A New Addressing and Forwarding Architecture for the Internet

by

Cong Guo

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2011

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The current Internet routing and addressing architecture is facing a serious scalability problem. The default free zone (DFZ) routing table size grows at an increasing and potentially alarming rate. The Internet architecture uses a single namespace - the IP address, to express two functions about a network entity: its identifier and locator. This overloading of semantics leads to the scalability problem as a consequence of multihoming, traffic engineering, and nonaggregatable address allocations. The current Internet architecture does not inherently support emerging features such as mobility either.

This thesis presents a simple addressing and forwarding architecture (SAFA) for the Internet. SAFA separates the locator namespace from the ID namespace so that the locators can follow the hierarchies in the Internet topology and be aggregated. The locators are allocated dynamically and automatically. The hierarchical format of locators gives end systems more control over the route selection. A straightforward forwarding scheme is designed based on the hierarchical addressing scheme. The meshed part of the Internet topology is integrated into the forwarding procedure through a special forwarding table. With a rendezvous service that maps from IDs to locators, SAFA also provides scalable support for mobility, multihoming and traffic engineering. Our work also includes an Internet topology study and a prototype implementation of the architecture. The evaluation results suggest that SAFA would be feasible in the current Internet if deployed.

Acknowledgements

I would like to thank all the people who made this possible.

I would like to thank my supervisor Martin Karsten for his encouragement and help. I could not have completed the work without his insightful advice and guidance.

Dedication

This is dedicated to my family.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

> "Any problem in computer science can be solved with another layer of indirection, but that usually will create another problem."
>
> David Wheeler

Today's Internet routing system is facing a serious scalability problem. Routing tables (also referred to as the Routing Information Base, or RIB) in the default-free zone (DFZ) already contain more than 330,000 prefixes and continue to grow super-linearly [1]. The forwarding information base (FIB), which is computed from RIB, also suffers from such a growth. Moreover, an increase in the routing table size (number of routable prefixes) also increases churn, since the number of networks that can fail or trigger a route change increases. Because the present Internet routing protocol, the Border Gateway Protocol (BGP) [45] advertises route updates globally, the rate of churn becomes another problem [21]. The rapid growth of the DFZ RIB has caused concerns among network operators and researchers.

Address aggregation is a method of grouping specific related addresses into a more general address which represents the whole group. With address aggregation, the routing system just needs to maintain one entry for a group of entities instead of its constituents. Address aggregation is the only known practical approach to control the growth of the DFZ RIB. In the current Internet, IP addresses are designed to represent the network attachment points of entities. They ought to have topological relevance so that the IP prefixes can be aggregated. However, they are also used to identify entities in network communication, and thus they should be independent of locations. This is the commonly known ID/locator problem. The overloading of IP address semantics is thought to be the reason for prefix de-aggregation that leads to the uncontrolled growth of DFZ RIB size. The scalability problem is solvable, for the immediate future, by adding more

memory, but an alternative routing and/or addressing architecture is no doubt worth investigating. On one hand, research on this topic will be a long-term effort. Early actions should be taken to avoid rushed work under deadline pressure. On the other hand, such a kind of research could bear fruit for something other than raw scaling, e.g., better support for traffic engineering, multihoming and mobility.

A simple addressing and forwarding architecture (SAFA) for the Internet is proposed in this thesis. Rather than adding another layer of indirection and making systems more complicated, SAFA aims to consider the essentials for an Internet architecture and provide a simple yet comprehensive design. In addition, SAFA inherently supports significant communication patterns and engineering techniques such as multicast, multihoming, traffic engineering, and mobility. Our architecture also gives end users a greater control over route selection.

The main objective of this thesis is to study a potential direction of Internet addressing and forwarding, not to provide a ready-to-use system. This thesis makes the following contributions:

- Analyzing essential requirements and basic components of an Internet architecture
- Designing a simple but comprehensive addressing and forwarding architecture for the Internet
- Evaluating the architecture using an Internet topology study
- Implementing a prototype of the architecture.

The remainder of the thesis is organized as follows. The problem and requirements are explained in Chapter 2. The network model used in this thesis is also introduced in this chapter. Chapter 3 discusses previous related proposals and introduces some work related to the Internet topology as a necessary background. The architecture design is presented in detail in Chapter 4. Chapter 5 presents a study of the Internet topology and evaluates the proposal based on the results of the study. A prototype implementation is described in Chapter 6. Chapter 7 concludes the thesis and illustrates some areas for future work.

Chapter 2

Problem Statement

2.1 Scalability of the Internet Routing System

The scalability problem of the current addressing and routing system is a major concern for the Internet community. This problem has two aspects. One is the increasing DFZ RIB size. The number of entries was about 150,000 in 2005; in 2010 this number has reached more than 330,000 [1], and could reach 2 million within 10 years [37]. The super-linear growth of the DFZ RIB size means both software and hardware challenges for the backbone networks of the Internet. Its implications include the time cost of recomputing the FIB, the BGP convergence time, and cost and power consumption of the hardware needed in the DFZ. The other aspect of the scalability problem is the increasing rate of BGP updates (i.e., BGP churn rate). It has a negative impact on routing convergence, since updates frequently necessitate a re-computation and download of the FIB. It is shown that BGP churn even increases at a much faster pace than the routing table size [21]. Even though the size of the RIB is bounded by the given address space size and the number of reachable hosts (still a very huge number), the amount of protocol activity required to distribute dynamic topological changes is not. Part of the growth of DFZ RIB size results from the natural evolution of the Internet, while a large portion is due to the de-aggregation of IP prefixes. A workshop held by the Internet Architecture Board (IAB) identified four main factors as the main driving forces behind the rapid growth of the DFZ RIB: non-aggregatable address allocations, multihoming, traffic engineering, and business events [37]. The first three of them are technical factors involved in this thesis.

IP addresses are designed to be aggregatable at first, but in practise, many prefixes are not allocated according to the network topology and cannot be aggregated at all. There are two kinds of IP address allocation policies: provider independent (PI) and provider allocated (PA). A PI ad-

dress space is assigned by a regional Internet registry (RIR) directly to an end-user organization. PI prefixes are injected into the routing tables directly and cannot be aggregated. Customers generally prefer to have a PI space because this can provide them agility in selecting ISPs and helps them avoid the need to renumber when changing ISPs. Site renumbering can be quite difficult. It is true that lots of end systems get addresses through DHCP (Dynamic Host Configuration Protocol [19]) today, but renumbering does not only mean modification of a modest number of routers and servers and update of some DNS records. For some networks, numerous necessary changes make renumbering effectively impossible in reality. This is because IP addresses are often used for other purposes like access control and status tracking. They are even hard-coded into applications sometimes.

Multihoming is typically used to describe the case in which a site is served by more than one transit provider [9]. Multihoming provides backup routing, i.e., Internet connection redundancy. Multihoming can be realized through either PA or PI address spaces. For sites using a PI address space, their prefixes, which cannot be aggregated must be present in the routing and forwarding tables of all their providers. For sites with a PA address space, each prefix allocated by one provider can only be aggregated by that provider. However, the prefixes are advertised by other ISPs in addition to the primary ISP. Due to the longest prefix matching rule, the primary ISP has to de-aggregate the customer's prefix to keep the customer's traffic flowing through itself instead of others.

Traffic engineering (TE) is the act of arranging for certain network traffic to use or avoid certain network paths. It is inter-domain traffic engineering that produces non-aggregatable prefixes. At the inter-domain level, if the address range requiring traffic engineering is a portion of a larger PA address aggregate, network operators are forced to de-aggregate otherwise aggregatable prefixes in order to steer the traffic of the particular address range to specific paths. In this case, the addresses do not carry topological information but work as identifiers. The support for multihoming and traffic engineering is discussed in detail in Section 2.4.

Besides the major factors above, several other issues also accelerate the growth of routing tables. The first one is a general concern with IPv6 deployment. IPv6 shares the same semantics with IPv4. The deployment of IPv6 lifts the constraint that the IPv4 address space has put on the IPv4 RIB growth. In the absence of a scalable routing strategy, the rapid DFZ RIB size growth problem today can potentially be exacerbated by IPv6's much larger address space. Second, it is commonly believed that hardware technology will continue to scale at a rate that surpass the growth rate of routing information handled by DFZ routers because of Moore's law. However, Moore's Law applies specifically to the high-volume portion of the semiconductor industry, while the low-volume, customized silicon used in core routing is well off Moore's Law's cost curve [37]. The third issue is the misalignment of costs and benefits in today's routing system. An AS that performs de-aggregation achieves a benefit, but the global Internet incurs the cost of carrying

the additional prefixes.

2.2 Components of an Internet Architecture

When researchers designed the Internet decades ago, they clarified basic components of the current Internet architecture [44, 46]. "The 'name' of a resource indicates what we seek, an 'address' indicates where it is, and a 'route' tells us how to get there." "Forwarding is the relaying of packets from one network segment to another by nodes in a computer network." However, as IP addresses are used to identify resources, the definitions of these concepts become fuzzy. Moreover, the apparently straightforward process of forwarding turns out to be surprisingly complex. This section reconsiders basic components of an Internet architecture and their relationships, and clarifies their essential functions.

2.2.1 Overloading of IP Address Semantics

The difficulties discussed in Section 2.1 fundamentally arise from the overloading of IP address semantics (both IPv4 and IPv6). An IP address serves two principal functions: host or network interface identification and location addressing. At first they are designed as locators, including a network part and a host part. The network part that represents the hierarchy of networks can be aggregated. However, currently IP addresses are also used to identify end points. As introduced in the previous section, the PI prefixes effectively become identifiers of networks. Another typical example is the session management of the transmission control protocol (TCP). TCP does not include a session identifier, and both endpoints identify the session using the client's IP address and port number. IP addresses are also used on higher layers directly. They are used for access control in systems like firewall and for status tracking in many web applications.

There is an inherent incongruence between the two roles of, locator and identifier. The function of locators is to carry topological information and reflect changes of network attachment points dynamically. However, identifiers are typically assigned independent of topological structures and should be stable. IDs are difficult to be aggregated in the routing system. To solve the scalability problem, locator and identifier functionalities must be separated. In fact, most proposals to scale the routing system are based on the ID/locator split idea. Details of relevant proposals are introduced in the next chapter.

2.2.2 Basic Components

This section reconsiders the concepts in an Internet architecture and identifies five basic components based on essential requirements. For the convenience of discussion, we first define some concepts and terms which are used throughout this thesis.

Entity : An entity is an independent unit in network communication that contains both application states and communication states.

The term ”entity” is deliberately chosen to be abstract. An entity might be a host, a router, a network interface. The basic question of network communication is how to make data reach a specific entity or entities.

Initiator : An initiator is the entity that initiates a communication procedure.

Responder : A responder the entity that communicates with the initiator.

A typical communication procedure is as follows. First, an initiator should be able to identify the responder no matter where it is. Then it needs to find out where the responder is in some way. Here a data structure is required to describe where the destination entity is and its format must be recognized by all entities. In addition, entities need a mechanism to exchange information about how to get to others, and select one route if there are multiple choices. Finally, since not all the entities are adjacent directly, the initiator sends data to its neighbor according to the route if necessary and the entities on the route relay the data towards the responder. Then the responder sends response data to the initiator in a similar way. The above intuitive reasoning leads to the following observation: an Internet architecture essentially involves five basic components, an identifier namespace to identify entities, an addressing scheme to locate entities, a rendezvous service to map identifiers to locators, a routing component to propagate route information and a forwarding component to handle the actual delivery.

Identifier

An intuitive design of the identifier namespace is to assign a globally uniform identifier (ID) to each entity. However, even with the same basic function, i.e., identifying entities, IDs have different requirements in different situations. For example, in terms of today’s Internet, a ”name”, which is a unique human-understandable symbol for network resources, can be considered as a kind of identifier. In some scenarios like instant messaging, access control and user status tracking, IDs do not have to be human-understandable. A numerical format is more efficient. Currently, IP addresses are used in these kinds of scenarios. Different network applications can have different identifier namespaces based on their own requirements.

Locator

This namespace can also be referred to as ”address”. Because of the fuzziness of the IP address semantics, for clarity, this thesis uses the term ”locator” to refer to this namespace and ”address” is only used for ”IP address”. A locator is used to describe the attachment point of an entity in the Internet topology. Locators also provide routing directives to entities. If locators can be aggregated according to the topology, the routing component just needs to maintain one entry for a group of entities instead of its constituents before aggregation. Thus the scalability problem is solved. The locator namespace should be able to describe different units in the Internet topology. Current IPv4 addresses have a problem in this aspect. An autonomous system (AS) is a connected group of one or more IP prefixes run by one network operator which has a single and clearly defined routing policy. ASes are the unit of routing policy in BGP, and routing information is represented by AS paths. However, IPv4 addresses cannot express such an unit. Thus another namespace ”AS number” is used to fix this problem.

Rendezvous Service

Rendezvous service is the component maintaining mappings from entities' IDs to their locators. It consists of *rendezvous servers* for entities to look up others' locators when initiating network communication. This component can also provide other functions like forwarding first packets for entities. Rendezvous service is a significant component in the ID/locator split. The DNS service in today's Internet is a kind of rendezvous service. If there are multiple identifier systems employed in the Internet, there should be multiple corresponding rendezvous services to meet various requirements respectively.

Routing Component

The routing component is responsible for exchanging information amongst entities about how to reach others. Its main tasks include route exploration, route selection, and dynamic reachability information maintenance.

Forwarding Component

Forwarding is the process by which an entity, on receiving data, decides where to forward the data, and then actually forwards them. The forwarder determines the next hop to forward a packet by looking up the destination locator in forwarding tables, while the data in the forwarding tables come from the routing component.

This thesis focuses on the design of addressing and forwarding schemes, and the general requirements of other components are also investigated. The detailed study of other components is future work.

2.3 Network Model

This section defines the network model of the Internet topology used in this thesis. The actual Internet topology is a meshed graph with some hierarchies (referred to as the core component and the tree component in [22]). The network model defined in this section captures hierarchical and non-hierarchical structures in the Internet topology based on routing policies of networks.

First, some clarification of terms needs to be made. In the current Internet addressing system, the term "prefix" as it is used here is equivalent to "CIDR block" [27], and an AS is a connected group of one or more IP prefixes. In this network model, the term "network" is an abstract concept. It can be a prefix, or an autonomous system (also referred to as a routing domain). SAFA does not put a limit on the organization of addressing unit.

The Internet is considered as a directed graph G with networks as its vertices $V(G)$. If there are links between two networks A and B, their corresponding vertices u and v are adjacent in the graph and there are two edges between them, (u,v) and (v,u). Two kinds of annotations are used to label the edges, up and down. An edge from vertex u to vertex v, (u,v) can be either an up edge or a down edge.

Definition 1 *The largest subset of vertices such that every two vertices are adjacent to each other is the core of the Internet,* $C \subseteq V(G)$.

We assume that there is only one such subset.

Definition 2 *A network that accepts only traffic to itself is a stub vertice. S is the set of stub networks.* $S \subseteq V(G)$.

Definition 3 *Let* $u,v \in V(G)$. *If the edge from u to v* (u,v) *is a down edge, v is a terminating vertex of u,* $v \in T(u)$, *and all the terminating vertices of v are terminating vertices of u,* $T(v) \subseteq T(u)$.

Definition 4

(1)Let $u \in \overline{S}, w \in S$. *If u and w are adjacent, the edge* (u,w) *is a down edge.*

(2)Let $u,v \in \overline{S}$. *If u and v are adjacent and v accepts traffic from u only when the traffic goes to either itself or its terminating vertices, the edge* (u,v) *is a down edge.*

Definition 5

(1)Let $u \in \overline{C}, w \in C$. *If u and w are adjacent, the edge* (u,w) *is an up edge.*

(2)Let $u,v \in \overline{C}$. *If u and v are adjacent and v accepts from u traffic that goes through its up edges, the edge* (u,v) *is an up edge.*

Recursively, we can define all the edges based on Def.4 (1) and Def.5 (1). All the edges between core vertices are down edges.

The network model can be extended to include internal entities of networks. For end points like hosts and mobile devices, they only accept traffic to themselves from other entities, so they are stub vertices. The definitions of edges also apply to these internal entities.

The relationship between two networks can be defined based on the edges between the vertices representing them. If there is an up edge from v to u and there is a down edge from u to v, u is a *provider* of v while v is a *customer* of u. If the two edges between two vertices are both down edges, the two networks have a *peering* relationship; if the two edges are both up edges, the relationship between them is *sibling*.

The provider-customer relationship leads to hierarchies in the graph. A hierarchical structure starts at a vertex which is a provider, and includes the customers of the provider, recursively its indirect customers, and the edges between these vertices. However, the whole Internet topology is not a strict tree structure. First, there are multiple core vertices instead of a single root of this structure. Second, one vertex may be simultaneously connected to multiple providers. Most importantly, there are peering relationship and sibling relationship between vertices. These edges cross different hierarchies. In addition, we assume there is no provider-customer cycle in the graph. In practice, a provider network will not acquire transit service from a network that is its own customer, or its indirect customer.

2.4 Other Considerations

Besides the fundamental communication requirements, an addressing and forwarding architecture for the Internet should support significant communication patterns and features inherently. The communication patterns and features arise from realistic applications. Some of them are factors that contribute to the scalability problem of the current Internet architecture, such as multihoming and traffic engineering. This section reviews the requirements from these issues.

2.4.1 Mobility

The mobility in the current Internet practice can be categorized into four classes: fast endpoint mobility, slow endpoint mobility, slow network mobility, and fast network mobility. Fast endpoint mobility occurs when an entity moves relatively rapidly, changing its network layer attachment point. Maintenance of session continuity is a goal. Mobile devices like cell phones are

typical examples of this class. Virtual machine mobility in data centers also belongs to this class because the virtual machines always work as responders and hope to maintain their availability during the movement. Slow endpoint mobility means some individual entities want to move, but are not concerned about maintaining session continuity. All the records involving their locators should be changed. Slow network mobility means the whole network wishes to change its attachment points to the Internet. Renumbering is involved in this case, too. Fast network mobility also exist in the Internet, e.g., the in-flight networks. These networks change their attachment points to the Internet on a global scale rapidly.

Fast endpoint mobility involves a large number of mobile entities. IPv4 addresses work as both IDs and locators, but the namespace is not large enough. Service providers currently use NAT to meet the demand. Providing Internet connectivity to a mobile entity is twofold, access control and mobility management. Cellular networks achieve global access control using unique identifiers and a global provider database, but today's Internet does not have an inter-provider access control mechanism. This issue depends on business relationships between service providers and this is beyond the scope of this thesis. As for mobility management, all the existing solutions can be broadly classified into two basic approaches. The first method is dynamic routing. In the current IP system, a mobile entity keeps its IP address regardless of its location changes. The routing system continuously keeps track of mobiles' movements and reflect their current positions in the routing table. IP addresses are used both to identify and locate mobile entities. Currently, because the whole network must be informed of every movement by every mobile, this approach does not scale well for a large number of mobile entities. The other method is to set up an anchor point. The anchor point has a stable locator which is used to identify mobile entities connected to the anchor point. The anchor point maintains mobile entities' current locators and forwards traffic to them. Most current systems utilize this approach, such as the home agent in Mobile IP and the GGSN in GPRS. As for fast network mobility, the Connexion service [20] for in-flight networks took the dynamic routing approach, where BGP is used to propagate airplane location updates. However, this service has stopped while current in-flight networks are overlaid on cellular networks. An airplane is treated as a mobile node and the devices inside get their private IP addresses through NAT. In other words, an extra layer is introduced for in-flight networks.

Renumbering is the major operation for slow endpoint mobility and slow network mobility. The problem is how to reduce the workload produced by renumbering. First, the assumption that locators are permanent should be avoided and the unnecessary use of locators should be reduced. Second, the locators should be configured automatically and dynamically to avoid manual effects and reduce mistakes [34].

2.4.2 Multicast

Multicast means the delivery of data to a group of entities. Multicast is widely deployed in enterprises, commercial stock exchanges, and multimedia content delivery networks. A multicast group address is actually an identifier of a group of topologically independent locations, instead of a locator. The group address does not determine the location of the receiver(s). Thus, to support multicast, the forwarding infrastructure ought to employ translation functions. To be compatible with the unicast forwarders, multicast packets should share the same format with unicast packets. In the current Internet architecture and proposals that support multicast, certain prefixes are reserved to identify multicast groups.

Another character of multicast is that the sender(s) need to send only one copy of data while the forwarders duplicate the packets when necessary. Therefore a management protocol is required. First, the receivers need to register in the multicast group. Second, when a receiver joins a multicast group, the forwarders serving that receiver's network need to know that the receiver has joined so that they can arrange for multicast traffic destined for that group to reach it. The management protocol is in charge of setting up the multicast forwarding state in necessary forwarders.

2.4.3 Multihoming

In the network model, a multihomed vertex has multiple up edges originating from it. Multihoming provides backup links or network redundancy to increase the reliability of network applications. In addition, in mobile environments, multihoming can help to solve the problem of migrating between different types of networks while roaming. Multihoming requires publishing multiple routes that point to the same entity in the routing system. At the same time, the primary provider should be distinguished from others. Multihoming is a significant factor of IP address disaggregatioin. BGP uses a prefix as the identifier of the multihomed entity, and thus the prefix can not be aggregated. To solve this problem, ID and locator namespaces should be split and the multihoming information, i.e., the mapping between one ID and multiple routes should be published via a rendezvous service.

2.4.4 Traffic Engineering

The goal of traffic engineering is to select better routes when forwarding. First, there should be enough information to decide which route is better. The granularity of addressing should be flexible so that locators can describe routes as needed. Routing protocols should provide

enough information. Second, it is easy for forwarders to direct flows to the routes selected. This section focuses on inter-domain traffic engineering. Intra-domain traffic engineering is easy because network operators have control over what their own routers do. There are already mature techniques for intra-domain traffic engineering. New Interior Gateway Protocols (IGPs) can take advantages of previous work. Inter-domain traffic engineering is significantly more complicated than intra-domain traffic engineering since it involves multiple network operators.

Inter-domain traffic engineering has two typical tasks, balancing load across multiple links from/to a neighboring domain and directing traffic from/to a different neighbor. The techniques can be classified into two categories according to the directions, inbound and outbound. Inbound traffic engineering requires mechanisms or attributes that allow networks to show their preferred links for the flows into them. It is not easy for the current IP/BGP system to achieve the inbound TE. Tricks like prepending the AS path are used. Outbound traffic engineering is easier but also needs to consider the influence on the neighbors. Three TE example applications are as follows:

Congested edge link: The links between domains are common points of congestion in the Internet. Upon detecting an overloaded edge link, an operator can change the inter-domain paths to direct some of the traffic to a less congested link.

Upgraded link capacity: Operators of large backbones frequently install new, higher-bandwidth links between domains. Exploiting the additional capacity may require routing changes that divert traffic traveling via other edge links to the new link.

Violation of peering agreement: An AS pair may have a business arrangement that restricts the amount of traffic they exchange; for example, the outbound and inbound traffic may have to stay within a factor of 1.5. If this ratio is exceeded, an AS may need to direct some traffic to a different neighbor.

In addition, the TE techniques should also limit the influence of neighboring networks and reduce the overhead of global routing changes.

2.4.5 Source Routing

Source routing means the forwarding path is specified by the end systems. Currently, end systems have no control over the forwarding of their packets at the inter-domain level. However, user choice is beneficial for the creation of a healthy and competitive ISP market [14, 52]. Source routing is also required in some special situations like avoiding network censorship. To support source routing, the addressing scheme should be able to represent the entire route in packet headers. The header overhead of the route representation is a concern. With source routing, a provider no longer has the control to pick the cheapest next hop to reach a destination, so the providers should be compensated in some way.

2.5 Summary

This section summarizes the design goals for an Internet architecture.

First of all, the locator namespace should be separated from the ID namespace. It is possible that multiple ID namespaces and rendezvous service systems are employed. The locator namespace should be independent of concrete designs of ID namespaces and rendezvous services. The ID/locator separation has two benefits. Locators can be assigned in an aggregatable way to scale the routing component. Moreover, each entity can have multiple locators and selects one to use according to specific requirements. The mapping from one ID to multiple locators is provided by a rendezvous service. Thus support for multihoming is achieved without causing the scalability problem. Once one locator is no longer reachable, the multihoming entity can use another one to restore connectivity quickly.

Second, given that locators do not cover the meshed parts of the Internet topology, the architecture design should contain mechanisms that allow the forwarding component to make use of this kind of links.

Third, locators should be allocated dynamically and automatically. Thus, locators can reflect the changes of entities' attachment points in the Internet topology agilely through updates in the rendezvous service. Meanwhile, the automatic allocation can reduce the cost and possible mistakes of manual operations during renumbering. Previous work also suggests that manual work during renumbering should be avoided [34].

Fourth, source routing is a beneficial technique for the Internet. To support source routing, the source locator in each packet header should represent the forwarding routes that end systems select. The forwarding scheme should also include source-locator-based forwarding.

Finally, global routing information is necessary for route selection at the inter-domain level. However, it is not necessary to make local routing events globally visible like what BGP does. The routing information distribution should be restricted.

Chapter 3

Background and Related Work

There are kinds of proposals that aim to cover the problems of the current Internet architecture. This chapter analyzes some typical ones and presents an overall comparison between them based on the basic components and design goals presented in Chapter 2. Understanding the Internet topology, in particular, the AS-level Internet topology is very important for the evaluation of new Internet addressing and routing proposals. Therefore this chapter also reviews previous work on the characteristics of the Internet topology. Since network operators do not want to publish their relationships with neighbor networks, AS relationship data can only relay on inference. Important AS relationship inference algorithms are reviewed in Section 3.2.2.

3.1 Internet Architecture Proposals

A number of recent Internet architecture proposals revolve around the idea of ID/locator split. Meanwhile, they also have their respective characteristics. This section introduces a representative subset of this kind of approaches. On the other hand, some research efforts aim to resolve the scalability problem by designing a new inter-domain routing protocol. An approach in this direction is also introduced in this section.

3.1.1 HIP

The Host Identity Protocol (HIP) [38] separates identifier and locator namespaces. The separate ID namespace in HIP is called Host Identity, which is a globally uniform namespace. An host identity is based on an asymmetric key pair. The public key is called Host Identifier (HI), while

the host keeps the private key for self-authentication. The length of identifiers from different algorithms may be different, so all of them are hashed to a global Host Identity Tag (HIT) with a fixed length (128 bits). HI is a flat namespace without any hierarchy. HIP inserts a host identity layer into packet headers, between the network layer and the transportation layer. HIP still utilizes IP addresses as its locators. HIP can use only IPv6 headers because of the length of HIT. It continues to use IPv6 protocols for routing and packet forwarding.

An altered version of DNS with two new types of records works as the rendezvous service which maps HITs onto IP addresses [39]. Because DNS cannot update its records rapidly, the HIP architecture introduces another rendezvous mechanism called rendezvous servers to support mobility management [32]. Rendezvous servers will forward the first several HIP packets for initiators, while the rest of packets will be transferred between initiators and responders directly. If the locator of one host changes, the host needs to notify its rendezvous server.

HIP requires host changes, but the design still utilizes IPv6 addresses as locators, and makes no change in routing and forwarding components. HIP supports endpoint mobility and multi-homing. There is no traffic engineering or multicast consideration in the proposal. Its advantage is the cryptographic IDs which introduce the authentication mechanism into the Internet architecture.

3.1.2 LISP

The Locator/Identifier Separation Protocol (LISP) [24] uses two namespaces: Endpoint Idenfitiers (EIDs), which are assigned to only endpoints like hosts, and Routing Locators (RLOCs), which are assigned to only entities (primarily routers) that make up the global routing system. According to the specification, an EID should be routable within the domain of the entity, while RLOCs are allocated according to the network topology strictly and support prefix aggregation. In the pilot network of this project, IP addresses (both IPv4 and IPv6) are used as EIDs and RLOCs. LISP packets carry EIDs in the ”inner-header” while use RLOCs as outer source and destination locators. Two kinds of network elements are responsible for encapsulation and decapsulation: the Egress Tunnel Router (ETR) and the Ingress Tunnel Router (ITR). ITR accepts IP packets from systems and performs LISP encapsulation. If necessary, it will perform EID-to-RLOC mapping queries to resolve the destination RLOC. ETR accepts packets whose destination locator is its own RLOC and sends decapsulated IP packets to site end systems.

Multiple rendezvous service systems mapping EIDs to RLOCs have been presented. The LISP-ALT (Alternative Logical Topology) [23] is introduced in this section, which is the design from the same research group. ALT is a kind of overlay network, which is based on BGP and

Generic Routing Encapsulation (GRE) [25]. ALT makes use of BGP to maintain the reachability information of EID prefixes, to advertise and aggregate EID prefixes. ALT does not store RLOCs information. The mappings are stored in ETRs, while ALT just forwards initial data packets (called Data Probe packets) and reply messages for ITRs and ETRs respectively. All subsequent packets are sent directly between the ITR and the ETR. The topology of ALT is a tree-like hierarchical structure. The ALT routers aggregate the prefix data from the lower layers and inform the higher layer routers. LISP does not introduce new forwarding and routing mechanisms. BGP is still used for routing. Note that because BGP reveals complete path information, LISP can handle peering links between domains, so it does not have special locators for peering-links.

The strategy LISP uses is called ”Core-Edge Separation”. Local routing is based on IDs, while each packet flow goes through an encoder which attaches the locator for routing within the core. The ID namespace of LISP may not be aggregatable. BGP is used for the maintenance of EID information in ALT, so the scalability problem is actually moved to the rendezvous service. In addition, the status synchronization of the rendezvous service is also a problem. One of the authors discusses this issue in [36].

Getting locators at data-plane time is another problem, and this is actually a common problem of all the ID/locator split proposals. One significant advantage of LISP is that it does not need to make changes to hosts. LISP is a network-based solution and does not require any host change. It is applicable for both IPv4 and IPv6. Moreover, LISP supports traffic engineering and mobility.

3.1.3 NIRA

The New Internet Routing Architecture (NIRA) [52] focuses on two problems of the present Internet routing system: the lack of user choice and the scalability problem. It is actually based on the ID/locator split, even though the authors make no such claim. In NIRA, to bootstrap a communication, an initiator needs to know the responder’s ”name”. The design contains a rendezvous service called Name-to-Route Lookup Service (NRLS) which maps the name of a responder to the route segments. Their design does not clarify the concept of ”name”.

A provider-rooted addressing scheme is adopted in the proposal. Locators are allocated by providers and can be aggregated. Locators are 128 bits in length. Every 16-bit piece represents a domain layer, so one locator supports only 8 layers, including the end-point layer. Domains can have private address space rooted at themselves, and the locators for peers can be allocated from this space.

The rendezvous service of NIRA, NRLS can be considered as an enhanced DNS. The servers store optionally topology information as well as the locators. NIRA does not architecturally

constrain how record updates should be done, which means it does not handle the fast update explicitly. Because the locators of NRLS servers are hard-coded, NIRA requires that the root NRLS servers reside in the Core, which consists of all the Tier-1 providers to make sure that a resolver can always reach a root server.

The forwarding of NIRA is still based on the longest prefix match. A router keeps three logical forwarding tables: the uphill forwarding table, the downhill forwarding table, and the bridge forwarding table, serving the providers, customers and peers respectively. A new inter-domain routing protocol is introduced in this architecture, the topology information propagation protocol (TIPP). TIPP operates outside the Core of the Internet. TIPP uses separate messages for locator information and topology information. Locator information is the prefixes. The topology information is represented by a set of link records identified by two domain identifiers. The locator propagation part is straightforward: notify customers of prefix updates. The part that distributes topology information is a policy-controlled link state protocol. There are two types of control: scope enforcement and information hiding. Scope enforcement means a domain can only send messages downward a provider hierarchy. Information hiding supports policy routing.

NIRA handles addressing, routing and forwarding, but does not have much consideration on the ID namespace and the rendezvous service. One obvious advantage of this proposal is source routing, and it also takes multihoming into account. However, there is no consideration for traffic engineering, mobility and multicast.

3.1.4 Pip

Paul's Internet Protocol (Pip) [26] is an internet protocol intended as the replacement for IPv4 and is another specific instance of the ID/locator split.

Pip uses 64-bit global unique IDs. Pip IDs are hierarchically structured, but they are treated as flat, and the hierarchy is solely for the purposes of insuring uniqueness. [26] leaves whether or not Pip IDs should contain significant organizational hierarchy information as an open issue, because such a kind of hierarchy would complicate the assignment. Both the source ID and the destination ID are carried in Pip headers.

The addressing scheme of Pip is provider-rooted. The Pip Header encodes locators as a series of separate numbers, one number for each level of hierarchy. A variable number of 16-bit FTIFs (Forwarding Table Index Fields) are carried in each Pip packet header. Note that a single "number" may in fact be more than 16 bits in length, and encoded as multiple FTIFs. The low-order part of each FTIF indicates the relationship of the FTIF with the next FTIF: vertical, horizontal, or extension. The vertical relationship means hierarchically above or below.

The horizontal relationship indicates hierarchically unrelated, through which Pip can handle the peering links naturally.

The Pip architecture uses DNS as the rendezvous service directly. In Pip, the information stored in DNS does not change often. Hosts inform communication peers of their new locators directly when they change their locations. Meanwhile, a service called mobile host servers is introduced to handle mobility management. When both entities in a communication session are mobile and cannot exchange packets at all, they can query each other's mobile host servers, whose locators are stored in the DNS records.

Pip's forwarding is based on the so-called FTIF chain. There is an active FTIF field in the packet header telling the router which FTIF to use. At the top-level of the Pip locator hierarchy, a path-vector routing algorithm is used. At any level below the top level, it is a local decision as to what routing algorithm technology to run.

3.1.5 HLP

Hybrid Link-state Path-vector routing protocol (HLP) is not based on the locator/ID split idea [48]. It leverages the hierarchies based on provider-customer relationships in the Internet topology, and routes at the granularity of AS's instead of prefixes. HLP uses link-state routing within each provider-customer hierarchy while it employs path-vector routing across hierarchies. Thus, HLP prevents local routing events such as routing updates, configuration errors, and policy enforcement from being propagated globally. HLP can reduce the churn resulting from renumbering and mobility, but it does not support these features completely, because it does not solve the semantic overlapping of IP addresses.

The paper claims that HLP can support traffic engineering. However, the design makes use of a (AS, prefix) mapping table to achieve the prefix-level route selection. Obviously, there still exists prefix de-aggregation in the mapping table. No detailed mechanisms of this table have been provided in the paper. With the forwarding based on prefix matching, each border router needs to query this mapping table, and it should be updated dynamically, so the maintenance cost is still large.

3.1.6 Discussion

Multiple proposals for the future Internet architecture employ globally uniform and permanent identifiers [24, 38, 26, 41, 40]. It is a straightforward design. Global uniqueness enables the entities to roam everywhere while preserving their ongoing communication sessions. However,

Table 3.1: Comparison of Different Approaches

Approach	ID/locator split	Aggregatable Locator	Automatic Allocation	Source Routing	Restricted Info Distribution
LISP	Yes	Yes	No	No	No
HIP	Yes	Yes	No	No	No
NIRA	Yes	Yes	No	Yes	Yes
PIP	Yes	Yes	No	Yes	N/A
HLP	No	No	No	No	Yes

such a namespace also complicates the identifier design and management. Due to the number of possible entities, if a globally uniform identifier namespace is employed, its size would be very large. The storage would become a concern and thus a hierarchical structure should be employed. When an entity moves out of its original organization, its ID cannot be aggregated. The management and storage of the "unaggregatable" IDs would be another scalability problem.

Some of the designs like [38, 24] still use BGP as the routing component. BGP uses a flat routing structure which impairs its scalability. Local routing events can be seen globally. It is fundamentally hard to isolate routing events. Moreover, the resulting interdependence between ASes makes the entire Internet vulnerable to localized security or configuration problems; a single configuration error or compromised router can affect the rest of the network.

DNS is successful as a rendezvous service mapping domain names to IP addresses. Some proposals reuse DNS as their only rendezvous service. However, DNS cannot update rapidly, so it cannot be used in the scenarios that require fast update, such as mobility. Proposals like [26, 38] already notice this problem and try to modify DNS.

In Table 3.1, a qualitative comparison of different approaches is made based on the requirement analysis in Section 2.5. It can be seen that none of the approaches meet all the design goals. NIRA meets most of the requirements. However, NIRA is not a complete solution. It neither provides a clear explanation of ID namespace(s) nor takes features like mobility and traffic engineering into account. The architecture presented in this thesis SAFA satisfies all the design goals listed. Moreover, SAFA takes the advantages of the proposals surveyed above such as source routing in NIRA and the stack-like locator in Pip.

3.2 AS-Level Internet Topology

The knowledge of the Internet topology plays a critical role in many research and operational tasks ranging from network resilience study to peer selection or data center location. The design and evaluation of a new Internet addressing and forwarding architecture also relies on the Internet's topological structure. All the Internet architecture proposals should prove that they can work well with the current topology and foreseeable changes. Autonomous systems are the unit of routing policy in BGP. This section discusses the research efforts mining data that capture information about ASes and exploring properties of associated graphs on the AS-level.

3.2.1 Internet Topology Model

ASes have different business relationships with each other. The relationships introduce routing policies that have a profound influence on how traffic and economic value flows through the Internet. Gao's work [28] categorizes three main types of relationships between ASes in general: provider-customer, peering, and sibling. In the provider-customer category, an AS (customer) pays the other AS (provider) for the access to remote parts of the Internet. In the peering category, the two ASes exchange traffic without paying each other. The sibling relationship usually happens between two ASes that belong to the same organization and is relatively rare in today's Internet.

The following selective export rules arise from the three business relationships [49]:

Exporting to a provider: In exchanging routing information with a provider, an AS typically exports its routes and routes of its customers, but will not export routes learned from other providers or peers.

Exporting to a peer: In exchanging routing information with a peer, an AS exports its routes and the routes of its customers, but will not export routes learned from other providers or peers.

Exporting to a customer: An AS can export its routes, routes of its customers, and routes learned from other providers and peers to its customer.

Denote a link from a customer to a provider with a -1, a link between peers with a 0, and a link from a provider to a customer with a +1. Given that if an edge is not exported to an AS, it will not be used to forward traffic for that AS, the following conclusion emerges from the export rules:

If every domain obeys the export policies, then every inter-domain forwarding path must belong to one of these two types for some $M, N \geqslant 0$:

1) Type-1: -1,... (N times), +1,... (M times).

2) Type-2: -1,... (N times), 0, +1,... (M times).

The first stage of a Type-1 path contains only customer-provider links (uphill segment) and the second stage contains only provider-customer links (downhill segment). The second type captures all paths which traverse exactly one peering link. The single peering link must appear in between the uphill and the downhill portions of any path. This path model is referred to as valley-free model.

Besides the AS-path model, existing research efforts aim to model the whole AS-level Internet topology. One of the most cited papers, [22], presents three power-laws of the Internet topology. This paper models the Internet as an undirected graph. First, ASes are ranked according to their outdegrees. They show the power-law relationship exists between the outdegree of a node and the rank of the node. The frequency of an outdegree value and the outdegree also have such a relationship. The third pair is the eignvalue of a graph and the order of the eignvalue. However, in [13] it is argued that the degree distribution in the Internet does not obey a strict power-law distribution, just heavy-tailed.

[12] proposes a model using recursive decomposition called k-shell. The process starts by removing nodes with degree 1 until no more such nodes remain, and assigns them to 1-shell. In the same manner, the nodes with degree 2 up to k are removed until all the nodes have been assigned to one of the shells. The nodes in the $k_{max} - shell$ form the nucleus of the Internet.

[50] also models the Internet topology recursively, but in a different way. First, the authors define the maximal clique that contains the highest-degree node as the core of the Internet. Then, they define the first layer to contain all the nodes that are neighbors of the core. Recursively, layer n contains all the neighbors of layer $n-1$ except the nodes already defined. The model has a total of 6 layers, with the core as layer 0. The size of the core in this model they measured was 20-25 ASes.

3.2.2 AS Relationship Inference

In reality, network operators determine routing policies according to the business relationship between ASes. However, public data from various inter-domain routing data sources show only AS adjacency. Network operators consider the details of their business relationships as proprietary information and do not generally make them public. Therefore, Internet researchers have to rely upon indirect AS relationship inference algorithms to compute an approximation.

Gao's pioneering work [28] is the first algorithm that infers ISP business relationships using information from publicly available BGP routing tables. Gao defines the AS paths that obey the

valley-free model as valid paths. The algorithm first identifies the AS with the maximum degree (the number of ASes connected to a given AS) in a given path as the top provider. Then each AS link in the path is assigned a relationship following the assumption that the path is valid. Peering and sibling links are identified based on AS degrees.

Subramanian et al. slightly relax the problem by not inferring sibling links, and provided a more elegant mathematical formulation based on the concept of valid paths [49]. Assuming maximization of the number of valid paths as a natural objective, this paper formulates the AS relationship inference problem as a combinatorial optimization problem: given an undirected graph G derived from a set of BGP paths P, assign the edge type (provider-customer or peering) to every edge in G such that the total number of valid paths in P is maximized. The problem is called the type-of-relationship (ToR) problem. This work conjectures that ToR is NP-complete, and provides a heuristic solution (referred to as SARK). The SARK approach takes as input the BGP tables collected at different vantage points and computes a rank for every AS. This rank is a measure of how close to the graph core an AS lies. The heuristic then infers AS relationships by comparing ranks of adjacent ASes. If the ranks are similar, the algorithm classifies the link as peering relationship, otherwise it is provider-customer relationship.

Di Battista et al. [17] and Erlebach et al. [11] independently prove that the ToR problem is indeed NP-complete. More importantly, both papers demonstrate that peering links cannot be inferred in the ToR problem formulation and develop mathematically rigorous approximate solutions to the ToR problem but inferring only provider-customer links. Note that neither [49] nor [28] offers a solution to the problem of reliable identification of peering links due to their low accuracy as demonstrated by Oliveira et al [42]. Di Battista et al. develop a new peering relationship inference technique in [17]. They show that the problem of finding a maximal set of peering links that do not introduce invalid paths in P is equivalent to the Maximum Independent Set (MIS) problem.

In addition to its inability to infer peering links, there are other issues with the ToR formulation that is identified in [49]. In particular, for some links either relationship (customer-to-provider or provider-to-customer) results in the same number of invalid paths. As a result, ToR labels such links randomly, classifying them as c2p or p2c with 50%-50% probability. In some cases this approach leads to obviously incorrect inferences, e.g., well-known large providers are inferred as customers of small ASes. This issue is resolved in [18] by using multiobjective optimization techniques incorporating both the notion of valid paths and AS importance as reflected in AS degree. To increase the reliability of peering link determination, the authors introduce link weights based on AS degree and turn the MIS problem into the Maximum Weight Independent Set (WMIS) problem [18].

3.2.3 Topology Characteristics

This section introduces previous work studying characteristics of the Internet topology, such as the depth of provider-customer hierarchies, multihoming degree and the average hops of routes. These topology properties are used for the evaluation of the Internet architecture designs. The HLP paper emulates the effect of churn/fault on an Internet topology computed from real routing data to prove the protocol's scalability [48]. The input to the emulator is an AS topology and the set of inter-AS relationships. The NIRA researchers measure the number of address prefixes allocated to each domain based on an AS-level topology inferred using the same algorithm [52]. The result shows that 90% of the domains have less than 20 prefixes and this number does not grow with the size of Internet. They also measure the number of forwarding entries a router has. About 90% of the domains have less than 100 forwarding entries. [48] and [52] make use of the inference result from [49].

A measurement study reports some growth trends of hierarchical structures of the Internet from 1998 to 2007 [15]. First their results show that the Internet grew exponentially in terms of the number of ASes and links over the ten years. The average AS path length has remained practically constant at 4.2 AS hops. During the same period, the multihoming degree defined as the number of providers of a given AS had been increasing. This implies that the Internet retains a certain hierarchical structure, and that the depth of that structure does not seem to vary with the size of the Internet. This paper classifies the ASes into four types according to their business type and size. The four types are enterprise customers(EC), small transit providers (STP), large transit providers (LTP) and content/access/hosting providers (CAHP). This study finds that the population of ECs that have few customers shows a strong growth trend. The CAHP population, even though small in absolute number, has also been growing significantly. However, the LTP population remains a constant number and the growth rate of STPs is not significant after 2001. The number of transit providers is stabilizing. Another interesting result this paper reports is the evolution of the median peering degree for the four AS types. ECs and STPs have median peering degree of zero. During the ten years they studied, the median peering degree of CAHPs has increased significantly from 2 to 10. The largest number and the highest growth rate is for links of the type CAHP-STP.

Some reports focus on properties of the undirected graph of ASes, ignoring the relationships between them. A study measured the average degree and effective diameter of the Internet AS graph and concluded that the AS graph is densifying [33]. Two other measurement studies [35, 47] consider the evolution of the Internet topology in the period 1997-2001 with respect to several microscopic and macroscopic properties. Siganos et al. [47] observe the exponential growth of the Internet during that time period, and shows that a rich-get-richer form of preferential attachment leads to exponential growth in the number of edges. Magoni et al. [35] examine the

evolution of some global Internet characteristics and find exponential growth in the number of ASes and links during that time period.

Chapter 4

Architecture Design

4.1 Architecture Overview

This chapter presents the design of the simple addressing and forwarding architecture (SAFA) for the Internet. SAFA separates the locator namespace from the ID namespace. Each entity can have multiple locators representing different routes towards it. The locators of an entity can be looked up in the rendezvous service using the entity's ID. This thesis does not design a specific ID namespace. The requirements of ID systems come from higher layers, which depend on specific protocols and/or applications. With proper rendezvous services, SAFA can support various ID namespaces.

The locators are allocated according to the provider-customer hierarchies in the Internet topology. Thus they can be aggregated to solve the scalability problem. The locators are allocated dynamically and automatically using leases. A SAFA locator is a stack of fixed-length integer labels. Except for the core networks, labels are allocated by providers locally. The explicit locator format facilitates source routing. Each locator represents an absolute route from the core towards an entity. They can be used anywhere in the Internet forwarding infrastructure. SAFA also has an optional private addressing mechanism. Private locators represent relative routes between entities. By default, the addressing and forwarding schemes introduced in following chapters are for the "public" locators, while the private addressing mechanism is introduced particularly in Section 4.2.4.

The forwarding scheme is based on the addressing scheme. The forwarding route is represented by the combination of the source locator and the destination locator. The whole forwarding process contains three stages: uphill, optional bridge and downhill stage, which use their

own forwarding tables respectively. In general, the next hop lookup is mainly based on the destination locator. If end systems want to use source routing, the uphill forwarding is done via multi-field matching of the source locator. The route representation of SAFA does not contain any peering locator explicitly. The forwarding through peering links are achieved by looking up the destination locator in a separate forwarding table.

The rendezvous service associates the identifier namespace(s) with the locator namespace in SAFA. Entities can determine which locators to publish in the rendezvous service, so that they have more control over inbound route selection. The rendezvous service also plays an important role in scalable support for mobility. There ought to be various rendezvous service systems for various ID namespaces and various applications. In addition, the locator allocation protocol can provide only local route information among neighbors. Entities need global routing information to select better inter-domain routes. Therefore a routing component is still required. The routing component also needs to take charge of dynamic attributes of routes and failure management. SAFA focuses on addressing and forwarding schemes. This thesis only discusses the requirements of the rendezvous service and routing component. The detailed designs will be done in the future work .

4.2 Addressing Scheme

> "Addressing can follow topology or topology can follow addressing. Choose one."
>
> Rekhter's Law

4.2.1 Locator Allocation

SAFA adopts an addressing scheme that follows the Internet topology. In particular, we make use of the hierarchies in the Internet's topology (see Section 2.3) to achieve locator aggregation. In SAFA, a locator is a stack of fixed-size integer labels. The networks in the core are assigned a unique label each by a centralized authority, such as IANA, and their labels are at the top of the locator stack. Then at the inter-domain level, each network allocate a local label to its customer with its own locators as prefixes. The provider does not have to inform customers of all its locators. It can select some of its locators according to its routing policy. All the border routers of a network share the locator data about their network through an internal protocol or a central database. As in the network model, here a network refers to an autonomous system or a so-called IP prefix. At the intra-domain level, the operators have more flexibility in addressing.

Each entity can be assigned a label. If available labels are not sufficient, operators can split one network into subnets and add a layer internally.

Using integer labels guarantees that locators are not overlapping. The allocation also obeys the non-looping rule, i.e., a network is prohibited from accepting a locator A if it has another locator that is the prefix of A. One disadvantage of this provider-rooted addressing is that when a network changes its provider, all the prefixes from this provider need to be changed dynamically. To reduce the cost and manual mistakes of renumbering, all the locators in SAFA are allocated automatically based on a lease. The allocation protocol can be considered as an extended version of DHCP (See Chapter 6 for the specification of the automatic allocation protocol).

This addressing scheme delegates as much control as possible to local agreement and requires only minimal global administrative coordination. Moreover, it does not require network operators to publish more internal topology information about their networks than current IP addressing and BGP routing.

4.2.2 Locator Format

This section specifies the design rationale of the locator format. In principle, a locator could be either fixed length or variable length. For a provider-rooted addressing scheme, it is difficult to decide the size of a fixed-length locator. If the size is too small (IPv4 currently), the available locators are not sufficient. If the size is too big (like IPv6), there will be a waste of both the locator space and packet header space. Currently according to [29], /48 prefixes are allocated to subscribers by RIR and /64 prefixes are allocated to subnets. It means each subscriber can have 65536 subnets. In reality, a lot of prefixes allocated are not used. The depth of provider hierarchy varies at different parts of the network, and the size of a network also varies. According to $L = n \times l$ where L is the locator length, n is the number of layers, and l is the length of each layer. Given a fixed L, either n or l might not be enough.

The locator format in SAFA is a variable-size stack of fixed-length labels. By allowing variable-length locators, the overall network does not have to settle for an either too small or too big locator space. The matching of fixed-length labels is more efficient than the matching of variable-length ones. The size of a label has implications on the overall header size and data structures for forwarding tables. It will typically be a power of 2. Fundamentally, this proposal is independent of the actual size that would be standardized for labels. 16 bits should be a reasonable choice. According to our study in Chapter 5, this number is sufficient at both intra-domain level and inter-domain level.

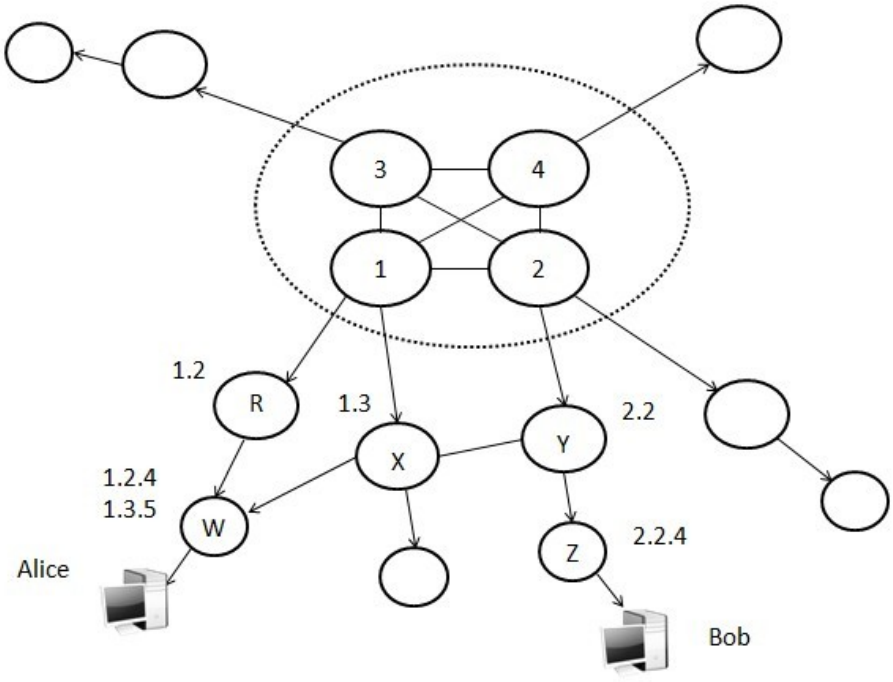

Figure 4.1: An example of provider-rooted hierarchical addressing

4.2.3 Example

Figure 4.1 shows an example of the strict provider-rooted hierarchical addressing scheme.

We represent a locator and a locator prefix using a notation similar to IPv4, where "." is a separator that separates every 16-bit label. For example, 1.2.3 designates a locator that has 3 labels. The first label has the value 1, and the second label has the value 2. This example shows four core networks each of which obtains a globally unique label, from 1 to 4. For the convenience of illustration, we use letters to identify some networks. In reality, they are not used in the addressing scheme. The core network 1 allocates labels 2 and 3 to network R and X respectively, so their complete locators would be 1.2 and 1.3 and their common customer W would have two locators, 1.2.4 and 1.3.4. A host in the network W, Alice, also has two locators. Similarly, network Z can get a locator 2.2.4. In this example we assume that all the prefixes are allocated to the customer, but in practise, a provider can just allocate some of its own locators as prefixes according to its policy.

4.2.4 Private Locators

Private Locators represent relative routes between entities and cannot be used globally. This mechanism can be used to optimize some local communication applications. When a network change its provider, the corresponding prefixes need to be changed, but the labels it allocated to internal entities do not. Based on this observation, network applications at the intra-domain level do not need to use complete locators. The prefixes from providers can be stripped. Then when the network change a provider, even static configurations of the applications using private locators do not need to be changed. Thus the cost of renumbering can be reduced considerably.

At the inter-domain level, private locators increase flexibility to accommodate special connectivity scenarios efficiently. For instance, networks with the peering relationship can allocate private labels to each other. With this kind of private locators, forwarding packets between peering networks or through a route consisting of multiple peering links just requires single-field matching. The packets using private locators should be distinguished from normal packet. This can be achieved by setting a special bit in the top label, because there are only a few core networks and they do not need all the 16-bit space.

4.3 Packet Forwarding

4.3.1 Route Representation

A route representation scheme has four basic design requirements. First, forwarding operations based on the representation scheme should be efficient; second, a receiver should be able to generate a reverse route representation to send a reply from a packet it receives easily; third, the number of bytes in a packet header to represent a domain-level route should be minimized; finally, for inter-domain routing, the policy checking overhead using a route representation should be low.

Our route representation is based on our addressing scheme. A typical route is represented by the source locator and destination locator carried in packet headers. We assume that all the routes obey the valley-free model introduced in Section 3.2.1. Hence an entire route can be separated into three segments: uphill segment, optional bridge segment, and downhill segment. Source locators represent the uphill segment and destination locators represent the downhill segment. There is no explicit bridge part in our route representation scheme. Each forwarder holds a separate bridge forwarding table. Once an entry to a packet's destination locator is found, the forwarder will forward the packet through the peering link. Because the bridge forwarding is

achieved through destination locator matching, even if there are multiple peering links in the bridge stage, our architecture can handle that without increasing header overhead.

Our route representation scheme meets the basic requirements. The downhill forwarding stage just needs single-field matching. Uphill forwarding is always done by default forwarding. There are not many entries in a bridge forwarding table. Thus the forwarding is efficient. The reverse route representation can be generated via source locators easily. (In SAFA, the reverse route can also be obtained from the rendezvous service.) The length of locators is always shorter than that of IPv6. Because locators implicate the relationships between domains, the policy checking is convenient, too.

4.3.2 Forwarding Tables

Forwarding tables that contain (locator, next hop) entries are used to determine the next hop to forward a packet. The forwarding procedure of SAFA involves three forwarding tables: downhill forwarding table, uphill forwarding table and bridge forwarding table. The downhill forwarding table contains the labels assigned to customers. Its lookup is done through only single-label matching. The uphill forwarding table manages the entity's locators from providers which are also uphill routes. The bridge forwarding table maintains the next hop towards destinations outside the hierarchy such as peers. The uphill and bridge forwarding tables both use multi-label prefix matching. Note that the uphill forwarding table does not need the longest prefix matching because of the non-looping allocation principle. In SAFA, the data in forwarding tables are initialized by the locator allocation, and updated by both routing protocols and the locator allocation protocol.

The uphill and bridge forwarding tables can be stored in accelerated hardware like ternary content-addressable memory (TCAM) [30]. TCAM allows three possible values to be stored in a memory cell, i.e. 0, 1, or x (don't care). Variable-length prefixes are converted to fixed-length words by padding don't care bits on the right. The input locator is compared with all the prefixes stored in the TCAM array in parallel. The major advantages of TCAM-based lookup engine are twofold: (1) the simplicity of the system design, and (2) fast lookup rate brought by the parallel comparison. The major disadvantages of TCAM are its relatively high cost and high power consumption. In commercially available TCAM devices, the word length can be configured to 36,72, 144, or 288 bits. Locators in SAFA usually have fewer than 128 bits, so a word length of 144 bits will be enough. According to our simulation result in Chapter 5, the size of the tables is not very large. Small size TCAM (like 1Mb) which is not so expensive can store 6,400 entries, which are sufficient for most of cases.

4.3.3 Forwarding Algorithm

This section describes the forwarding algorithm that works with our route representation scheme and forwarding tables. The forwarding process has three stages: uphill, optional bridge, and downhill stage. Once receiving a packet, the entity determines the current forwarding stage using the destination locator. If one of the entity's locators is the prefix of the destination locator, the forwarding is at the downhill stage. Otherwise, the entity looks up the destination locator in the bridge forwarding table. If there is an entry for the prefix of the destination locator, the forwarding is at the bridge stage. If there is no match in the bridge forwarding table, the forwarding is at the uphill stage.

The downhill forwarding depends on only one label in the stack rather than the entire locator since the labels are allocated to customers locally. The label to look up is the one following the matched prefix in the uphill forwarding table. According to the valley-free model, there is at most one peering link in a route and the bridge forwarding would be just before the downhill forwarding. Entities make use of the bridge forwarding table to find the point turning to downhill stage. At the uphill stage, if the end systems request source routing, the packets are forwarded according to the source locator; otherwise, if knowing nothing about the destination, entities just forward the packets to a default provider.

The forwarding algorithm is shown in Peseudocode 1.

Algorithm 1 Packet Forwarding

src: the source locator
dst: the destination locator
match_prefix(l, T): check if any locator in table T is the prefix of locator l
lookup(l, T): find the next hop according to the prefixes of locator l in table T
single_lookup(a, T): find the next hop according to a single label a in table T

```
1:  pre ←match_prefix(dst, UP)
2:  if pre ≠ nomatch then
3:     label ← get_next(dst, pre)
4:     if label = NIL then
5:        forward(kernel)
6:     else
7:        nexthop ← single_lookup(label, DOWN)
8:        forward(nexthop)
9:     end if
10: else
11:    nexthop ← lookup(dst, BRIDGE)
12:    if nexthop ≠ nomatch then
13:       forward(nexthop)
14:    else
15:       if srf = true then
16:          nexthop ← lookup(src, UP)
17:          forward(nexthop)
18:       else
19:          forward(default)
20:       end if
21:    end if
22: end if
```

4.4 Rendezvous Service and Routing

This thesis does not provide a concrete design of a rendezvous service or routing protocols, but they still play critical roles in the Internet architecture. The following analysis shows the requirements for the two components to cooperate with our addressing and forwarding architecture.

4.4.1 Rendezvous Service

The rendezvous service, a distributed mapping system, plays an important role in our architecture. The success of DNS and middle boxes for mobility management in the cellular networks has proven that it is feasible to deploy this kind of system in the Internet. The requirements for this kind of service are summarized as follows.

High scalability: the system needs to scale up to possible billions of entries in the Internet, so it is not reasonable that every rendezvous server stores all the mappings, especially if a globally uniform ID namespace is employed.

Fast lookup: in an ID/locator split architecture, packets cannot be forwarded until the lookup is completed, so the lookup latency is significant for good performance; meanwhile, the loss of initial packets should be avoided as much as possible.

Fast update: mapping entries ought to adapt quickly with locator changes, so the consistency of caches, if there is any, is a problem.

Resilience to attacks: the rendezvous service can be a potential target for attacks, and updates to the rendezvous service or query replies from the rendezvous service must be authenticated.

Note that one rendezvous service system does not have to meet all the requirements. Different rendezvous systems can be designed for different applications, for example, one for web applications like current DNS systems and one used for mobility management.

The rendezvous service also help to solve the scalability problem of routing updates. The updates of route availability cannot be avoided. In SAFA, the updates are delegated to the rendezvous service. This prevents the churn from obstructing the routing, especially in the core of Internet.

4.4.2 Routing Subsystem

In SAFA, the locator allocation procedure and rendezvous service already share some responsibilities of routing. First, an entity assign locators to its customers according to its preference, so that it can have control over the outbound paths of uphill traffic. Second, when an entity receives locators assigned to it, it learns the uphill routes to use. The process that an entity chooses locators to publish in rendezvous service is also inbound route selection. However, all these mechanisms provide only local route information among neighbors. Entities need global routing information to select better inter-domain routes. Therefore a routing component is still required.

The routing component supports restricted distribution of routing information, both to reduce resource consumption associated with such distribution and to permit information hiding.

In addition, entities cannot determine whether the dynamic attributes of a route satisfies its requirement. The routing component ought to take charge of the distribution of dynamic routing information. The protocols can be proactive and/or reactive. Failure management is another significant task of the routing component. When the route specified in a packet header is unavailable, the router should try its best to send a control message to inform the original sender.

4.5 Other Considerations

This section introduces how SAFA supports some important communication patterns and satisfies some realistic engineering requirements. Use cases are presented in this section to show potential designs with SAFA.

4.5.1 Mobility

Section 2.4 discusses four kinds of mobility in the Internet. This section shows what support SAFA provides, but the design of SAFA does not put any limit on the specific systems.

Mobile Nodes

How to support more and more hand-held gadgets as well as other types of mobile devices in the global Internet becomes an important aspect in the design of the Internet architecture. As introduced in Section 2.4, there are two concerns about including these mobile nodes into the Internet. First is the locator namespace for such a large number of entities. In today's cellular networks, hierarchies already exist, so the addressing scheme of SAFA can adapt to the mobile entities. For instance, a cell site that creates a cell in a cellular network can be considered as an addressing unit. For other wireless Internet access technologies like Wi-Fi, the mobile entities can be considered the same as wired hosts. The other concern is mobility management. SAFA already provides infrastructure for the two typical methods of mobility management, anchor points and dynamic routing. Currently, anchor points are both rendezvous servers and forwarders. They record the dynamically changing locators of mobile entities and redirect the traffic for them. In SAFA, special forwarding entries can be set up at the anchor points to redirect the flows to the mobile nodes. However, after the ID and locator namespaces are separated, high layer protocols

and/or applications no longer need the locators of anchor points as stable IDs. Dynamic routing is simple in SAFA. Mobile nodes can update their locators in the rendezvous service as they roam and change their locators. Packets can be forwarded between the two associated entities directly without traversing the anchor points. The updates happen in the rendezvous service and have no impact on the routing and forwarding subsystems. In this case, the rendezvous service must be able to update its records rapidly.

Network Renumbering

Network renumbering, for either the entire site or some subnetworks is considered as one of the fundamental design goals [34]. An efficient renumbering is easy to achieve in SAFA. First, the locators are allocated automatically, so the renumbering process does not need manual efforts. Second, one locator in SAFA is hierarchical and its intra-domain part is allocated locally. Therefore only the involved prefix and not the entire locator needs to be changed during the renumbering process. If the applications use only the intra-domain part of a locator, they do not need to make any change. For example, in Figure 4.1, assume Bob in the network Z is assigned a label 3, and then its current locator is 2.2.4.3. When the network Z changes its provider from Y to X and is assigned a label 2, the local labels do not need to change, and the address of Bob will be 1.3.2.3. At last, using locators as IDs is not encouraged in SAFA. Only when the assumption that the locators are permanent is avoided, the cost of renumbering can be reduced fundamentally.

4.5.2 Multihoming

In SAFA, locators are no longer used to identify entities. Each entity can have multiple locators, all of which follow the provider-customer hierarchies and can be aggregated respectively. The mapping from one identifier to multiple locators is published in the rendezvous service. With necessary attributes, the entities, generally the responders, can show their preference for the locators, so the primary provider can be distinguished from other providers easily. High layer protocols can establish connections using their ID systems and cache multiple locators from the rendezvous service. If the primary route fails, they can use another one with the session continuity maintained. Meanwhile the locators are aggregated in the routing and forwarding subsystems respectively. Therefore, multihoming will not cause a scalability problem anymore.

4.5.3 Traffic Engineering

This section still focuses on inter-domain traffic engineering. For intra-domain traffic engineering, since current routing protocols can still work under SAFA, current TE techniques can be adopted to the new architecture. This thesis does not include a specific traffic engineering tool, but the architecture has provided fundamental supports to future designs. The unit of locators in SAFA is flexible, so entities can choose locators according to TE requirements without de-aggregation. Moreover, in combination with a rendezvous service, entities can show inbound TE preference.

To balance load across multiple links from/to a neighboring network, the inbound and outbound traffic engineering need similar mechanisms. One protocol is required for two adjacent networks to exchange information about the links. Both the two sides make a negotiation and decide the link. The addressing scheme allows the protocol to distinguish the links. To accept traffic from a different neighbor, the inbound TE tool can change the records in the rendezvous service, changing the preference attributes or even add/remove the locators published. The outbound TE tool can direct the traffic to a different peer via updating the bridge forwarding table. In SAFA, it is impossible to switch the traffic to a different downhill customer domain with a fixed destination locator, or at a real time scale. However, according to [10], the inter-domain traffic engineering tasks fall within the medium time scale category, i.e. minutes to days, so there is no real-time switch requirement. In addition, all the updates mentioned above will not propagate globally or affect the routing in backbone networks, since SAFA transfers the control from routing subsystem to the rendezvous service and localize the influence through the addressing scheme.

4.5.4 Multicast

To support global multicast, SAFA preserves the topmost bit of the top label to identify multicast packets. Thus multicast group IDs share the same format with normal locators. At first the multicast packets are forwarded as unicast packets using their multicast group IDs. And then forwarders supporting multicast translate multicast group IDs into specific destination locators and duplicate packets when necessary. All the forwarders supporting multicast have a special table maintaining the mappings from a multicast group ID to multiple actual locators. In addition, a multicast management protocol is required. With this protocol, entities can join a multicast group. More importantly, this protocol sets up special forwarding entries for multicast group IDs in the bridge forwarding table of forwarders along the routes. In the forwarders that need to make the duplications, the mapping from a multicast group ID to multiple specific locators are set up in the special multicast table. Destination rewriting is also done in these duplication forwarders.

Not all the multicast services are global. If the multicast group works only in a certain scope, the multicast address can be local instead of global. Since the locators in SAFA are hierarchical, the management protocol can set up special entries only in necessary local forwarders. This procedure is similar to MPLS supporting multicast [43].

Chapter 5

Internet Topology Study

5.1 Introduction

This chapter presents a study of the AS-level Internet topology which is part of the evaluation of SAFA. This study aims to understand the current status and the evolution of the Internet topology. The results show that SAFA can work well with the current topology and foreseeable changes. The study consists of two sections. First, critical topological characteristics relevant to the provider-customer hierarchies are measured. Second, the potential benefits for forwarding tables are estimated quantitatively through a simulation based on real routing data. The study utilizes a snapshot of the AS-level Internet topology, annotated with relationship information for the interconnection between ASes. Given that the ground truth is not available, the study can only rely on collections of BGP routing data and inference results.

In general, three types of Internet routing data sources are available for Internet topology studies, BGP routing tables and updates, traceroute-based tools, and RIR WHOIS databases. The Route Views project at the University of Oregon [8] collects real time BGP tables and updates data from BGP routing tables of multiple geographically distributed BGP routers. RIPE-RIS (Routing Information Service) [4] is another project that provides Internet routing information from global routing tables. This study collects AS-path data from BGP table dumps obtained from both Route Views and RIPE. The constraints of data collected through traceroute-based tools are their limited coverage and the inability to accurately converting router paths to AS paths. DIMES [6] is a promising Internet topology measurement project, but it provides only AS link data without original AS path data that are required in the study. The WHOIS data provide the registration information of network operators, including inter-AS connectivity and routing

policies. The registration is done by the network operators on a voluntary basis. It is well known that the data are incomplete and outdated, but the data can still be used to verify inference results.

There are two sets of public AS-relationship inference data. This study uses the data from [5] (referred to as CAIDA data), which are believed to achieve a high level of accuracy and consider the sibling relationship. Another public AS-relationship inference result is provided by the Internet research lab of UCLA [3] (referred to as UCLA data). Their raw data are from monitors at Tier-1 ASes. According to the introduction on the website, provider-customer links are inferred based on the valley-free model, but how to infer peering relationship is unclear. The sibling relationship is not considered in this work at all. The data set used in [48, 52], which is based on the algorithm in [49] is not available anymore.

The CAIDA data in January 2010 is selected for this study. The inference data contain 33258 ASes and 75001 links. The corresponding Internet routing data collected from Route Views and RIPE contain 33562 ASes and 71247 inter-AS links.

5.2 Internet Topology Analysis

5.2.1 Pilot Study

There are some problems with both CAIDA data and UCLA data, making them cannot be used in the study directly. First, both of them have relationship cycles. In our network model, a relationship cycle means two vertices that are not adjacent can reach each other through only down edges. For example, network A is a provider of network B; network B is a provider of network C; and network C is a provider of A. In reality, a relationship cycle means a large ISP get transit service from its customer, which is unlikely the case. Second, in the inference results, some hierarchies have too many levels (more than 20). If the deep hierarchies exist, some long AS paths should be seen in the real AS-path data, but this does not happen according to our measurement. The distribution of AS path lengths from real routing data in this study is shown in Figure 5.1. The maximal value is only 13. The average AS path length is 4.15. According to the measurement in [15], the distribution of AS-path length has been the case. Even if these deep hierarchies really exist, they are hardly, if not never, used for packet forwarding.

Relationship cycles conflict with the no-looping principle of the addressing scheme (see Section 4.2.1). Deep hierarchies mean long locators in packet headers and increase the space cost. To study the causes of the relationship cycles and deep hierarchies, we run a naive inference algorithm based on the valley-free model. The input are a dense core which is from the Tier 1

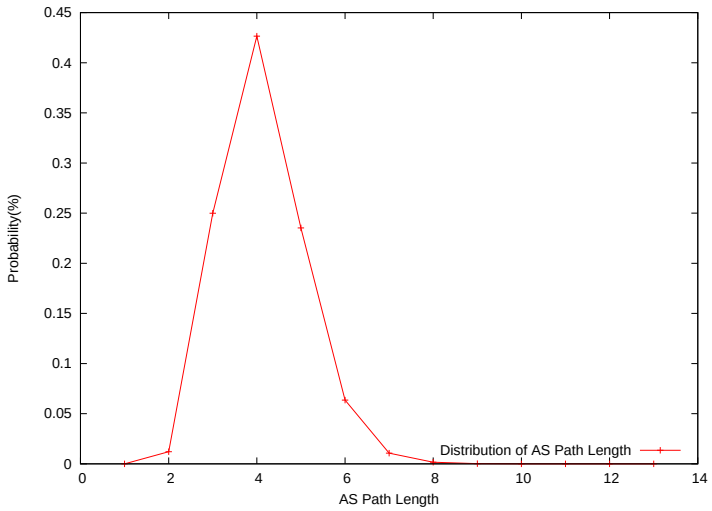

Figure 5.1: Distribution of AS-path length

network list [7] and the real AS-path data. The relationship between core networks are peering. The relationship between core networks and their neighbors are left unclassified. In the AS paths containing core networks, all the links before the first direct neighbor of core networks are considered as customer-to-provider links while all the links after the last direct neighbor of core networks are considered as provider-to-customer links. Then classify the AS paths without core networks using classified links. The links before customer-to-provider links are customer-to-provider links and the links after provider-to-customer links are provider-to-customer links. Ignore all the links left unclassified. When conflicts happen, a voting mechanism like [28] is employed. The final inference result is determined by the majority of paths.

The naive inference result also contains cycles and deep hierarchies. Through case studies we find it is difficult to get an accurate Internet topology due to the limitation of the simplified relationship model. [18] demonstrates that maximizing the number of valid paths can lead to incorrect inferences. The ultimate reason of the relationship cycles and deep hierarchies is that the simplified AS relationships used in the model doesn't really capture the full complexity of the Internet. In some cases ASes have different relationships in different geographic regions, and can even have different relationship policies on a per prefix level. Some have even more complex relationships. We compare the classified results from our algorithm with the corresponding results in CAIDA's data, 97% of the results are the same. The 3% difference suggests that these algorithms are inaccurate, but the majority of the results are probably correct.

Table 5.1: Maximum and Average Number of Customers

	0601	0606	0701	0706	0801	0806	0901	0906	1001
Maximum	2348	2332	2342	2621	2595	2528	2213	2412	2611
Average	12.6	12.7	12.9	12.9	13.1	13.1	13.5	13.7	13.9

5.2.2 General Characteristics

This section shows the statistics of the Internet topology collected from the CAIDA data.

Core

In the network model, the core of the Internet is defined as the largest subset of vertices such that every two vertices are adjacent to each other. It is assumed that there is only one such set. The addressing scheme of SAFA assumes that the size of the core is not large so that routing and forwarding in the core is efficient. According to the definition, the topology snapshot in this study contains 10 core networks.

In today's Internet, a Tier 1 network is one that can reach every other network in the Internet without purchasing IP transit or paying settlements. Tier 1 networks are located at the top of the provider-customer hierarchies. In our network model, the core networks do not need transit service either (have no up edges from them). Therefore, the scope of the core is similar to Tier 1 networks. [7] provides a list of Tier 1 networks which contains 12 ASes. The subgraph of the 12 vertices is completely connected. We define this subgraph as the dense core. The upper bound of the core of the Internet can be estimated by counting all the networks connected to the dense core as core networks. There are 3646 vertices in such a subgraph. The subgraph is not completely connected, but vertices can reach each other within 3 hops.

Each label in a SAFA locator is 16 bits. Even the upper bound of the size of the core is much fewer than 2^{16}, so the assumption that the core is small would be the case. We can make use of the space left in the top label to identify special packets like multicast packets.

Customers

The number of customers implicates the number of entries in the downhill forwarding tables. The number also determines whether 16-bit space is sufficient for a label. Figure 5.2 shows the distribution of the number of customers each AS has. The average number of customers each provider has is 14. The maximum number is 2611, far from 2^{16}. This number is also much fewer than the number of entries in the current DFZ RIB/FIB. The result shows that at the inter-domain level, the customer number is not a problem.

At the intra-domain level, network operators have more control over subnetting. If a network has more than 2^{16} internal entities and customers, it can be split into multiple subnetworks inter-

nally similar to what is done today. Even if mobile devices are taken into account, our addressing scheme still works. Hardware infrastructures already provide the hierarchical structure needed in the addressing scheme. For example, in the current cellular networks, a cell site can be used as an addressing unit. We do not have public user data to study the distribution of simultaneous users per cell site. However, this number is bounded by the subscriber capacity of cell sites that can be estimated. The simultaneous subscribers a cell site can support is calculated as follows [51]:

$$N_{subscriber} = \frac{K*S*SpectralEfficiency*Bandwidth*BusyHourLoading}{D}$$

K is a over-subscribe factor whose practical value is typically 15. S is the number of sectors one cell site has which is typically 3. D is the QoS data rate, which is usually 1.0 Mbps. The BusyHourLoading is typically set at 60%. A study shows the subscriber capacity of such an average cell site under HSPA, HSPA+, WiMAX and LTE [51]. The maximum subscriber number under 100MHz system bandwidth is less than 2500, far smaller than the space of a label 2^{16}. The maximum number is achieved by WiMAX. WiMAX is a promising candidate for 4G, so at least in the immediate future, 16 bits are enough for addressing in cellular networks.

Providers

Multihoming is a factor that affects the number of both locators and routes a network can have. Theoretically, if there are h levels in the hierarchy, and at each level, a network on average has p providers, then the number of locators an entity has scales as p^h. Business relationships and routing policies limit the exponential growth. Both p and h are not very large, and providers will not allocate all their prefixes to customers. Figure 5.3 shows the distribution of the multihoming degree, defined as the number of providers of a given AS. Although there are 60% ASes having more than one provider, the 90th percentile is only 3, which means that most of them have only two or three providers.

Figure 5.4 shows the average multihoming degree for all ASes and two classes of ASes stubs and non-stubs separately from January 2006 to January 2010. The average multihoming degree has been increasing for economic, reliability and performance reasons, but the average number is not very large due to the cost. The increase is not obvious for stub networks. Transit networks have been increasing their average multihoming degree from 2.5 to 3.5.

Peers

Peering links play an important role in the inter-domain forwarding process, even though they are outside the hierarchies. It is important to access the quantity of peering links and their positions in the Internet topology. In SAFA, the bridge forwarding table is used to handle special ”shortcut” entries. This bridge forwarding table will mainly store entries for peering links. Given the forwarding algorithm of SAFA, too many such entries will obstruct the forwarding speed.

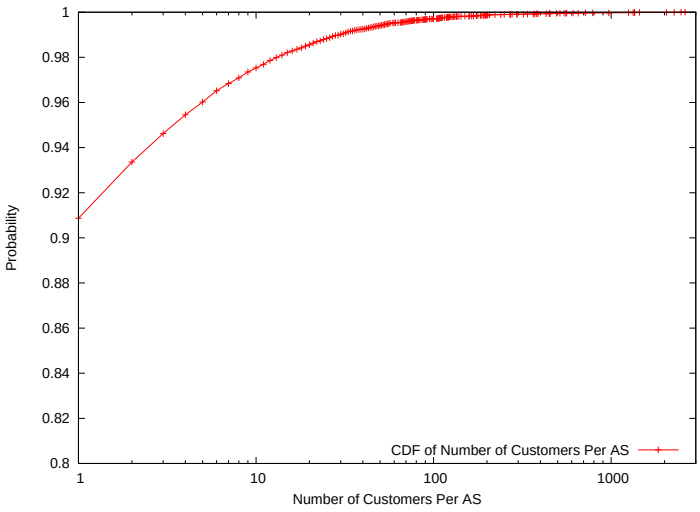

Figure 5.2: Distribution of customer number

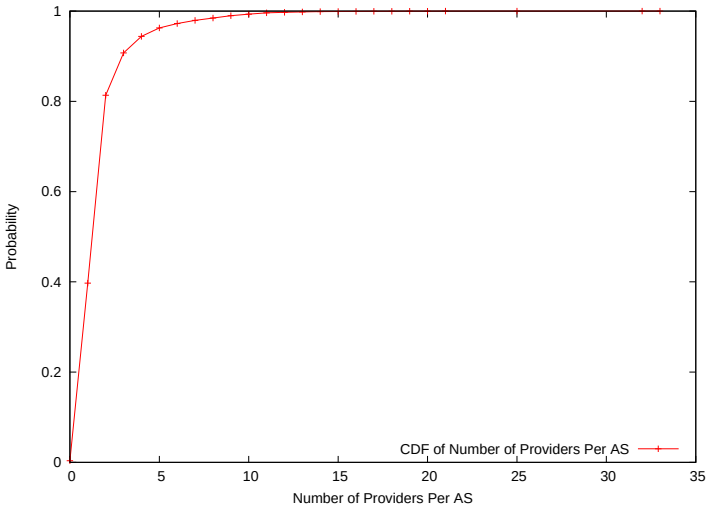

Figure 5.3: Distribution of provider number

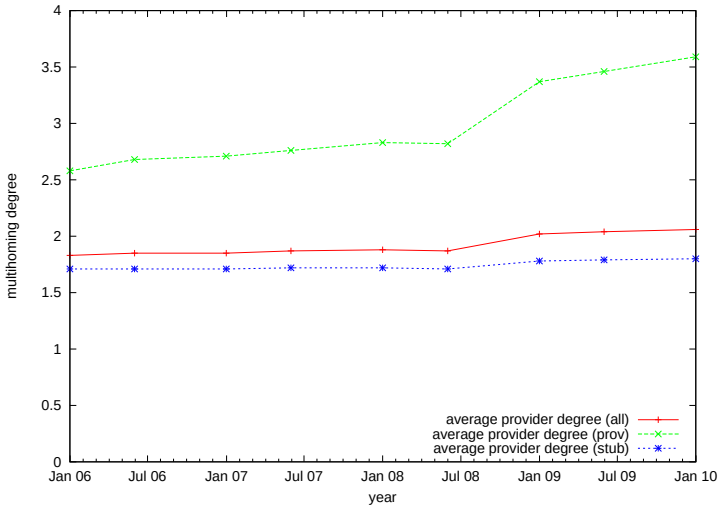

Figure 5.4: Evolution of multihoming degree

Figure 5.5 shows the distribution of the number of peers each domain has. More than 95% domains do not have any peer, and the 99% percentile is only 5. According to the conclusion of [42], there is a substantial number of invisible peer links interconnecting ASes at lower tier and around the edge of the Internet. Therefore further work is required here, but at least the core of the Internet would not suffer from large bridge forwarding tables.

5.2.3 Simulation

A simulation of the locator allocation process is carried out on the Internet topology. One goal of this simulation is to estimate the size of uphill forwarding tables and bridge forwarding tables in SAFA. The number of entries in the uphill forwarding table of an entity is also the number of locators it has. The bridge forwarding table of an entity may contain entries set up by protocols for special purposes, but its size mainly depends on the total number of locators of the network's peers. The locator allocation along the hierarchies is simulated. The number of locators each AS has is computed. The sum of the locator number of an AS's peers is considered as the approximation of the size of its bridge forwarding table.

Another goal of this simulation is to collect the statistics of the provider-customer hierarchies. First, we define the "layer" of a network as the minimum length of its locators. The layer of a core network is one. The layer is the shortest distance from a network to the core, that is, the shortest distance to networks outside its own hierarchies. The shortest AS-path length is also

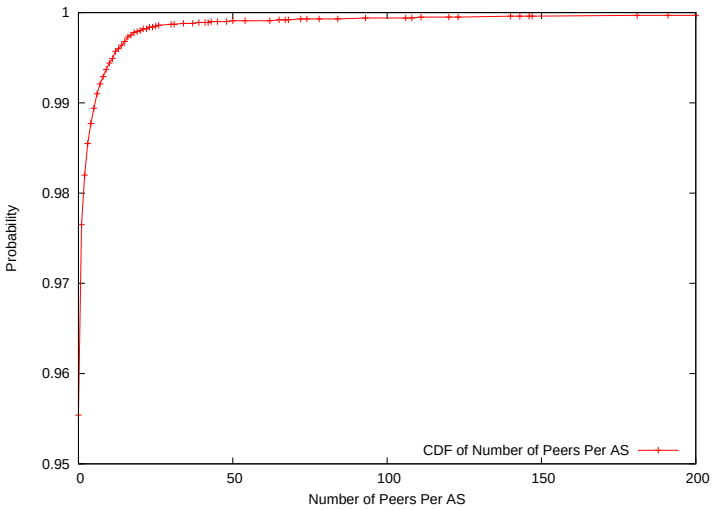

Figure 5.5: Distribution of peer number

a significant metric in the inter-domain routing. Therefore this concept is used to describe the position of a network in the Internet.

We combine the allocation simulation with real world AS-path data to get rid of the unreasonable relationship cycles and deep hierarchies. A concept "AS2link" is introduced to achieve this purpose. AS2link is defined as a set of two connected links $X \rightarrow Y \rightarrow Z$ existing in real AS paths. During the locator allocation procedure, if AS2link "$X \rightarrow Y \rightarrow Z$" does not exist, Y does not allocate its prefixes from X to Z. The observations from the simulation are as follows.

Figure 5.6 shows the number of locators allocated to each domain as a cumulative distribution. It can be seen that 90% of the domains have less than 20 locators, and the 99th percentile is only 50. However, the largest number is 207. A few hand-debugging cases suggest that the tail part of the distribution may be caused by inference errors. Note that the actual number of locators would be less than the simulation result, because the providers do not necessarily allocate all their prefixes to the customers simultaneously. Figure 5.5 shows the cumulative distribution of the number of peering entries in a domain's bridge forwarding table. More than 95% of the domains do not have peers and the 99th percentile is 43. For most of the domains, the size of the bridge forwarding table is quite small and will not have a significant effect on the performance of the forwarding algorithm. The maximum number of peering entries is 3496, which is not too large compared to the size of the current DFZ RIB.

Figure 5.8 shows the distribution of the domains' layers in both 2005 and 2010. The result shows that with an obvious growth of the number of ASes, the hierarchies do not expand very

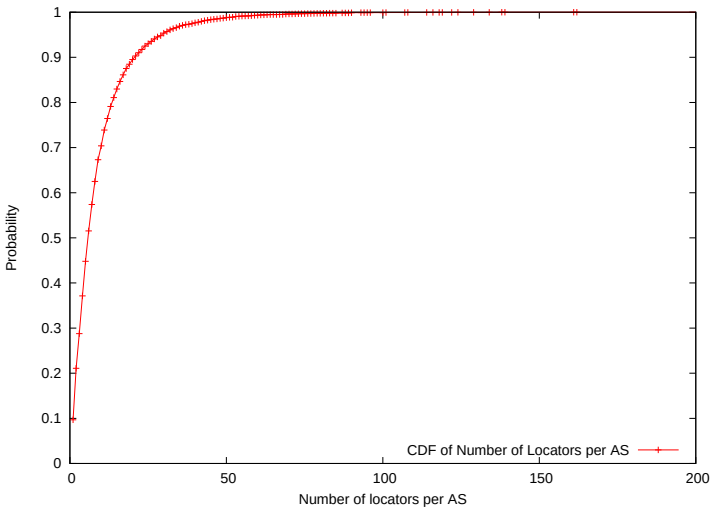

Figure 5.6: Distribution of locator number

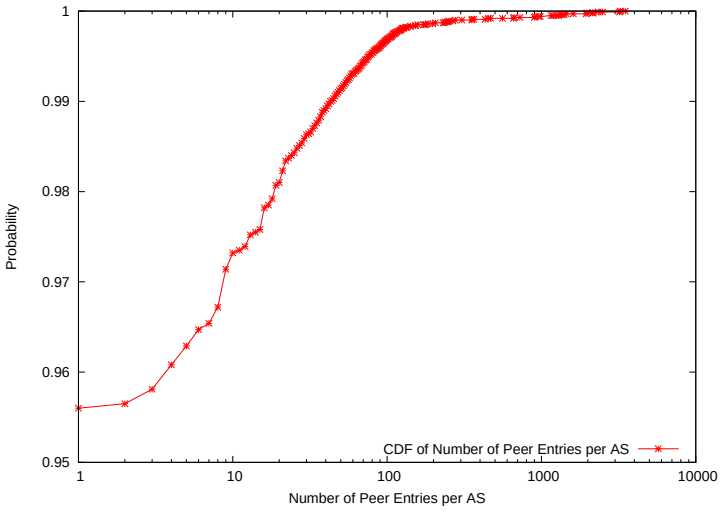

Figure 5.7: Distribution of peering entry number

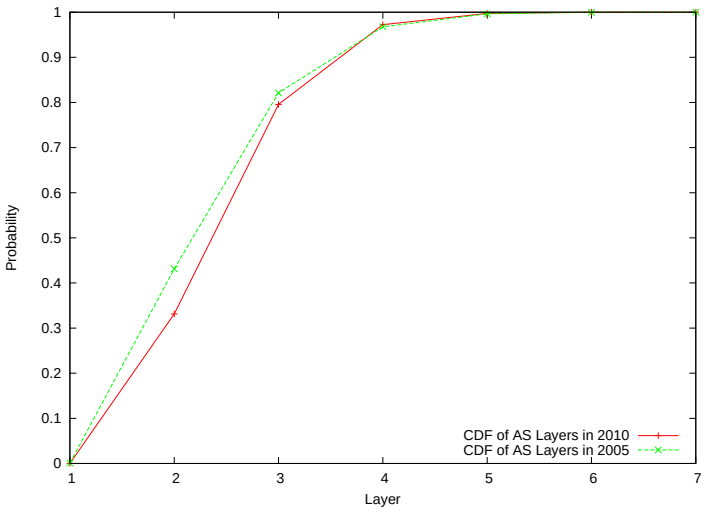

Figure 5.8: Evolution of the distribution of layers

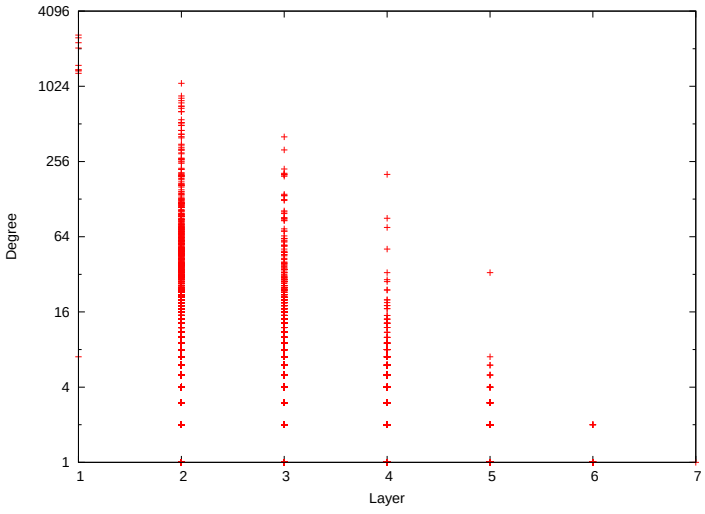

Figure 5.9: AS degree by layer

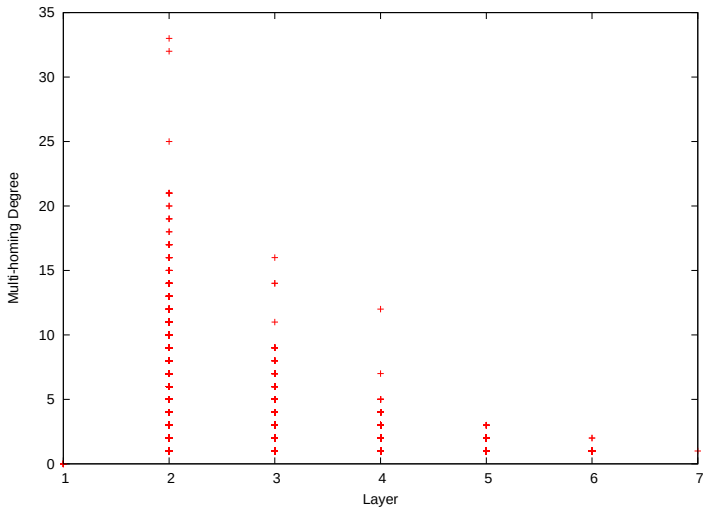

Figure 5.10: Multi-homing degree by layer

much. The Internet topology is becoming more dense. This characteristic means that even if the Internet continues growing, SAFA is still practical in the foreseeable future. Figure 5.9 shows the degree values of domains at different layers. The two attributes do not have a strict correlation but their trends are coherent. Figure 5.10 shows that many large transit networks at layer 2 have a high multihoming degree. This reflects the purpose of multihoming: increase the network reliability.

Figure 5.11 shows that the number of locators does not increase obviously with the increase of the node's layer. In other words, there is no exploration implicated by p^h (see Section 5.2.2). Figure 5.12 is the number of peering entries of domains at different layers. Some domains at high layers have more than 1000 peering entries, much more than most of the domains. However, such an order of magnitude will not cause a serious performance problem.

5.3 Flat Internet

Recent papers report that the peering links keep on increasing and the Internet topology is becoming flat [31, 16]. According to our analysis, this trend has not threaten our hierarchical addressing scheme yet. First, the hierarchical structures are still an important part in the new Internet logical topology observed in [31] (see Figure 5.13) The Internet topology is far from a fully meshed structure. The growth of peering links mainly results from the increase of peering

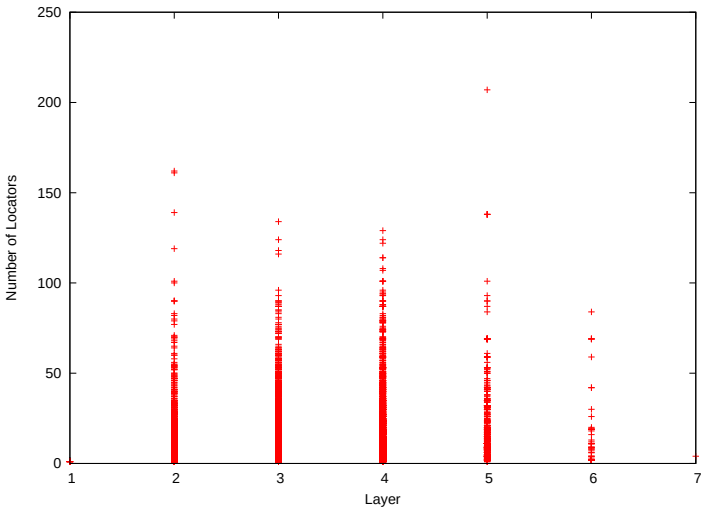

Figure 5.11: Locator number by layer

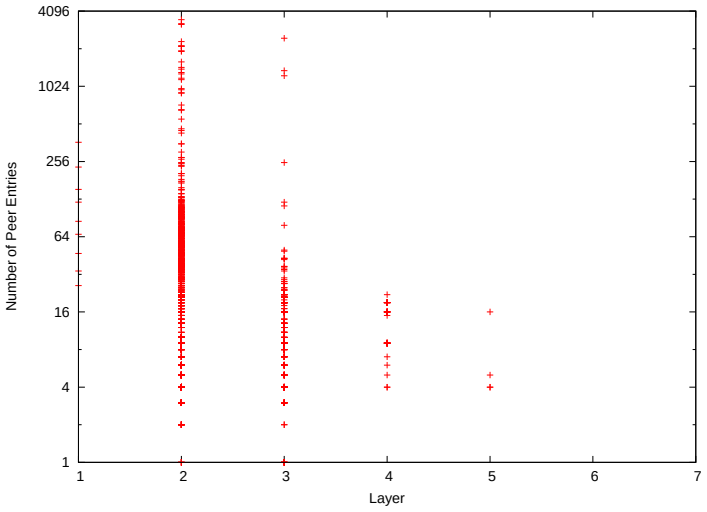

Figure 5.12: Number of peering entries by layer

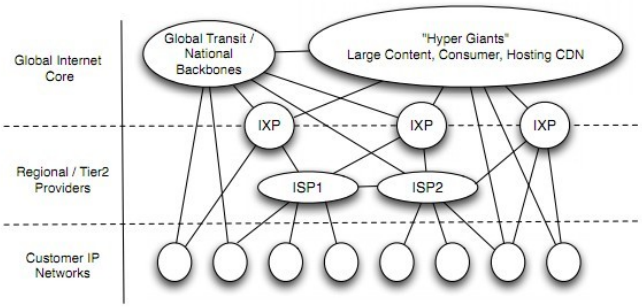

Figure 5.13: Emerging new Internet logical topology

links between content delivery networks (CDNs) and small transit providers. This increase does not happen between any kind of ASes. Actually large transit providers still hold strict peering policies. Second, according to these recent results, the flattening trend is more reflected in the change of traffic fraction. The traffic going through the links between CDNs and their peers keep on increasing. The addressing scheme is based on the topological structure, so the change of the traffic distribution does not affect the addressing scheme. Third, SAFA already takes peering links into account. With special bridge forwarding entries, traffic can traverse a chain of peering links, while NIRA can make use of only one peering link in the route. Given the fact that the number of the core networks is much smaller than the space provided by a 16-bit label, the CDNs can also acquire top level labels. This can make the forwarding between them and their peers more efficient.

Chapter 6

Prototype System

This chapter presents a proof-of-concept prototype of SAFA. The prototype system is part of the evaluation, with the objective to understand protocol design and software implementation better. The implementation involves the packet header, forwarding component and locator allocation component. The forwarding component is implemented based on Section 4.3. Section 6.1 introduces the packet header design and the interactive locator allocation protocol. Section 6.2 illustrates the implementation of forwarding infrastructure and the addressing allocation protocol.

Note that the prototype implementation is different from a ready-to-deploy system. The deployment of a clean-slate Internet architecture involves a lot of stakeholders, and should be a incremental process. It requires compatibility between the new architecture and the legacy IP system. The prototype does not address this problem.

6.1 Specification

6.1.1 Packet Header

A SAFA packet header carries source and destination locators, flags to support forwarding, and necessary fields for other protocols. Figure 6.1 shows the general format of a SAFA packet header. Because locators in SAFA have variable lengths, the length values should be stored in each packet header. Based on the Internet topology study, the length of a destination locator, i.e., the size of the label stack would not be too large, so both of the length values take 4 bits. The

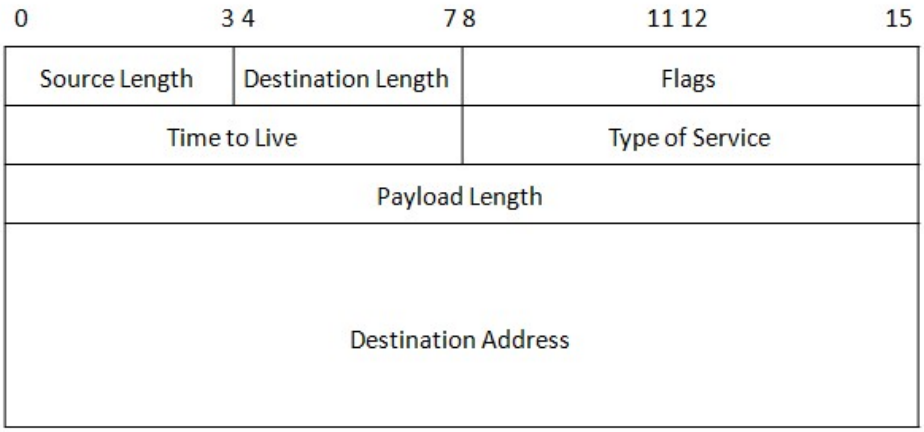

Figure 6.1: SAFA packet header

following 8 bits are for forwarding flags. Only the Source Routing flag which takes one bit is compulsory. The 7 bits left are for optional flags and future extensions.

Variable-length locators are believed to be less efficient compared to fixed-length locators because the comparison and lookup operations of a variable-length locator always require multiple memory accesses. Thus we adopt a trailer design to reduce the cost from variable-length locators. In particular, the header part carries only the first 4 labels of the destination locator while the left variable-length part is appended at the end of the packet, called a trailer. If the destination locator has less than 4 labels, the left bits are filled by 0. The source locator is also put in the trailer since source routing is an optional operation. Therefore the length of a SAFA packet header keeps fixed. With the trailer design, a variable-length locator is extended or shortened to a fixed-length word which can be fetched through one or two accesses. The number of memory accesses is reduced. With a proper kind of hardware like TCAM, the comparison and lookup operations are more efficient. The reason why the value 4 is chosen is because the topology study shows such a space is sufficient for most of the packets to store their destination locators. According to the topology study in Chapter 5, more than 90% AS paths in the real routing data have a length less than 6 (see Figure 5.1) and the layer of more than 95% ASes is not more than 4 (see Figure 5.8).

The other fields in the header: time to live, type of service, data length have the same semantics as in the current IP headers. The data length does not include the length of the trailer.

6.1.2 Automatic Locator Allocation Component

The merits of automatic and dynamic locator allocation have been introduced above. This section introduces the specification of the locator allocation component. The main functions of this component include requesting locator information from providers/peers, assigning locators to customers/peers, and maintaining relevant forwarding tables.

The automatic locator allocation is an interactive process. There are two types of relationships involved, i.e., customer-provider and peering, which are relevant to allocation of locators and setup of bridge forwarding entries respectively. The locator allocation protocol is an interactive protocol like DHCP, so for the convenience of discussion, in the left part of this section, the entity that requests a locator is named a client while the entity that responds is called a server, no matter what business relationship they have. The allocation is lease based. Typically the lease time depends on the device type of the client.

Messages

There are six types of messages used in the allocation component: REQUEST message, REPLY message, ACK message, CONFIRM message, RELEASE message and UPDATE message. All kinds of messages share a common header. The format of the messages is shown in Figure 6.2. The fields include the type of the message, the role or subtype of the message, the type of the application, and necessary authentication information. Some messages need to carry prefixes data of variable lengths. The messages have different roles because the protocol needs to handle two types of relationships, i.e., customer-provider and peering. The details of the fields can be found in Table 6.1.

The function of each type of messages is as follows.

REQUEST: A REQUEST message is used to request a new locator from the server or to renew a locator. For the former purpose, the client provides the relationship, and its device type, its hardware address and/or its ASN. For the latter purpose, it sends the locator it wants to renew instead of its hardware address.

REPLY: A REPLY message is used to send the lease to the client. The lease contains the label assigned to the client, selected prefixes of the server, lease time. If there is no available locator, the 'role' field of the message is set to DENY.

ACK: An ACK message is used for the client to accept or reject the lease offer.

CONFIRM: If the client accepts the lease offer, the server sends a CONFIRM message to notify the client so that the client can start timing.

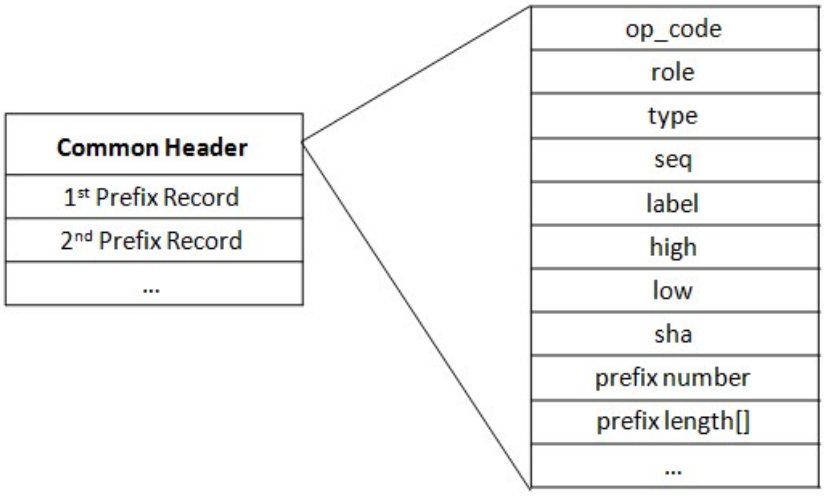

Figure 6.2: Fields of message packets

RELEASE: A client sends RELEASE messages to a server when it actively terminates the connection and gives up locators from that server.

UPDATE: When the locators of the server change, it sends UPDATE messages to its clients to update their records.

Note that for border routers of an AS, ASN is required to facilitate policy-based allocation, while for internal routers and hosts, this field is not necessary. The purpose of the round of ACK and CONFIRM is to start timing accurately after the entries in forwarding tables are available .

Interaction Protocol

The locator allocation of SAFA is an interactive procedure. This section is a summary of the protocol exchanges between clients and servers. This summary refers to the messages described in the previous section. Figure 6.3 shows the client-side finite state machine and Figure 6.4 shows the server-side finite state machine.

1. The client sends a REQUEST message to the sever, indicating the relationship, its device type and necessary identity information such as message sequence number, its hardware address and/or ASN. If the client does not receive a REPLY message after a pre-set time (T1 in Figure 6.3), it resends the REQUEST message with a new sequence number.

Table 6.1: Locator Allocation Messages

Field	REQUEST	REPLY	ACK	CONFIRM	UPDATE	RELEASE
op_code	OP_REQUEST	OP_REPLY	OP_ACK	OP_CONFIRM	OP_UPDATE	OP_RELEASE
role	CUSTOMER or PEER or C_RENEW or P_RENEW	role from REQUEST or DENY	role from REPLY	role from ACK	INSERT or REMOVE	CUSTOMER or PEER
type	GENERAL or MOBILE or SERVER	type from REQUEST	0	0	0	0
seq	sequence number from the counter	sequence number from REQUEST	sequence number from REPLY	sequence number from ACK	sequence number from the counter	0
label	0 or label to renew	label to client	label from REPLY	label from ACK	label to client	label to release
high	high 16 bits of ASN	0	0	0	0	0
low	low 16 bits of ASN	lease time	0 or private label to server	0	0	0
sha	hardware address of client	hardware address of server	0	0	0	0
prefix number	0	number of prefixes	0 or number of prefixes	0	number of prefixes	0
prefix length[]	0	values of prefix lengths	0 or values of prefix lengths	0	values of prefix lengths	0

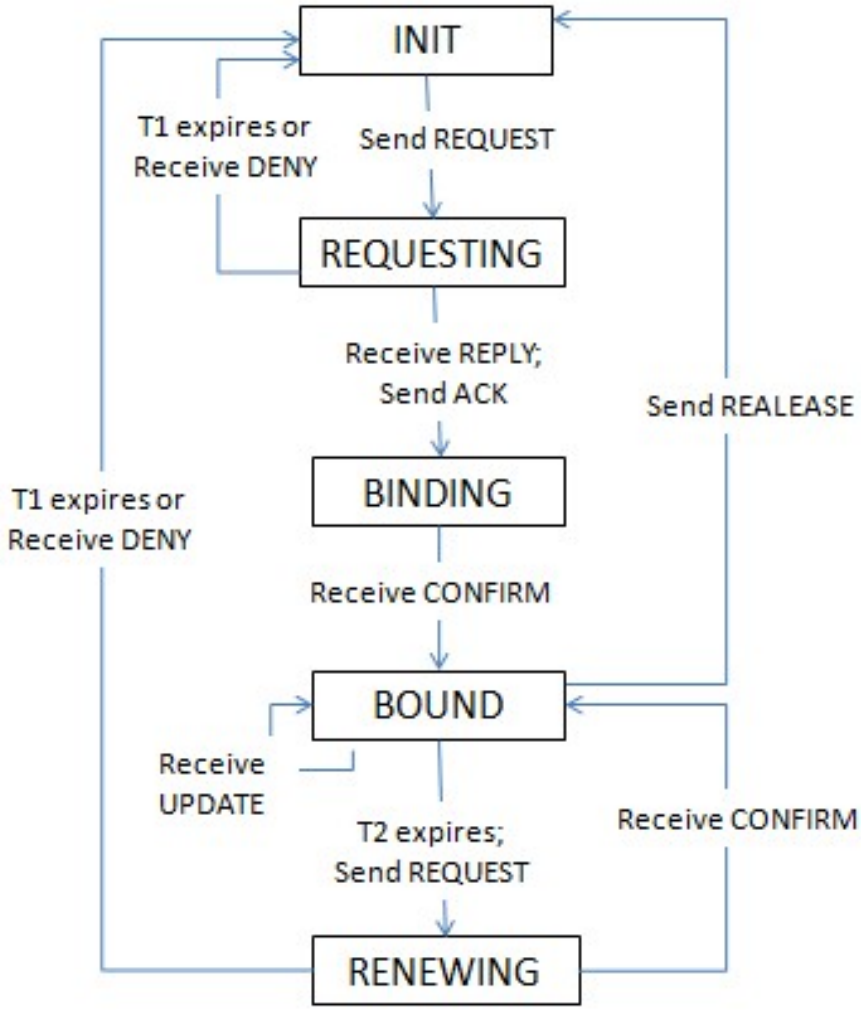

Figure 6.3: Client-side finite state machine

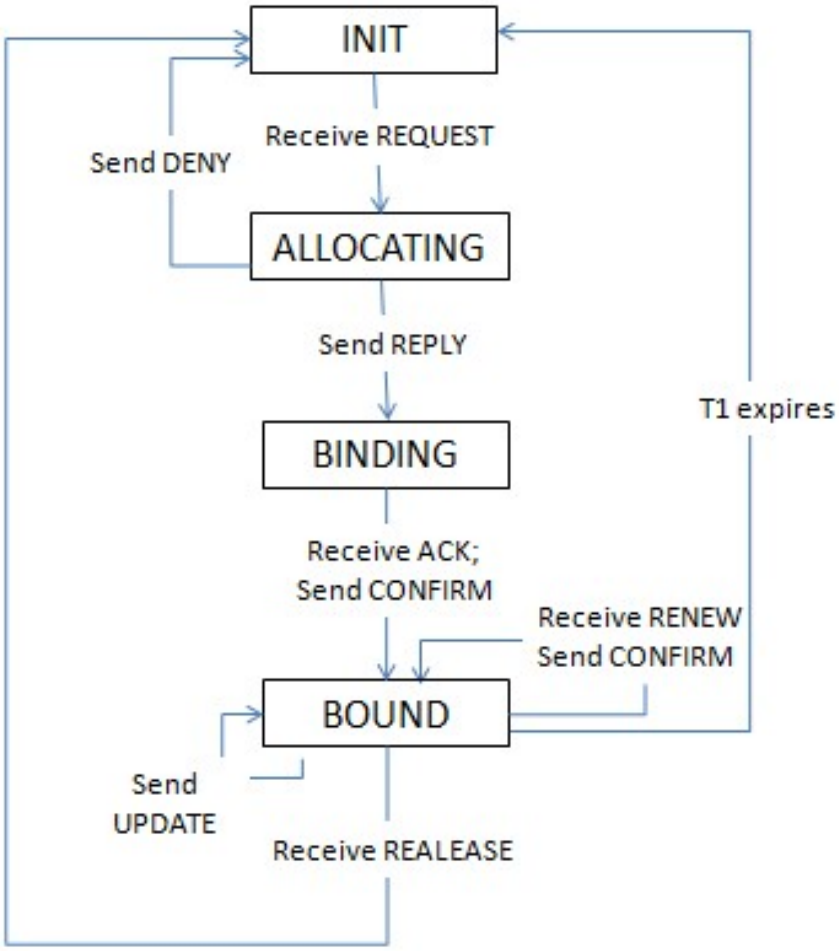

Figure 6.4: Server-side finite state machine

2. After receiving the REQUEST message, the server allocates a label to the client according to its policies, puts the lease into the lease table and puts the client's hardware into the ARP table but does not activate the records. Then the server sends a REPLY message containing the lease information and all its prefixes to the client. If there are no available locators, the sever will send a deny REPLY to the client.

3. If the client accepts the lease, it puts the locators assigned to it into the uphill forwarding table, puts the sever's hardware address into the ARP table and writes the lease into a temporary table. The client does not activate the records either. The client sends an ACK to the server to accept the lease. If the relationship between the client and the server is peering, the client needs to allocate labels and provide its own prefixes to the server in the ACK messages.

4. Once the server receives the ACK message, it will activate the records in tables, start the lease-time timer (T1 in Figure 6.4)and send a CONFIRM message to the client. If the relationship is peering, the server needs to store the locators from the client into the bridge forwarding table.

5. When the client receives the CONFIRM message, it also starts the lease-time timer (T2 in Figure 6.3) and activates the records in tables.

6. When the lease-time timer expires, if the client wants to renew the locator, it sends out another REQUEST message with the locator to renew. The server will send a CONFIRM message to respond to the client.

7. If the client wants to release the locator before the lease time is out, it sends a RELEASE message to the server.

8. When the prefixes of the server change, it will send an UPDATE message to the client to update its records.

When the client actively releases the locators allocated to it, the server assumes the client has finished hold_over operations. The private locator allocation in the peering part is optional. The major objective of this procedure is to construct the bridge forwarding table.

6.2 Implementation

The prototype of SAFA is implemented in a modular software router called Click [2]. A Click router acts in a flexible and configurable way. Its configuration is based on interconnected modules called elements which control every aspect of the operation of the router: communicating with devices, packet modification, queueing, dropping policies, packet scheduling, etc. The router configuration results from gluing elements together in a plain text configuration file using a simple script language. Click can work either at user level or as a kernel module, overriding

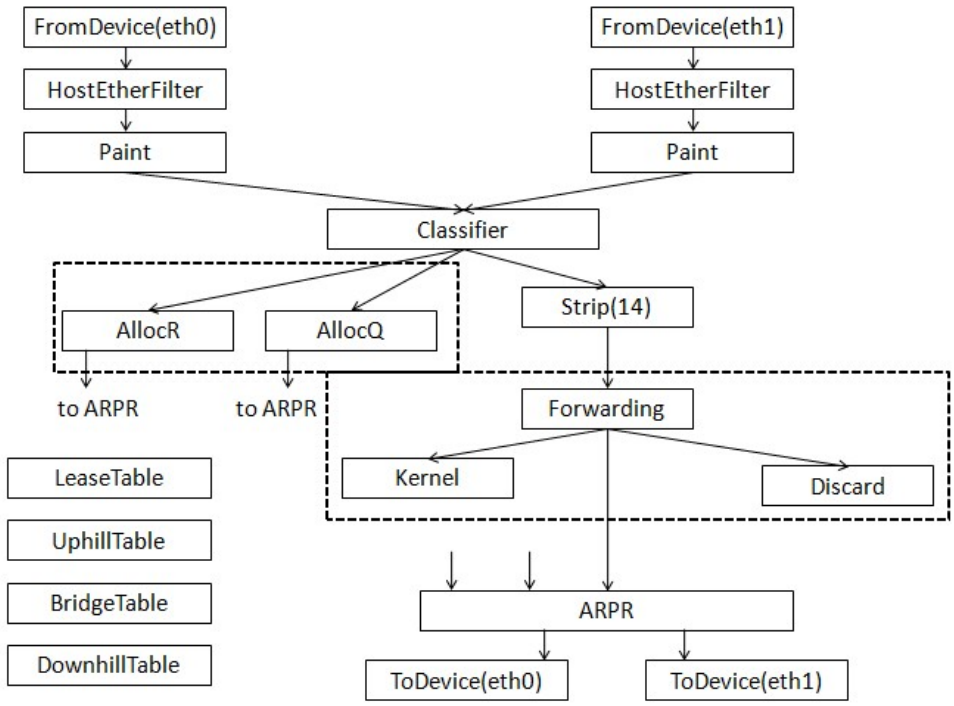

Figure 6.5: The configuration graph

the original protocol stack of FreeBSD or Linux. Click can also cooperate with the network simulator NS2. Elements in a simulation environment can be used directly in other modes without any change.

As modularity is the main advantage of a Click router, it is possible to write new modules in C++ with the desired behavior. Figure 6.5 shows a typical router that runs SAFA. However, the packet header and address format files of IP (both IPv4 and IPv6) do not belong to the configurable part. The corresponding format files of SAFA have to be put into the common included fold, too. Therefore they are not shown in element configure files.

Figure 6.5 is the actual configuration graph of Click. Each label represents an element in Click. The router contains four parts: locator allocation, forwarding, forwarding tables and interfaces to higher and lower layer protocols. The FromDevice and ToDevice elements are interfaces between hardware drivers and the software router. ARPR is the element maintaining the MAC addresses of entities connected to each port of the Click router. LeaseTable is used for locator lease management. Every lease contains port and timing data. The three other separate tables are forwarding tables. The two circled parts are the locator allocation and forwarding

components. Classifier is a filter element that classifies the coming packets according to the ether_type field in the ethernet header into two categories: locator allocation packets (including request and reply packets) and normal SAFA packets.

When a packet arrives at a system, it is passed to the Classifier element with its input port marked by the Paint element. The port number information is used for locator allocation. If it is a packet for locator allocation negotiation, it is sent to the corresponding element. AllocR handles request and ACK messages, while AllocQ sends out requests and handles reply and confirm messages. If it is a normal SAFA packet, it is sent to the forwarding component. The forwarding component includes three elements. Kernel is the interface to higher layer protocols, used by end points. Discard handles the packets broken or delivered incorrectly. Forwarding implements the forwarding algorithm introduced in Section 4.3.3.

Note that this is a general configuration and there are variants for different routers. A distinction can be drawn between routers in an AS that are connected only to other routers within the AS, versus those that connect to other ASes. Routers in the former group are usually called internal routers, while the latter group are called border routers in BGP. Typically, the configuration in Figure 6.5 can be used in border routers while internal routers can employ intra-domain addressing policies and routing protocols which means a different set of elements. In addition, the configuration of routers in the core networks is a little different from others. This kind of routers do not request for locators. There are only two forwarding tables. One is the downhill forwarding table, while the other one is used for forwarding between the core networks. Both of them use single-label matching. Therefore, routers in the core networks can have their specific elements.

The forwarding tables can be implemented using any data structure and look up algorithm that satisfy the specification in Section 4.3.2. In the prototype system, the bridge forwarding table is implemented using a normal radix look up algorithm. All the locators from neighbor entities are stored in a radix tree. Each radix contains key and pointers to children. The radix nodes storing output port number are set real. In addition, a table with port numbers as indexes is implemented to remove radixes efficiently when leases expire. Each slot of the table contains a linked list of pointers to corresponding radixes. The two data structures are shown in Figure 6.6. The filed circles represent real radixes. The data structures of the uphill forwarding table are similar to the ones of the bridge forwarding table.

We run a series of tests in NS2 to verify our implementation and improve design details. It is easy to set up and change the topology and collect time statistics in the simulation environment. The Click elements can be transferred to a real platform directly. Therefore, the tests are carried out in the NS-Click environment. The topology in Figure 4.1 is set up for our tests. This topology has typical structures in the Internet including hierarchies, a peering link and a multihoming

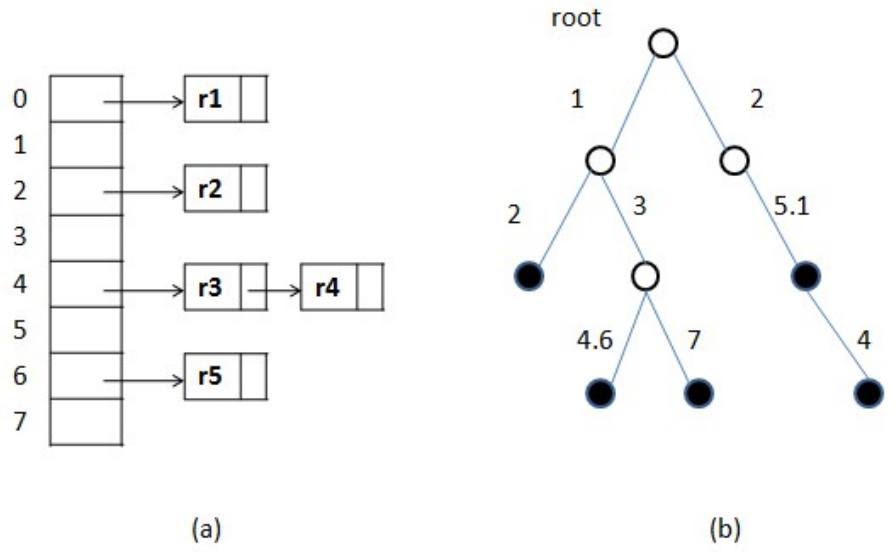

Figure 6.6: Data structures for the bridge forwarding table: (a) port table (b) radix tree

node. Delay of all the links in the tests is set to 1 ms. First, the packet forwarding along different routes is tested. The matching in the bridge forwarding table does not put obvious additional time cost. After that the negotiation time of the locator allocation is measured repeatedly. The standard deviation of the average time between different nodes is quite low. The mean value of all the negotiation time is 4.22 ms. The propagation of locator updates is also tested. There are two kinds of updates: a new prefix is announced as available; a previously available prefix is withdrawn. Topology changes are injected at a pre-set time point. We define the *convergence latency* as the time between the injection of the change and the receiving of the last confirm message. The convergence latency of updates originating from R and X (see Figure 4.1) is measured. The updates from X cause changes in not only uphill forwarding tables of its customers but also the bridge forwarding table of Y. The average convergence latency at both X and R is 6.07ms. Note that the convergence latency is not the counterpart of the convergence time of BGP. In SAFA, the topology changes will also lead to the updates in the rendezvous service. This time needs to be taken into account, too.

The design of the addressing and forwarding schemes leaves many details. This implementation helps to learn more lessons about the design details. First, variable-length locators are always believed to be less efficient than fixed-length ones because of more memory accesses. This is even one of the reasons that Pip is not adopted. In our packet header design, the des-

tination locator is extended (or shortened) to a fixed-length field to improve the efficiency. In our software lookup algorithm the improvement is not obvious, but with a proper hardware like TCAM, most of lookup operations can have the same efficiency as the ones of fixed-length locators. Second, in the interactive allocation protocol, there are a number of parameters that can be tuned according to users' requirements. The values of the lease time can be chosen according to the type of the client devices. For stable costumers and peers, a long lease time can reduce unnecessary renew cost. For temporary clients like mobile devices, a short lease time can reduce the waste of locators. Moreover, the allocation can also be implemented to be policy-based. Servers do not have to export all their locators to clients. Instead, they export the information according to a policy parameter, which is determined by business relationships. Third, if multiple links lead from one network to another, there are two options for each side of the multi-link interconnect: either two distinct labels can be assigned or a local multi-path routing strategy needs to be used, similar to existing solutions. In the former case, extra connectivity is potentially exposed to the rest of the Internet, but handled normally by the addressing scheme. In the second case, the extra connectivity is handled locally and transparent for the rest of the Internet. Finally, like DHCP, the locator allocation protocol can carry MAC address information, but a protocol like ARP is still needed for LAN communication.

Chapter 7

Conclusion and Future Work

This chapter summarizes the contributions of this work and discusses directions for future research.

This thesis presents the design of SAFA, a new Internet addressing and forwarding architecture. The architecture employs a hierarchical addressing scheme that delegates as much control as possible to local agreement and requires only minimal global administrative coordination. In combination with a rendezvous service, the addressing scheme gets rid of the scalability problem of the current Internet routing system. The forwarding process based on the addressing scheme is straightforward and efficient. With the assistance of the rendezvous service, SAFA can also support communication patterns and engineering techniques such as mobility, multihoming and traffic engineering. SAFA gives more control over route selections to end systems. The architecture does not require network operators to publish more internal topology information about their networks than current IP addressing and BGP routing.

SAFA considers essential requirements of the Internet, and takes the experience of previous similar work into account. SAFA is a pure ID/locator split architecture, not a core/edge split solution like LISP. Also different from traditional designs like HIP, SAFA does not contain a uniform ID namespace. The requirements of identifier systems arise from higher layer protocols/applications. There should be various ID systems in the Internet. With a variable-length format, the addressing scheme of SAFA is more flexible than the one of NIRA's. Without the bridge segment in the route representation, the header overhead of SAFA is lower. The forwarding procedure of SAFA keeps simple. There is no complicated label types and instructions like Pip. In addition, the rendezvous service plays a more significant role in SAFA than other proposals. It is used to support features like route preference and mobility management.

This thesis also extracts five essential components of an Internet architecture and presents an

Internet topology study and a proof-of-concept prototype to evaluate the design of this architecture.

The work presented in this thesis focuses on addressing and forwarding. Unavoidably, some parts of the architecture are preliminary, and require future study. This thesis discussed what functionalities routing protocols and rendezvous service should provide, but does not propose detailed algorithms. How to implement the routing subsystem in an efficient way is not answered. The details of rendezvous services like server placement, cache mechanisms and compatibility with DNS are all interesting topics. Another interesting area is the deployment of SAFA. Incremental deployment and address translation are necessary because of the large size of legacy systems. Questions like how the routing protocols and rendezvous services cooperate with each other are also worth studying. Beyond the functions of traditional network layer, this architecture can be extended to the transport layer. The end system would become a virtual network and port numbers can be expressed as an additional layer in the addressing hierarchy. Different from the layering abstraction, transport layer services can just piggyback on the underlying scheme. Above the network layer, one could feasibly operate with service modules that provide specific functionality, such as reliability, flow control, congestion avoidance, same order delivery and so on. Applications could pick and configure these service modules according to their needs, instead of being required to choose between preconfigured transport layer protocols. The extension of SAFA is also one of the future study topics.

References

[1] BGP Routing Table Analysis Reports. http://bgp.potaroo.net. 1, 3

[2] Click Modular Router. http://read.cs.ucla.edu/click/. 58

[3] Internet Topology Collection. http://irl.cs.ucla.edu/topology/. 39

[4] RIPE Routing Information Service. http://www.ripe.net/data-tools/stats/ris/routing-information-service. 38

[5] The Cooperative Association for Internet Data Analysis. http://www.caida.org. 39

[6] The DIMES Project. http://www.netdimes.org/new/. 38

[7] Tier 1 Networks on Wikipedia. http://en.wikipedia.org/wiki/Tier_1_carrier. 40, 41

[8] University of Oregon Route Views Project. http://www.routeviews.org/. 38

[9] J. Abley, K. Lindqvist, E. Davies, B. Black, and V. Gill. IPv4 Multihoming Practices and Limitations. RFC 4116 (Informational), July 2005. 4

[10] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. Overview and Principles of Internet Traffic Engineering. RFC 3272 (Informational), May 2002. Updated by RFC 5462. 36

[11] Giuseppe Di Battista, Thomas Erlebach, Alexander Hall, Maurizio Patrignani, Maurizio Pizzonia, and Thomas Schank. Computing the types of the relationships between autonomous systems. *IEEE/ACM Trans. Netw.*, 15(2):267–280, 2007. 22

[12] S. Carmi, S. Havlin, S. Kirkpatrick, Y. Shavitt, and E. Shir. A model of internet topology using k-shell decomposition. *Proceedings of the National Academy of Sciences*, 104(27):11150, 2007. 21

[13] Q. Chen, H. Chang, R. Govindan, and S. Jamin. The origin of power laws in internet topologies revisited. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 608–617. IEEE, 2002. 21

[14] D.D. Clark, J. Wroclawski, K.R. Sollins, and R. Braden. Tussle in cyberspace: defining tomorrow's internet. *ACM SIGCOMM Computer Communication Review*, 32(4):347–356, 2002. 12

[15] A. Dhamdhere and C. Dovrolis. Ten years in the evolution of the internet ecosystem. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 183–196. ACM, 2008. 23, 39

[16] A. Dhamdhere and C. Dovrolis. The internet is flat: Modeling the transition from a transit hierarchy to a peering mesh. In *Proceedings of the 6th International Conference*, page 21. ACM, 2010. 48

[17] G. Di Battista, M. Patrignani, and M. Pizzonia. Computing the types of the relationships between autonomous systems. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 156–165. IEEE, 2003. 22

[18] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, et al. As relationships: Inference and validation. *ACM SIGCOMM Computer Communication Review*, 37(1):29–40, 2007. 22, 40

[19] R. Droms. Dynamic Host Configuration Protocol. RFC 2131 (Draft Standard), March 1997. Updated by RFCs 3396, 4361, 5494. 4

[20] A. Dul. Global ip network mobility using border gateway protocol (bgp). *Boeing White paper*, 2006. 10

[21] A. Elmokashfi, A. Kvalbein, and C. Dovrolis. On the scalability of BGP: the roles of topology growth and update rate-limiting. In *Proceedings of the 2008 ACM CoNEXT Conference*, page 8. ACM, 2008. 1, 3

[22] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM Computer Communication Review*, volume 29, pages 251–262. ACM, 1999. 8, 21

[23] D. Farinacci. LISP Alternative Topology (LISP+ALT). Internet-Draft draft-fuller-lisp-alt-06, Internet Engineering Task Force, March 2011. Work in progress. 15

[24] D. Farinacci, V. Fuller, D. Oran, D. Meyer, and S. Brim. Locator/ID Separation Protocol (LISP). Internet-Draft draft-farinacci-lisp-12, Internet Engineering Task Force, April 2011. Work in progress. 15, 18, 19

[25] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina. Generic Routing Encapsulation (GRE). RFC 2784 (Proposed Standard), March 2000. Updated by RFC 2890. 16

[26] P. Francis. Pip Near-term Architecture. RFC 1621 (Informational), May 1994. 17, 18, 19

[27] V. Fuller and T. Li. Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan. RFC 4632 (Best Current Practice), August 2006. 8

[28] Lixin Gao. On inferring autonomous system relationships in the internet. *IEEE/ACM Trans. Netw.*, 9(6):733–745, 2001. 20, 21, 22, 40

[29] IAB and IESG. IAB/IESG Recommendations on IPv6 Address Allocations to Sites. RFC 3177 (Informational), September 2001. 27

[30] L. Karthik, R. Anand, and V. Srinivasan. Algorithms for advanced packet classification with ternary cams. In *Proceedings of ACM SIGCOMM*, pages 193–204, 2005. 30

[31] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. Internet inter-domain traffic. *ACM SIGCOMM Computer Communication Review*, 40(4):75–86, 2010. 48

[32] J. Laganier and L. Eggert. Host Identity Protocol (HIP) Rendezvous Extension. RFC 5204 (Experimental), April 2008. 15

[33] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2–es, 2007. 23

[34] T. Li. Design Goals for Scalable Internet Routing. RFC 6227 (Informational), May 2011. 10, 13, 35

[35] D. Magoni and J.J. Pansiot. Analysis of the autonomous system network topology. *ACM SIGCOMM Computer Communication Review*, 31(3):26–37, 2001. 23

[36] D. Meyer and D. Lewis. Architectural Implications of Locator/ID Separation. Internet-Draft draft-meyer-loc-id-implications-01, Internet Engineering Task Force, 2009. Work in progress. 16

[37] D. Meyer, L. Zhang, and K. Fall. Report from the IAB Workshop on Routing and Addressing. RFC 4984 (Informational), September 2007. 3, 4

[38] R. Moskowitz and P. Nikander. Host Identity Protocol (HIP) Architecture. RFC 4423 (Informational), May 2006. 14, 18, 19

[39] P. Nikander and J. Laganier. Host Identity Protocol (HIP) Domain Name System (DNS) Extensions. RFC 5205 (Experimental), April 2008. 15

[40] E. Nordmark and M. Bagnulo. Shim6: Level 3 Multihoming Shim Protocol for IPv6. RFC 5533 (Proposed Standard), June 2009. 18

[41] Mike O'Dell. GSE - an alternate addressing architecture for IPv6, 1997. 18

[42] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang. The (in) completeness of the observed internet as-level structure. *IEEE/ACM Transactions on Networking (ToN)*, 18(1):109–122, 2010. 22, 44

[43] D. Ooms, B. Sales, W. Livens, A. Acharya, F. Griffoul, and F. Ansari. Overview of IP Multicast in a Multi-Protocol Label Switching (MPLS) Environment. RFC 3353 (Informational), August 2002. 37

[44] J. Postel. Internet Protocol. RFC 791 (Standard), September 1981. Updated by RFC 1349. 5

[45] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard), January 2006. 1

[46] J.F. Shoch. Inter-network naming, addressing, and routing. In *COMPCON, IEEE Computer Society, Fall*, 1978. 5

[47] G. Siganos, M. Faloutsos, and C. Faloutsos. The evolution of the internet: Topology and Routing. *University of California, Riverside technical report*, 2008. 23

[48] L. Subramanian, M. Caesar, C.T. Ee, M. Handley, M. Mao, S. Shenker, and I. Stoica. HLP: a next generation inter-domain routing protocol. *ACM SIGCOMM Computer Communication Review*, 35(4):13–24, 2005. 18, 23, 39

[49] Lakshminarayanan Subramanian, Sharad Agarwal, Jennifer Rexford, and Randy H. Katz. Characterizing the internet hierarchy from multiple vantage points. In *INFOCOM*, 2002. 20, 22, 23, 39

[50] S.L. Tauro, C. Palmer, G. Siganos, and M. Faloutsos. A simple conceptual model for the internet topology. In *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, volume 3, pages 1667–1671. IEEE, 2001. 21

[51] Geng Wu and Caroline Chan. WiMAX, 3G and LTE: A Capacity Analysis, 2010. 42

[52] Xiaowei Yang, David Clark, and Arthur W. Berger. Nira: a new inter-domain routing architecture. *IEEE/ACM Trans. Netw.*, 15(4):775–788, 2007. 12, 16, 23, 39