in churn. It has proven hard to find an automated method for identifying these level shifts and finding their root cause. Instead, we make an in-depth analysis of the most dominant level shifts. We focus our analysis on the AT&T and Level-3 monitors.

AT&T The AT&T time series involves several clear level shifts, in addition to a long period of increased activity spanning one and a half year from January 2004 to June 2005. In our detailed analysis, we manually identified five distinct level shifts.

The first activity period is the long period of increasing trend from December-11-2003 to March-01-2005. The second level shift started immediately after the first activity period and lasted for one month. The third and fourth level shifts took place from February-15-2006 to March-31-2006 and from July-31-2006 to September-25-2006 respectively. Finally, the fifth level shift took place from August-19-2009 to October-16-2009.



(a) Churn contribution from the most active prefixes during four level shifts at the AT&T monitor.

(b) Median interarrivals for the most active prefixes during Period-1.

Figure 8.12: Analyzing level shifts at the AT&T monitor

Fig. 8.12a shows the fraction of total churn contributed by each prefix during our five activity periods, sorted by the activity level of each prefix. From this plot we observe in the first four periods that there is a very small set of prefixes that contributed the majority of churn during each period. In the fifth period on the other hand, a relatively larger set of prefixes was responsible for the majority of churn with a less contribution per prefix. For comparison, we also include a curve for the churn in 2008, which does not contain any level-shifts. This clearly shows the abnormality of the level shift periods.

During period 1, a small set of 148 prefixes (i.e 0.1% of the total number of prefixes) contributed 49.8% of the total churn. We investigated the activity patterns of these prefixes by examining the inter-arrival times of their updates. Figure 8.12b shows the median inter-arrival for the updates of the 148 most active prefixes, together with their 10th-90th percentile range. Notice that the prefixes can be classified into three groups based on their median updates inter-arrival times.

The first group consists of prefixes with median update inter-arrival time at 58 seconds. When investigating their update patterns we find that these prefixes belong to AS 21617 and are reached using the path {7018,701,21617} where 7018 is the monitor AS number. During this period this group of prefixes flapped up and down almost every minute. It is reasonable to believe that this long-lasting and high-frequency flapping pattern is caused by a flapping link or misconfiguration.

The prefixes that fall into the second group have their median update inter-arrival time at 65 seconds. By examining them closely we find that they are originated by either the monitor AS (i.e. 7018) or its direct customers. During this period this group of prefixes exhibited a change which is either a withdrawal or a re-announcement approximately every minute. Moreover, the 90th percentile of the update inter-arrival times is approximately equal to the median, which confirms a strict periodicity in these updates. Group two and one differ in that most of the updates which belong to the former are equally spaced in time. This makes us believe that these updates are caused by an anomaly that changes the path selection at regular intervals, rather than a flaky link or some adaptive load balancing method that would give a more irregular pattern.

The last group includes prefixes that have their median update inter-arrival time at 196 seconds. We find that these prefixes belong to AS 1938, and were changing their AS path between {7018 10888 24 11537 20965 2200 1938} and {7018 10888 11537 20965 2200 1938}. It is difficult to spot the root cause in this case. However, in the FT and Level-3 datasets we observe similar flapping patterns that involve changing some prefixes' next hop from AS 24 (NASA) to other ASes. Therefore, this activity might be caused by some instability in/near AS 24 that lasted for a long time.

Period 2 started immediately after the end of period 1, and lasted for one month. There is a small set of 170 prefixes that generates 71.7% of the total churn during this period. The main cause of this level shift is a small set of prefixes belonging to General Electric's AS (AS 80). These prefixes continuously flapped between the direct route {7018 80} and a longer route with AS 1239 (Sprint) as a next hop, i.e.{7018 1239 80}. Note that AS 80 is a stub AS and does not announce many prefixes. Still, the frequency of the route changes is high enough to create this radical increase in churn.

In period 3 and 4, we find that the level shifts are caused by leaking of private AS numbers into the global routing system. Private AS numbers (ranging from 64512 to 65535) are used to divide large ASes into multiple smaller domains connected by eBGP, or they can be assigned to stub ASes that want to use BGP with their upstreams but don't want to be part of the global routing system. Private AS numbers should always be removed from routing updates that are sent to the global BGP mesh. During these two level shifts, updates containing private AS numbers are responsible for 54.3% and 70.5% of total churn respectively.

Period 5 is different than the other four periods in two respects. First, it involved a larger number of prefixes (2030). Second, the daily churn was much higher during the shift period, about 1.8 million updates per day. We observed that a set of prefixes reached by AT&T through AS7132 (SBIS-AT&T Internet service) and AS2685(AT&T Global Network Services) had flapped with a high frequency during the shift period – approximately every 80 seconds. A discussion in the NANOG mailing list pointed to this level shift and observed the same flapping behavior [NAN09].

Level-3 The data shows a clear level shift in the Level-3 time series from March-01-2006 to August-31-2006. By doing similar analysis as in the case of AT&T, we find that the increased activity can be attributed to a set of flapping prefixes, which changed their AS-PATH continuously from {3356 3561 4134 X} to {3356 1239 4134 X} or vice versa, where X represents the rest of the AS path. Here we see how AS 3356 (Level-3) alternates between the two different neighboring ASes 3561 (Savvis) and 1239 (Sprint) to reach AS 4134 (China-Backbone). Note here that Savvis is owned by Level-3 and hence the route through Savvis is preferred. When this route is lost, Level-3 will select the backup route through Sprint. The frequency of this flapping for each prefix is between once every 10 minutes and once every 20 minutes. However, China-Backbone is a major transit provider, and Level-3 selects it as the preferred path for more than 2000 destination prefixes. Hence, a single change will trigger a large number of updates.

We also identified a second level shift in the Level-3 time series, that took place from June-15-2010 to 31-July-2010. We find that this level shift is caused by continuous flapping in reaching prefixes originated by AS9808(Guangdong Mobile Communication) and its customers.

The above analysis shows that level shifts are usually caused by specific failures or misconfigurations in or near the monitored AS. The left column in Fig. 8.13 shows the churn time series after filtering out all updates attributed to the level shift generators described above. Level shifts are clearly anomalies that last for long period of time. Mitigating and avoiding such anomalies can be achieved through better monitoring and configuration management systems. There is a need for tools that are able to detect level shifts.

In the following section we discuss and analyze different statistical properties and trends in the churn time series after removing the effect of level shifts.

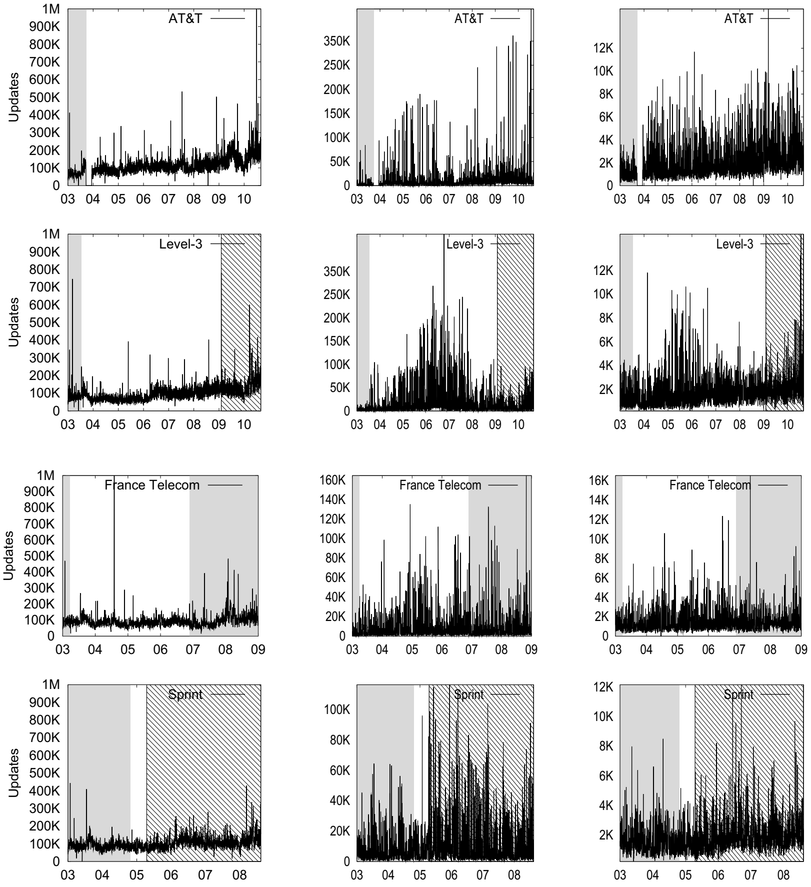


Figure 8.13: Baseline daily total churn (left), 1-minute peak churn per day in the raw time series (middle), and 1-minute peak churn per day in the baseline time series (right) at our four monitors.

8.7 The Growth of Baseline Churn

In this section, we analyze the growth of the churn time series after removing duplicate updates, large events, and the level shifts of the previous section. We refer to this time series as the churn "baseline". We also analyze the time series of peak churn, measured from the busiest 1-minute period of each day.

8.7.1 Baseline churn

After removing churn caused by anomalies and effects that are not related to the long term evolution of churn, we are left with the baseline churn, which is a much smoother time series and shows more correlation across monitors (see Fig. 8.13). The Kendall's tau rank correlation coefficient between the AT&T, Level-3, and Sprint baseline time series is around 0.5, which is almost double the highest value observed in the raw time series (0.25). This increase suggests that our approach has managed to filter most of the effects that are quite local to each monitor. The cross-correlation between the three North Americans monitors and FT is lower around 0.40. Geographical locality and presence could be a plausible explanation for this. Note that we should not expect our monitors to be tightly correlated, since they have different topological connectivity. Nevertheless, we expect to observe a higher correlation in the baseline, because we have identified and filtered out a significant amount of the local effects.

The application of linear regression on the baseline time series results in a low Pearson's correlation coefficient (0.03 to 0.42, depending on the monitor), since even the baseline churn contains some spikes and small level shifts. Therefore, we rely on non-parametric statistics, which is more robust to outliers, and in particular on the Mann-Kendall statistical test for trend detection. The Mann-Kendall test reports that there is a statistically significant increasing trend in the baseline time series in all four monitors at a 90% significance level. Actually, both the non-parametric and parametric (linear regression) tests give similar estimates for the slope of the increasing trend. Table 8.5 presents the estimated slopes in additional updates per day.

The same table also shows the estimated relative churn increase during the study period. This figure is calculated from the estimated slope using the median daily churn rate for the first 3 months as starting point. The two estimation techniques are in reasonable agreement with each other. Note that the estimated increase covers a period of six years in FT and Sprint case, while it spans seven years and eight months in AT&T and Level-3. During the first six years the daily churn grew by about 50% in AT&T and 69% in Level-3, which indicates a faster growth than in FT and Level-3. Interestingly, the estimated increase reported in Table 8.5 shows a faster growth in both AT&T and Level-3 during the last 20 months (about 30%) than in the first six years. The significant differences between the monitors is not surprising, since different monitors have different sets of customers and peers and different internal configuration.

A main observation from this analysis is that the increase in the baseline churn is relatively slow compared to the growth of the routing table size. The data presented in Fig. 8.1 shows that the number of routable prefixes has increased by 168% over our study period, while the baseline churn has increased by about 100%.

This implies that the *daily number of updates per prefix in the baseline time series has decreased over our study period.* We are planning to investigate the reasons behind this decrease in our future work. One potential explanation is better network management practices at most stub networks. The "densification" of the Internet [DD08] may have also helped, as it provides additional routing paths when the preferred route is lost. A recent study by Huston [Hus10] concluded that BGP churn increases at a much slower pace than the routing table size, in agreement with the findings in this chapter.

8.7.2 Daily Peak Activity

The churn rates presented so far are daily averages. The peak churn rates in shorter timescales may be more important in terms of the processing load imposed on routers. Here, we examine the growth of the peak daily churn rate, measured as the *maximum 1-minute churn on each day.* We refer to this time series as the "daily peak churn".

The plots in the second and third columns of Fig. 8.13 show the daily peak churn in the raw time series and in the baseline time series, respectively. A first observation is that the daily peak activity in the raw time series is much higher than in the baseline time series: on average, there is an order of magnitude difference between the two time series across all monitors, and on some days the difference can reach up to two orders of magnitude. This confirms that local effects and protocol misconfigurations are responsible for most of the peak churn that routers have to process.

Table 8.5: Baseline churn growth: Mann-Kendall slope estimate in updates per day, and estimated relative churn increase during our study period. The parametric estimates are also shown.

Monitor	AT&T	Level-3	FT	Sprint
M-K slope	33.82	28.65	7.79	14.38
Est. increase	101.2%	103.7%	20.0%	29.4%
Lin. regr. slope	40.06	31.55	6.50	16.42
Est. increase	119.9%	114.2%	15.5%	33.3%

Table 8.6: Daily peak churn growth

Monitor	AT&T	Level-3	FT	Sprint
Raw peak churn				
M-K slope	2.22	0.81	0.27	-
Est increase	168.4%	171.3%	50.5%	-
Baseline peak churn				
M-K slope	0.50	0.42	0.10	0.29
Est increase	100.0%	113.4%	20.9%	39.3%

The Mann-Kendall test reports an increasing trend in the raw and baseline daily peak churn across all four monitors. The exception is the raw time series at the Sprint monitor, where no trend could be detected. Table 8.6 presents the slope and the relative estimated increase at each monitor. In order to compare all four monitors, we compute the M-K slope of the raw and baseline peak churn during the first six years [3]. During this period, the M-K slope of the AT&T baseline peak churn (0.32) was about third the M-K slope of the AT&T raw peak churn (1.03). We observe a similar trend in the Level-3 monitor with the M-K slope of the baseline peak churn at 0.43 and the M-K slope of the raw peak churn at 1.57. The modest growth in the FT and Sprint monitors is probably due to the use of rate-limiting timers. The noisy nature of the raw time series makes it difficult to get accurate growth trends, and so these numbers should be viewed only as rough estimates.

We observe that the estimated relative growth in daily 1-minute peak churn rate is somewhat higher for the raw time series than for the baseline. This indicates that the impact, in terms of peak churn, of duplicates and other local effects increases with time. For the baseline time series, the increase in

[3] We don't include the last year and half because Sprint's and France Telecom's monitors were unavailable.

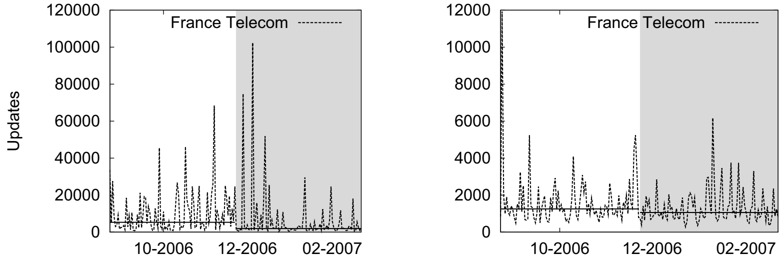


Figure 8.14: Churn before and after MRAI timer is turned on at FT monitor (left: raw peak churn, right: baseline peak churn).

the daily 1-minute peak level is comparable to the increase in the total daily churn.

Finally, we investigate to what extent the daily peak churn is influenced by the use of rate-limiting timers. We compare the median daily peak churn calculated in a three month window immediately before and after each change in the rate-limiting configuration at the FT, Level-3, and Sprint monitors. Fig. 8.14 shows the churn in the 3-month period before and after the MRAI timer was turned on in late 2006 at the FT monitor, for the raw and baseline time series (the horizontal lines in the figures show the median level of churn). We find that the rate-limiting timer has no clear effect on the daily peak churn in the baseline time series. However, in the raw time series, there is a clear increase in the peak churn when the rate-limiting timer is turned off. The peak churn increases by a factor 1.1 and 1.2 in the first and second transitions in Sprint, 2.0 and 0.0 in the first and second transitions in Level-3, and 3.7 and 2.8 in the first and second transitions in FT.

These findings show that the effect of the rate-limiting timer is much stronger on the raw time series than on the baseline. This implies that the rate-limiting timer is mostly effective at filtering out duplicate updates and local effects. We also observe that there are significant spikes in the daily peak churn both before and after deploying rate-limiting timer. This observation is in agreement with our earlier findings in Sec. 7.2. Such spikes are normally caused by underlying events in the transit paths that affect a large number of destination prefixes simultaneously.

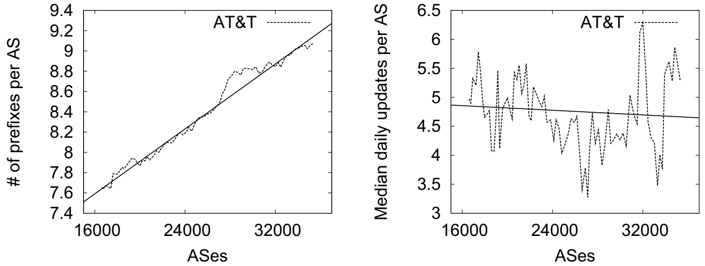
8.8 Discussion

A main conclusion of our analysis is the surprisingly slow growth in the baseline churn time series in comparison to the large increase in the number of prefixes. This observation raises questions about possible reasons behind the slow increase, factors that drive the baseline churn, and more importantly its implications on routing scalability.

Our analysis of churn evolution has focused on the temporal side of the growth. Another important measure is to evaluate churn growth with respect to the Internet size. This size can be measured in different ways, like DFZ routing table size, number of ASes, number of domains, and number of hosts. In this context we are interested in evaluating both the increase in the DFZ table size and the baseline churn with respect to the Internet size. Hence, we choose the number of ASes as a measure for the network size.

Figure 8.15a illustrates the evolution of the number of prefixes per AS as measured at the AT&T monitor. These values are sampled once every month in the period between (January-2004 to September-2010). The number of prefixes per AS increases linearly as the number of AS increases; it has grown slowly from 7.6 to 9.0. A recent work by Carpenter [Car09] illustrated a fixed relationship between the routing table size and number of AS, i.e, the number of prefixes per AS is fixed. Our estimation does not show that fixed relationship but rather a slow increase. Figure 8.15b shows the evolution of the median daily churn per AS as measured at the AT&T monitor. The number of updates contributed by each AS is almost constant. The results from Level-3 monitor, which was available during the same period, are very close to what reported above. Furthermore, replacing median by average daily churn does not change the observed fitting.

The number of ASes has increased by 112.8% between Jan-2004 and Sep-2010. In the same period, the DFZ table size has grown by 152%, while the baseline churn has increased by about 103.7%. We observe that *the increase in baseline churn is very similar to that of the number of ASes*. While we can not identify the causes behind the emergence of this relationship, we believe that it is important for the scalability of the global routing system. More research is needed before we can tell whether this is a fundamental property of the Internet.



(a) Prefixes per AS (b) Median daily updates per AS

Figure 8.15: Global routing scalability

8.9 Summary

This chapter has investigated the evolution of churn at four monitors located in the core of the Internet over a period up to seven years and eight months. The corresponding time series are very bursty, with large churn spikes and level-shifts. We have performed an in-depth analysis of the time series in order to identify and explain the main sources of churn.

We have found that up to 40% of route announcements are redundant and they are not needed for correct protocol behavior. We also identified the underlying reasons for the most severe level-shifts in churn. These are normally caused by configuration mistakes or other anomalies in or close to the monitored AS. Our findings suggest that the most effective short-term solutions for limiting churn will be protocol improvements that filter out redundant updates, and methods that can detect (long-lasting) configuration mistakes and other anomalies that result in sustained high churn levels.

This study has also shown that *there is a long-term increasing trend in the identified baseline churn*, but at the same time, *the growth rate is relatively low*. We find that the *churn rate increases more slowly than the number of prefixes* in the routing table. While the routing table grew about 168% during our study period, the baseline churn rate grew at most by about 103.7%.

We also investigated the daily 1-minute peak churn rate, and found that this is an order of magnitude higher in the raw time series compared to the baseline. These time series are very noisy, but they appear to be slowly growing with time.

In the next part of this thesis, we examine our conclusions and their implications. Further we sketch possible future work directions.

Part IV

Epilogue

Chapter 9

Conclusions

Throughout this thesis, we have studied factors that influence the evolution of BGP churn. Our work sheds light on the impact of topology growth under different failure types on the experienced level of churn. We have also investigated the effect of update rate-limiting, and different implementations choices. Further, we have presented the most comprehensive measurement study so far of churn evolution at the core of the Internet. In this study, we have shown that the increase in the baseline churn is relatively slow, and will not pose a serious scalability problem in the foreseeable future. The observed growth contradicts previous studies that reported an alarming growth in churn. The rest of this chapter presents our conclusions in details.

9.1 The Role of Topology Growth

We grouped factors that influence churn into three categories. Namely, topology properties; types of routing events that take place in the network; and routing protocol implementations and configurations. Then, we asked several "what-if" questions about the impact of topology growth under different failure types on the experienced level of churn.

Addressing such wide range of hypothetical scenarios cannot be performed by doing measurements in the current Internet. To this end, we have proposed a flexible model for simulating BGP. It consists of a topology generator that produces AS-level graphs which are annotated with business relationships; and a light-weight BGP simulator that is capable of capturing routing dynamics and scaling to network sizes of thousands of nodes. We have validated the topology generator and illustrated its ability in reproducing various known properties of the AS-level Internet topology. Besides, we have compared the performance and correctness of our simulator when simulating BGP dynamics, with that of the widely used SSFNET simulator. This benchmarking confirms that our simulator significantly outperforms the SSFNET simulator in terms of processing time and memory requirements, while producing similar results.

We have further employed our framework in examining the role of topology growth on the scalability of BGP. We have started by looking at the number of updates received at nodes at different locations in the AS hierarchy after a prefix failure event at the edge of the network. For different node types, we have identified the most significant sources of churn, and described how different factors contribute to increased churn as the network grows. We have shown that nodes at the top of the AS hierarchy experience both the highest churn in absolute terms, and the strongest increase as the network grows. We have further looked into the impact of a single link failure at a multihomed stub on routing scalability, and demonstrated that certain topology growth scenarios scale differently depending on failure types. Besides, we have investigated the impact and visibility of transit link failures between core nodes, by calculating the number of affected nodes and source-destination pairs.

Using our flexible topology model, we have explored scalability in several plausible or educational ”what-if” scenarios for the growth of the AS-level topology. We have shown that *the most important topological factor deciding the number of updates generated is connectivity in the core of the network*. In particular, the number mid-tier transit providers and the multihoming degree of these nodes plays a crucial role, since *transit nodes in the mid-level of the Internet hierarchy have a special role in multiplying update messages.*

Another important finding is that peering links play a very different role than transit links with respect to scalability. *The peering degree in the Internet does not influence churn.* We have also shown that the depth of the hierarchical structure in the Internet plays a significant role. *If we are moving towards an Internet in which customers and content providers at the edges prefer to connect to mid-tier ISPs, the number of BGP updates at tier-1 nodes will be much higher than if they prefer to connect to tier-1 ISPs.* Finally, we have demonstrated that densification through increased multihoming degree will have a different impact on routing scalability depending on its location and the failure type. *While densification at the edge increases churn after a* *prefix failure, it gives reduced churn after a link failure. On the other hand, a denser core reduces the scope and impact of a transit link failure but it increases churn because of both edge prefix and edge link failures.*

The observed densification in the Internet [DD08] can potentially impact the occurrence probability of different failure types. We expect the probability of a complete failure of an AS or a prefix to decrease. An increased multihoming will reduce the likelihood that such failures are caused by a failure of a single link. Furthermore, a denser network means that there are more inter-AS links. It is reasonable to expect more links to fail if we assume a constant failure pattern over time, but at the same time the importance of a link decreases due to the potential availability of alternative routes. Combining these possibilities and our findings concerning failures of edge links in Sec. 6.4 indicate that densification will likely have a positive impact on BGP churn scalability.

Recent measurement studies [GALM08, LIJM^{+}10], showed that the AS-level topology is becoming flatter. The observed flattening is a direct consequence to the proliferation of settlement-free peering between content/access providers and transit providers that is made possible by the increase in the number of IXPs. This increased peering may not influence BGP churn directly; recall that the peering degree in the Internet does not influence churn. However, it results in an increasing path diversity which will likely have a positive impact on BGP churn scalability. We believe that further measurement work is needed to quantify the impact of the topology densification and flattening on churn evolution in the Internet.

9.2 The Role of Update Rate-Limiting

Our work has also explored how different BGP rate-limiting implementations and configurations affect the level of churn. We have employed measurements to investigate differences between rate-limiting implementations and timer's values. Measurements were performed on two parallel BGP sessions, with and with- out rate-limiting respectively, to three routers in a stub AS. *Our measurements have shown that the sustained level of churn is strongly reduced when enabling the rate-limiting timer*, and we have explained how the Out-Delay implementation (used in Juniper routers) gives a stronger reduction than MRAI timers (used in Cisco routers). Using emulation on the measured churn time series, *we have shown that the reduction is significant for both im-*

plementations already at low timer values (which keep the convergence delay acceptable). With an OutDelay of 30 seconds, churn is reduced by as much as two thirds.

Using data from a large number of RouteViews monitors, we have investigated the update inter-arrival pattern in BGP sessions that are not rate limited. *We have found a strong similarity in the distribution of inter-arrival times across monitoring sessions in different parts of the Internet, and across different years.* This observed universality has allowed us to formulate an expression that quantifies the expected reduction in churn levels for different rate-limiting implementations and timer values. *This expression is able to predict the churn reduction observed in our measurements within a few percent.* Our model can be useful for network operators in deciding a rate-limiting configuration that gives the right balance between churn reduction and convergence delay.

In addition, we have used our simulation framework to study the role of a particular aspect of the rate-limiting mechanism - whether explicit route withdrawals are subject to the MRAI timer. *We have shown that rate-limiting explicit withdrawals leads to a significant increase in churn after a prefix failure at a stub network*, because of the effect of path exploration. This difference grows with network size, and is stronger in well-connected networks.

9.3 Churn Growth at the Core of the Internet

This thesis has investigated the evolution of churn at four monitors located in the core of the Internet over a period up to seven years and eight months. The corresponding time series are very bursty, with large churn spikes and level-shifts. We have performed an in-depth analysis of the time series in order to identify and explain the main sources of churn.

We have found that up to 40% of route announcements are redundant and they are not needed for correct protocol behavior. We have also identified the underlying reasons for the most severe level-shifts in churn. These are normally caused by configuration mistakes or other anomalies in or close to the monitored AS. Surprisingly, some of these incidents went unnoticed for several months. *Our findings suggest that the most effective short-term solutions for limiting churn will be protocol improvements that filter out redundant updates, and methods that can detect (long-lasting) configuration mistakes and other anomalies that result in sustained high churn levels.*

This part of our work has also shown that *there is a long-term increasing trend in the identified baseline churn, but at the same time, the growth rate is relatively low.* We have found that the churn rate increases more slowly than the number of prefixes in the routing table. While the routing table grew by about 168% during our study period, the baseline churn rate grew at most by about 103%. There can be several reasons why we only see a slow increase in the baseline churn compared to the growth of the routing table size. On one hand, configuration management systems and experience are improving. Also, the increasing connectivity in the Internet can play a positive role, since more failures can be handled locally if an alternate route is known.

We have also investigated the daily 1-minute peak churn rate, and found that this *is an order of magnitude higher in the raw time series compared to the baseline.* These time series are very noisy, but they appear to be slowly growing with time.

To sum up, this thesis contributes important findings about churn evolution at the core of the Internet; a methodology for dissecting BGP churn time series; a systematic investigation of different factors behind churn evolution; and new tools for generating AS-level topologies and simulating BGP. Our work is a first step towards fully characterizing the contribution of different factors to BGP churn. Following efforts can potentially employ our simulation framework and methods for that purpose. The next chapter highlights possible future directions.

Chapter 10

Future directions

This thesis work can be improved upon and taken forward in several directions. We highlight some of them in the following.

Measuring and modeling inter-domain routing events. In Chapter 6, we have examined two well-defined routing events at the periphery of the network. It is also interesting to explore other events that occur in the core of the network. Moreover, other events' characteristics such as frequency and duration of failures present another dimension to investigate. To this end, we need first to model and characterize different routing events and failures. These characterizations can be further combined to create a realistic mix of routing events that reproduces measured BGP updates data. This mix can be used as an input workload for testing router's ability to cope with churn. To model different routing events and failures, we can start by measuring BGP updates at several vantage points and then applying a root cause analysis method [CSK03, FMM$^+$04]. However, such approach necessitates addressing several challenges in advance that are related to the representativeness of the measurement data and the accuracy of the root cause analysis inference.

iBGP configurations. All simulations in this thesis make simplifying assumptions by modeling ASes as atomic nodes and thus ignoring intra-AS topology and iBGP configurations. We have done that because we wanted to reduce the complexity of our simulations since we investigate effects that are unrelated to intra-AS connectivity. In addition, intra-AS topologies vary between ASes, and there is no reasonable model that can be used to reproduce them. The lack of this model is partly caused by the reluctance of network operators about revealing internal details of their networks. However, exploring the impact of different internal topologies and iBGP configurations on inter-domain churn will help in improving our understanding of BGP churn evolution. To address this, we first need to support iBGP in our simulation framework. Then we can ask several "what-if" questions about different iBGP configurations.

Churn evolution. We have investigated churn evolution from one perspective, i.e., the core of the Internet. It is also interesting to explore other perspectives, for example stub networks and access providers. However, these networks inherently vary widely in terms of connectivity, which questions the representativeness and generality of the respective conclusions. Hence, such measurement studies could be presented as case studies that report on ASes with specific properties. This can be achieved by examining measurement data in the light of topology properties of monitored networks. Another interesting problem is related to investigating the evolution of BGP churn inside different ASes i.e. iBGP churn. Unfortunately, obtaining representative data in this respect is hard, since ISPs are reluctant to share internal data. Furthermore, it is not clear whether such historical data exists in the first place.

Statistical analysis of churn time series. In this thesis, in order to gain insights about various components of BGP churn time series, we have preferred to use an exploratory approach over a black-box statistical one in analyzing churn time series. Our approach has proved suitable in dissecting churn components and characterizing the growth in the baseline churn. Combined with the insights we have gained so far, we believe that applying more rigorous statistical methods can potentially reveal more details about properties of anomalies and characteristics of churn, for example underlying statistical distributions and their evolution over time. An interesting observation in this respect is the universal pattern in BGP updates inter-arrival times that we have identified in Chapter 7. Statistical analysis is necessary for modeling churn time series, and for predicting future levels of churn.

Our approach in analyzing churn involves several manual steps, which makes the task of investigating more monitors tedious. One possibility, is to use time series analysis methods for detecting anomalous periods that contain large spikes and level-shifts. The authors in [PVA$^+$09] have proposed employing a set of time series analysis techniques for this purpose. We can use our results to calibrate and improve such techniques.

The growth of the baseline churn. One of our central findings is the observed slow growth in the baseline churn at the core of the Internet. A potential explanation is better network management practices at most stub networks. The "densification" of the Internet may also play an important role as it provides additional routing paths when the preferred route is lost. Pointing out the exact causes behind the observed growth will help improving our understanding of the current routing architecture and consequently in designing better architectures in the future.

Bibliography

[700] AS 7007. http://www.merit.edu/mail.archives/nanog/1997-04/msg00444.html.

[AB02] Reka Albert and Albert L. Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47, 2002.

[ACL00] William Aiello, Fan Chung, and Linyuan Lu. A random graph model for massive graphs. In *STOC '00: Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 171–180, New York, NY, USA, 2000. ACM.

[AFBB02] David G. Andersen, Nick Feamster, Steve Bauer, and Hari Balakrishnan. Topology inference from BGP routing dynamics. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, IMW '02, pages 243–248, New York, NY, USA, 2002. ACM.

[ALD$^+$05] Joe Abley, Kurt Lindqvist, Elwyn Davies, Benjamin Black, and Vijay Gill. IPv4 multihoming practices and limitations. RFC4116, Jul 2005.

[BA99] Albert L. Barabasi and Reka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, October 1999.

[BBAS04] Anat Bremler-Barr, Yehuda Afek, and Shemer Schwarz. Improved BGP convergence via ghost flushing. *IEEE Journal on Selected Areas in Communications*, 22(10):1933–1948, 2004.

[BCC00] Tony Bates, Ravi Chandra, and Enke Chen. BGP Route Reflection - an alternative to full mesh iBGP. RFC2796, Apr 2000.

[BCK$^+$10] Randy Bush, Brett Carr, Daniel Karrenberg, Niall O'Reilly, Ondrej Sury, Nigel Titley, Filiz Yilmaz, and Ingrid Wijte. IPv4 address allocation and assignment policies for the RIPE NCC service region. ripe-498, Oct 2010.

[BFF05] Olivier Bonaventure, Clarence Filsfils, and Pierre Francois. Achieving sub-50 milliseconds recovery upon BGP peering link failures. In *Proceedings of the 2005 ACM conference on Emerging network experiment and technology*, CoNEXT '05, pages 31–42, New York, NY, USA, 2005. ACM.

[BFM$^+$05] Hagen Bohm, Anja Feldmann, Olaf Maennel, Christian Reiser, and Rudiger Volk. Network-wide inter-domain routing policies: Design and realization. Presentation at the NANOG34 Meeting, April 2005.

[BFZ07] Hitesh Ballani, Paul Francis, and Xinyang Zhang. A study of prefix hijacking and interception in the Internet. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '07, pages 265–276, New York, NY, USA, 2007. ACM.

[BGT04] Tian Bu, Lixin Gao, and Don Towsley. On characterizing BGP routing table growth. *Computer Networks*, 45(1), May 2004.

[BNkc02] Andre Broido, Evi Nemeth, and kc claffy. Internet expansion, refinement, and churn. *European Transactions on Telecommunications*, January 2002.

[BOR$^+$02] Anindya Basu, Chih-Hao Luke Ong, April Rasala, F. Bruce Shepherd, and Gordon Wilfong. Route oscillations in I-BGP with route reflection. In *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '02, pages 235–247, New York, NY, USA, 2002. ACM.

[BSH] Tony Bates, Philip Smith, and Geoff Huston. CIDR report. http://www.cidr-report.org/as2.0. ”Online; accessed 15-Dec-2010”.

[CAI10a] CAIDA. Archipelago measurement infrastructure. http://www.caida.org/projects/ark, 2010. "Online; accessed 15-Dec-2010".

[CAI10b] CAIDA. Skitter. www.caida.org/tools/measurement/skitter, 2010. "Online; accessed 15-Dec-2010".

[Car09] Brian E. Carpenter. Observed relationships between size measures of the Internet. *SIGCOMM Comput. Commun. Rev.*, 39:5–12, March 2009.

[CDZK05] Jaideep Chandrashekar, Zhenhai Duan, Zhi-Li Zhang, and J. Krasky. Limiting path exploration in BGP. In *INFOCOM*, pages 2337–2348, 2005.

[Cha96] Chris Chatfield. *The Analysis of Time Series: An Introduction.* Chapman & Hall/CRC, fifth edition, 1996.

[CID10] CIDR report. http://www.cidr-report.org/as2.0, 2010. "Online; accessed 28-Dec-2010".

[CMU$^+$10] Luca Cittadini, Wolfgang Muhlbauer, Steve Uhlig, Randy Bush, Pierre Francois, and Olaf Maennel. Evolution of Internet address space deaggregation: Myths and reality. *IEEE Journal on Selected Areas in Communications*, 2010.

[Cow10] James Cowie. China's 18-Minute Mystery. http://www.renesys.com/blog/2010/11/chinas-18-minute-mystery.shtml, Nov 2010.

[CR06] Rami Cohen and Danny Raz. The Internet dark matter - on the missing links in the AS connectivity map. In *INFOCOM*, 2006.

[CSK03] Matthew Caesar, Lakshminarayanan Subramanian, and Randy H. Katz. Towards localizing root causes of BGP dynamics. Technical Report UCB/CSD-04-1302, U.C.Berkely, November 2003.

[CTL96] Ravishanker Chandra, Paul Traina, and Tony Li. BGP Communities Attribute. RFC1997, Aug 1996.

[CZZZ10] Pei-chun Cheng, Xin Zhao, Beichuan Zhang, and Lixia Zhang. Longitudinal study of BGP monitor session failures. *SIGCOMM Comput. Commun. Rev.*, 40:34–42, April 2010.

[DCK$^+$04] Zhenhai Duan, Jaideep Chandrashekar, Jeffrey Krasky, Kuai Xu, and Zhi-Li Zhang. Damping BGP Route Flaps. In *IEEE International Performance Computing and Communications Conference*, 2004.

[DD08] Amogh Dhamdhere and Constantine Dovrolis. Ten years in the evolution of the Internet ecosystem. In *IMC 2008*, 2008.

[DFJG01] Zihui Ge Daniel, Daniel R. Figueiredo, Sharad Jaiswal, and Lixin Gao. On the hierarchical structure of the logical Internet graph. In *in Proc. SPIE ITCOM*, pages 208–222, 2001.

[DIM10] The dimes project. http://www.netdimes.org, 2010. ”Online; accessed 15-Dec-2010”.

[DKF$^+$07] Xenofontas Dimitropoulos, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Young Hyun, kc claffy, and George Riley. AS relationships: inference and validation. *SIGCOMM Comput. Commun. Rev.*, 37:29–40, January 2007.

[DKVR08] Xenofontas Dimitropoulos, Dmitri Krioukov, Amin Vahdat, and George Riley. Graph annotations in modeling complex network topologies. *arXiv:0708.3879*, 2008.

[dL] Cedric de Launois. GHITLE: Generator of Hierarchical Internet Topologies using LEvels. http: //ghitle.info.ucl.ac.be/.

[DPP03] Giuseppe Di Battista, Maurizio Patrignani, and Maurizio Pizzonia. Computing the types of the relationships between autonomous systems. In *IEEE INFOCOM 2003*, pages 156–165, 2003.

[DR06] Xenofontas Dimitropoulos and George Riley. Efficient large-scale BGP simulations. *Elsevier Computer Networks, Special Issue on Network Modeling and Simulation*, 50(12):2013–2027, 2006.

[DS04] Shivani Deshpande and Biplab Sikdar. On the impact of route processing and MRAI timers on BGP convergence times. In *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, 2004.

[DSK08] Xenofontas A. Dimitropoulos, M. Ángeles Serrano, and Dmitri V. Krioukov. On Cycles in AS Relationships. *CoRR*, abs/0807.0887, 2008.

[Dub99] Rohit Dube. A comparison of scaling techniques for BGP. *SIGCOMM Comput. Commun. Rev.*, 29:44–46, July 1999.

[ER59] Paul Erdos and Alfred Renyi. On random graphs. I. *Publ. Math. Debrecen*, 6:290–297, 1959.

[FB05] Nick Feamster and Hari Balakrishnan. Detecting BGP configuration faults with static analysis. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2*, NSDI'05, pages 43–56, Berkeley, CA, USA, 2005. USENIX Association.

[FCM$^+$09] Anja Feldmann, Luca Cittadini, Wolfgang Mühlbauer, Randy Bush, and Olaf Maennel. HAIR: hierarchical architecture for Internet routing. In *Proceedings of the 2009 workshop on Re-architecting the internet*, ReArch '09, pages 43–48, New York, NY, USA, 2009. ACM.

[FFF99] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the Internet topology. In *ACM SIGCOMM*, pages 251–262, 1999.

[fir] FIRE - Future Internet Research and Experimentation. http://cordis.europa.eu/fp7/ict/fire.

[FJB05] Nick Feamster, Jaeyeon Jung, and Hari Balakrishnan. An empirical study of "bogon" route advertisements. *SIGCOMM Comput. Commun. Rev.*, 35:63–70, January 2005.

[FLYV93] Vince Fuller, Tony Li, Jessica Yu, and Kannan Varadhan. Classless inter-domain routing (CIDR): an address assignment and aggregation strategy. RFC1519, Sep 1993.

[FMM$^+$04] Anja Feldmann, Olaf Maennel, Z. Morley Mao, Arthur Berger, and Bruce Maggs. Locating internet routing instabilities. In *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '04, pages 205–218, New York, NY, USA, 2004. ACM.

[GAG$^+$03] Geoffrey Goodell, William Aiello, Timothy G. Griffin, John Ioannidis, Patrick Drew McDaniel, and Aviel D. Rubin. Working around BGP: An Incremental Approach to Improving Security and Accuracy in Interdomain Routing. In *NDSS*, 2003.

[GALM08] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. The flattening Internet topology: natural evolution, unsightly barnacles or contrived collapse? In *Proceedings of the 9th international conference on Passive and active network measurement*, PAM'08, pages 1–10, Berlin, Heidelberg, 2008. Springer-Verlag.

[Gao01] Lixin Gao. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking (TON)*, 9(6), December 2001.

[gen] The Global Environment for Network Innovations (GENI). http://www.geni.net.

[GGRW03] Joel Gottlieb, Albert Greenberg, Jennifer Rexford, and Jia Wang. Automated provisioning of BGP customers. *Network, IEEE*, 17:44–55, Nov 2003.

[GMZ03] Christos Gkantsidis, Milena Mihail, and Ellen W. Zegura. Spectral analysis of Internet topologies. In *INFOCOM*, 2003.

[GP01] Timothy G. Griffin and Brian J. Premore. An experimental analysis of BGP convergence time. In *ICNP*, 2001.

[GR00] Lixin Gao and Jennifer Rexford. Stable Internet routing without global coordination. In *Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS '00, pages 307–317, New York, NY, USA, 2000. ACM.

[GSW02] Timothy G. Griffin, F. Bruce Shepherd, and Gordon Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Trans. Netw.*, 10:232–243, April 2002.

[GW99] Timothy G. Griffin and Gordon Wilfong. An analysis of BGP convergence properties. In *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, SIGCOMM '99, pages 277–288, New York, NY, USA, 1999. ACM.

[GW02a] Timothy G. Griffin and Gordon Wilfong. On the correctness of IBGP configuration. In *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '02, pages 17–29, New York, NY, USA, 2002. ACM.

[GW02b] Timothy G. Griffin and Gordon T. Wilfong. Analysis of the MED Oscillation Problem in BGP. In *Proceedings of the 10th IEEE International Conference on Network Protocols*, ICNP '02, pages 90–99, Washington, DC, USA, 2002. IEEE Computer Society.

[HA06] Geoff Huston and Grenville Armitage. Projecting future IPv4 router requirements from trends in dynamic BGP behaviour. In *ATNAC*, Australia, December 2006.

[HFJ$^+$09] Hamed Haddadi, Damien Fay, Almerima Jamakovic, Olaf Maennel, Andrew W. Moore, Richard Mortier, and Steve Uhlig. On the importance of local connectivity for Internet topology models. In *the 21st International Teletraffic Congress*, pages 1–8, September 2009.

[HFKC08] Yihua He, Michalis Faloutsos, Srikanth V. Krishnamurthy, and Marek Chrobak. Policy-aware topologies for efficient interdomain routing evaluations. In *IEEE INFOCOM 2008 Mini-Conference*, Phoenix, AZ, USA, April 2008.

[HM07] Xin Hu and Z. Morley Mao. Accurate Real-time Identification of IP Prefix Hijacking. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, SP '07, pages 3–17, Washington, DC, USA, 2007. IEEE Computer Society.

[HSFK07] Yihua He, Georgos Siganos, Michalis Faloutsos, and Sirkanth Krishnamurthy. A systematic framework for unearthing the missing links: Measurements and impact. In *NSDI*, Cambridge, MA, USA, April 2007.

[Husa] Geoff Huston. The BGP instability report. http://bgpupdates.potaroo.net/.

[Husb] Geoff Huston. BGP Routing Table Analysis Reports. http://bgp.potaroo.net. "Online; accessed 15-Dec-2010".

[Hus02] Geoff Huston. Analyzing the internet BGP routing table. *The Internet Protocol Journal*, 4(1), 2002.

[Hus10] Geoff Huston. BGPin 2009 (and a bit of 2010). Presentation at ARIN XXV meeting, TORONTO, Canada, April 2010.

[IAN10] The Internet Asigned Numbers Authority (IANA). http://www.iana.org, 2010. "Online; accessed 15-Dec-2010".

[int10] Internet World Stats. www.internetworldstats.com/stats.htm, 2010. "Online; accessed 10-Dec-2010".

[ipa] IPv4 address report. http://www.potaroo.net/tools/ipv4. "Online; accessed 15-Dec-2010".

[Jak08] Paul Jakma. Revised default values for the BGP 'minimum route advertisement interval'. IETF Internet Draft, November 2008.

[JMY$^+$08] Dan Jen, Michael Meisel, He Yan, Dan Massey, Lan Wang, Beichuan Zhang, and Lixia Zhang. Towards A New Internet Routing Architecture: Arguments for Separating Edges from Transit Core. In *Proceedings of the Seventh ACM Workshop on Hot Topics in Networks (HotNets-VII)*, October 2008.

[Ken38] Maurice G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[KFR06] Josh Karlin, Stephanie Forrest, and Jennifer Rexford. Pretty Good BGP: Improving BGP by Cautiously Adopting Routes. In

Proceedings of the Proceedings of the 2006 IEEE International Conference on Network Protocols, pages 290–299, Washington, DC, USA, 2006. IEEE Computer Society.

[KkcFB07] Dmitri Krioukov, kc claffy, Kevin Fall, and Arthur Brady. On compact routing for the Internet. *Computer Communications Review*, 37(3), July 2007.

[KKK07] Nate Kushman, Srikanth Kandula, and Dina Katabi. Can you hear me now?!: it must be BGP. *SIGCOMM Comput. Commun. Rev.*, 37:75–84, March 2007.

[KKKM07] Nate Kushman, Srikanth Kandula, Dina Katabi, and Bruce Maggs. R-BGP: Staying Connected in a Connected World. In *4th USENIX Symposium on Networked Systems Design and Implementation*, Cambridge, MA, April 2007.

[KL51] Solomon Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 1951.

[KLMS00] Stephen Kent, Charles Lynn, Joanne Mikkelson, and Karen Seo. Secure Border Gateway Protocol (S-BGP). *IEEE Journal on Selected Areas in Communications*, 18:103–116, 2000.

[LABJ00] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed Internet routing convergence. In *ACM SIGCOMM*, pages 175–187, August 2000.

[LAJ99] Craig Labovitz, Abha Ahuja, and Farnam Jahanian. Experimental Study of Internet Stability and Backbone Failures. In *Proceedings of the Twenty-Ninth Annual International Symposium on Fault-Tolerant Computing*, FTCS '99, pages 278–, Washington, DC, USA, 1999. IEEE Computer Society.

[LAWD04] Lun Li, David Alderson, Walter Willinger, and John Doyle. A first-principles approach to understanding the Internet's router-level topology. In *ACM SIGCOMM*, pages 3–14, Portland, OR, 2004.

[LAWV01] Craig Labovitz, Abha Ahuja, Roger Wattenhofer, and Srinivasan Venkatachary. The impact of Internet policy and topology on delayed routing convergence. In *INFOCOM*, Anchorage, AK, USA, April 2001.

[LBCX02] Anukool Lakhina, John W. Byers, Mark Crovella, and Peng Xie. Sampling biases in IP topology measurements. In *In IEEE INFOCOM*, pages 332–341, 2002.

[LBU09] Anthony Lambert, Marc-Olivir Buob, and Steve Uhlig. Improving internet-wide routing protocols convergence with MRPC timers. In *CoNEXT '09: Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 325–336, New York, NY, USA, 2009. ACM.

[LCMF08] Yan Li, Jun-Hong Cui, Dario Maggiorini, and Michalis Faloutsos. Characterizing and modelling clustering features in AS-Level Internet topology. In *INFOCOM*, pages 271–275, 2008.

[LGGZ08] Yong Liao, Lixin Gao, Roch Guerin, and Zhi-Li Zhang. Reliable interdomain routing through multiple complementary routing processes. In *Proceedings of the 2008 ACM CoNEXT Conference*, CoNEXT '08, pages 68:1–68:6, New York, NY, USA, 2008. ACM.

[LGW$^+$07] Jun Li, Michael Guidero, Zhen Wu, Eric Purpus, and Toby Ehrenkranz. BGP routing dynamics revisited. *Computer Communications Review*, April 2007.

[LIJM$^+$10] 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 SIGCOMM*, SIGCOMM '10, pages 75–86, New York, NY, USA, 2010. ACM.

[LK99] Averill Law and W. David Kelton. *Simulation Modeling and Analysis.* McGraw-Hill Higher Education, 1999.

[LKF07] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph Evolution: Densification and Shrinking Diameters. *ACM Trans-*

actions on Knowledge Discovery from Data (ACM TKDD), 2007.

[LMJ97] Craig Labovitz, G. Robert Malan, and Farnam Jahanian. Internet routing instability. *SIGCOMM Comput. Commun. Rev.*, 27(4):115–126, 1997.

[LMJ99] Craig Labovitz, G. Robert Malan, and Farnam Jahanian. Origins of Internet routing instability. In *Proceedings IEEE INFOCOM 1999*, New York, NY, March 1999.

[LMP$^+$06] Mohit Lad, Dan Massey, Dan Pei, Yiguo Wu, Beichuan Zhang, and Lixia Zhang. PHAS: a prefix hijack alert system. In *Proceedings of the 15th conference on USENIX Security Symposium - Volume 15*, Berkeley, CA, USA, 2006. USENIX Association.

[LT06] Nenad Laskovic and Ljiljana Trajkovic. BGP with an adaptive minimal route advertisement interval. In *IPCCC*, 2006.

[MBGR03] Z. Morley Mao, Randy Bush, Timothy G. Griffin, and Matthew Roughan. BGP beacons. In *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 1–14, New York, NY, USA, 2003. ACM.

[Mey08] David Meyer. The Locator/Identifier Separation Protocol (LISP). *The Internet Protocol Journal*, 11(1), Mar 2008.

[MGVK02] Zhuoqing Morley Mao, Ramesh Govindan, George Varghese, and Randy H. Katz. Route flap damping exacerbates Internet routing convergence. *SIGCOMM Comput. Commun. Rev.*, 32(4):221–233, 2002.

[MHK$^+$07] Priya Mahadevan, Calvin Hubble, Dmitri Krioukov, Bradley Huffaker, and Amin Vahdat. Orbis: rescaling degree correlations to generate annotated internet topologies. In *SIGCOMM*, pages 325–336, 2007.

[Mil84] D. L. Mills. Exterior gateway protocol formal specification. RFC904, 1984.

[MKF$^+$06] Priya Mahadevan, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Xenofontas Dimitropoulos, kc claffy, and Amin Vahdat. The Internet AS-level topology: three data sources and one definitive metric. *SIGCOMM Comput. Commun. Rev.*, 36(1):17–26, 2006.

[MKFV06] Priya Mahadevan, Dmitri Krioukov, Kevin Fall, and Amin Vahdat. Systematic topology analysis and generation using degree correlations. In *SIGCOMM*, pages 135–146, 2006.

[MLMB01] Alberto Medina, Anukool Lakhina, Ibrahim Matta, and John Byers. BRITE: An approach to universal topology generation. In *Proceedings of IEEE MASCOTS*, pages 346–353, 2001.

[MMB00] Alberto Medina, Ibrahim Matta, and John Byers. On the origin of power laws in Internet topologies. *SIGCOMM Comput. Commun. Rev.*, 30:18–28, April 2000.

[Moy98] John Moy. OSPF version 2. RFC2328, 1998.

[MWA02] Ratul Mahajan, David Wetherall, and Tom Anderson. Understanding BGP misconfiguration. In *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '02, pages 3–16, New York, NY, USA, 2002. ACM.

[MXZ$^+$05] Xiaoqiao Meng, Zhiguo Xu, Beichuan Zhang, Geoff Huston, Songwu Lu, and Lixia Zhang. IPv4 address allocation and the BGP routing table evolution. *SIGCOMM Comput. Commun. Rev.*, 35:71–80, January 2005.

[MZF07] David Meyer, Lixia Zhang, and Kevin Fall. Report from the IAB workshop on routing and addressing. http://tools.ietf.org/id/draft-iab-raws-report-02.txt, April 2007.

[NAN09] www.mail-archive.com/nanog@nanog.org/msg15962.html, October 2009.

[Ng04] James Ng. Extensions to BGP to support secure origin BGP (soBGP). Internet draft draft-ng-sobgp-bgp-extensions-02.txt, Apr 2004.

[O+90] Dave Oran et al. OSI IS-IS intra-domain routing protocol. RFC1142, 1990.

[OPW+10] Ricardo Oliveira, Dan Pei, Walter Willinger, Beichuan Zhang, and Lixia Zhang. The (in)completeness of the observed Internet AS-level structure. *IEEE/ACM Trans. Netw.*, 18:109–122, February 2010.

[OZP+06] Ricardo Oliveira, Beichuan Zhang, Dan Pei, Rafit Izhak-Ratzin, and Lixia Zhang. Quantifying path exploration in the Internet. In *IMC*, Rio de Janeiro, Brazil, October 2006.

[PAMZ05] Dan Pei, Matt Azuma, Dan Massey, and Lixia Zhang. BGP-RCN: improving BGP convergence through root cause notification. *Comput. Netw.*, 48:175–194, June 2005.

[PCC10] Packet Clearing House. http://www.pch.net, 2010. "Online; accessed 15-Dec-2010".

[PJL+10] Jong Han Park, Dan Jen, Mohit Lad, Shane Amante, Danny McPherson, and Lixia Zhang. Investigating occurrence of duplicate updates in BGP announcements. In *PAM*, pages 11–20, 2010.

[PK08] Alex Pilosov and Tony Kapela. Stealing the Internet: An Internet Scale Man-In-The-Middle Attack. Presentation at Defcon 16, Aug 2008.

[PPU05] Alin C. Popescu, Brian J. Premore, and Todd Underwood. Anatomy of a Leak: AS9121. NANOG 34, May 2005.

[PVA+09] B. Aditya Prakash, Nicholas Valler, David Andersen, Michalis Faloutsos, and Christos Faloutsos. BGP-lens: patterns and anomalies in internet routing updates. In *KDD*, pages 1315–1324, 2009.

[QdSFB09] Bruno Quoitin, Virginie Van den Schrieck, Pierre François, and Olivier Bonaventure. IGen: Generation of router-level internet topologies through network design heuristics. In *Proceedings of the 21st International Teletraffic Congress*, September 2009.

[QHL05] Jian Qiu, Ruibing Hao, and Xing Li. The optimal rate-limiting timer of BGP for routing convergence. *IEICE Transactions*, 88-B(4):1338–1346, 2005.

[QU05] Bruno Quoitin and Steve Uhlig. Modeling the routing of an autonomous system with C-BGP. *IEEE Network*, 19(6), November 2005.

[qua] Quagga project website. http://www.quagga.net/.

[RF06] Anirudh Ramachandran and Nick Feamster. Understanding the network-level behavior of spammers. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '06, pages 291–302, New York, NY, USA, 2006. ACM.

[rip] RIS Routing Beacons. www.ripe.net/ris/docs/beacon.html.

[RIP08] RIPE NCC. YouTube Hijacking: A RIPE NCC RIS case study. www.ripe.net/news/study-youtube-hijacking.html, Mar 2008.

[RIR10a] American Rregistry for Internet Numbers (ARIN). https://www.arin.net, 2010. "Online; accessed 15-Dec-2010".

[RIR10b] Reseaux IP Europeens (RIPE). http://www.ripe.net, 2010. "Online; accessed 15-Dec-2010".

[RL93] Yakov Rekhter and Tony Li. An architecture for IP address allocation with CIDR. RFC1518, Sep 1993.

[RL95] Yakov Rekhter and Tony Li. A border gateway protocol 4 (BGP-4). RFC1771, March 1995.

[RLH06] Yakov Rekhter, Tony Li, and Susan Hares. A border gateway protocol 4 (BGP-4). RFC4271, January 2006.

[rou] Routeviews project page. http://www.routeviews.org.

[RRG] Internet Research Task Force, Routing Research Group. trac.tools.ietf.org/group/irtf/trac/wiki/RoutingResearchGroup.

[RRG10] http://www.mail-archive.com/rrg@irtf.org/msg02714.html, March 2010.

[RRIS] RIPE's Routing Information Service. http://www.ripe.net/ris/.

[RWI10] RIPE WHOIS database. http://www.db.ripe.net/whois, 2010. "Online; accessed 15-Dec-2010".

[RWXZ02] Jennifer Rexford, Jia Wang, Zhen Xiao, and Yin Zhang. BGP routing stability of popular destinations. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement*, pages 197–202, New York, NY, USA, 2002. ACM.

[SCE$^+$05] Lakshminarayanan Subramanian, Matthew Caesar, Cheng Tien Ee, Mark Handley, Z. Morley Mao, Scott Shenker, and Ion Stoica. HLP: a next generation inter-domain routing protocol. In *Proceedings SIGCOMM*, pages 13–24, Philadelphia, USA, August 2005. ACM.

[Scu07] John G Scudder. [idr] re: [rrg] re: BGP path hunting, MRAI timer and path length damping. Message to the IDR mailing list, http://www1.ietf.org/mail-archive/web/idr/current/msg02415.html, jun 2007.

[SFF02] Georgos Siganos, Michalis Faloutsos, and Christos Faloutsos. The Evolution of the Internet: Topology and Routing. *University of California, Riverside technical report*, 2002.

[SFPB09] Virginie Schrieck, Pierre Francois, Cristel Pelsser, and Olivier Bonaventure. Preventing the unnecessary propagation of BGP withdraws. In *Proceedings of the 8th International IFIP-TC 6 Networking Conference*, NETWORKING '09, pages 495–508, Berlin, Heidelberg, 2009. Springer-Verlag.

[sid] Secure Inter-Domain Routing (sidr). http://datatracker.ietf.org/wg/sidr/charter.

[SMWA04] Neil Spring, Ratul Mahajan, David Wetherall, and Thomas Anderson. Measuring ISP topologies with rocketfuel. *IEEE/ACM Trans. Netw.*, 12:2–16, February 2004.

[SP06] Philip Smith and Christian Panigl. Ripe routing working group recommendations on route-flap damping. http://www.ripe.net/docs/ripe-378.html, May 2006.

[Spe04] Charles Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 15:72–101, 1904.

[ssf] SSFNet website. http://www.ssfnet.org/.

[STR06] Srihari R. Sangli, Dan Tappan, and Yakov Rekhter. BGP Extended Communities Atribute. RFC4360, Feb 2006.

[TMS01] Paul Traina, Danny McPherson, and John G. Scudder. Autonomous Sesteem Confederations for BGP. RFC3065, Feb 2001.

[TMS07] Paul Traina, Danny McPherson, and John G. Scudder. Autonomous system confederations for BGP. RFC5065, Aug 2007.

[TMSV03] Renata Teixeira, Keith Marzullo, Stefan Savage, and Geoffrey M. Voelker. Characterizing and measuring path diversity of Internet topologies. In *Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS '03, pages 304–305, New York, NY, USA, 2003. ACM.

[Tuk77] John Wilder Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[U.S03] U.S. Department of Homeland Security. The National Strategy to Secure Cyberspace, 2003.

[Var04] George Varghese. *Network Algorithmics*. Morgan Kaufmann, San Fransisco., 2004.

[VC07] Quaizar Vohra and Enke Chen. BGP support for four-octet AS number space. RFC4893, May 2007.

[VCG98] Curtis Villamizar, Ravi Chandra, and Ramesh Govindan. BGP Route Flap Damping. RFC 2439, Nov 1998.

[VGE00] Kannan Varadhan, Ramesh Govindan, and Deborah Estrin. Persistent route oscillations in inter-domain routing. *Computer Networks*, 32:1–16, Jan 2000.

[VIRZZ05] Ricardo V.Oliveira, Rafit Izhak-Ratzin, Beichuan Zhang, and Lixia Zhang. Measurement of highly active prefixes in BGP. In *Proceedings IEEE Globecom*, 2005.

[VQB09] Laurent Vanbever, Bruno Quoitin, and Olivier Bonaventure. A hierarchical model for BGP routing policies. In *Proceedings of the 2nd ACM SIGCOMM workshop on Programmable routers for extensible services of tomorrow*, PRESTO '09, pages 61–66, New York, NY, USA, 2009. ACM.

[WADL04] Walter Willinger, David Alderson, John C. Doyle, and Lun Li. More "normal" than normal: scaling distributions and complex systems. In *Proceedings of the 36th conference on Winter simulation*, WSC '04, pages 130–141. Winter Simulation Conference, 2004.

[Wax88] Bernard M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, 1988.

[WGJ+02] Walter Willinger, Ramesh Govindan, Sugih Jamin, Vern Paxson, and Scott Shenker. Scaling phenomena in the Internet: Critically examining criticality. *National Academy of Science*, 99:2573–2580, 2002.

[WGWQ05] Feng Wang, Lixin Gao, Jia Wang, and Jian Qiu. On understanding of transient interdomain routing failures. In *Proceedings of the 13TH IEEE International Conference on Network Protocols*, pages 30–39, Washington, DC, USA, 2005. IEEE Computer Society.

[WJ02] Jared Winick and Sugih Jamin. Inet-3.0: Internet Topology Generator. Technical Report UM-CSE-TR-456-02, EECS, University of Michigan, 2002.

[WMRW05] Jian Wu, Zhuoqing Morley Mao, Jennifer Rexford, and Jia Wang. Finding a needle in a haystack: pinpointing significant BGP routing changes in an IP network. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2*, NSDI'05, pages 1–14, Berkeley, CA, USA, 2005. USENIX Association.

[WMW+06] Feng Wang, Zhuoqing Morley Mao, Jia Wang, Lixin Gao, and Randy Bush. A measurement study on the impact of routing events on end-to-end Internet path performance. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '06, pages 375–386, New York, NY, USA, 2006. ACM.

[WRCS10] Daniel Walton, Alvaro Retana, Enke Chen, and John Scudder. Advertisement of multiple paths in BGP. Internet draft, draft-ietf-idr-add-paths-04.txt, Aug 2010.

[WZMS07] Jian Wu, Ying Zhang, Z. Morley Mao, and Kang G. Shin. Internet routing resilience to failures: analysis and implications. In *CoNEXT '07*, pages 1–12, New York, NY, USA, 2007. ACM.

[WZP+02] Lan Wang, Xiaoliang Zhao, Dan Pei, Randy Bush, Daniel Massey, Allison Mankin, S. Felix Wu, and Lixia Zhang. Observation and analysis of BGP behavior under stress. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, IMW '02, pages 183–195, New York, NY, USA, 2002. ACM.

[XGF07] Jianhong Xia, Lixin Gao, and Teng Fei. A measurement study of persistent forwarding loops on the Internet. *Comput. Netw.*, 51:4780–4796, December 2007.

[ZKL+05] Beichuan Zhang, Vamsi Kambhampati, Mohit Lad, Daniel Massey, and Lixia Zhang. Identifying BGP routing table transfers. In *MineNet '05: Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, pages 213–218, New York, NY, USA, 2005. ACM.

[ZLMZ05] Beichuan Zhang, Raymond Liu, Daniel Massey, and Lixia Zhang. Collecting the Internet AS-level topology. *SIGCOMM Comput. Commun. Rev.*, 35:53–61, January 2005.

[ZPMZ05] Beichuan Zhang, Dan Pei, Daniel Massey, and Lixia Zhang. Timer interaction in route flap damping. In *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, pages 393–403, Washington, DC, USA, 2005. IEEE Computer Society.

[ZPW$^+$01] Xiaoliang Zhao, Dan Pei, Lan Wang, Dan Massey, Allison Mankin, S. Felix Wu, and Lixia Zhang. An analysis of BGP multiple origin AS (MOAS) conflicts. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, IMW '01, pages 31–35, New York, NY, USA, 2001. ACM.

[ZZM$^+$05] Xiaoliang Zhao, Beichuan Zhang, Daniel Massey, Andreas Terzis, and Lixia Zhang. The impact of link failure location on routing dynamics: A formal analysis. In *ACM SIGCOMM Asia Workshop*, April 2005.

[ZZM$^+$07] Ying Zhang, Zheng Zhang, Zhuoqing Morley Mao, Charlie Hu, and Bruce MacDowell Maggs. On the impact of route monitor selection. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 215–220, New York, NY, USA, 2007. ACM.

Appendix A

List of Acronyms

AS Autonomous System

BGP Border Gateway Protocol

CIDR Classless Inter-Domain Routing

DFZ Default Free Zone

eBGP external Border Gateway Protocol

EGP Exterior Gateway protocol

FIB Forwarding Information Base

IAB Internet Architecture Board

IANA The Internet Assigned Numbers Authority

iBGP internal Border Gateway Protocol

IGP Interior Gateway Protocol

IP Internet Protocol

MED Multi-Exit Discriminator

MHD Multihoming Degree

MRAI Minimum Route Advertisement Interval

RFD Route Flap Damping

RIB Routing Information Base

RIR Regional Internet Registry

TCAM Ternary Content Addressable Memory

TCP Transmission Control Protocol