INTERNET-SCALE REACTIVE ROUTING AND MOBILITY

A Thesis

Presented to

the Faculty of California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Daniel Nelson

June 2009

COMMITTEE MEMBERSHIP

TITLE: Internet-Scale Reactive Routing and Mobility

AUTHOR: Daniel Nelson

DATE SUBMITTED: June 2009

COMMITTEE CHAIR:	Dr. John Bellardo
COMMITTEE MEMBER:	Dr. Hugh Smith
COMMITTEE MEMBER:	Dr. Clark Turner

Abstract

Internet-Scale Reactive Routing and Mobility

by

Daniel Nelson

Since its commercialization, the Internet has grown exponentially. A large variety of devices can communicate creating advanced services for a diverse ecosystem of applications. However, as the number of Internet hosts has grown, the size of routing tables required to correctly route data between them has also increased exponentially. This growth rate necessitates increasingly frequent upgrades to routing device hardware, providing them with additional memory for fast-access storage of route information. These upgrades are both physically and fiscally untenable, and a new Internet routing solution is necessary for future growth.

This research focuses around an incrementally deployable, reactive routing system that is scalable to projected Internet growth. It requires no hardware or software updates to Internet routers, and offloads processing to end hosts and the network's edge. Within this framework, routers can make accurate decisions about optimal data paths; incurring no increase in path length over the current routing system.

A new architecture for IP Mobility is considered as a case study within this routing system, and compared with existing standards and implementations. The new architecture eliminates the triangle routing problem, while providing legacy hosts with connectivity to mobile devices. This mobility solution can integrate with a variety of hierarchical reactive routing systems with little overhead.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

According to First Research, Internet Routing hardware is a sixty billion dollar industry [8]. This is significant growth from an industry that started with 2 hosts on the ARPANET in 1969 [14]. Along with its expansion, the architecture of the Internet has changed significantly over time. In its infancy, nodes were connected in a tree-like structure. A major shift in architecture came with the commercialization of the Internet, when it transitioned to a more mesh-like structure [5]. While this new mesh structure is better suited to the complex relationships of independent organizations that make up the Internet, it also presents more difficulties in routing data correctly and efficiently.

Prior to 1995, Internet routes were aggregated in ”classes” of networks. These classes simplified routing by dividing the address space into three types of subnets, designated class A, B, and C, with A being the largest and C being the smallest [9]. By dividing the address space permanently, the availability of each type of subnet was fixed. Unfortunately, as the Internet expanded, the demand for class C subnets outstripped supply, and it became clear that a more flexible solution was necessary. That solution was Classless Inter Domain Routing (CIDR), which allows for routing of arbitrary size subnets.

CIDR was introduced in 1995, and performed well between 1995 to 1998. Network administrators used CIDR to advertise very specific routes to small subnets, allowing for richer routing policies. These routing policies began to take up more and more routing table space. After 1998, the size of the routing table began to grow exponentially, and continues to do so today [16].

Internet core routers must match the growth rate of the routing table with increasing amounts of system memory. For these routers, most of the routing table is stored in specialized high speed memory which allows for quick route lookups based on packet information. The specialized router memory costs significantly more than commodity PC memory because of its unique design and speed, which makes router memory upgrades a significant monetary investment. There is also concern within the Internet routing community about whether hardware will be able to match the growth of the routing table into the future. For these reasons, many experts are calling for a new routing solution which provides the scalability to address growth into the future.

This thesis presents a new method for Internet-scale routing that solves the current scalability problems, migrates processing to the network edge, and provides useful abstractions for future applications. It utilizes existing Internet technologies that are currently deployed globally to provide routing services, and requires no hardware or software upgrade to Internet core routers. Small network topological additions are necessary, which can be made using generic, inexpensive hardware.

Chapter 2 will discuss historical work in Internet Routing as well as current proposals for moderating routing table growth. Also, a brief overview of the current IP mobility standard will be given, in preparation for a more complete discussion of mobility in the context of a hierarchical routing system. Chapter 3 will discuss the details of the system's implementation, and Chapter 4 will look at the specific optimizations and changes that can be applied to IP Mobility. Finally, an analysis of the system's operation is included with a detailed discussion of throughput and latency. The current routing system is compared with both the hierarchical routing system proposed here and an implementation of mobility within the new routing system.

Chapter 2

Previous Work

The Internet is a large network composed of private networks owned by various public and private institutions. It is thus very easy to conceptually simplify the Internet to a two-layer network architecture, where the upper layer is inter-organization space, and the lower consists of the institution-run component networks [31].

Within the Internet infrastructure, there are a number of protocols that provide routers with information on logical network topology. This information allows the routers to forward data from source host to destination host, and enables global connectivity for most of the Internet. There are a unique set of problems that a network as diverse as the Internet must overcome to provide service to all potential applications. Features such as IP Mobility, VPN Tunneling, and load balancing all rely on the current architecture to provide services that build on basic connectivity. This chapter will give an overview of the current Internet ecosystem and present alternative proposals for solving the routing table size problem.

2.1 Internet Architecture

Because of the distinctly different natures of the layers of Internet architecture, different routing protocols are used at each layer [31]. The following sections provide a high-level overview of operations at both layers and share insight into design decisions that have resulted in the Internet's current state.

2.1.1 Network Level

The lower layer of the general Internet architecture is composed of the individual organizations that own part of the overall backbone. These organizations are commonly referred to as Autonomous Systems (ASes), reflecting their ability to make decisions independent of the rest of the ASes on the Internet. Each AS is assigned its own AS Number (ASN), which is used to identify it on a global scale. ASes and ASNs are virtually invisible to the average end user; ASNs are not currently included in any packet that reaches end users, and private networks can be established without any knowledge about or presence of an AS.

Because of the nature of ASes, AS internal routing can be accomplished by any means that the network administrators deem necessary. Thus, there are a multitude of methods used for internal routing, such as RIP and OSPF, but the area currently under the most scrutiny is Inter Domain routing, or routing on the global level between ASes.

2.1.2 Global Level

Routing on a global level is policy based, rather than a strict mathematical problem [15]. ASes use the Border Gateway Protocol (BGP) to distribute routing information to each other, as specified in RFC 4271 [21]. BGP is different from many other routing protocols in that it distributes complete path information for each advertised subnet. When sending a packet along a BGP-advertised route, the sender knows which ASes the packet will go through, which allows for fine-grained control of the intermediate networks between end-hosts on the Internet. For example, a policy instituted at the Pentagon might dictate that no Internet traffic should go through North Korea. BGP allows the Pentagon to enforce this policy on outgoing traffic by not selecting routes with ASNs that include North Korean networks.

2.1.3 Business Relationships

Routing policies on the global level are also affected by the business relationships between ASes [40]. There are two main forms of these business relationships: Peering and Transit. Peering is simply a mutual agreement between two ASes, usually of similar magnitude, where each AS will carry traffic sourced from the other without monetary cost. As mentioned in Weiss and Shin, the primary motivation for these types of agreements is that each organization sees the benefits of peering as outweighing the costs of peering [40].

A transit relationship usually takes place between a larger organization and a smaller organization, where the smaller purchases transit from the larger. In these relationships, the AS purchasing transit will be charged for data it sends through the mutual connection. Because of this additional cost, ASes with both types of agreements may prefer routes using peering links over transit providers' routes, even if they result in more inefficient paths.

Any two ASes that are interconnected have an agreement or contract defining the parameters of their connection, and, as described previously, traffic is routed based on these contracts. It is therefore important that any inter-domain routing protocol be able to maintain and enforce business policies while providing connectivity. BGP is well suited for this task, and has become the industry standard inter-domain routing protocol [21].

2.2 Routing Protocols

Routing protocols simplify the distribution of path data between routers. Various routing protocols are designed for various purposes; BGP serves for distributing inter-domain routes, while other, smaller-scale protocols distribute routes within ASes. The following sections provide an overview of some common Internet routing protocols.

2.2.1 RIP

The origins of Routing Information Protocol, or RIP, date back to 1957 [33]. As such, it is a well established routing protocol. It is defined by the Internet Engineering Task Force in RFCs 2453 and 4822 [19, 22]. RIP is a Distance-Vector routing protocol, meaning that a set of IP addresses may be advertised through a RIP session with an associated "distance" metric specifying how far away the host is from the advertising router. No explicit knowledge regarding network structure other than the next hop is transferred through the protocol. This makes RIP unsuitable for large-scale inter-domain routing, because organizations would be unable to enforce routing policies without knowing the specific networks its data travels through. RIP is proactive in broadcasting routes to its neighbors; it ensures that all routers have destination routes to all RIP-routed networks.

2.2.2 OSPF

OSPF is a link-state routing protocol. All links connected to OSPF enabled routers are monitored, with changes in link status being broadcast across the entire network [32, 18]. Based on the link state messages generated by OSPF routers, each router builds a identical graph of the overall network topology, and uses this graph to compute routes. It is critical that these graphs contain exactly the same information, because routing loops can result if two routers build conflicting forwarding tables. Unlike RIP, OSPF can operate on a hierarchical network of limited size, usually smaller than an AS. The information distributed through OSPF could allow for BGP-like business policies to be enforced on OSPF networks, but is impractical on the Internet scale. On larger networks, OSPF link state update packets consume a significant amount of network capacity, an the level of overhead becomes unacceptable.

2.3 IP Mobility

In addition to the features provided by routing protocols, new Internet protocols and specifications provide additional functionality that will drive future applications. Many of these features are not currently well known by the end users of the Internet, but they have great potential. It is critical that any new inter-domain routing proposal enable these features, if not enhance them. IP Mobility is one such standard.

IP Mobility provides inter-domain mobility for end users. RFC 3344[20] specifies a means by which communications may resume uninterrupted without changing connection parameters after a mobile end host migrates from one location on the Internet to another. The RFC suggests the creation of ”agents” which exist on every network that supports mobility and handle the connection transfer across domains.

If a mobile host moves in the midst of communicating with another host, the agent in the network from which the mobile host moved will assume responsibility for forwarding packets to the new home of the mobile host. The mobile host is responsible for notifying its "home agent" of its new location.

Packets are forwarded between the home agent and the mobile host through a "tunnel"[20]. A tunnel encapsulates a packet by adding a new set of addresses around the original packet. When a mobile node moves from one network to another, it maintains its old IP address while acquiring a new address from its new network. When the mobile host notifies its home agent of its new location, it includes the new IP address in the update. When the home agent receives a data packet for the mobile host, it encapsulates the packet with the new IP address, leaving the original packet untouched. The destination network is responsible for decapsulating this data packet and delivering it to the mobile host.

This method can result in serious inefficiencies. In the worst case, the mobile host's home location is relatively distant from the other end of its connection, and its roaming location is close to the packet source. Even if both computers are physically in the same room, packets will be sent from the source to the home agent, and then forwarded back to the mobile host, instead of being sent directly between the two hosts. This is called the "triangle routing problem" because of the triangular paths made between the two end hosts and the home agent. Figure 2.1 shows an example of a situation where IP Mobility performs poorly. If Host A communicates with Mobile Host through its Home Agent, the connection traverses two low bandwidth links, decreasing connection performance. This efficiency issue significantly hampers adoption of Mobile IP in its current form [34].

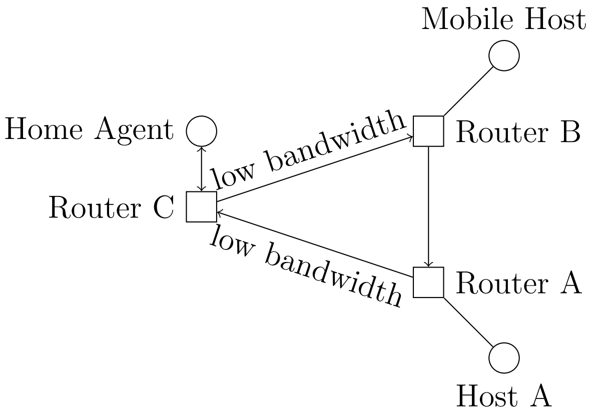


Figure 2.1: The Triangle Routing Problem

2.4 Current Issues

The Internet currently routes packets successfully on both the intra-domain and inter-domain levels, but shadows of problems lurk on the horizon. Recent BGP data publicly available from a number of ASes show the quantity of routes advertised is increasing exponentially over time [16]. The current BGP routing table is larger than 500 Megabytes in text format, and the exponential growth rate requires additional router memory. These increased requirements incur high hardware upgrade costs. Ideally, a more scalable routing solution should take the place of BGP and provide similar connectivity while keeping new hardware costs minimal. This problem is not new, and a significant amount of research has already been done in this area.

2.4.1 Related Work

Simplification of routing tables was first investigated in 1977 by Kleinrock and Kamoun [11]. As mentioned in their paper, their work was performed shortly after the birth of packet switching, but they recognized the potential expansion of computer networks to "even possibly thousands of nodes." They foresaw the future memory costs and bandwidth overhead of unoptimized routing on such a network, and determined that a highly scalable system is necessary to ensure long term network operability. They proposed a concept called "Hierarchical Routing," which enables tree-like groupings of related addresses when distributing and storing routes. The actual physical structure of the network is not required to be hierarchical; the system works well on distributed topologies. They also present a key metric for quantifying performance of hierarchical networks called "stretch," which is the ratio of actual path length to the optimal path length. This metric has since gained great significance and usage in the routing protocol design community.

In 1988, Tsuchiya presented his paper on the so-called "Landmark Hierarchy" [38]. Like Kleinrock and Kamoun, his network provided hierarchical routing, but with an underlying physical topology much closer to today's Internet. He provided a set of routing algorithms for distributed computation of hierarchical routes across multiple hosts.

Much of today's work surrounding compressing the routing table while maintaining connectivity is given the title "compact routing." Cowen[4] formulated proof of a method of compact routing that uses less than linear space at each node of a compactly routed network. Thorup and Zwick[36] further reduced the space requirements of compact routing schemes while maintaining stretch factors comparable to Cowen's. More recent work by Abraham[1] describes possible future directions of compact routing through better algorithms and higher efficiency techniques.

Researchers have also presented various compact routing-based solutions that interface with the current system to solve the routing table size problem. One approach, suggested by Subramanian et al.[30] in 2004 proposes modifying the "granularity" of distributed routes to the AS level, essentially reducing the total number of routes stored in memory. They combine the introduction of larger route granularity with a form of hierarchical routing, and introduce it as a replacement for BGP. This new protocol requires that all Autonomous Systems on the Internet publicly disclose their business relationships with each other. This may be looked on unfavorably by many organizations, but other recent research shows that inter-AS relationships can be deduced currently from AS information propagated through BGP [29].

Krioukov's overview of large-scale routing[12, 13] reaffirm the need for decreased routing table size, but state that AS-granularity routing is unacceptable due to its exponential growth rate, as data suggested in 2004. More recent data tends to contradict that AS growth rates are as fast as Krioukov claims, leaving the area open for study. Krioukov does make an important point when he describes the need for a location-identifier split in addressing. In the current Internet architecture, node identification is based on IP address. This prevents mobility, in the sense that a computer's identity is based on its location, and not on more stable attributes of that computer. The concept of the location identifier split is critical for abstracting away a computer's physical location on the Internet and providing routing services to hosts regardless of their network connection point. This is also a key feature for Mobility, where it is necessary to identify a host based on factors other than location.

While a great deal of research has gone into compact routing, a relatively small amount of theoretical research has looked into changing the overall routing paradigm. A concept called "reactive routing" suggests that routers query the network to find the packet's destination, rather than maintaining all routes in memory simultaneously [26]. This has the advantage of maintaining fresh data on often-used routes. At the same time, if a router never uses a route; it will never look it up, and hence not waste any memory. This has largest advantages for routers that only route to a few destinations; the more destinations a router must forward packets to, the more memory it must use. Feamster et al.[7] note that reactive routing improves performance in networks with path failures, and provide data on conditions in which this performance boost is observed. Unfortunately, most of their research deals with peer-to-peer overlay networks on top of the IP layer, but their observations on reliability are still valid.

2.4.2 Metrics

There are various metrics that describe qualities of routing protocols in quantifiable terms for efficient comparison. When determining the applicability of a routing protocol to a specific situation, it is important to take these metrics into account.

The most important metric in Internet routing describes whether the routing scheme provides complete connectivity between all pairs of end hosts, despite the presence of physical links and restrictive business policies. The Internet Routing Task Force, a committee of the Internet Engineering Task Force, lists "timely routing to all reachable destinations" first as a requirement for inter-domain routing systems [5]. Another key routing requirement is the ability to handle a large number of nodes and deal well with scaling of the network. This requirement is becoming increasinly important, as discussed in the introduction.

In their document, the IRTF describes the need for a decentralized solution to Internet routing. Previous Requests for Comments (RFCs) published by the Internet Engineering Task Force (IETF) state that any new solution designed for the Internet should not use a centralized system [17]. With centralized solutions, growth at the network edge requires an equal amount of growth in the systems that provide service to the network edge. If the network edge were to expand exponentially, a centralized service would have to match that growth. However, by placing control and information at key locations around the network edge, the services scale automatically at the same rate as the edge.

Finally, the document requires the "Provision of a Credible Environment"[5]. Security is a hot-button issue on the Internet, with ever-present threats of human and software attack against Internet resources. Any protocol or system designed for routing on the Internet must be capable of providing integrity and security services for routing. Attackers should not be able to inject routes at will or misrepresent routes for any purpose. If a protocol cannot provide these features, it will surely be rejected as a viable system.

Specialized metrics are used when designing protocols within the fields of compact and hierarchical routing. Stretch, as presented by Kleinrock and Kamoun, is a key factor in determining the performance of a compact routing system, and much work has focused on minimizing both routing table size and stretch simultaneously. Granularity is another important metric, indirectly related to stretch. As routes become more fine-grained, stretch decreases, but routing table sizes increase as well.

A less quantifiable metric for systems is deployability. The problem of Internet routing table growth is very real, and any solution to this problem must be deployable across the entire Internet. The IPv6 transition debacle[39] demonstrates the difficulties of simply "flipping the switch" from one protocol to another on the Internet. A usable protocol should have a smooth implementation path from the current design to the new design, with incentive for the private companies that provide Internet infrastructure to upgrade their own systems to support the new technology.

Thus, a solution to the problem of exponentially increasing numbers of Internet routes must fit within the constraints of both regular Internet routing protocols as well as with those of Compact Routing solutions. It must be able to preserve the business relations at the AS level for the transit and peering agreements to remain

Name	Connectivity	Decentralized	Security	Stretch	Deployability	Features
APT	1	2	2	2	1	1
CRIO	2	2	2	1	1	1
LISP	2	2	3	2	1	2
IPvLX	2	2	2	2	1	1
Ivip	2	2	1	2	1	2
Six/One	2	2	1	2	2	1
TRRP	2	2	2	2	1	2

Table 2.1: Breakdown of Routing Proposals (scale of 1 (worst) - 3 (best))

valid. It must also provide complete network service to all hosts while reducing the routing table growth rate.

2.4.3 Current Proposals

The IETF Routing Research Group (RRG) is currently entertaining a number of proposals to solve the problem of growing routing tables [27]. The majority of proposals involve implementing a location identifier split as described in Krioukov. The following summary of current proposals should give some light as to the state of the art in modern routing research. Table 2.1 contains a summary of protocol compliance with the above metrics.

APT

The APT proposal by Jen et al.[2] specifies a form of tunneling requiring no modifications to end hosts. It splits the network in two segments, the transit space, and the delivery space. Internet Service Providers (ISPs) make up the transit space, while edge networks and ISP clients make up the delivery space. Each space has its own set of addresses. When an end host sends a packet, it is encapsulated by the ISP in a standard UDP packet. The address lookup between delivery space and transit space is performed by the encapsulating router against a dedicated system within each network which has all routes in memory.

There are a couple issues with this proposal that make implementation difficult. The separation of the IP space into two areas could possibly raise issues with connectivity between address spaces not being available; it is questionable whether the transit addresses can be referenced from the delivery space. Also, a single device on each network maintaining all the mappings as specified could present problems as the edge scales. Finally, the number of encapsulation/decapsulation devices hosted by each ISP must also scale as the size of the network edge increases, potentially leading to problems in the future.

CRIO

CRIO, or Core Router-Integrated Overlay seeks to decouple network hierarchy from addressing hierarchy [41]. The researchers claim to have reduced the FIB size of a Tier 1 Internet Service Provider in simulations by 10 times with very little path length penalty. By decoupling addressing hierarchy, they remove the requirement that all nodes within a site contain IP addresses from their own assigned IP space. Each router simply sends packets to its neighbor with the closest address to its destination. If necessary, a router may use a GRE or MPLS tunnel to forward the packet to its ultimate correct destination.

This proposal is interesting, but it makes policy-based routing more difficult overall; tunnels may be used to send traffic where the sender did not intend. Also, the tunneling mechanism may result in packets ”backtracking” once encapsulated. The proposal was not significantly clear what would happen on large networks. Finally, this proposal did not mention explicit support for advanced features (such as mobility) within the protocol itself, leaving the proposal in need of clarification before it is considered more in depth.

LISP

LISP is by far the most popular proposal on the RRG as of March 2009 [27, 6]. LISP is a relatively straightforward system which achieves its goals through encapsulation. The protocol stacks of end hosts on the LISP Internet would be no different from those on current end hosts; encapsulation routers would be responsible for all tunneling of packets over the Internet. The encapsulating address would be an address reflective of the node's position in the network hierarchy, while the encapsulated address would be reflective of the node's identity. LISP uses UDP packet encapsulation, where original packets are captured from the IP layer up, and encapsulated in another UDP packet with a special LISP prefix header. In determining the mapping between node identity and the node's position, LISP uses a special new protocol.

Only two encapsulation headers are permitted in this scheme, which may prevent further extensions of the protocol. LISP's form of encapsulation is slightly more inefficient than IP-over-IP encapsulation in packet size. The assignment scheme for network-hierarchy addresses remains unsolved. Finally, the lookup scheme proposed by LISP is an entirely new system which requires new infrastructure. Overall, the concepts presented by the LISP system are interesting, but its system is unproven, which may result in slow adoption.

IPvLX

IPvLX is a dual-protocol IPv4/IPv6 solution. All nodes have an IPv6 address that uniquely identifies them on a global scale, while some have an IPv4 address for routing purposes [35] . All IPv6 packets traversing inter-domain space are first encapsulated in an IPv4 packet with the IPvLX extensions. The sender and receiver nodes are responsible for looking up the IPv4 addresses with which to encapsulate their packets; which helps distribute the encapsulation workload between the hosts on the edge of the network. The basic concept is that the IPv4 addresses for routing nodes will allow aggregation of many addresses into one ”destination” IP address which reduces the load on routing tables.

Unfortunately, this scheme has a few drawbacks. First, the reliance on IPv6 could be a problem in the immediate future, as IPv6 has not been as widely adopted in the United States. Also, it forces all end hosts to support DNS as part of their routing stack, which could break compatibility. Finally, they do not present a transition method by which a smooth switch could be made from the current Internet structure to the new system. Without good solutions to these problems, this proposal is less desirable.

Ivip

Ivip, standing for Internet Vastly Improved Plumbing, is another proposal suggesting core-edge separation of the Internet [10]. Again, like the other proposals, it suggests encapsulation of packets, but with the goal of eventually modifying the IP packet format to include extensions for the new routing system. In its favor, the proposal does discuss a means for mobility using the protocol.

This solution proposes using a system for distributing information which is not fully described, except as an abstract database of sorts. Before such a proposal is to be considered for adoption, the database design must be completed. Similarly, the proposal lacks a complete description of the path to adoption. With more information, this proposal could be viable.

Six/One

The Six/One proposal, unlike many of the other proposals, only uses one IPv6 address space, using subnets to partition and route packets [28]. Most of the processing is moved to the network edge, with the exception of routers changing packet addresses in transit, which may incur some core processing overhead. Six/One uses a global database of undefined type to perform transit address lookups.

Like IPv6, the transition path to Six/One relies on vendors building in support for Six/One into a significant portion of their hardware. Also, Six/One requires IPv6, which is not yet widely deployed. This reliance on new hardware makes deployment uncertain, as shown with IPv6 [39].

TRRP

Tunneling Route Reduction Protocol (TRRP) is a simple proposal for using DNS to distribute gateway addresses globally, essentially providing reactive routing capabilities to end users [37]. They provide a very methodical step-by-step implementation plan, and describe their use of tunnels to cut down on the number of Internet routes very well. The use of reverse lookups on IP addresses to determine destination encapsulation is particularly enlightened.

While TRRP's use of DNS is admirable, their use of TXT records may severely hamper their implementation. TXT records are designed for general use, and a more specific resource record type may provide more compatibility with existing uses of TXT records and less conflict with future uses of TXT records. Also, the authors do not discuss in detail the number of tunnels required for average use and the operating system overhead incurred. Finally, multiple layers of encapsulation are impossible in this system.

2.4.4 Summary

A great deal of work has been done in the area of packet routing. Solutions for intra-domain routing such as RIP and OSPF work well for smaller networks, and BGP has successfully provided routing capabilities for the global Internet. Unfortunately, none of these protocols is able to provide the scalability necessary for future Internet growth while allowing current business practices to continue. A number of proposals have been made that suggest new inter-domain routing systems, but all face shortcomings. There is currently a need for a viable inter-domain routing system that provides scalability and support for business decisions in routing. IP Mobility support within inter-domain routing is nonexistent, and within a proper routing system design IP Mobility could be improved substantially.

Chapter 3

Implementation

All of the previously discussed solutions contain valuable concepts, but all are rendered undeployable due to some shortcoming. The solution proposed here strives to meet all of the criteria outlined by the IRTF to the best extent possible. Because of the complexity of a complete routing solution for the Internet, some details are out of the scope of this thesis. For instance, the transition strategy is a highly complex problem and contains many sub-issues. These details are currently being researched, implemented, and tested, and the results will be discussed in a separate thesis published by Bryan Clevenger. This discussion will focus specifically around the reactive routing solution for the Internet, current Internet infrastructure systems leveraged for routing, and enhancements to protocols made possible by this system.

3.1 Routing Framework

The key to efficient routing is a hierarchical structure, much like that proposed by Kleinrock and Kamoun. This approach also lends itself to providing a Location-Identifier split as described by Krioukov. The critical issue faced by many modern proponents of Hierarchical Routing is defining the structure of the hierarchy within the modern Internet. The new hierarchy must support current business relationships, and not needlessly subjugate one organization under another. Thus, a simple and fair solution to defining the hierarchy is found in using the set of Autonomous Systems as the first level of nodes within the hierarchy. Every Autonomous System is equal within the hierarchy, and they are free to specify their own internal routing structure autonomously. This essentially reduces the granularity of globally distributed routing information to the transit AS-level.

Within this architecture, BGP continues distributing routes for the first level of the hierarchy. Core AS routers advertise their AS number through BGP in the form of a single /32 IPv4 route. The mapping between the ASN and the IPv4 address advertised is beyond the scope of this paper. Core routers will route packets based on AS-level information, rather than on current CIDR subnets advertised through BGP.

3.2 Encapsulation

In using Hierarchical Routing, packets traversing the network must be encapsulated with routing information for each level of the hierarchy. Because compatibility with existing routing equipment is critical, the encapsulation must be routable by current systems. For this reason, GRE was chosen as the best option for encapsulation. Not only is GRE well supported by many vendors, it also allows for variable types of encapsulated data; any protocol can be encapsulated by an IPv4 packet. Also, GRE also allows for multiple encapsulations, where a GRE packet can be encapsulated by another GRE packet. Levels in the hierarchy are referred to as “strata;” the highest level (outermost encapsulation) is strata 0. Since the AS is considered the first level of the hierarchy, strata 0 should carry AS-level routing information. The source address of the strata 0 header is the source ASN, and the destination address is the destination ASN. For a flat hierarchy within a destination AS, only two layers are required, with the strata 0 header populated as described above, and the strata 1 header being the original header sent from the packet source. Because scalability is critical, end hosts should encapsulate their own traffic to distribute processing.

Once a packet reaches its destination AS, it must be decapsulated. Ideally, this function should be performed by the router delivering it to the final network. Unfortunately, this function is not supported in current router firmware, and will likely take time to implement and be deployed. For this reason, dedicated decapsulation hosts are the recommended short-term solution. Internet core routers receiving an encapsulated packet with a destination address matching the local IP-mapped ASN should forward it to their local decapsulation device. The decapsulation device will decapsulate the packet, multiple times if necessary, then forward it to its target. Thus, end hosts send their packets encapsulated, and receive them decapsulated.

3.3 Lookups

This system requires modification to the procedure of sending a packet from one host to another across AS-boundaries. End hosts are now responsible for encapsulating a packet as necessary to ensure it reaches its destination. After constructing the standard IP header for a packet, the host must determine if encapsulation is necessary, and if so, determine the appropriate encapsulating source and destination addresses. This is done through a reactive lookup system, as suggested in many other proposals.

DNS is used here for reactive distribution of host specific routing data, for several reasons. First, control of DNS is distributed to organizations by a recognized central

Prio	Pref	Strata	Flags	Type	Data

Table 3.1: Structure of ENCAP Record

authority. Organizations are free to modify their own entries as they deem necessary, allowing distributed, autonomous control. Secondly, DNS is already widely deployed and well understood. It is critical for many common Internet activities, such as web browsing and email. Thirdly, DNS is already used in establishing some connections, such as in web browsing. It is trivial to return encapsulation information along with IP address lookups on server names. Finally, DNS can be made secure through such technologies such as DNSSEC, which allows for signing of records, preventing spoofing attacks. It would be possible to use an entirely new system for routing data lookups rather than DNS, but the overhead in designing, testing, and deploying such a new system may prove impractical.

An end host can look up appropriate strata 0 source and destination addresses by performing a reverse DNS lookup on the original source and destination addresses of the packet. These typically take the form of ZZZ.YYY.XXX.WWW.in-addr.arpa for an IPv4 address WWW.XXX.YYY.ZZZ, and are currently used to find the DNS name of an IP address. In this solution, routing records are distributed within DNS through a new record type, called the ENCAP record. ENCAP records can be returned as results from a forward or reverse DNS lookup on an IP address, and contain 6 fields, detailed in Table 3.1 and described below.

3.3.1 ENCAP Records

As shown in Table 3.1, there are a number of fields distributed within an ENCAP record. The Data field itself is variable length, and specifies the encapsulation address.

The two-byte Type field defines the type and length of data within the Data field. Currently, two types are implemented: ASN, and A. ASN signifies that a four byte ASN is contained in the data field, and the packet should be encapsulated using the IPv4 mapped address. The A record corresponds with the standard DNS A record, and signifies that a four byte IPv4 network address is contained in the Data field.

The one-byte Flags field of the ENCAP record contains various informational flags from the server. All bits in this field are reserved for future use. The one-byte Strata field specifies the level of encapsulation for which the Data field is valid, within the range 0-255. A correct response from a DNS server with maximum strata n must contain all strata $0...n-1$ as well.

The one-byte Priority and Preference fields allow administrators to provide failover and load balancing capabilities for encapsulation addresses. The Priority field provides a strong mandate; if several results having the same strata designation are returned, the client should choose the result having the highest priority, and fail over to those with lower priorities. The Preference field provides a weaker suggestion for load-balancing purposes. This field is used to make an encapsulation addressing decision in the case that the strata and priority fields are the same between several results. For i results of a given strata and priority, the decision algorithm sums the preference values of all i results, and chooses one result j from the group with probability $pref(j)/sum(pref(i))$. This allows great flexibility in weighting, providing administrators with the ability to load balance through the encapsulation itself.

3.3.2 Considerations

In using an OSI Layer 7 protocol as a lookup for an OSI Layer 3 addressing scheme, a circular dependency is being introduced. It would be possible to reinvent a new naming service that operates at Layer 3 or lower which would replace DNS and support ENCAP records; however, a new naming system would take a long time to gain acceptance and reach widespread adoption.

DNS, in its current state, already involves similar circular dependencies. In order for a host looking up a name to contact the authoritative DNS server for a domain, it must know the name and IP address of that server. This introduces a circular dependency when the host must contact an Internet root nameserver, but a lookup would be required to do so. To solve this problem, DNS resolvers use a ”hints” file with the names and IP addresses of the root nameservers. As the lookup continues, each nameserver provides contact information for the next in the chain until the authoritative nameserver is reached.

The encapsulation address problem can be solved similarly. Since DNS resolvers already have the ”hints” file with names and addresses for root nameservers, a trivial change could include encapsulation data for the root nameservers as well; giving resolvers the resources necessary to begin a DNS lookup for a destination address. Again, each nameserver should provide information for the next in the chain, but in this case, the encapsulation data for future nameservers is also included in the response. This breaks the circular dependency by solving the base case.

Security within DNS is a subject of great concern. Recent attacks against the DNS system have revealed the widespread effects of record poisoning, which would be exacerbated by this routing system, were not appropriate measures taken. For adoption, this solution requires a significant move towards DNSSEC across the Internet. By certifying that records are not being spoofed, routing can be made more secure.

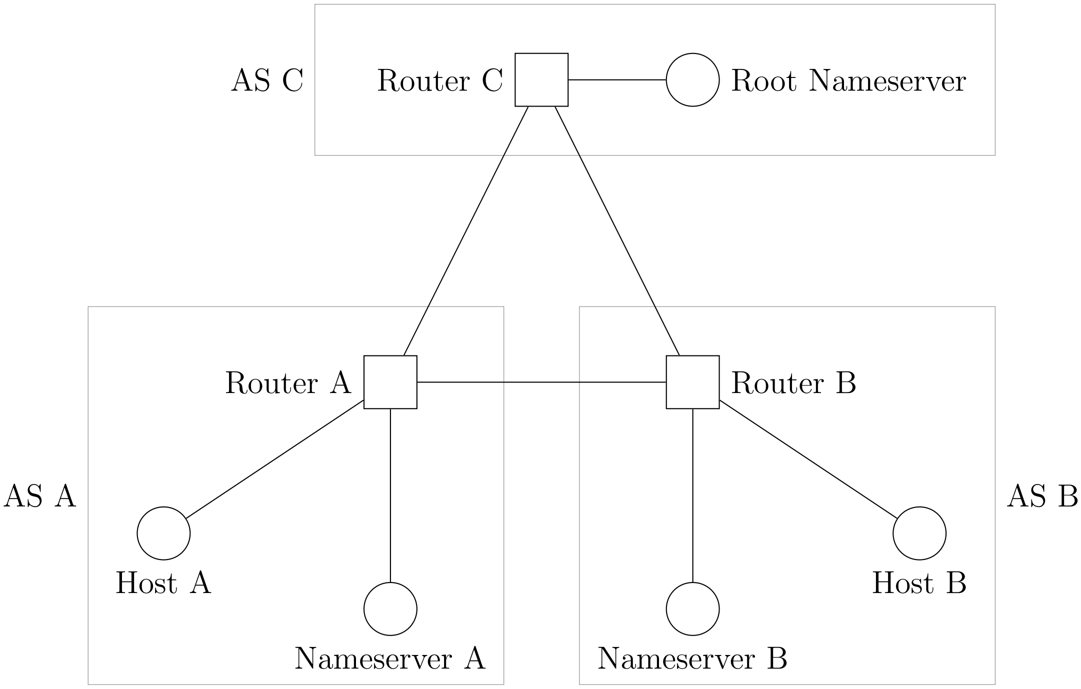
3.4 Adoption Path

The path to adoption of this system is subject matter of too large a scope for this thesis. Research is currently underway at Cal Poly to present a clear path to adoption providing incentive for all network participants at all stages of implementation. While much of the adoption strategy is left out of this thesis, support for legacy devices is critical to this solution.

Because legacy hosts cannot perform encapsulation for themselves, dedicated encapsulation devices can provide encapsulation and routing services for legacy hosts. These devices participate in DNS lookups and encapsulate packets as necessary. Dedicated encapsulation devices are critical for legacy device support in the context of this new routing scheme.

3.5 Example Transaction

To better illustrate the functionality of this system, a brief example is given of a packet's path. The topology of the example network is given in Figure 3.1. In this example, Host A initiates two-way communication with Host B using the new route lookup system. Each of the dotted boxes in the diagram represent a single Autonomous System. Each router shown has BGP sessions established with the other routers in other Autonomous Systems. Each AS has a nameserver that can provide authoritative reverse lookup results for its AS and serves as the recursive resolver for hosts within its AS. It is assumed that all hosts within an AS can contact their nameserver/recursive resolver without the need for encapsulation of any kind. Within an AS, encapsulation is not necessary, and the network operates directly with S1 addresses. This is believed to be a relatively safe assumption, given the availability



All Hosts/Nameservers perform encapsulation as necessary

Routers perform decapsulation

Figure 3.1: Example Network

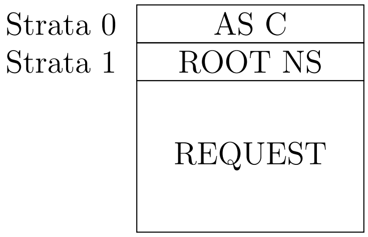


Figure 3.2: Encapsulated Request to Root Nameserver (Destination Addresses)

of recursive DNS resolution. Prior to the transaction taking place, each router will have distributed its ASN in the form of a /32 route to all other routers, so that each router has a single route for each AS in the form of an IPv4-mapped ASN. Each router also has a complete set of routes for its home network; i.e. Router A has routes to Host A and Nameserver A. Router A should not have direct routes to Nameserver B, or to the Root Nameserver. All nameserver caches are assumed to be empty at the start of this transaction, and routers are assumed to have firmware that supports decapsulation.

To initiate contact with Host B, Host A sends a recursive lookup request to Nameserver A, querying for Host B's ENCAP information.

Nameserver A's cache is empty, so it must start recursing at the root nameserver. To send a packet to the root nameserver, it must know the ENCAP records for the root nameserver. These are provided in the "hints" file discussed earlier that breaks the circular dependency. Nameserver A encapsulates its request for Host B's ENCAP records with destination addresses as shown in Figure 3.2, and sends the request to the Root Nameserver. Router C receives the encapsulated packet, decapsulates the request, and forwards it to the Root Nameserver for processing.

The Root Nameserver does not have ENCAP records stored for Host B, but it does have Nameserver B's record listed as the authoritative nameserver for that IP

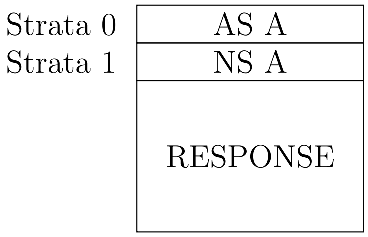


Figure 3.3: Encapsulated Response from Nameserver (Destination Addresses)

address space. Therefore, the Root Nameserver responds with Nameserver B's name as the authoritative nameserver, and includes Nameserver B's IP address and ENCAP records. It is critical that the Root Nameserver respond with Nameserver B's ENCAP records, because they are necessary for contacting Nameserver B and continuing the resolution.

Before sending the response, the Root Nameserver must determine the encapsulation data for Nameserver A with which to encapsulate the response. In this example, Nameserver A's encapsulation data is stored on the Root Nameserver, because the Root Nameserver is Nameserver A's direct parent in the resolution tree. Were Nameserver A further down the tree, the Root Nameserver would conduct a recursive resolution of Nameserver A's ENCAP records. Once the Root Nameserver sends the encapsulated response as shown in Figure 3.3, it is decapsulated by Router A, and forwarded to Nameserver A.

Nameserver A now continues its recursive lookup by contacting Nameserver B. It uses the ENCAP records included in the response from the root nameserver to encapsulate the query packet, which traverses the network, is decapsulated by Router B, and forwarded to Nameserver B.

Nameserver B has the appropriate encapsulation data for Host B in its authoritative records, so it prepares a response with that data. However, for Nameserver

B to send its response to Nameserver A, Nameserver B must know Nameserver A's encapsulation records. Therefore, Nameserver B contacts the Root Nameserver for Nameserver A's encapsulation records. It must use the Root Nameserver's encapsulation data stored in the hints file again, to prevent circular dependencies. Router C is again responsible for decapsulating the packet destined for the Root Nameserver.

The Root Nameserver returns Nameserver A's ENCAP records to Nameserver B. Now that Nameserver B can respond to Nameserver A's original query for Host B's ENCAP records, it does so. Router A decapsulates Nameserver B's response, and forwards it to Nameserver A. In turn, Nameserver A responds to Host A's recursive query for Host B's ENCAP records. Host A now encapsulates the original packet, and sends it to to Host B.

Before Host B sends a response packet, it will query Nameserver B for Host A's ENCAP records. Nameserver B will then query the Root Nameserver for Host A's ENCAP records, using the Root Server's ENCAP data from its own hint file. Again, the query is decapsulated by Router C and forwarded to the Root Nameserver.

The Root Nameserver, after looking up Nameserver B's encapsulation data, will send its encapsulated response to Nameserver B, signaling that Nameserver A is the authoritative nameserver for Host A's IP address. The packet is decapsulated en route by Router B, and passed to Nameserver B.

Nameserver B can retrieve Nameserver A's encapsulation records from its cache due to previous interaction with Nameserver A, and immediately send the query to Nameserver A. The query is decapsulated en route by Router A. Nameserver A, being authoritative for Host A, has Host A's ENCAP records in memory, and due to previous interaction with Nameserver B, should be able to send an encapsulated response to Nameserver B by way of decapsulation at Router B. Nameserver B can then pass Host A's encapsulation record to Host B, and Host B can encapsulate and send the response to the original packet.

At this point, Nameserver A and B have cached many of the critical encapsulation records to greatly increase the speed of future network operations. As long as Host A and Host B maintain an uninterrupted, non-mobile connection, they should cache the encapsulation records for their connections, making future DNS lookups only necessary for new connection establishment and refreshing caches should they expire.

Chapter 4

Mobility

As Krioukov discusses in his survey of networking, a Location-Identifier split is critical for an efficient implementation of Mobility. A mobile host must be addressable by the same identifier, no matter its location in the network, and routers must be able to route packets correctly even as the host moves. This solution to the expanding route table presented above enables a simple solution for Mobility with zero stretch.

It is important to distinguish between the different types of mobility. One form of mobility is also known as "wireless hand-off," and discusses maintaining connectivity with a host as it moves between wireless access points. Another form of mobility is local-network mobility, where a mobile host transitions between networks controlled by the same organization. Both of these forms of mobility usually take place within a single AS, and inter-domain routing does not play a direct role. The third type of mobility, in the context of an inter-domain routing system, allows a host to transition between networks in different Autonomous Systems while maintaining its active connections. This form of mobility can be improved greatly by a new routing architecture, and will be discussed here.

4.1 Mobile Agent Requirements

Correct functionality of IP mobility within this system requires the mobile host and its home network to meet certain requirements. First, the mobile host must be able to retain its IP address from its home network while acquiring a new one from its new host network. However, the mobile host should not retain any routes from its home network while roaming; all routes should use the IP address from its host network when sending packets. Also, the home network must not re-assign the mobile node's IP address obtained from the home network. The uniqueness of these IP addresses must be preserved.

Another requirement is that hosts wishing to become mobile must be direct participants in the new routing system. Legacy hosts using encapsulation and decapsulation services of their host network cannot become mobile, as mobile hosts are required to decapsulate some packets on their own.

4.2 Routing Framework Extensions

To support mobility, several features must be supported by the routing system. The routing system presented in the previous chapter supports all of these features. However; this form of mobility is not limited to the structure presented in the previous chapter.

4.2.1 Route Updates

For packets to reach a mobile host, routing tables must be updated to reflect the new location of that host. When using a reactive lookup system, the route distribution servers must be updated by the mobile host to respond to queries with the mobile host's new location. DNS already provides the capability for remote record updates through its DDNS service, which can be used to update ENCAP records. After connecting to the foreign network and acquiring an IP address there, a mobile host has full Internet connectivity through its new IP address. To activate Mobility, the mobile host can then contact its home DNS server using DDNS and provide it with new ENCAP records that will route data to its new location.

When a mobile host leaves its home network and moves to a foreign host network; it must update its encapsulation. The mobile host's encapsulation system must reflect that of its new host network. If n levels of encapsulation are used to locally identify a host on the foreign network, the mobile host can be identified by its **home** IP address through $n+1$ levels of encapsulation. For instance, the strata 0 information of the mobile host will be updated to reflect its new host AS. The strata 1 information should be the local address of the mobile host on the foreign network. The "home" address of the Mobile Host becomes strata 2. The decapsulation process as outlined in Chapter 3 continues as normal, with the network decapsulator removing the strata 0 header. The mobile host is responsible for removing the strata 1 header and interpreting the strata 2 header.

For Mobility to work, the mobile host's authoritative AS DNS server must accept a location update from the mobile host. By controlling record updates permissions, an organization can retain control over which IP addresses are allowed to be mobile. These features are already available in at least one major DNS server. IP addresses cannot be "stolen" from an address space without the operators permission, as mobility requires explicit permission and configuration by the network administrator.

4.2.2 Timeout Behavior

Mobility introduces a new set of unique challenges when dealing with lost packets and connection timeouts. If a stationary host in communication with a mobile host stops receiving responses, there are several possible reasons. Examples of such reasons include: the mobile host turning off unexpectedly, a temporary network fault, or migration of the mobile host. In any of these cases, the stationary host should perform a lookup at the mobile host's authoritative server to obtain the latest routing information for the mobile host.

The DNS protocol does not currently have a flag in the query packet to signify that the lookup should be carried out while ignoring caches. This means that deep lookups must be carried out manually by contacting the authoritative nameserver for the mobile host and requesting data. If the routing data has been updated since last contact with the mobile host, the encapsulation addresses should be modified accordingly, and communication re-established. If the routing data has not been updated, timeout behavior should be observed, and DNS should be periodically re-queried in the event that the host was in transit.

A more graceful alternative to timeouts that trigger route updates is desirable. Because the mobile host is still able to contact the stationary host, it can set an IP Options flag in its outgoing packet that signifies the mobile host has moved. This can short-circuit the timeout, and forces an immediate lookup on the mobile host's location at the authoritative nameserver, minimizing packet loss.

4.2.3 Legacy Hosts and Mobility

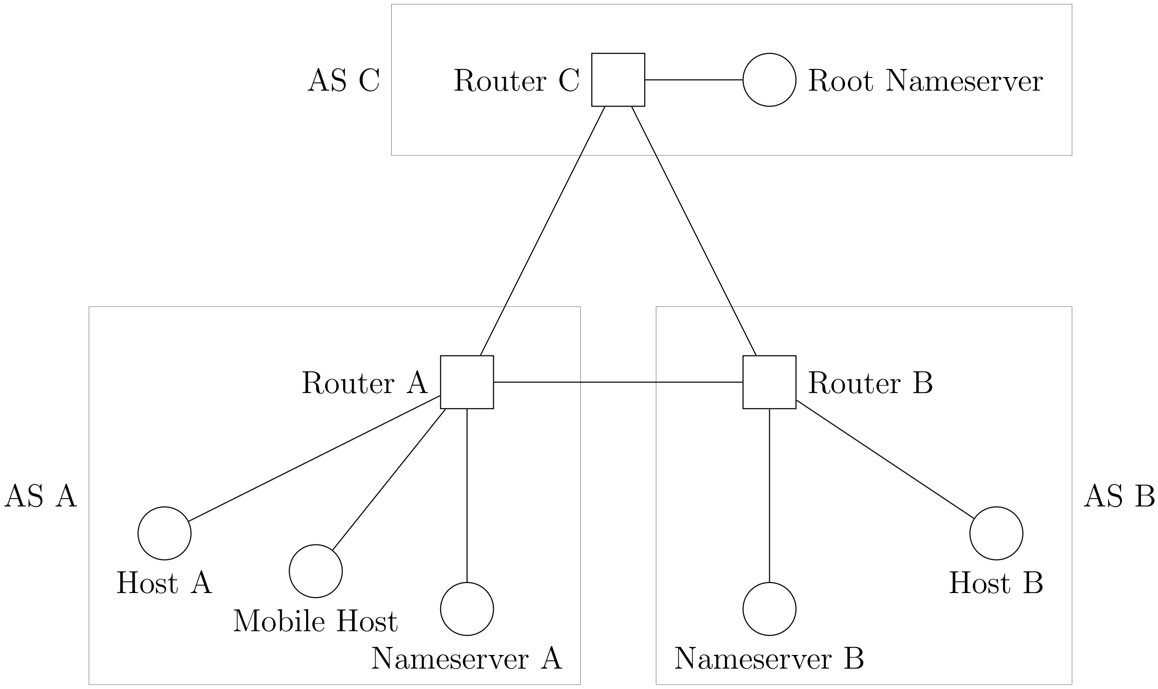
If a legacy host and a mobile host within the same AS are communicating when the mobile host moves, the connection must be maintained. Since the legacy host by definition does not support DNS-based encapsulation, another host on the network must perform the encapsulation on behalf of the legacy box and forward it to the mobile host. For this solution to work, the "encapsulation server" must emulate the mobile host's continued presence in the network through Proxy ARP, if on the same subnet as the mobile host.

In a larger network, where one encapsulation server serves many subnets, a more complex solution must be implemented. Currently, routers from a major vendor support intra-AS mobility [3]. These routers can be configured to allow the encapsulation server to participate in this mobility on behalf of the mobile host. This solves the problem, providing access to mobile hosts for all hosts in a home network.

4.3 Mobility Example

In Figure 4.1, the initial state of the network is shown. The Mobile Host is in its Home AS, and is routable by an IP address assigned to AS A. Host B is able to initiate a connection with Mobile Host in much the same way as described above in the discussion of non-mobile (static) hierarchical routing architecture. Nameserver A must be configured to allow Mobile Host to update its own records so Mobile Host can traverse the network freely.

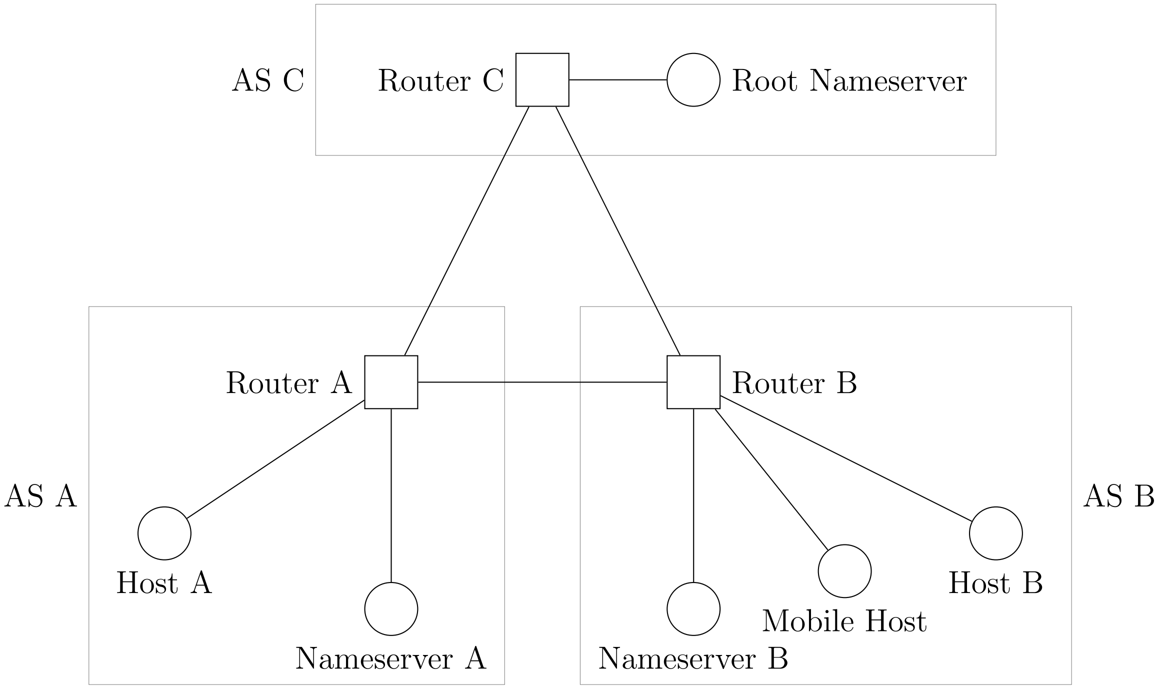
Let Mobile Host reconnect to AS B while maintaining its connection with Host B. Figure 4.2 shows the state of the network after Mobile Host switches physical networks. In doing so, Mobile Host acquires an IP address from AS B, while main-



All Hosts/Nameservers perform encapsulation as necessary

Routers perform decapsulation

Figure 4.1: Initial State of Mobility Example Network



All Hosts/Nameservers perform encapsulation as necessary

Routers perform decapsulation

Figure 4.2: Final State of Mobility Example Network

taining its original globally routable IP address from AS A on a virtual interface. Mobile Host now depends on Nameserver B as its local resolver, but Nameserver A is still authoritative for AS A's IP address. Router B serves as Mobile Host's default gateway.

At this point, the logical connection with Host B is interrupted, since packets addressed to Mobile Host are still routed to AS A. Mobile Host, at this point, is able to contact Nameserver A using its IP address assigned by AS B. Mobile Host updates the ENCAP records for Mobile Host's home IP address. It sets the first layer of encapsulation to be AS B's IPv4 address-mapped ASN, and the second layer of encapsulation to be Mobile Host's IP address belonging to AS B. Mobile Host then sends a packet to Host B with IP options set to signify that Mobile Host's ENCAP records have changed. Host B looks up Mobile Host's new encapsulation information, and adjusts its encapsulation procedure to match.

4.3.1 Local Connection Considerations

If Host A and Mobile Host are communicating when Mobile Host moves to AS B, their connection will be severed. This is because Host A does not use encapsulation to communicate with Mobile Host while both are in AS A. There are two possible solutions to re-establish communication between these hosts. These two solutions are not mutually exclusive, they could be used in parallel to provide a fail-over system.

Solution 1

As discussed earlier, dedicated encapsulation devices allow legacy hosts to use mobility in the new routing system. One such device could be added to AS A. This device would participate in local routing protocols, such as RIP or OSPF, advertising a route to the Mobile Host. It would also provide Proxy ARP service for hosts on the same subnet, advertising the IP address of the mobile host. If the device received a packet for the Mobile Host, it would encapsulate it and send it to the Mobile Host properly encapsulated. In this sense, the dedicated encapsulation device would impersonate the Mobile Host, forwarding all datagrams to it.

Solution 2

Mobile Host could again make use of the IP Options field to signify to Host A that it has moved, and a reactive route lookup and encapsulation are now necessary. Since Mobile Host is always able to send packets to Host A, the message would be received, and after Host A updates its records for Mobile Host, bidirectional communication is again established.

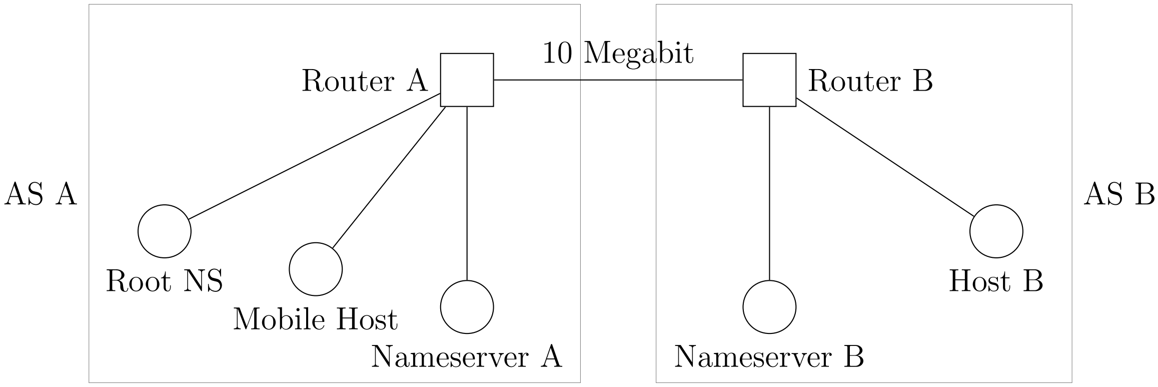
Chapter 5

Results

For evaluation of this reactive routing system, a small testbed was created. The details of the testbed are described below, followed by the results of several tests.

5.1 Testbed Details

The testbed differs little from previous diagrams of example networks. A full diagram of the experimental network is included here as Figure 5.1. Of particular note is the migration of the Root Nameserver to AS A. This is of relatively no consequence to the lookup process, except that recursive lookups conducted by Nameserver A do not need to be encapsulated to reach the root nameserver. Another detail that differs from the previous examples is the location of decapsulation. Because current router firmwares used in the testbed do not support decapsulation, the nameservers here are also used as decapsulation devices. In turn, the Nameservers forward the decapsulated contents of the packets as necessary within the AS.



All Hosts/Nameservers perform encapsulation as necessary

Nameservers perform decapsulation

Figure 5.1: Testbed used for generating results

5.2 Hardware

In implementing the system, equipment was used from Cal Poly's Cisco Advanced Networking Lab. Two Cisco 3640 routers running IOS 12.4(21) were used as Routers A and B. All hosts and nameservers are Dell Optiplex GX1s, with Pentium III processors clocked at 500 MHz. Two Cisco Catalyst 2900 series switches provide the backbone of the network between the router and the hosts. During normal operation, all links operate at 100 Megabits, with the exception of the link between Router A and Router B, which operates at 10 Megabits. The hardware used for testing is admittedly old, but the age of hardware has no impact on the correctness of the results, only the magnitude of the numbers.

System	Number of Lines Modified/Added
Bind 9.6.0-P1	400
PacketHandler	980
Linux Kernel 2.6.26	4

Table 5.1: Implementation Complexity

5.3 Software

In implementing a testbed to evaluate the various aspects of this system, a preliminary suite of software was developed. A brief summary of the tools is included below.

Since the method of route distribution is DNS-based, ISC Bind was a clear choice as the DNS server. The ENCAP resource record type was added, along with additional logic in the A, AAAA, and NS records to include the new resource record with a lookup for those types. Including the ENCAP record when other types of records are returned speeds the lookup process by caching the ENCAP record at the resolver before they are needed, and breaks the circular dependencies discussed earlier. Minor changes to the core logic of Bind were required to accept reverse mapped encapsulation records into the cache when returned as additional data. In all, roughly 400 lines of code were changed/added, with only minor logic changes. In this testbed, the nameservers run Bind-9.6.0-P1.

Because one goal of the routing architecture is deployablity on current operating systems, much of the encapsulation process is done in userspace. The encapsulation itself is performed by a user application developed jointly with Bryan Clevenger and John Bellardo called "PacketHandler." PacketHandler uses the Linux NetFilter system to siphon packets from the kernel routing process that are in need of encapsulation or decapsulation. Decapsulated packets are stripped of their AS encapsulation and re-routed by the kernel routing table. For packet encapsulation, PacketHandler captures the packet before it is routed by the kernel and performs the DNS lookups necessary to determine the AS-level destination. After encapsulating the packet as dictated by the ENCAP records, PacketHandler forwards the packet to the appropriate gateway.

In the development of PacketHandler, an interesting behavior was discovered in the Linux kernel that generated incorrect routing decisions. The code in the kernel responsible for writing IP packet headers set the destination IP address of ICMP and IP response packets to the source address of the kernel route, not the source address of the original packet. In some edge cases, the source address of the route in the kernel does not match the source IP of the packet itself. The PacketHandler code relies on correct behavior according to RFC 792 [24], which states that the destination IP address of an ICMP packet should be "The source network and address from the original datagram's data" [24, 25, 23]. Given that the Linux kernel made these wrong routing decisions, a slight modification (four lines) was made to the Linux kernel to rectify this error.

Also, since encapsulation is done outside the kernel, Maximum Transmission Unit measurements are rendered invalid. PacketHandler automatically calculates the size of encapsulation overhead, and if the size of the encapsulated packet exceeds the MTU, it will return an ICMP MTU exceeded message to the sender with an acceptable MTU based on the encapsulation overhead. This automates the process of adjusting the MTU based on each connection's differing encapsulation overhead.

5.4 Quantitative Results

In determining performance of the new routing system, it is important to profile the performance of the underlying hardware using the current (legacy) routing system.

In the context of the diagram in Figure 5.1, Routers A and B were given routes to all hosts on all networks, and measurements were carried out between Hosts A and B to determine raw network performance. In all latency measurements presented in this thesis, 300 ICMP requests were sent between two hosts, and the completion times are presented as a cumulative density function (CDF). A value of 0.5 on the Y axis signifies that 50 percent of responses were received before the corresponding X axis time value.

Similarly, the bandwidth measurements are presented as a CDF of completion time of a 20 megabyte file transfer. One host was set up as an FTP server, while the other used wget to retrieve files from the FTP server over the network. Each experiment involving the 20Mb file transfer was executed 100 times.

5.4.1 Baseline Performance

The data plots in Figure 5.2 represent the latency between Host A and Host B. In one plot, the ARP caches on Hosts A and B were cleared before each ICMP request, demonstrating the increased latency generated by performing ARP lookups.

Figure 5.3 shows a cumulative density function (CDF) of file transfer completion times for a 20 Mb file. The "Empty ARP Caches" label signifies that the ARP cache was cleared on Hosts A and B before the transfer began. As can be seen, clearing the ARP cache before the transfer does not significantly impact completion time of the transfer. Some anomalous high completion times with full cache initial conditions appear at the 95th percentile, but these results do not detract from the overall conclusion regarding the effect of ARP caches on transfer times.

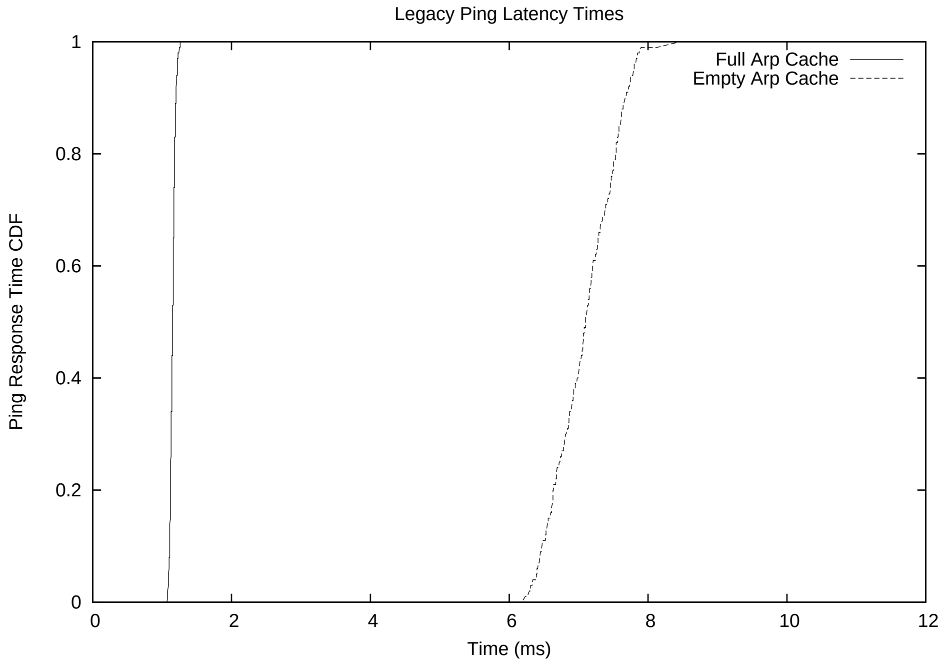


Figure 5.2: Baseline Latency Profile of Testbed

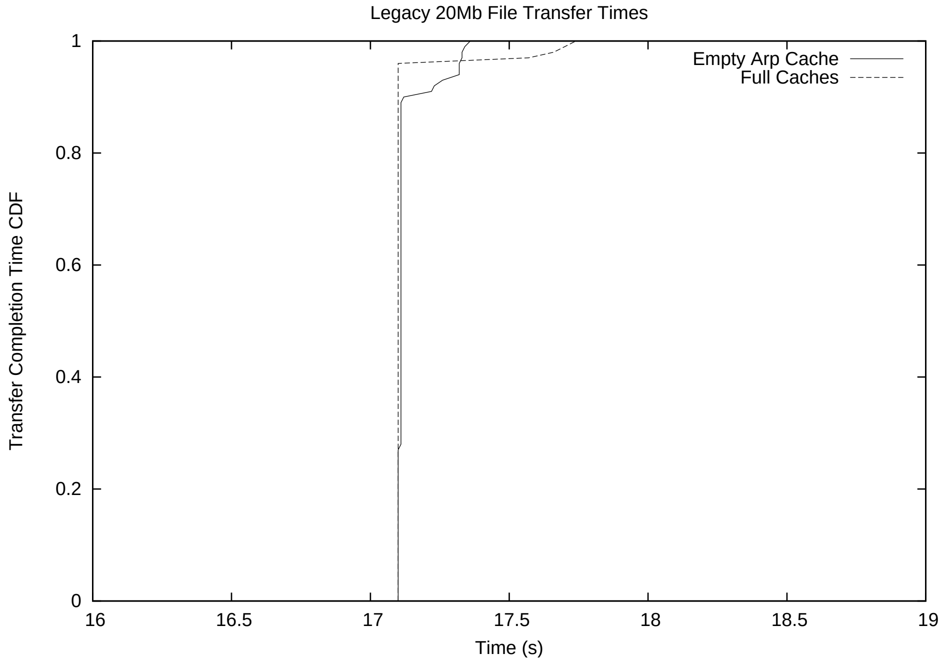


Figure 5.3: Baseline Bandwidth Profile of Testbed

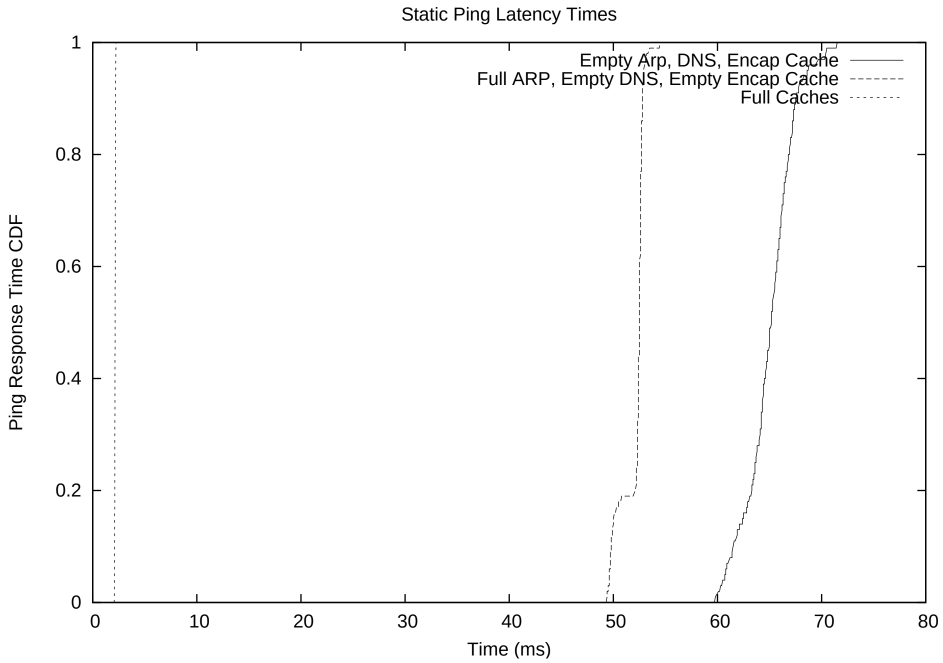


Figure 5.4: Static Hierarchical Routing Latency

5.4.2 Static Hierarchical Routing Performance

After completing a profile of the underlying hardware, the full hierarchical routing system was set up on the testbed as shown in Figure 5.1. For the following graphs, all hosts are static, mobility is not yet considered. Within the testbed, there are three primary forms of caching. The first is ARP caching, as previously discussed in terms of the legacy routing scheme. The second is DNS caching, taking place at the AS-level DNS servers. This functionality is currently included in Bind, allowing multiple hosts using the same nameserver to share query results for greater speed. The third type of caching is the Encap cache, which is located within the PacketHandler encapsulation process and stores encapsulation information on a per-destination basis. The Encap cache is designed to remove the need for performing a high-latency DNS request for every outgoing packet.

Figure 5.4 compares ICMP latency across a static hierarchical routing structure with various levels of caching. Three cases were chosen to represent possible different network conditions. The first case represents a new host joining a new network, and connecting to a previously uncontacted host. In this scenario, all three caches would be empty, and data would have to be requested to fill them. The second case, with empty Encap and DNS caches and full ARP caches, represents an established host within a network connecting to a previously uncontacted host. An outgoing packet will trigger requests for the encapsulation information, and both caches must be filled before the packet can be sent. The third case, with full caches, represents the continuation of a connection, or the average case latency for a packet within a stream. This case is logically equivalent to the "Full ARP Cache" scenario in the legacy routing system. As can be seen, there is a significant cost in latency when performing initial lookups for DNS and Encap records. Once the caches are populated, however, latency decreases significantly.

In contrast to the above discussion of initial packet latency, the file transfer times for a static hierarchical routing structure show very little variation with respect to the initial state of caches. In Figure 5.5, for all three cases described above, transfer times very closely match, suggesting that the initial state of the cache does not make a significant difference in throughput. The latencies of the initial lookups are simply absorbed by the TCP stream, minimizing their cost. As can be seen by comparing Figures 5.5 and 5.3, the total transfer time does increase, and a more complete discussion of the increase will be included later in this section.

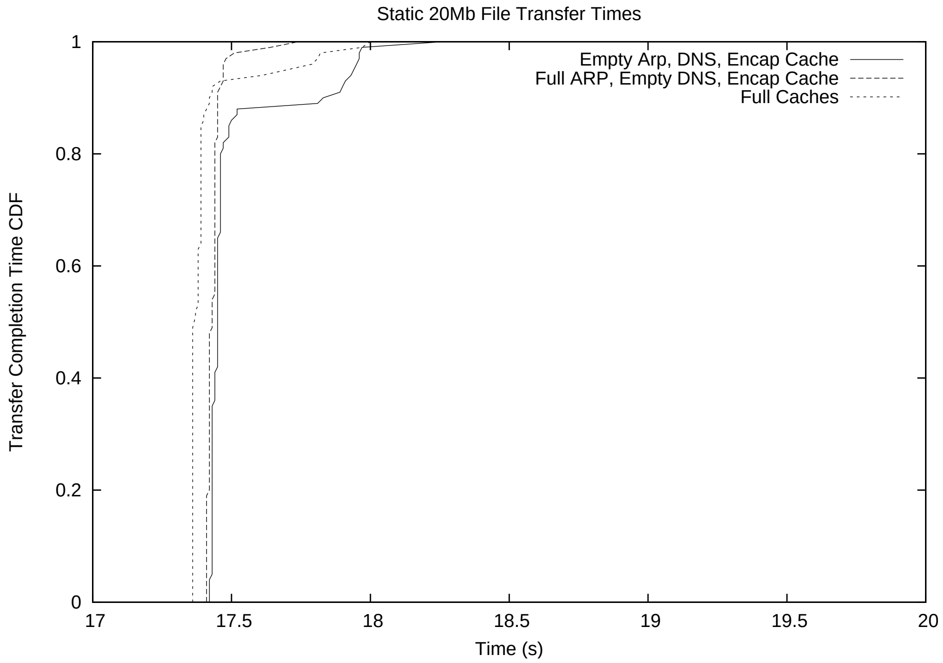


Figure 5.5: Static Hierarchical Routing Bandwidth

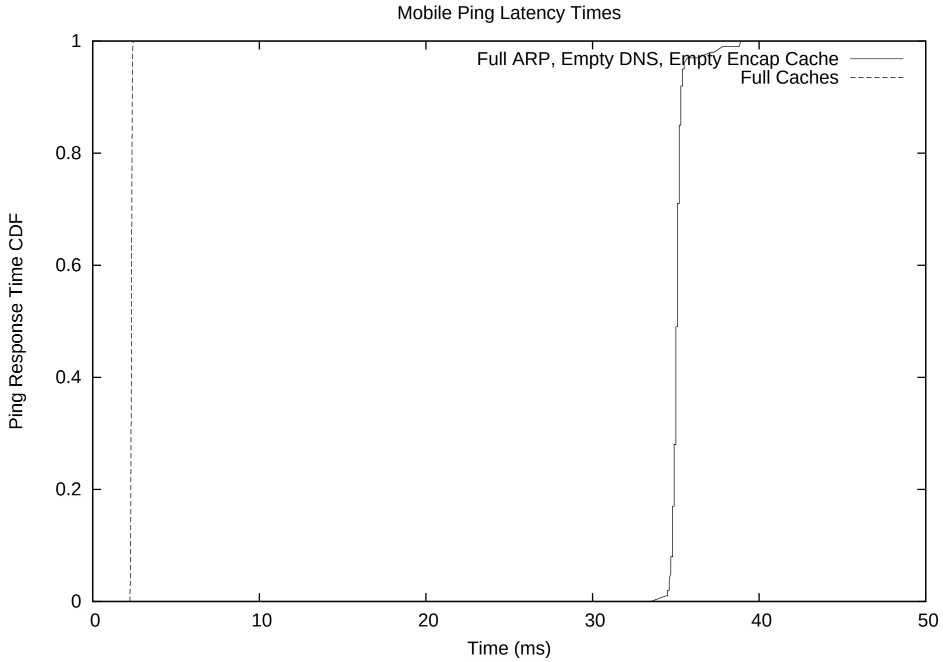


Figure 5.6: Mobile Hierarchical Routing Latency

5.4.3 Mobile Hierarchical Routing Performance

To test the performance of mobility, Mobile Host logically switched ASes, and the tests performed on the Static framework were re-run. No change in physical topology occurred. In previous tests, Mobile Host had its logical home in AS A, while Host B had its logical home in AS B. For this set of experiments, Mobile Host's logical home was changed to AS B, while maintaining it's physical connection to Router A. Thus, Mobile Host's home IP address belongs to AS B, while maintaining a secondary IP address in AS A's space. Mobile Host is therefore roaming in AS A while communicating with Host B using Mobile Host's AS B home IP address. As described above, Host B performs double-encapsulation to reach Mobile Host at its AS B home IP address, and Mobile Host performs single-encapsulation to respond.

Figure 5.6 shows a profile of packet latency between Mobile Host and Host B while Mobile Host is "roaming" in AS A. The classification of different types of connections follows those of the static hierarchical routing system, with the exclusion of the "ARP Cache Clear" case. Linux Kernel behavior necessitated the addition of a static ARP entry in the Mobile Host's table, precluding valid results from this test. The results for the two remaining tests tend to follow the trend of the static case, with full caches minimizing latency, and empty caches causing a significant increase in latency.

Figure 5.7 shows the bandwidth profile of a mobile host in the hierarchical routing system. Again, the results mirror those in the static host bandwidth profile, with initial cache state having a minimal impact on overall file transfer time. Measurements after clearing the ARP cache were also precluded during these tests due to kernel behavior.

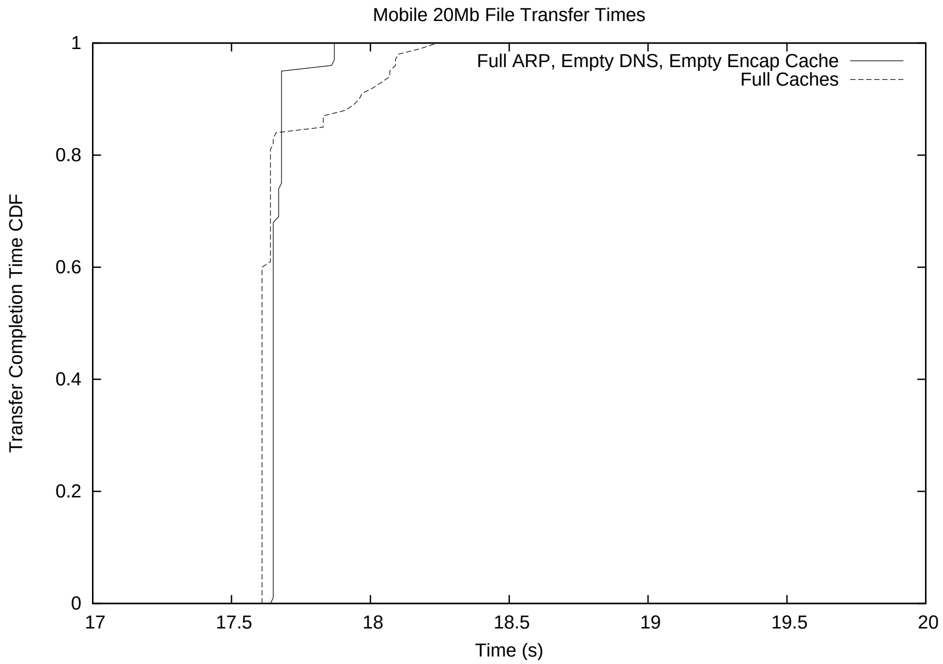


Figure 5.7: Mobile Hierarchical Routing Bandwidth

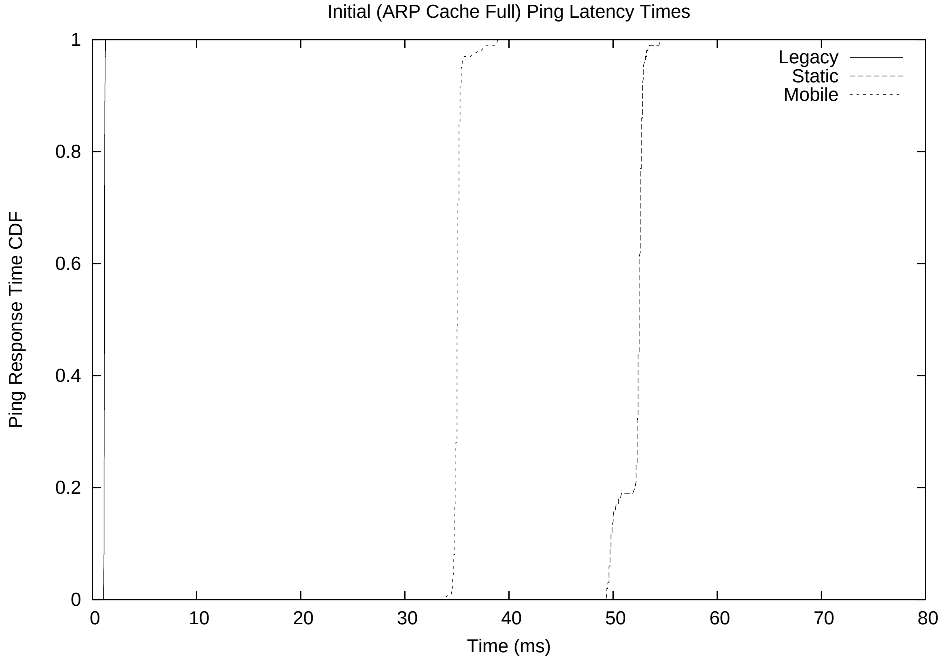


Figure 5.8: Comparison of Initial Latency (DNS and Encap Caches Cleared)

5.4.4 Comparison Data

Figure 5.8 compares the initial round-trip latency of ICMP packets between the three systems. The static and mobile hierarchical routing solutions lag significantly behind the legacy code, largely because of the latency of the extra DNS lookups. As the graph shows, there is less latency in the mobile case than in the static case. This s counter intuitive, but makes logical sense in the following context.

In the static case, Nameserver A is authoritative for Mobile Host, and Nameserver B is authoritative for Host B. In the mobile case, Nameserver B is authoritative for both Host B and Mobile Host due to the logical shift. When Mobile Host initiates a connection to Host B, it must perform parallel DNS lookups on the source and destination addresses of the packet. In the static case, Nameserver A is authoritative for the source address, and Nameserver B is authoritative for the destination address. In the mobile case, Nameserver B is authoritative for both addresses. For both cases, at least one request must traverse the link between Router A and B. Because all requests are done in parallel, the overhead of more than one request traversing that link are negligible. Once the resolution is complete, the packet is sent to Host B.

Upon Host B receiving the packet, it responds and initiates requests for the source and destination addresses of the response packet. In the mobile case, Nameserver B is authoritative for both IP addresses, removing the need for a full lookup to Nameserver A. The static case still requires traversing the link back to Nameserver A for information. This saved DNS lookup is responsible for the mobile case's lower latency. If, however, the mobile host is in communication with a host from outside its home network, this performance gain will not be observed, and performance will be similar to the static case. In the median case, mobile hierarchical routing will provide little to no initial packet latency cost over static hierarchical routing.

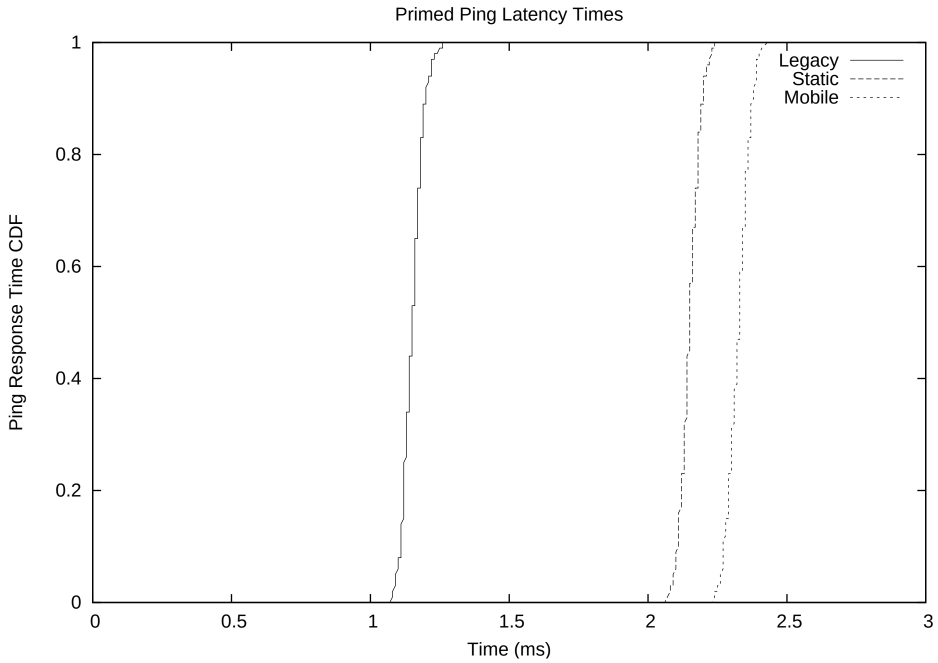


Figure 5.9: Comparison of Primed Connection Latency (All Caches Populated)

Figure 5.9 compares the full-cache round-trip latency of ICMP packets. Again, the comparison takes place across the three systems, and the results are similar to that presented in the initial latency chart. The round trip packet latency for the legacy system is significant less than for the static and mobile hierarchical routing systems. Here, mobility introduces little cost over the static system. The small increase in latency is due to the additional decapsulation step required on the mobile host. Since PacketHandler is a user-space utility, passing the packets through it for encapsulation and decapsulation requires multiple kernel-space to user-space transitions. These transitions require significant overhead, and are believed to constitute the majority of the latency increase shown in figure 5.9. With a kernel module that includes encapsulation and decapsulation directly in the IP stack, it is likely that this latency increase will shrink.

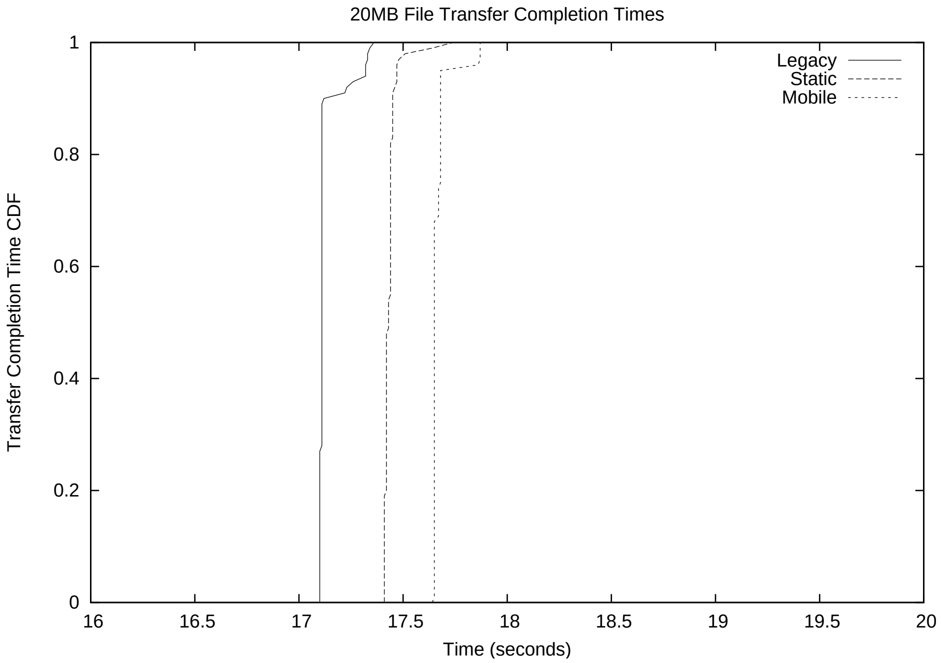


Figure 5.10: Comparison of File Transfer Times

The impact of these latency differences is investigated in figure 5.10. For this test case, the DNS and encapsulation caches were cleared in the hierarchical routing solutions before the file transfer began. Legacy file transfer again completes transfers before the hierarchical routing solution. Mobility incurs a small cost over static hierarchical routing. Much of the difference in time between the static and mobile solutions here is due to userspace transitions and delay in the kernel-userspace queuing. The extra layer of encapsulation used by mobility causes additional overhead in the data stream by increasing the header size, and increasing the total amount of data that must be transferred. This contributes to the higher transfer time of the mobility solution.

Packet count measurements were taken when measuring bandwidth of the system, and are shown in Figure 5.11. The results show that the per-packet overhead introduced by the hierarchical routing scheme has a tangible result on packet count.

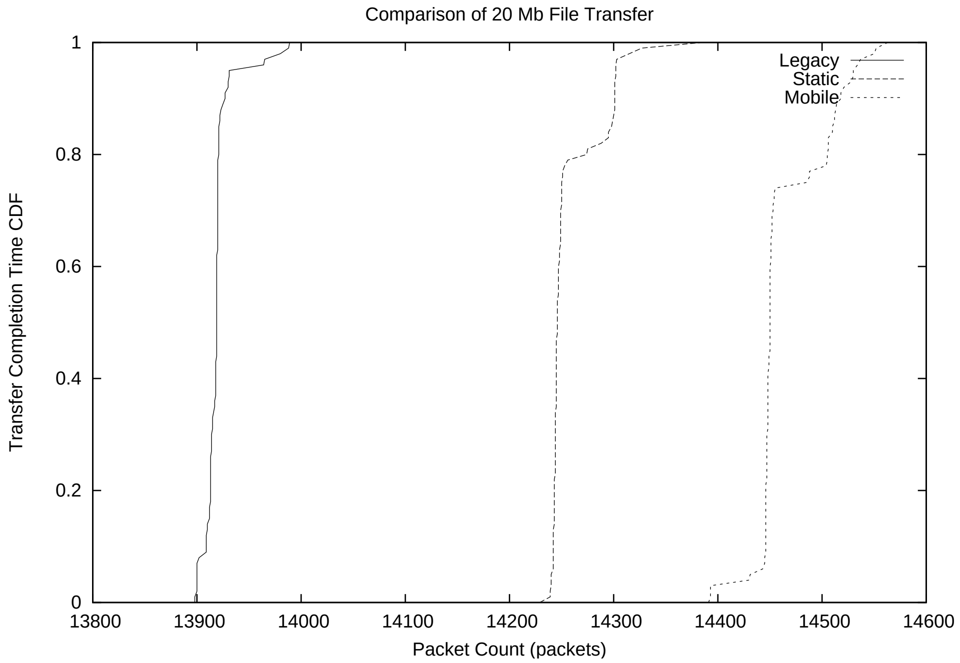


Figure 5.11: Comparison of File Transfer Packet Counts

Test	Legacy System	Static Solution	Mobile
Init. Latency (Caches Empty)	1.15 ms	52.06 ms	35.11 ms
Stream Latency (Caches Full)	1.15 ms	2.15 ms	2.33 ms
Transfer Time (Caches Full)	17.12 s	17.41 s	17.68 s
Packet Count (Caches Full)	13919	14257	14464

Table 5.2: Average Performance Measurements

The added header displaces a small amount of data from each packet, resulting in a higher overall packet count. The additional overhead caused by mobility further increases the packet count as expected. Increasing the MTU of the network would mitigate this problem, as it would allow for more data to be sent in each packet.

5.4.5 Summary

The hierarchical routing system in its current state introduces varying forms of overhead to data transfer to allow for minimal forwarding table size in core routers.

The effects of this overhead are most visible in initial packet latencies, where DNS and Encapsulation caches must be populated; and in packet counts, where additional headers decrease throughput. However, within the hierarchical routing system, mobility becomes a trivial problem, and can easily be supported with minimal overhead. When integrated with the kernel network stack, the latency and throughput issues caused by the kernel to userspace transitions will decrease, if not disappear, and performance differences between the hierarchical routing system and the raw hardware will diminish. IP Mobility is greatly improved by hierarchical routing, with little overhead.

Chapter 6

Conclusion

The system presented here provides a new, scalable approach to Internet routing. By changing the granularity of routing and the metric by which routing table size is determined, the routing table growth rate can be slowed significantly, saving money for ISPs globally. This solution moves processing to the network edge, instead of centralizing it in the core routers of Autonomous Systems. Reactive routing provides as-needed routes, dynamically adjusting the size of the cached routing table as needed. By embracing these more scalable principles, the growth rate of the Internet can continue unhampered by router hardware constraints. Additionally, now that mobility is a low-cost operation on top of the new routing system, it can be revisited as a viable tool for widespread use. Experimental results show that mobility has virtually no bandwidth cost over the baseline hierarchical routing implementation. Latency costs can easily be minimized by streamlining the implementation and removing the kernel-space to user-space transition. This initial implementation demonstrates the power of hierarchical routing in a global context.

These concepts have the potential to prevent BGP forwarding tables from growing too large for core routers and to give the Internet a more structured topology for future growth. These principles can guide the Internet into the future, providing global connectivity to an ever-expanding Internet.

Bibliography

[1] I. Abraham, C. Gavoille, D. Malkhi, N. Nisan, and M. Thorup. Compact name-independent routing with minimum stretch. 2008.

[2] APT: a practical transit mapping service. http://tools.ietf.org/html/draft-jen-apt-01.

[3] Cisco IOS IP configuration guide, release 12.2 - configuring IP addressing - cisco systems. http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfipadr.html#wp1001249.

[4] L. J. Cowen. Compact routing with minimum stretch. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 255–260. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 1999.

[5] E. Davies and A. Doria. Analysis of Inter-Domain routing requirements and history. http://smakd.potaroo.net/ietf/idref/draft-irtf-routing-history/index.html.

[6] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis. Locator/ID separation protocol (LISP). http://tools.ietf.org/html/draft-farinacci-lisp-12.

[7] N. Feamster, D. G. Andersen, H. Balakrishnan, and M. F. Kaashoek. Measuring the effects of internet path faults on reactive routing. In *Proceedings of the 2003*

ACM SIGMETRICS international conference on Measurement and modeling of computer systems, pages 126–137. ACM New York, NY, USA, 2003.

[8] Computer networking equipment manufacturing. http://www.firstresearch.com/Industry-Research/Computer-Networking-Equipment-Manufacturing.html, 2009.

[9] G. Huston. Scaling the internet's routing system - ISOC member briefing #3. http://www.isoc.org/briefings/003/, 2001.

[10] Ivip - scalable routing and addressing architecture for the internet. http://www.firstpr.com.au/ip/ivip/.

[11] L. Kleinrock and F. Kamoun. Hierarchical routing for large networks. *Computer Networks*, 1(3):155–174, 1977.

[12] D. Krioukov, K. Fall, and A. Brady. On compact routing for the internet. 2007.

[13] D. Krioukov, K. Fall, and X. Yang. Compact routing on internet-like graphs. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 1, 2004.

[14] B. Leiner, V. Cerf, D. Clark, R. Kahn, L. Kleinrock, D. Lynch, J. Postel, L. Roberts, and S. Wolff. Internet society (ISOC) all about the internet: History of the internet. http://www.isoc.org/internet/history/brief.shtml, 2003.

[15] R. Mahajan. Negotiation-based routing. In *Proc. of Workshop on Internet Routing Evolution and Design (WIRED)*, 2003.

[16] Route-Views BGP reports. http://bgp.potaroo.net/bgprpts/rva-index.html.

[17] RFC 1126 - goals and functional requirements for Inter-Autonomous system routing. http://tools.ietf.org/html/rfc1126.

[18] RFC 2328 - OSPF version 2. http://tools.ietf.org/html/rfc2328.

[19] RFC 2453 - RIP version 2. http://tools.ietf.org/html/rfc2453.

[20] RFC 3344 (rfc3344) - IP mobility support for IPv4. http://www.faqs.org/rfcs/rfc3344.html.

[21] RFC 4271 - a border gateway protocol 4 (BGP-4). http://tools.ietf.org/html/rfc4271.

[22] RFC 4822 - RIPv2 cryptographic authentication. http://tools.ietf.org/html/rfc4822.

[23] RFC 4884 - extended ICMP to support Multi-Part messages. http://tools.ietf.org/html/rfc4884.

[24] RFC 792 - INTERNET CONTROL MESSAGE PROTOCOL. http://tools.ietf.org/html/rfc792.

[25] RFC 950 - internet standard subnetting procedure. http://tools.ietf.org/html/rfc950.

[26] U. Roth. Reactive routing. http://wiki.uni.lu/secan-lab/Reactive+Routing.html, 2009.

[27] Routing research group - IRTF. http://trac.tools.ietf.org/group/irtf/trac/wiki/RoutingResearchGroup.

[28] Six/One: A Solution for Routing and Addressing in IPv6. http://users.piuha.net/chvogt/pub/2007/draft-vogt-rrg-six-one-01.txt.

[29] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the internet hierarchy from multiple vantage points. In *INFOCOM 2002. Twenty-First*

Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, volume 2, 2002.

[30] L. Subramanian, M. Caesar, C. T. Ee, M. Handley, M. Mao, S. Shenker, and I. Stoica. Towards a next generation inter-domain routing protocol. In *Proceedings of Third Workshop on Hot Topics in Networks (HotNets-III)*, 2004.

[31] C. Systems. Exploring autonomous system numbers - the internet protocol journal - volume 9, number 1 - cisco systems. http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_9-1/autonomous_system_numbers.html.

[32] C. Systems. Internetworking technology handbook - open shortest path first (OSPF) - cisco systems. http://www.cisco.com/en/US/docs/internetworking/technology/handbook/OSPF.html.

[33] C. Systems. Internetworking technology handbook - routing information protocol (RIP) - cisco systems. http://www.cisco.com/en/US/docs/internetworking/technology/handbook/RIP.html.

[34] The TCP/IP guide - mobile IP efficiency issues. http://www.tcpipguide.com/free/t_MobileIPEfficiencyIssues.htm.

[35] F. Templin. The IPvLX architecture. http://tools.ietf.org/html/draft-templin-ipvlx-08.

[36] M. Thorup and U. Zwick. Compact routing schemes. In *Proceedings of the thirteenth annual ACM symposium on Parallel algorithms and architectures*, pages 1–10. ACM New York, NY, USA, 2001.

[37] TRRP. http://bill.herrin.us/network/trrp.html.

[38] P. F. Tsuchiya. The landmark hierarchy: a new hierarchy for routing in very large networks. *ACM SIGCOMM Computer Communication Review*, 18(4):35–42, 1988.

[39] D. Waddington and F. Chang. Realizing the transition to IPv6. *Communications Magazine, IEEE*, 40(6):138–147, 2002.

[40] M. B. Weiss and S. J. Shin. Internet interconnection economic model and its analysis: Peering and settlement. *Netnomics*, 6(1):43–57, 2004.

[41] X. Zhang, P. Francis, J. Wang, and K. Yoshida. Scaling IP routing with the core router-integrated overlay. In *IEEE International Conference on Network Protocols (ICNP)*, 2006.