EDGE-AWARE INTER-DOMAIN ROUTING PROTOCOL FOR THE MOBILITYFIRST FUTURE INTERNET ARCHITECTURE

BY SHRAVAN SRIRAM

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Dipankar Raychaudhuri

and approved by

New Brunswick, New Jersey

October, 2015

ABSTRACT OF THE THESIS

Edge-Aware Inter-Domain Routing Protocol for the MobilityFirst Future Internet Architecture

by Shravan Sriram

Thesis Director: Dipankar Raychaudhuri

This thesis presents the design and evaluation of an edge-aware inter-domain routing (EIR) protocol for the MobilityFirst Future Internet Architecture. The EIR protocol provides enhanced inter-domain routing capabilities for wireless/mobile usage scenarios including wireless edge peering, dynamic network formation and mobility, multipath and multi-homing support. The EIR protocol design proposed here is based on abstractions of internal network topology and state of ASes in terms of aNodes and vLinks with this information being flooded through Network State Packets (nSPs) across the Internet. A technique called "telescopic flooding" in which nSP forwarding rates are reduced as a function of hop-distance from the originating node is introduced in order to control the overhead. These nSPs are used to construct the global network topology with some information about the structure and capabilities of each autonomous system(AS), making it possible to realize a variety of routing algorithms corresponding to the use cases mentioned earlier. Further, EIR is designed to work in conjunction with late binding of names to addresses and in-network storage in order to provide robust services in environments with dynamic mobility and disconnection. The proposed EIR protocol was validated through both large-scale simulations and ORBIT testbed emulations using the Click software framework implementation. The evaluations prove the feasibility of the protocol in terms of flooding overhead, convergence time and inter-domain forwarding table sizes. Also, evaluation of mobility service scenarios with migration of clients and networks across domains were performed and the results demonstrate the benefit of exposing the network state and the performance enhancements that can be achieved through the EIR protocol and related routing algorithms.

Acknowledgements

I would like to express my deepest gratitude to my advisor, Prof. Dipankar Raychaudhuri for his support and invaluable guidance through this project. His encouragement and counsel has helped me shape up this work. I am highly grateful to Ivan Seskar and Prof. Roy Yates for their insights during weekly meetings. I would like to thank Prof. Marco Gruteser and Prof. Yanyong Zhang for being on my thesis committee and providing invaluable suggestions on my work. I would also like to thank Shreyasee Mukherjee for all the discussions we had that has helped structure most of the work.

Table of Contents

Abstract ii

Acknowledgements iv

List of Tables ix

List of Figures x

1. Introduction 1

1.1. The Need For Edge Awareness 3

1.1.1. Wireless edge peering 3

1.1.2. Dynamic network formation and mobility 4

1.1.3. Multipath support 4

1.1.4. GNRS assisted global roaming support 5

1.1.5. Optimal mobile data offloading 6

1.2. Related Work 7

1.3. Thesis Organization 8

2. Overview of the MobilityFirst Architecture 9

2.1. Naming and Name Resolution 9

2.2. GSTAR Routing Protocol 12

2.3. Hop-by-Hop Transport Protocol 14

3. Protocol-design 15

3.1. Design Goals 15

3.1.1. Separation of the Name-Address Binding 15

3.1.2. Efficient Utilization of Route Diversity 16

3.1.3. Enable Aggregation 16

3.1.4. Fast Convergence Time 17

3.1.5. Robustness to Incorrect Routing Information 17

3.1.6. Local Policy Enforcement based on Link Type 17

3.1.7. Satisfying QoS Requirements by Routing Efficiently Using the Available Link Metrics 18

3.2. Protocol Components 18

3.2.1. Abstracting the organisational information of an AS 18

3.2.2. Organizing the Aggregated Information in Network State Packets 20

3.2.3. Telescopic flooding of nSPs 22

3.2.4. Providing Late-Binding support through the GNRS to ensure data delivery 23

3.2.5. Local policy setup by the use of Fast-Paths 23

3.2.6. On-Demand Query support 24

3.3. Policy Framework in EIR 24

3.3.1. Business Relation Support Between ASes 24

3.3.2. Dynamic Traffic Engineering 25

3.3.3. Inter-AS Agreements for Tunnel Setup 25

3.3.4. Proliferation of the SID space 26

3.3.5. GNRS assisted Global Roaming Agreements 26

3.4. Routing in The Inter-Domain Framework 27

3.4.1. Unicast support 28

3.4.2. Multicast support 28

3.4.3. Multi-Homing support 29

4. label based fast-path setup 30

4.1. Label-based cut-through switching for the core network 30

4.1.1. Label-based cut-through for transit traffic within a domain . . . 30

4.1.2. Label-based cut-through across multiple domains 33

Single Inter-Domain Tunnel 33
Stitching Intra-Domain Tunnels 34
GNRS assisted tunnel setup 35
5. Router Design 37
5.1. Router Modules 38
5.1.1. Control Plane 38
Link-Probe handler: 38
Link State Advertisement handler: 38
Network State Packet handler: 39
Path-Injection Packet handler: 39
5.1.2. Data Plane 39
Chunking-up and aggregation of data blocks: 39
Address resolution: 40
Route Lookup: 40
Handling On-Demand Queries: 41
5.2. Router Decision Process 41
6. Evaluation 44
6.0.1. Overhead and scalability studies 44
Overhead vs. settling time tradeoff 44
Global routing overhead 45
Worst case update time 46
Routing table size 47
6.0.2. Mobility evaluations 48
Topology generation and probabilistic mobility 48
End host mobility support 50
Network mobility support through nSP propagation 51
7. Future Work and Conclusion 54

7.1. Future Work 54

7.2. Conclusion 54

References 56

List of Tables

3.1. Business relation types in nSPs 21

3.2. Comparative analysis of policy support in EIR and BGP 27

4.1. Different types of encapsulation possible 36

List of Figures

1.1. Dynamic network formation with wireless edge peering 5

1.2. Multipath support at the edge 5

1.3. GNRS assisted inter-domain roaming support 6

2.1. Separation of Identification and Network Location in the MobilityFirst Architecture [28] 10

2.2. Conceptual illustration of mechanisms of GUID to NA binding 12

2.3. Example scenario with routers running GSTAR 13

3.1. Representation of the router level topology of ASes in terms of aNodes and vLinks 16

3.2. Scenario describing the aggregation of late-binding capable routers into aNodes that are then broadcasted using nSPs 20

3.3. Structure of network state packets propagated across domains 21

3.4. Shape of different telescopic functions used for reducing nSP associated control overhead 22

3.5. Scenario illustrating the unicast delivery of chunks to a destination using EIR 28

4.1. Routing of transit traffic based on labels; the Border Routers push rules to all core routers along a path to forward traffic based on a label. The label is added to incoming traffic by the border router 31

4.2. Setting up of an inter-domain tunnel across multiple ASes 33

4.3. Stitching up of multiple intra-domain tunnels across multiple ASes into an inter-domain tunnel.The label and flag is swapped at every ingress border router 34

4.4. GNRS assisted tunnel setup and maintenance in a distributed framework 35

5.1. Elements and their organization in a click router instance 37

5.2. Routing tables at the routers in a sample topology 41

5.3. Routing decisions in a border router 42

5.4. Routing decisions in a core router 42

6.1. Overhead vs. settling time for constant-exponential-constant telescopic function for varying values of A, α and β 45

6.2. Outbound overhead at each router for the 2013 Internet topology due to nSP generation by 100-600 random ASes 46

6.3. Longest routing event update time in the global routing topology for different telescopic functions 47

6.4. Inter-domain table size at each border router for different levels of router to aNode aggregation 48

6.5. End-host mobility topology:(a) aNode distribution in the evaluation topology, (b) Connectivity graph of the aNodes 50

6.6. CDF of path stretch with and without late binding 51

6.8. Network mobility:(a) Distribution of the number of AS transitions for network mobility events, (b) percentage of packets that undergoes storage or rebinding along several bus traces in SFO 52

6.7. Taxi traces in SFO considered for evaluating network mobility 52

Chapter 1

Introduction

This work focuses on the design and evaluation of a new inter-domain routing protocol for the future mobile Internet. The archaic architecture of the Internet was not designed for the dynamism appearing in today's access patterns. Inexpensive mobile phones and wireless connections are driving the current trend in internet access and these are the major sources of dynamism that can take various forms, from the explicit end-host mobility, edge-network mobility, to multi-path and multi-homing introduced by the diversity in number of network interfaces available at the same time. The growth of Information Centric Networks (ICN) in the recent past has also placed an emphasis on the gradual transition from traditional host-to-host approach to content-centric networking. The quick changes in network boundary introduced by dynamic Ad hoc networks elevate delivery concerns. EIR satisfies the basic inter-domain routing protocol requirements of scalability, robustness, and support for flexible routing policies. Moreover, the design philosophy of the protocol is motivated by the following additional requirements:

- React faster to changes at the edge network
- Better support for multi-path, multi-homing and multi-network operation
- Provide room for imposing local policies

The proposed edge-aware inter-domain routing protocol is being developed as a part of the MobilityFirst future Internet Architecture project [23] aimed at a clean-slate redesign of the IP protocol architecture. In earlier works [20, 31] the GSTAR (generalized storage aware routing) protocol was proposed and extensively validated as a solution for intra-domain routing. The objective for this work is to explore and understand Internet scale routing mechanisms in view of emerging mobility services.

The EIR protocol proposed here is based on the following key features: (1) Selectively exposing the internal organization of an AS as an aggregated internal topology in terms of "aNodes" and "vLinks"; (2) telescopic flooding of network state packets in order to limit routing overhead; (3) late binding of object names to network addresses to counter the staleness induced by the telescopic flooding; (4) setting up of label-based cut-through paths within an AS for transit traffic; (5) support for specification of a broad range of routing policies; and (6) supporting On-Demand Querying for making informed forwarding decisions. The aNode and vLink abstractions are inspired by the pathlet routing protocol [13] and are intended to provide a mechanism for autonomous systems (ASes) to optionally express some details about their internal graph and wireless edge properties. The use of network state packets with telescopic flooding is meant to provide fast updates to nearby domains while gradually slowing down the rate at which these updates are propagated to farther ASes. Late binding is a key element of the design which is predicated on the use of names or "Globally Unique IDentifiers" (GUIDs) to identify all network attached objects, along with a logically centralized Global Name Resolution Service (GNRS) for dynamic binding of names to network addresses (see refs [23], [20], [31] for further details on these components of the MobilityFirst architecture).

Recent studies have shown that with endpoints becoming inherently mobile, they are likely to cross multiple ASes, sometimes within a short span of time [12]. The de-facto inter-domain routing protocol, BGP is agnostic to such changes and routes data based on IP prefixes only. Also, with a boom in the wireless field, there are numerous access technologies and hence devices connect to the internet in numerous ways. Providing information about these links can greatly improve delivery. In EIR, border routers perform link state routing based on the global aNode topology and this leads to several interesting use-cases that can now be inherently supported by the Internet architecture, such as multipath support, dynamic flow aggregation, etc. The routing mechanisms of MobilityFirst must also ensure an efficient way to forward data across the core network of a domain. While hop-by-hop segmented data transport and late-binding of endpoint addresses provides critical gains in wireless portions of the network, some overheads are incurred even in stable segments such as the core. If we know that a node will remain connected to the same access point for a period of time, we do not need to make routing decisions at every hop between the source and the destination. This is made possible through the setting up of label based cut-through paths within domains. Also, stitching up of these tunnels can result in fast inter-domain tunnels.

In this architecture, any router in the network can optionally query the GNRS for an updated name-to-address binding, thus enabling packets to be delivered correctly even when the routing protocol cannot keep up with the pace of dynamic changes at the edge. The availability of capabilities such as edge-awareness or late-binding in network routers imply an opportunity for enhanced policy specification capabilities that apply to services such as mobility, multihoming and multicast.

1.1 The Need For Edge Awareness

The proliferation of mobile Internet and the evolution of wireless technology has had a huge impact in the way devices access the internet. Customers are accessing the Internet in newer ways and improving the visibility of changes along the edge, drastically enhances data-delivery. In this section we highlight scenarios that motivate the need for edge awareness in inter-domain routing.

1.1.1 Wireless edge peering

Peering between autonomous domains in the Internet is one of the most important, yet least understood technique used in the Internet. Different ASes employ different types of peering agreements with neighbouring ASes and a recent report shows the presence of 75% more peering links than previously known [7]. There is also a lack of a clear structure to specify, infer, and instantiate peering relations between networks even though works[21] detailing possible choices exists. As a motivating example, consider the case of two small enterprise networks N_1 and N_2 which operate in geographically close locations (e.g. on different floors of a building) and have different Internet service providers ISP_1 and ISP_2. Due to the geographical proximity, some wireless routers in both networks can connect to each other, for example using the bridging-mode available in many enterprise WiFi APs [2]. In this case, N_1 and N_2 can establish a wireless edge peering link by exchanging the corresponding network specific policy requirements with each other. This wireless peering link would keep the two networks connected even if networks ISP_1 and ISP_2 both are undergoing failures, and can help one network to use the internet-connectivity of the other network in case either one of ISP_1 and ISP_2 has a link failure. We believe that wireless peering would be increasingly important for the mobile-dominant future Internet, especially for supporting disaster-recovery (when wired connections to ISPs might fail) and congestion (to maintain partial edge-connectivity when the main links become too congested) and a knowledge rich protocol like EIR can help make the right design decision.

1.1.2 Dynamic network formation and mobility

Another distinction of wireless/mobile networks is that of ad hoc network formation and mobility. For example, there are opportunities for network formation in disconnected islands of cars, as shown in Fig 1.1. In addition such dynamic vehicular networks are inherently mobile and might peer along the edge with different networks as they move. Today's BGP support for airlines and maritime vessels requesting connectivity would partially work [5] but cannot scale to tens of millions of such networks and allow for efficient peering wherever possible.

1.1.3 Multipath support

A typical mobile hand-held device can see multiple available networks (cellular or WiFi) at a time. Although the current business model prevents an user from utilizing multiple cellular service providers simultaneously, consider the increasingly popular “hetnet” mobile service in which a mobile device may be simultaneously connected (multihomed) to a dynamically changing set of cellular and WiFi networks. It is possible to consider a variety of service objectives for this scenario, ranging from “most economical” to “best

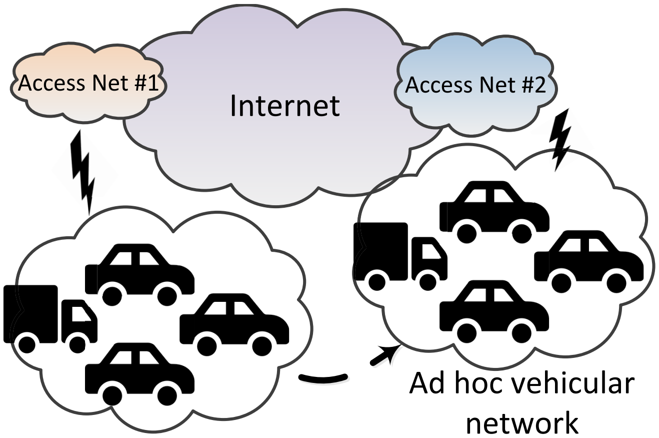


Figure 1.1: Dynamic network formation with wireless edge peering

interface" to "all interfaces". Intermediate solutions to support such connectivity do exist [25, 22, 15], but supporting network-wide multihoming has a very broad architectural implication. Since the cellular and WiFi networks will in general be in different Internet domains, completely different domains need to support multiple paths of connectivity for a single end-to-end flow as shown in Fig. 1.2. Accordingly, routers need to have visibility of the network graph and some awareness of edge network properties in order to make informed forwarding and/or multicast copy decisions.

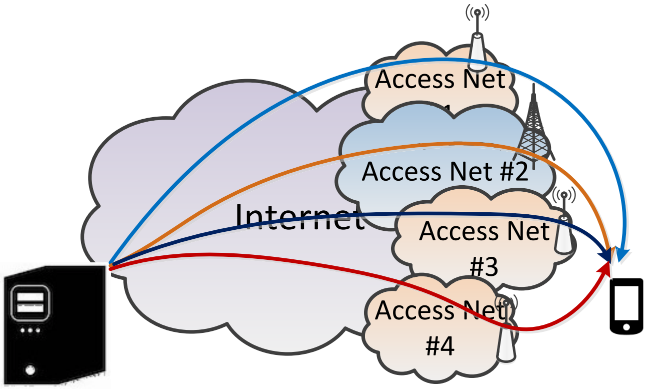


Figure 1.2: Multipath support at the edge

1.1.4 GNRS assisted global roaming support

Modifications can be applied to the current routing architecture while handling inter-domain traffic from clients that attach to ASes with roaming agreements. The initial

DTN approach to handle delivery failures due to end-host mobility is to cache the content. However, we can use a spray and wait algorithm and send copies to all the ASes that are part of the roaming agreement. In this approach, the optimal point where we create copies and hence spray the content can be same as the node where late-binding is performed. In this approach, data can be delivered towards a destination. When the chunk reaches an intermediate late-binding node, the GNRS is queried for the latest binding of the end-host as shown in Fig. 1.3. One can obtain a mobility attribute that is stored in the GNRS and make forwarding decisions appropriately. If the end-host is less mobile, we can forward packets to the destination aNode. However, if the device is highly mobile and is in one of the ASes that form the roaming agreement, the node performing the lookup could tag the chunk with an appropriate SID and spray the content to all the ASes participating in the agreement. At the ingress border routers, the chunk can be dropped or forwarded after a simple check for the end-host.

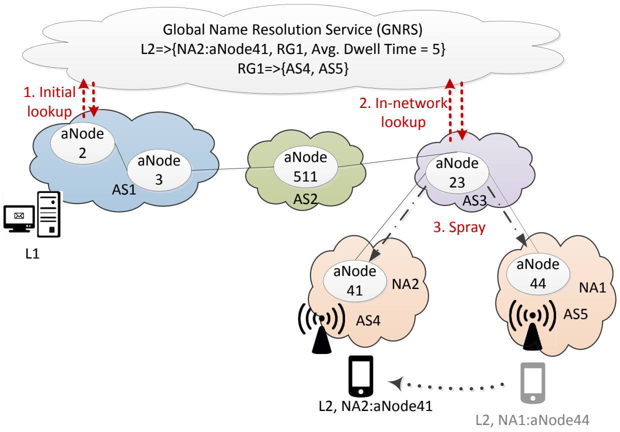


Figure 1.3: GNRS assisted inter-domain roaming support

1.1.5 Optimal mobile data offloading

The demand for data and the proliferation of private and public hotspots are major proponents of the idea of the “Wi-Fi First” service. Companies like Republic Wireless and Scratch Wireless are offering the Wi-Fi first service with Sprint network providing cellular backup in areas where there is no Wi-Fi coverage [4]. Since the Wi-Fi providers are usually in a different domain from cellular networks, better exposure of the organisation of the cellular networks will greatly help MVNOs(Mobile Virtual Network Operators) to offload expensive cellular data services to the most suitable cellular provider with whom the MVNO shares an agreement.

1.2 Related Work

There has been a considerable amount of work done in improving inter-domain routing which can be broadly classified into two categories: (1) extensions to BGP, and (2) clean-slate routing proposals.

Extensions to BGP: Proposals such as path splicing [19] and route-deflections [36] are loose source routing based schemes, where the end-hosts are assumed to be intelligent enough to decide and to explicitly choose a path alternative to the default BGP-computed route. [36] provides a limited choice of paths, whereas [19] provides path diversity, it however does not address the issue of scalability. MIRO [34] moves the decision of path choice from the end-host to the AS which could request alternate paths if it is *not satisfied* with the default BGP route. This handles scalability effectively, but reduces path diversity. The authors in [33, 17] propose similar failover path set-up techniques in order to reduce disconnectivities on link failures.

Clean slate routing: There has also been a growing interest in the Internet community to look for alternatives of BGP that could be incrementally deployed. For example, in the Locator/Identifier Separation Protocol (LISP) [10], tunnels are set up between egress points in an AS, similar to MPLS [26], and then BGP is used to deliver data based on these tunnels. A flat end-point ID is then used at the receiving AS to deliver to the final destination. This multi-AS tunnel setup could easily be emulated in EIR, with the difference being, tunnels and end-hosts are both identified by flat GUIDs. In addition, intra-AS aNode-level topology information provides a finer granularity of path selection in case of EIR. As mentioned before, the aNode-vLink abstraction in EIR is similar to the idea of vNodes in Pathlet [13]. However, our path-selection approach is quite different from Pathlet, which performs loose source routing. Instead EIR provides flexibility to choose end-to-end routes both to end-hosts as well as intermediate ASes through the use of SIDs. HLP [30] uses a hybrid link-state and path-vector approach where provider-customer sub-graphs use link-state routing for path-diversity and peer-peer use path-vector. This effectively improves scalability of the protocol. In contrast providing global view of multiple end-to-end paths provides additional path-diversity and allows EIR to realize policies beyond simple business relationships with the tradeoff being bigger forwarding table sizes. NIRA [35] offers more choice to end-users in choosing the exact set of transit ASes using a hierarchical provider-rooted address scheme. However, similar to HLP, the basic protocol provides limited support for policies other than business relationships.

1.3 Thesis Organization

The rest of the thesis is organized as follows: Chapter 2 provides a brief overview of the MobilityFirst architecture. Chapter 3 details of the goals of this protocol while chapters 3 and 4 describes design of the components employed to achieve these. In Chapter 6 we present detailed simulation results and finally conclude the thesis in Chapter 7 with a discussion on future works.

Chapter 2

Overview of the MobilityFirst Architecture

The MobilityFirst Future Internet Architecture is a clean slate design of the Internet. This architecture is designed on the premise that, mobile devices will far outnumber stationary ones for which the current design of the internet was proposed [9]. With the current boom in wearable technology that is now at the heart of every Internet of Things(IoT) related discussion, this event is fast approaching. The motivating use cases, protocol design challenges, key principles and prototype specifications are discussed in detail in the earlier works [27, 31, 20, 29]. Below we discuss the architecture diagram shown in Fig. 2.1 and some scenarios shown in Fig. 2.2 to highlight the key features of MobilityFirst.

2.1 Naming and Name Resolution

The distinguishing feature of the MobilityFirst architecture is that it proposes an efficient system that works on names and addresses by utilizing both high level network services and the lower-level routing protocols. This layer employs flat Globally Unique IDentifiers (GUIDs). This architecture implements a name based service layer that serves as the narrow waist of the protocol stack. In the current internet, the characteristics of IP addresses are as follows: (i) IP addresses are overloaded to signify both the identity and the location of an end-point, (ii) IP addresses are typically assigned to network entities. But in this architecture, we achieve a clear separation of the identifier and locator functionalities of an IP address through the employment of flat Globally Unique IDentifiers (GUIDs). The GUIDs serve as long lasting identifiers for a broad spectrum of objects ranging from a smartphone, service, vehicle, content, group of these objects and even a context. These GUIDs are assigned by one of the

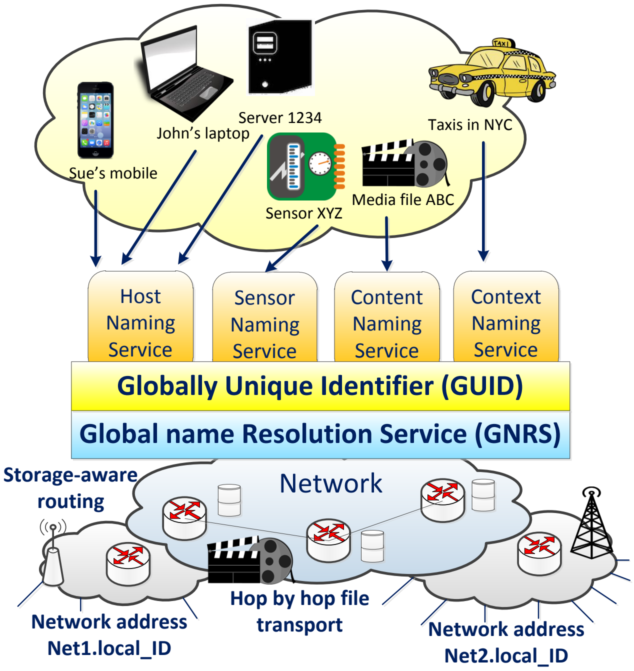


Figure 2.1: Separation of Identification and Network Location in the MobilityFirst Architecture [28]

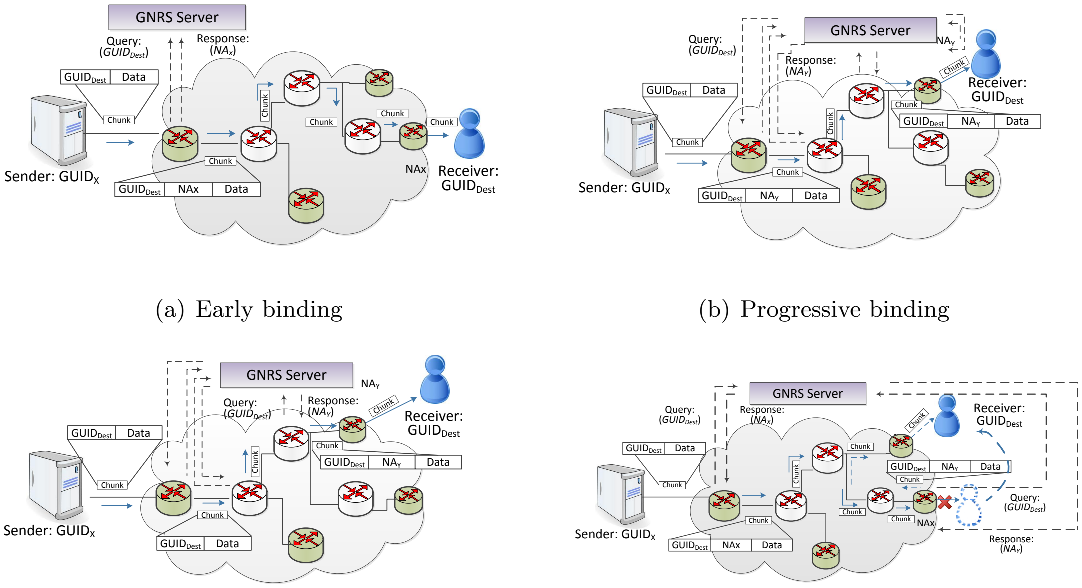
several NCSs(Name Certification Services) that exists and is derived by hashing the public key. This provides these GUIDs a self-certifying property. Hence, no external authority is needed for node authentication [6]. GUIDs are dynamically mapped to a set of network addresses that corresponds to the physical attachment points or locators corresponding to the current attachment points to the internet for the network objects. These mappings are stored in logically centralized Global Name Resolution Service (GNRS). But there are replicas of these mappings stored at different physical locations. More design, implementation and performance evaluation of the GNRS can be found in [31]. This results in a scalable system in which, packets can be routed based on GUIDs of the end host (network object) which can automatically be resolved to a NA or a set of NAs based on where the end device is located.

A component of MobilityFirst that is key to its scalability as explained before is the GNRS. The packets destined to an end device can be routed based on the destination GUID that can be resolved to the point of attachment by using the GNRS. In the MobilityFirst architecture there are different levels of identification. Where destinations can be given unique GUIDs or these destinations can be specified by a low level network address. The point where this resolution occurs characterizes the type of binding and can help during mobility differently. Fig. 2.2 illustrates how early, progressive, late and re-binding could help in mobility and interface management of a destination GUID.

In early binding, the destination GUID is resolved to the corresponding NA(s) at the router attached to the source, the packets are then routed based on the NA. This method produces a constant GNRS lookup delay for all transactions while making the need for subsequent lookups at consequent hops unnecessary. It is justified to use this procedure when there is not much mobility. However, In the case of progressive binding, there is a rebinding operation that occurs at every hop. This while allowing the chunk to always possess up-to-date information, increases the end-to-end delay due to the GNRS lookup at every hop. Late binding however provides better latency management where, the chunk is routed upon its GUID along the path towards its destination where at some intermediate router in the network, it binds the GUID to the network address. After the binding, the chunks are routed based on the NA and no further rebinding is needed till the destination is reached. The final case is rebinding where once the chunk reaches the destination, it rebinds to the most current NA.

In the experiments that were designed to evaluate the system performance while experiencing end-host mobility and edge-network mobility, a combination of early-binding with late-binding at an intermediate junction node(node with high degree) was employed. This approach helps in keeping the end-to-end lookup delay within acceptable limits while also helping in maintaining low path-stretch.

In Fig.2.2(d), the routers in green are the access routers. An end device is considered to move from being connected to an access router(this connection is represented by the network address NA$_X$) to a different access router whose connection is depicted as NA$_Y$. When host connects to the network for the first time, the access router sends a GNRS insert message that contains the network address to which the GUID is mapped to. When a different host tries contacting the newly connected host, there is an early binding event that occurs at the first hop router. Then the packet is routed to the appropriate access router, NA$_X$. While the packet is in transit, the end host moves and gets serviced by a different access router. At this point, the access router sends a



(a) Early binding

(b) Progressive binding

(c) Late binding

(d) Early binding with rebinding

Figure 2.2: Conceptual illustration of mechanisms of GUID to NA binding

GNRS update message with the new attachment point. But the chunk on reaching the access router to which the end-host is believed to be connected to, realises that the end host has moved and performs a rebinding. This is done by resending a GNRS query. For highly mobile devices, this can lead to an increase in path stretch and hence, the alternative of performing late-binding(depicted in Fig.2.2(c)) is preferred in such a case. In the late binding case, the GUID undergoes rebinding at an intermediate router(one with high degree). The choice of performing a rebinding on failure as opposed to late-binding can be made at the first hop router based on a mobility attribute stored in the GNRS.

2.2 GSTAR Routing Protocol

Since in this work, we enhance the protocols employed by the underlying routing layer, we will discuss the work on the intra-domain routing in detail. MobilityFirst has a well designed intra-domain touting protocol. It runs the Generalized Storage Aware Routing (GSTAR) protocol that provides complete visibility of how all the routers are linked within a domain [20]. This system employs both mobile ad-hoc networks

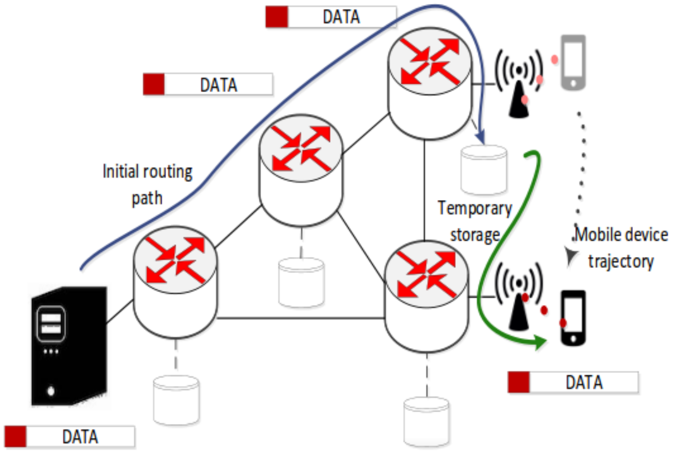


Figure 2.3: Example scenario with routers running GSTAR

(MANET) and delay-tolerant networks (DTN) components. The protocol has a global approach that works on names and addresses that utilizes low level routing protocols and higher level network services. It also has a local approach that involves intelligent buffer management, that reacts to link-quality and storage by utilising the in-network storage. In this protocol, all routers periodically flood fine-grained link quality metrics through Link State Advertisements(LSAs) that contain the short term and long term Expected Transmission Time (SETT and LETT) to their 1-hop neighbours. This information is used by a router in Dijkstra's shortest path algorithm [29] to compute the best next hop to the nodes in its domain. Then forwarding decisions are made based on factors like link availability and the relative values of SETT and LETT. Consider the intra-domain routing scenario described in Fig.2.3, A request for content from a mobile host is being serviced by a content-provider. There is a GNRS resolution that occurs at the first-hop router. Then the packet is routed along the shortest path to the point of attachment of the host. However, at the last-hop router, the packet gets stored because the device moves and attaches to a new node. This stored chunk is re-routed to the new attachment point after a subsequent GNRS resolution is performed. The information in LSAs is further aggregated and flooded through the internet in Network State Packets(nSPs) in a telescopic manner. The information in the nSPs is used in an internet-scale shortest path algorithm for performing efficient inter-domain routing.

2.3 Hop-by-Hop Transport Protocol

The Hop [18] paper discusses the transport protocol used in MobilityFirst in some detail. Hop is a clean slate transport protocol that is intrinsically different from TCP in three aspects. Firstly, The standard unit of data transmission at this layer is called a *chunk*. Before sending a chunk to its next hop, a sender sends a control message *CSYN*, on receipt of which the receiver readies itself by allocating appropriate memory for the incoming chunks. The receiver also sends a *CSYN-ACK*, which contains a bitmap of the packets of the chunk that it has correctly received. The router then employs in-network caching to temporarily cache in-transit chunks and reduce the overhead due to retransmission thus making the system robust to disconnections. The router finally employs a hop-by-hop backpressure through an ack-withholding mechanism. In contrast to end-to-end feedback, this per-flow mechanism is more robust and provides better resource utilization. In this ack-witholding mechanism, the router monitors the difference between <the number of received chunks for a source/destination pair>and <the number of chunks successfully transmitted to its downstream hop>and checks if this value is less than *H*. If the difference is greater, the router stops sending *CSYN-ACKs* to its upstream neighbour for newer chunks belonging to the same flow.

Chapter 3

Protocol-design

Edge-Aware Inter-Domain Routing is a comprehensive protocol that provides us the loquacity of a link state routing protocol while maintaining the overhead within bounds. The current Inter-Domain protocol sacrifices the diversity of paths available in-order to be concise in their route announcements thus reducing overhead. Also, the abundance of information available puts EIR in a good position to be able to make informed decisions under various mobility constraints thus improving the overall routing efficiency. The protocol achieves scalability through the aggregation of the underlying router topology of a domain into aNodes interconnected using vLinks and the flooding of this topology in a telescopic manner using nSPs. Fig. 3.1 shows an aggregated representation of the underlying router-level topology of an AS in terms of aNodes and vLinks. Most of the inter-domain routing algorithms and decision making are done at the border routers that are part of aNodes(e.g., BR1 in aNode21). Key features of EIR include the robustness to missing/incorrect routing information through the efficient use of the late-binding feature offered by MobilityFirst and the fast realisation of any routing churn through the telescopic flooding of nSPs while maintaining the control overhead in check.

3.1 Design Goals

3.1.1 Separation of the Name-Address Binding

This is the key requirement of the MobilityFirst Future Internet Architectures(FIA). In the current internet, the IP addresses are overloaded with the roles of both name and locator. In MobilityFirst, these roles are assigned to Globally Unique Identifiers (As

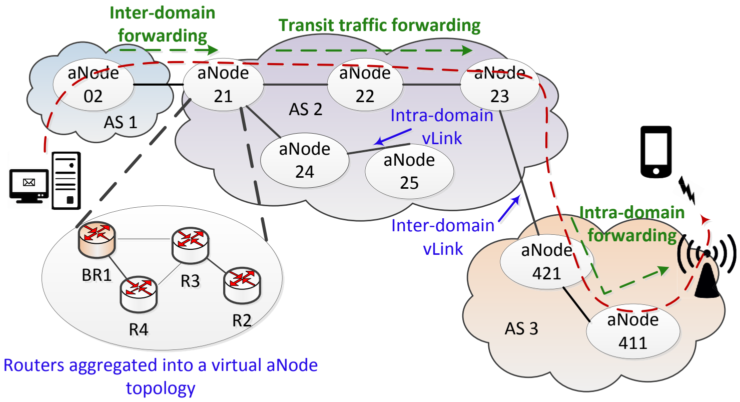


Figure 3.1: Representation of the router level topology of ASes in terms of aNodes and vLinks

defined in [23]) and Network Addresses respectively. The GNRS is consulted in order to get the most current binding and route packets in scenarios that involve mobility.

3.1.2 Efficient Utilization of Route Diversity

The exposure of the internal topology of an AS makes multiple alternative routes to a destination visible. Also tagging the aNodes with information that describes the characteristics of the technology used provides an opportunity to use a protocol like Multi-Homing efficiently. Also, the presence of an aggregated topology makes the selection of optimal bifurcation points for implementing Multi-Cast services easy.

3.1.3 Enable Aggregation

Route aggregation or route summarization is a key factor in achieving scalability. It is through this feature that fewer entries exist in the inter-domain forwarding tables of the current Internet. EIR achieves this form of scalability through the aggregation performed at the border routers where blocks of internal routers are abstracted into aNodes interconnected through vLinks.

3.1.4 Fast Convergence Time

An important factor used to assess the efficiency of any inter-domain routing protocol is convergence time. BGP is highly criticised for its slow global re-convergence. In BGP, this time is reducible by tweaking the timers to smaller values. In EIR, reducing the telescopic hold time can greatly improve the convergence time. However, since EIR is a link state protocol, one must be careful about what telescopic function and associated parameters are used in order to keep the overhead in check.

3.1.5 Robustness to Incorrect Routing Information

There is some research that states that in today's Internet the global routing tables never truly converge; instead, it gently wafts back and forth between states. This might lead to scenarios where, incorrect forwarding decisions are made. In EIR, any churn related information is eventually propagated by the link-state protocol. However, in scenarios when there is stale routing information due to the distance between the source and destination ASes, late-binding can be used to provide a more up-to-date route to the destination.

3.1.6 Local Policy Enforcement based on Link Type

The Link-Type information present in the nSP can be used to setup local routing policies between ASes and hence effect routing decisions based on the business relationships(Customer-Provider, Peer-Peer) between Domains. Another distinct characteristic of the link is the nature of the connection (Wired/Wireless). Routing intelligently based on this link type improves the data delivery rate without any need for storage or re-transmission. There can be certain links in a network with a poor capacity (Wireless instead of Ethernet). The prior knowledge of which could have helped mitigate the unnecessary packet loss. An example scenario is when we try to route packets to Maritime vessels[5]. These have a last hop as a wireless link that is inherently slower. The sending of packets at higher data rates during such instances unnecessarily increases the loss rate.

3.1.7 Satisfying QoS Requirements by Routing Efficiently Using the Available Link Metrics

The sender might need various performance guarantees depending on the application. This information can be made known by setting a specific Service IDentifier(SID) in the packets. The availability of global aggregated link metrics enable internal routers to be forced to route through paths that satisfy some of these requirements. For example, routers can compute global routes based on high bandwidth, low latency, high availability and so on. The nSP could also hold information like the presence of a storage capable router in an aNode that can be made use of by Content Delivery Networks to serve content to end-users with high availability and high performance.

3.2 Protocol Components

In EIR, the border routers hold large inter-domain forwarding tables. At these nodes, algorithms for aggregation of routers, propagation of the aggregated topology and initiation of fast-path setup are run. Also, the inter-domain routing decisions are made at the border routers. Based on the policy agreements between ASes, traffic seen at a node gets treated differently. Fig. 3.1 shows a scenario where traffic gets forwarded from a source attached to AS1 to a destination in AS3. The border routers decide weather the packet undergoes, inter-domain or intra-domain forwarding. There are also instances where transit traffic undergoes fast switching across the domain. The final model of EIR is able to satisfy most of the aforementioned goals and offer reliable data delivery through seamless integration of a few key components.

3.2.1 Abstracting the organisational information of an AS

The internal routers of an AS are abstracted into one or more aggregated nodes(aNodes) that are interconnected through Virtual Links(vLinks). The degree and criteria for aggregation of the routers into aNodes is under the discretion of each AS. An aNode can aggregate all the Storage-capable routers or Open Flow enabled routers internal to a domain. There are dedicated border routers within an AS that will perform the aggregation of the routers into aNodes. Since this happens on the border routers in a distributed manner the question of inconsistency arises. However, since the Link State Advertisements(LSAs) exchanged during the intra-domain routing[20] containing link quality for each of the node's 1-hop neighbours is what is used to perform the aggregation into aNodes, consistency issues get resolved. Using these LSAs the controller constructs a topology graph that includes all routers, links and link qualities within its domain. Also, if there is a centralized entity like an SDN controller that has a global view of the AS, it can manage the allocation of routers to aNodes and hence reduce the computational load on the border routers.

Once the border router is aware of the intra-domain topology, it can build up the aggregated network topology it wants to advertise to other domains, by clustering routers into aNodes and links into vLinks. While forming these clusters, the algorithm can place a few private routers in a specific blacklisted aNode in the AS graph. Doing so prevents information about these aNodes from being made public to the rest of the internet. Also, different policies can be applied for clustering such as:

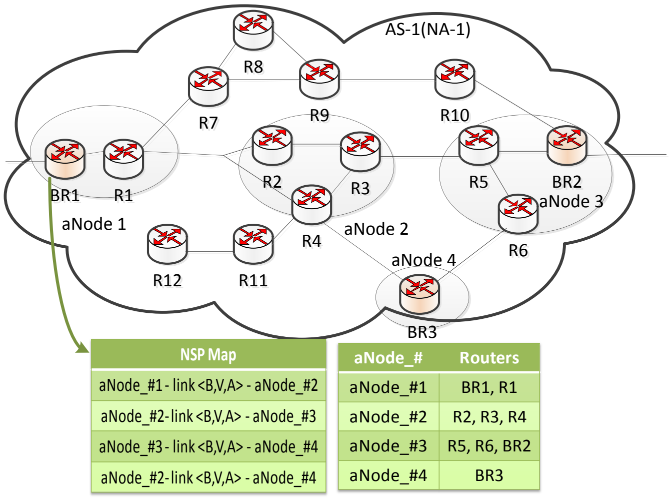
1. Group similar switches into aNodes (storage-capable switches, late-binding capable switches, etc.)

2. Aggregate switches based on geographic location and proximity

3. Aggregate links based on link quality information such as long-term estimated time of transmission (LETT) or bandwidth

For the second approach, since the entire network can be seen as a graph, different graph clustering algorithms can be used to form the aNodes and the vLinks. Markov Cluster Algorithm (MCL) is one such fast and scalable algorithm that can be run at the controller for creating the aNode-vLink topology [32]. The algorithm considers matrix entries as similarities and performs simple matrix operations of inflation and expansion which directly affects the granularity of aggregation and the affinity between clusters respectively. The final output of MCL consists of sets of nodes that are topologically aggregated.

Fig.3.2 illustrates an example scenario showing the aggregation of late-binding capable routers into aNodes. The Border routers maintain a log of the quality of links between the aNodes.



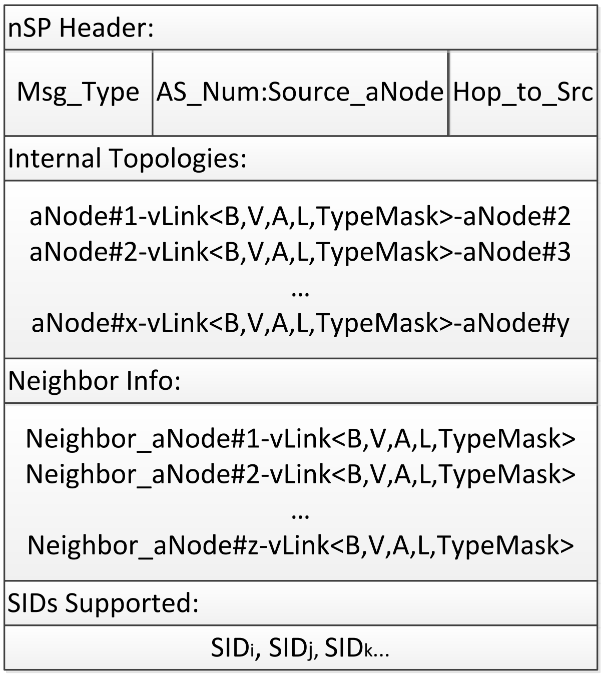
NSP Map
aNode_#1 - link <B,V,A> - aNode_#2
aNode_#2- link <B,V,A> - aNode_#3
aNode_#3 - link <B,V,A> - aNode_#4
aNode_#2- link <B,V,A> - aNode_#4

aNode_#	Routers
aNode_#1	BR1, R1
aNode_#2	R2, R3, R4
aNode_#3	R5, R6, BR2
aNode_#4	BR3

Figure 3.2: Scenario describing the aggregation of late-binding capable routers into aNodes that are then broadcasted using nSPs

3.2.2 Organizing the Aggregated Information in Network State Packets

Once the border routers(BR) have created a virtual topology for the domain using aNodes and vLinks, it must propagate it to other domains using network state packets (nSPs). These nSPs contain the bandwidth, availability, variability attributes between different aNodes in the topology. Also, different ASes employ different types of peering agreements with neighbouring ASes. These nSPs can contain an indicator that specifies the business relation that exists between border aNodes. This type field can also include weather the link is wireless or optical. Also, this type field can also convey information about aNode that forms the first vertex on a vLink(i.e, weather storage capable, late-binding capable). A vLink can satisfy multiple types at an instance, hence using a netmask is a good approach to represent these various types. Currently we use a 16-bit netmask with 8 possible type values. The various link types that the bits of the netmask represent are enumerated in 3.1. The BRs use the abstracted topology and the link information(B,A,V,Type) to create network state packets(nSPs) as shown in 3.3. Next, this packet is added to the $out - control - Q$ of the BR and sent to all its outgoing interfaces. In Fig.3.2, the nSP generated at border routers BR1, BR2 and BR3 contains information about how the aNodes are connected through vLinks. This information is flooded throughout the Internet in a telescopic manner.



nSP Header:

Msg_Type | AS_Num:Source_aNode | Hop_to_Src

Internal Topologies:

aNode#1-vLink<B,V,A,L,TypeMask>-aNode#2
aNode#2-vLink<B,V,A,L,TypeMask>-aNode#3
...
aNode#x-vLink<B,V,A,L,TypeMask>-aNode#y

Neighbor Info:

Neighbor_aNode#1-vLink<B,V,A,L,TypeMask>
Neighbor_aNode#2-vLink<B,V,A,L,TypeMask>
...
Neighbor_aNode#z-vLink<B,V,A,L,TypeMask>

SIDs Supported:

SID_i, SID_j, SID_k...

Figure 3.3: Structure of network state packets propagated across domains

Table 3.1: Business relation types in nSPs

Bitmask Position	link description
1	peer-to-peer edge
2	provider-to-customer edge
3	customer-to-provider edge
4	sibling-sibling edge
5	Wireless link
6	Optical link
7	link starts from a Storage capable anode
8	link starts from a late-binding anode

3.2.3 Telescopic flooding of nSPs

Once an nSP is received by the border routers of a domain, they need to be communicated to the rest of the internet. A component that is key for this process to scale efficiently is the use of a telescopic function at Border Routers. This dampens the rate at which the nSPs get flooded across the internet. When an nSP leaves an AS, it is buffered at the border router for a period of time that is determined by the telescopic function. This hold time is a function of the hop count. As an effect of this, ASes that are closer to each other exchange updates at a much quicker rate compared to the ones that are farther apart. These ASes that are farther away have a much stale information regarding the internal organisation of an AS that is far away. But this setback is handled by the use of the late-binding support provided by GNRS. The following equations define how the telescopic hold time varies as a function of the hop count.

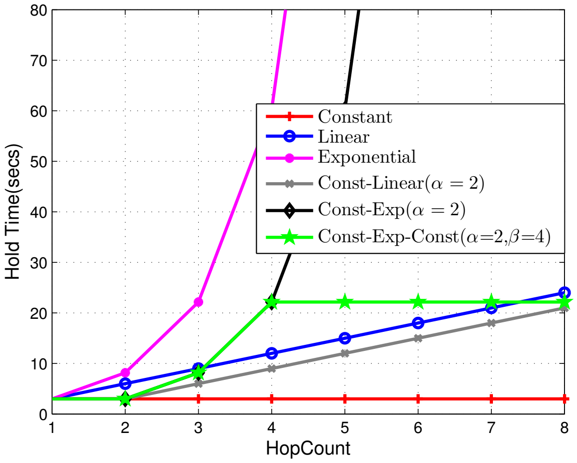


Figure 3.4: Shape of different telescopic functions used for reducing nSP associated control overhead

$$\text{Constant:} \qquad y_1 = A$$

$$\text{Linear:} \qquad y_2 = Ax$$

$$\text{Exponential:} \qquad y_3 = Aexp^{(x-1)}$$

$$\text{Constant-Linear:} \qquad y_4 = \begin{cases} A, \text{if } x < \alpha \\ A(x - \alpha + 1), \text{if } x \geq \alpha \end{cases}$$

$$\text{Constant-Exp:} \qquad y_5 = \begin{cases} A, \text{if } x < \alpha \\ Aexp^{(x-\alpha)}, \text{if } x \geq \alpha \end{cases}$$

$$\text{Constant-Exp-Constant:} \qquad y_6 = \begin{cases} A, \text{if } x < \alpha \\ Aexp^{(x-\alpha)}, \text{if } \alpha \leq x < \beta \\ Aexp^{(\beta-\alpha)}, \text{if } x \geq \beta \end{cases}$$

3.2.4 Providing Late-Binding support through the GNRS to ensure data delivery

This key feature has been made necessary with the inclusion of telescopic flooding. As an effect of using telescopic flooding, the ASes take a longer time to receive information about any mobility event that occurs in an AS that is much farther away. These ASes can route traffic towards the previously known attachment point of the destination and en-route perform an in-network name-address rebinding by performing a GNRS lookup to obtain the latest attachment point of the destination.

3.2.5 Local policy setup by the use of Fast-Paths

The EIR protocol has the provision for a border router initiated fast-path setup. In this procedure, the border routers compute paths based on bandwidth, link latency or any other local policy and inject these path information into internal routers along these paths by the use of route-injection messages. Each of these paths are assigned a unique label. At the internal routers, the computational complexity while routing packets is reduced as simple label based switching occurs.

3.2.6 On-Demand Query support

This is a very useful add-on to the EIR routing protocol. This is primarily used while routing packets across domains. Because of the delayed flooding, the information contained in an nSP could be stale. In this case, this querying feature provides us with more up-to-date information. While routing packets across domains, this feature can be used to obtain more recent link characteristics at the last hop and also the path quality across all the links along the path. Special queries can list all the storage cable aNodes or Late binding capable aNodes along the path. In this mechanism, the query from a border router for a particular aNode is responded with an acknowledgement from the closest router in the aNode for which the request was posted with the relevant information.

3.3 Policy Framework in EIR

One of the key features of BGP is its ability to account for a range of policies. The initial version of BGP performed simple path-vector routing. However, incremental modifications were added with a number of mechanisms for the support of policies adding to the overall complexity of the protocol. In EIR, there are a set of policy agreements set up between ASes that impacts how the routing takes place.

3.3.1 Business Relation Support Between ASes

The nSPs convey a large amount of information about the internal organisation of the ASes. But the vLinks that represent inter-domain links between ASes are tagged with useful business-relationships such as “customer-to-provider” or “provider-to-customer” or “peer-to-peer”. In the current architecture of the Internet, the business relationships between domains is not disclosed to all. This is used to enforce local policies. However, work done in [11] indicates that these relationships can be interpreted by intelligently parsing the BGP advertisements. In EIR we do not hide this information. The routing algorithms that run at the border routers, filter out some of the possible paths by making sure that the AS paths traversed remain valley-free. The property as described in [11] prohibits the traversal of a customer-to-provider or a peer-to-peer edge after traversing a provider-to-customer or peer-to-peer link while trying to reach a destination. Packets then use the best paths that satisfy the business agreements between ASes while routing towards a destination.

3.3.2 Dynamic Traffic Engineering

EIR is a link-state based routing protocol that runs a shortest-path algorithm across the aNode topology graph that it learns through link state flooding that occurs across the Internet. This can lead to the congestion of certain links. In EIR, ASes can constantly monitor internal congestion by observing the intra-domain LSAs. These can be used to determine congested paths. Anodes along such paths can then be omitted in the subsequent nSPs. The inter-domain routes on re-computation will avoid the congested links. Another use for the nSPs is in load-balancing. There might be multiple ingress links to an AS. The border router can manipulate the vLinks across these inter-domain links and bias the preference for a particular vLink over the others.

3.3.3 Inter-AS Agreements for Tunnel Setup

There are situations where, a company pays an ISP to support applications (having requirements that exceed the QoS offered by the best-effort Internet) between its branches across the Internet. However, this AS might not be the only one between both the branches. Then, the AS might have to purchase transit from multiple Tire-1 or wholesale national providers. In such a scenario a GNRS assisted inter-domain tunnel setup algorithm can be run across domains. This setup procedure can result in fast switching tunnels across ASes. However, there can be a scenario where an intermediate AS might not want to participate in the tunnel. In such cases, the provider carries traffic internally and hands it over to an uninterested external peer(through not necessarily the shortest path) as close as possible to the destination network. This practice is often referred to as cold-potato routing.

3.3.4 Proliferation of the SID space

The SID space was used while performing intra-domain routing to intimate the routing plane to make use of the available diversity in paths(Multipath) or different access technologies(multi-homed) or using the nearest content source(any-cast). The standard SIDs can also be used from an inter-domain perspective to determine how traffic is forwarded. Every value that the SID assumes can denote a different forwarding scheme varying from low latency to high availability or high bandwidth. The control plane plays a vital role in keeping track of all the possible SIDs and their impact on the forwarding logic. The lack of knowledge about a specific SID would force the use of a standard Estimated Time of Transmission(ETT) based shortest path. Using the SID space gives the end-user significant control on data forwarding methodologies unlike BGP. In BGP, which ISP to use may not be one's decision. If a customer trusts another company to host its Website, its ISP connections will usually come with it. Another use case is the corollary of the roaming scenario where, an intelligent aNode along the edge, tags the traffic from specific users based on metrics like host mobility, user-type(PC vs mobile phone).

3.3.5 GNRS assisted Global Roaming Agreements

This key feature discussed as one of the use-cases of EIR, states that ASes can easily enter global roaming agreements with each other and form an AS roaming group. In particular, a domain (hosting domain) that is willing to provide network connectivity for the clients of another domain (remote domain) can tag its incoming vLinks with "Roaming" to indicate its willingness to enter into a roaming agreement. Then, once the ASes enter such an agreement after agreeing upon the terms and conditions, the remote domain can register itself as one of the roaming partners of the hosting AS in the GNRS. When a remote domains client migrates to and associates with the hosting domain, the domain first verifies that the client belongs to the hosting domain using the previous binding stored in the GNRS. Once the verification is completed, the hosting

Type	Policy	BGP	EIR	Note for EIR
Business relationship	Local Pref	✓	✓	Bias vLink metrics
	Community attribute	✓	✓	Tag vLinks with relationship
Traffic engineering	Hot potato routing	✓	✓	Use AS hop count based forwarding
	Load balancing (using MED)	✓	✓	Cut-through paths and dynamic aNode formation
Scalability	Prefix aggregation	✓	X	nSP aggregation not supported
	Default routes	✓	✓	Use of ETT based forwarding table
	Route flap damping	✓	✓	Supported by modifying telescopic flooding
Others	User-initiated	X	✓	Use of SIDs
	Network-initiated	X	✓	Use of SIDs
	Global roaming	X	✓	Use of GNRS
	Blacklisting	X	✓	Stitching of inter-domain tunnels

Table 3.2: Comparative analysis of policy support in EIR and BGP

domain will allow up stream traffic from the client and update the GNRS with a GUID-to-address mapping for that client so that other network entities can reach the remote domains client.

Based on the above discussion, table 3.2 provides a summary of comparison between policies currently supported in BGP and the ones supported in EIR (refer to [8] for detailed description of BGP supported policies).

3.4 Routing in The Inter-Domain Framework

Like BGP, the routing in EIR works in harmony with the various policy agreements that exist between domains. This section discusses how the routing is handled in some of the scenarios supported by EIR. Since EIR is based on the link-state flooding of state in a telescopic manner, there can be some staleness in the routing information. In such scenarios, the late-binding is used to route correctly. Variants of Fig.3.5 will be used to describe some of the scenarios that EIR can support. In most of the scenarios, the way the transit traffic is handled remains the same. Once the chunks enters a transit-AS, the ingress border router tags this chunk as in-transit and the chunk thereafter follows a pre-determined fast-path.

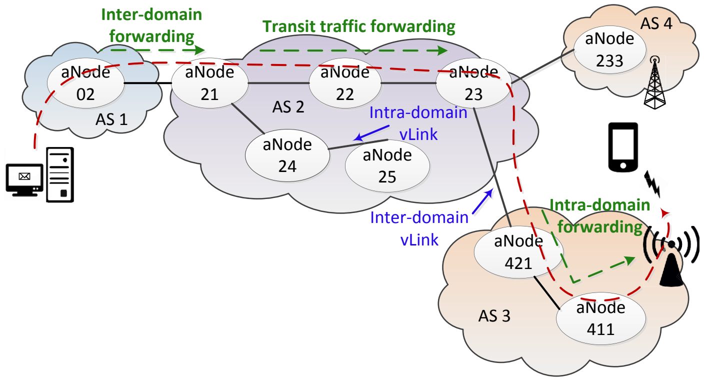


Figure 3.5: Scenario illustrating the unicast delivery of chunks to a destination using EIR

3.4.1 Unicast support

A scenario describing the delivery of unicast traffic to a destination by routing across domains is illustrated in Fig.3.5. Consider the scenario where traffic is generated from a client connected to an access-router in aNode2 that is destined to an end-host connected to an access-router in aNode411. In this scenario, the chunks are first delivered to the closest border router(CBR) in AS1. This CBR determines the best aNode-path in-order to reach the destination an AS3. The source or the edge aNode can tag the chunks with a specific SID that forces the use of the low-latency path or the high-bandwidth path. At the intermediate ASes(AS2 in Fig.3.5), preset fast-paths are used for quick forwarding of traffic. Once the chunk enters the AS where the destination is present, standard GSTAR forwarding occurs.

3.4.2 Multicast support

Consider the case where the data is being forwarded to a set of destinations that belong to a multicast group. These destinations might be within the same AS(destinations connected to aNode22 and aNode24 in Fig.3.5) or the attachment points of the destinations might belong to different ASes(destinations connected to aNode233 and aNode421 in Fig.3.5). In both these cases, the CBR decides, the best aNode(aNode along the longest common path) upto which the chunk can be routed without bifurcation. In the first case, the intra-domain routing takes over after the chunk enters AS2. However, in the scenario where the destinations are in different ASes, chunks are forwarded till aNode23, from where the copies of the chunk get routed to the different destinations independently.

3.4.3 Multi-Homing support

This is an area of active research over the past few years. A device might have access to the internet using multiple technology. A sample illustration is in Fig.3.5 where, an end-host has reachability to the internet through two parallel technologies. Such devices have an entry in the GNRS that enumerates the different attachment points(aNode411 and aNode233 in this case). In such a scenario, the CBR tags the traffic as MH-capable by setting the appropriate SID and sends the chunk towards an aNode that has good reachability to the destination through both technologies(aNode23 in this case). At this aNode, an On-Demand query is issued to both the aNodes containing the access-routers. Based on the response, the traffic can be sent to the better of the two technologies or be striped across both the technologies.

Chapter 4

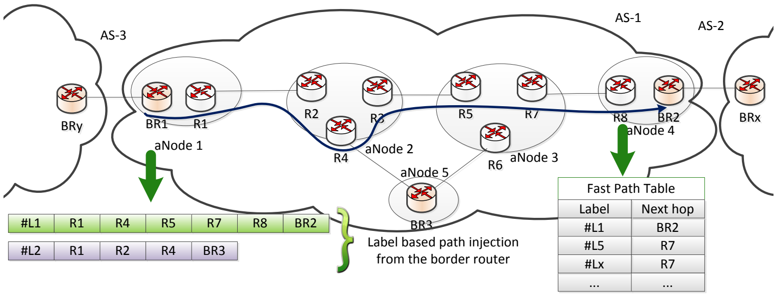
label based fast-path setup

4.1 Label-based cut-through switching for the core network

There are different flavours of layer-2 packet forwarding protocols. The ones that are most widely known are 'Store and Forward' and 'Cut-Through Switching'. In the latter approach, a switch does not wait for the entire packet to be received, and rather just forwards frames towards the destination soon after the destination address is read. This analogy can be applied to the routing layer. Upon receiving a chunk with a particular label at a router, it can immediately be forwarded to the appropriate next hop router. By this method ISPs can guarantee a promised QoS across their domain by switching traffic through certain predetermined tunnels rather than waiting for entire packets to be received before forwarding thus decreasing the overall delivery rate.

4.1.1 Label-based cut-through for transit traffic within a domain

ASes carrying inter-domain transit traffic can set up cut-through paths across its domain (from ingress to egress border routers) by the use of locally-unique labels. In a distributed routing scenario, each border router independently determines the transit paths based on local and transit policies and the advertised link characteristics of its network. Accordingly, it assigns a label (which is locally unique within the AS) and pushes the path and label information into a fast-path forwarding table at each internal router, using a route-injection packet. Once the path has been set up, the internal routers simply forward transit traffic based on the label an ingress border router attaches to every packet that is in transit. Incoming data at an ingress border router is marked as transit and encapsulated with the corresponding label of the path it is intended to transit through. For example in Fig. 4.1, border router BR_1 could choose



#L1	R1	R4	R5	R7	R8	BR2

#L2	R1	R2	R4	BR3

Fast Path Table

Label	Next hop
#L1	BR2
#L5	R7
#Lx	R7
...	...

Figure 4.1: Routing of transit traffic based on labels; the Border Routers push rules to all core routers along a path to forward traffic based on a label. The label is added to incoming traffic by the border router

the path $< R_1, R_4, R_6, R_7, R_8, BR_2 >$ to reach BR_2, based on transit policies. In this distributed scenario, BR_1 forwards a path-injection packet of the form shown in Fig. 4.1 along the path containing the generated label and the path information in terms of the routers along the path. The internal routers along the path create a fast-path entry in their fast-path table (similar to the one shown at R_8). The advantage of this scheme is two fold: (a) Internal routers do not need to perform any inter-domain route processing; and (b) Policies could easily be expressed by border routers by creating different labels and paths and injecting this information in the fast-path table.

The presence of a centralized controller greatly reduces the burden on the border routers during the process of path setup. As the controller has a snapshot of the internal organization of the AS and its also aware of the transit policies, it can take decisions on behalf of all the border routers in the AS. Accordingly, it can inject flow rules into the fast-path table that dictates the action the switch has to take when it encounters a packet with a particular label arriving at a particular port.

In a larger domain, the problem is to find a label for each pair of non-neighbor border routers such that there are no conflicts. The problem can be defined as follows:

Input: Set of tuples $< BR_i$, BR_j, metric $>$, BR_i, $BR_j \in$ *Border routers*;

Output: Set of tuples <*path, label*>;

The problem definition can be read as follows. Given a set of source and destination border routers and a metric, output the best path and the label assigned to it. The constraint of this algorithm is that the label-forwarding should not be ambiguous: an incoming label should not have two possible next-hops. This problem can be solved by distributed routers that exchange messages. However, it becomes challenging to ensure that the constraint stated above is respected. In contrast, this problem is easily solvable from a centralized control plane (see Algorithm 1).

```
Data: HashMap labelAssignment
//Assign a label to all paths between border routers
for all $BR_i \in$ border routers do
    for all $BR_j \in$ border routers do
        label = nextLabelAvailable();
        routers = computePath($BR_i$, $BR_j$, metric);
        key = < $BR_i$, $BR_j$, metric>;
        value = <label, routers>;
        labelAssignment.add(key, value);
    end
end
```

Algorithm 1: Label assignment for each pair of border routers.

However, In the distributed algorithm that we consider, the border routers can make use of the GNRS in order to avoid possible conflicts. On generating a label, the source border router can look for the label in the GNRS and if no conflict arises, it can insert the newly generated label into the GNRS. Note that there is no need to minimize the number of labels used, as the label-based paths are only used to connect border routers within the same domain. If we assume that tunneling is done using VLAN tagging or a similar approach, the number of labels is enough for any domain. Thus, Algorithm 1 simply uses the next label available.

Also, note that the algorithm takes a metric into consideration when computing the path between two border routers. This metric can be the shortest path, the highest bandwidth, the lowest jitter and so on. This is relevant because it allows every domain to provide different routes based on the information advertised through the network state packets. Thus, if a previous domain requests traffic to be routed through a high bandwidth path, the current Border Router(or controller) can achieve that using the appropriate label.

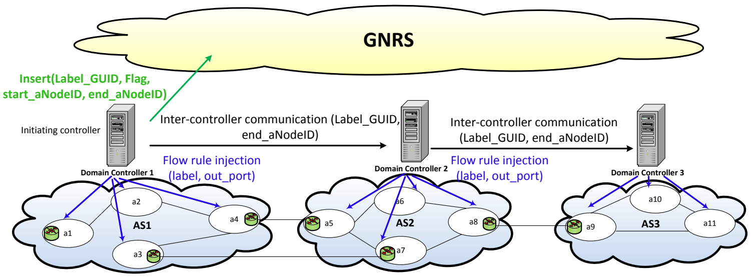

GNRS
Insert(Label_GUID, Flag, start_aNodeID, end_aNodeID)
Initiating controller
Inter-controller communication (Label_GUID, end_aNodeID)
Inter-controller communication (Label_GUID, end_aNodeID)
Flow rule injection (label, out_port)
Flow rule injection (label, out_port)
Domain Controller 1
Domain Controller 2
Domain Controller 3
a1
a2
a3
a4
AS1
a5
a6
a7
a8
AS2
a9
a10
a11
AS3

Figure 4.2: Setting up of an inter-domain tunnel across multiple ASes

4.1.2 Label-based cut-through across multiple domains

In order to set up cut-through paths across multiple domains, ASes are required to exchange label based information which then needs to be populated across the fast path tables of the internal routers along the tunnel-path within the participating ASes. The presence of a centralized controller enables two alternative approaches for such inter-domain tunnel setup. There is also an approach that can be followed in a distributed scenario that makes use of the GNRS.

Single Inter-Domain Tunnel

In this method, the Initiating Controller (InC) works in conjunction with the GNRS in order to setup and maintain an inter-domain tunnel that cuts-through ASes. The label that defines the tunnel is a GUID that is globally unique and the up-to-date mapping of its start and end point locators is stored in the GNRS. Consider the setup described in Fig. 4.2: If the controller of AS1 decides that it is worthwhile to setup a tunnel that links AS1-AS2-AS3, it first receives a unique GUID (L1) for the tunnel from the NCS. This is then mapped to the destination anodeID in AS3 and stored in the GNRS. The controllers of all the participating ASes agree upon setting up an inter-domain tunnel. In a fully SDN domain, the participating controllers inject flow rules into switches within their respective domains for setting up the tunnel denoted by L1. However, in a scenario where the controllers have an influence over just the border routers, the tunnel setup needs to follow a procedure similar to the intra-domain path setup mechanism discussed in the previous section. All the controllers also inject a

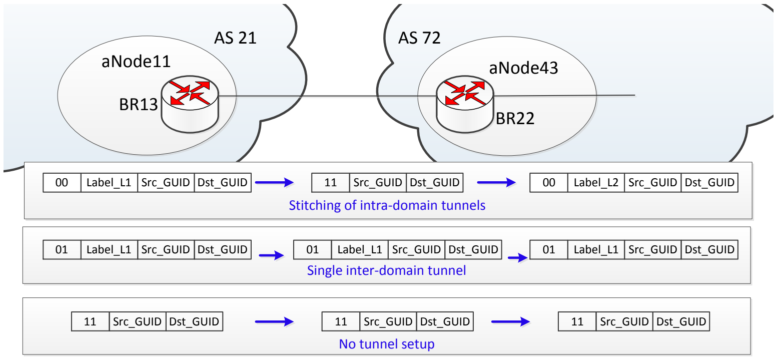

AS 21
aNode11
BR13
AS 72
aNode43
BR22
00 | Label_L1 | Src_GUID | Dst_GUID
11 | Src_GUID | Dst_GUID
00 | Label_L2 | Src_GUID | Dst_GUID
Stitching of intra-domain tunnels
01 | Label_L1 | Src_GUID | Dst_GUID
01 | Label_L1 | Src_GUID | Dst_GUID
01 | Label_L1 | Src_GUID | Dst_GUID
Single inter-domain tunnel
11 | Src_GUID | Dst_GUID
11 | Src_GUID | Dst_GUID
11 | Src_GUID | Dst_GUID
No tunnel setup

Figure 4.3: Stitching up of multiple intra-domain tunnels across multiple ASes into an inter-domain tunnel.The label and flag is swapped at every ingress border router

rule that dictates the egress border router to forward the packet with label L1 to the appropriate ingress border router of the neighboring AS. The InC can decommission a tunnel by deleting the mapping stored for L1 from the GNRS. In this scenario, the controllers of the participating ASes will periodically do a GNRS lookup for label (L1), the absence of which will trigger the removal of the corresponding entry from the flow tables.

Stitching Intra-Domain Tunnels

An important observation to note is that the individual ASes might already have cut-through paths setup within their own domain for transit traffic and the multi-domain cut-through may follow the path described by an already existing intra-domain tunnel, within each AS. This enables the possibility where, the inter-domain tunnels can be composed of a series of intra-domain tunnels. In this case, there is no single label (GUID) for naming the tunnel and labels need to be swapped at every ingress border router. Accordingly, there should also be a procedure by which a border router can differentiate an intra-domain tunnel from that of an inter-domain tunnel. This can be enforced by tagging the packet with a flag based on the path the packet is meant to travel as described in detail in table 4.1. Fig. 4.3 highlights the scenario where AS21 and AS72 are both transit ASes using the three different flags and appropriate labels for forwarding data.

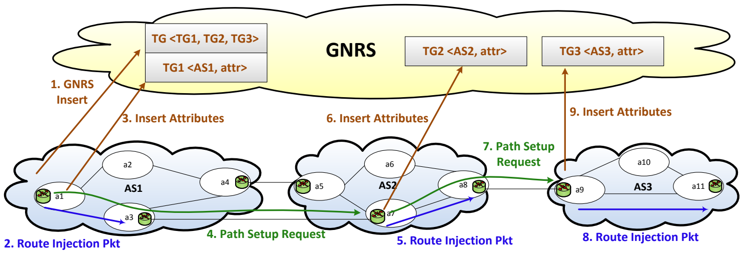


Figure 4.4: GNRS assisted tunnel setup and maintenance in a distributed framework

GNRS assisted tunnel setup

This method makes efficient use of the GNRS for tunnel setup and maintenance. Any border router(BR) that wants to initiate an inter-domain cut-through tunnel, generates a label for the inter-domain tunnel and a set of sub-labels for the intra-domain tunnel segments for each of the participating ASes. Consider the scenario illustrated in Fig. 4.4, where a BR in aNode a1 wants to setup an inter-domain tunnel between itself and a BR in a11. As a first step, the source BR inserts a mapping in the GNRS where the tunnel label <TG >is mapped to a set of intra domain labels that are to be assigned to the intra-domain tunnels <TG : TG1, TG2, TG3 >within the ASes involved. The source BR also inserts a mapping of the intra-domain tunnel label assigned for its AS to the AS number and the attributes of the tunnel, such as aggregate bandwidth, residual bandwidth, etc, in the GNRS. It then sends a path setup message containing the assigned tunnel sub-labels to the ingress BR of its neighboring domain involved in the tunnel. In addition, it also sends a route-injection packet to the internal routers within its domain to setup a cut-through path within its domain. On receiving the path-setup message, the ingress BR in the next domain, initiates a similar intra-domain path setup procedure. It also inserts the corresponding attributes for label TG2 in the GNRS, as shown. If any of the ASes along the path is not interested in participating in the inter-domain tunnel, it can delete the mapping for its intra-domain tunnel label from the GNRS. The source BR that initiated the tunnel periodically checks the GNRS for the sub-label mappings. If any entry is not present, the tunnel is decommissioned by deleting the entry for the tunnel label <TG >from the GNRS.

The egress border router on seeing a packet with a flag of 00, immediately knows

Table 4.1: Different types of encapsulation possible

Flag	header type
00	Packet traversing on an intra-domain tunnel
01	Packet traversing on an inter-domain tunnel
11	Packet not on a tunnel

that the packet is on an intra-domain tunnel and hence pops off the label and forwards the resulting packet with flag 11. However, if the flag is 01, it is just forwarded to the appropriate ingress border router in the neighboring AS, as depicted in Fig. 4.3. The advantage of stitching up of existing intra-domain tunnels is reduction in flow-rule injections for setting up cross-domain tunnels.

The cut-through switching techniques described above provide efficient traffic forwarding across the network core.

Chapter 5

Router Design

For evaluating the performance of Edge-Aware Inter-Domain Routing under several mobility scenarios, We built a prototype router(based on the click modular router described in [16]). The modular router system is used for developing network packet processing modules called elements. The way the elements of a router interact with each other are as shown in Fig.5.1.

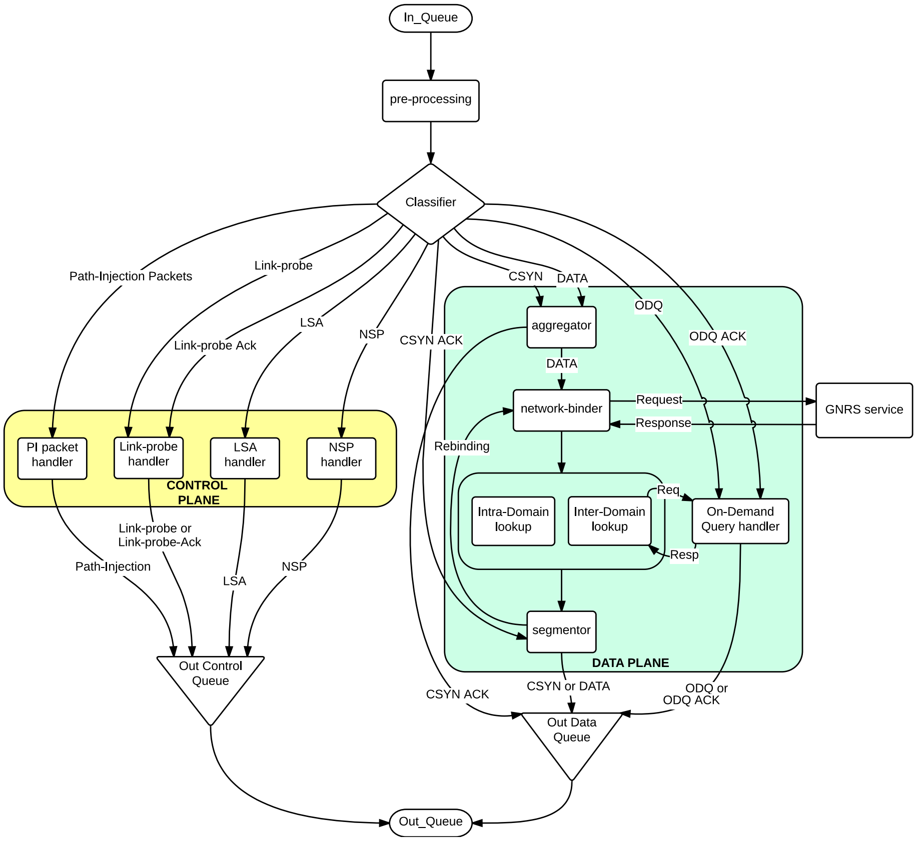


Figure 5.1: Elements and their organization in a click router instance

5.1 Router Modules

The modularity offered by Click has made it possible to clearly separate control traffic from data. Also, elements/modules of this model behave differently based on weather the router instance is that of a core router or a border router. Any packet that enters the router undergoes some pre-processing where invalid packets are dropped. Then these packets are classified based on a type field and is passed to an appropriate element for further inspection. In the sections that ensue, the terms element, module and component will be used interchangeably.

5.1.1 Control Plane

There are primarily four major functions that the modules in the control plane perform. The modules help collect/generate/re-broadcast the logs needed for the formation of all the forwarding tables and hence maintain an updated view of the network. The various control-plane elements and the requirements that they address are described in detail below.

Link-Probe handler:

This element exchanges link probes with neighbouring routers in the network and computes the short-term Estimated Time of Transit(SETT) based on when the acknowledgements were received. The long term value known as LETT is a sliding average of the past SETTs. The magnitude difference between the SETT and the LETT is used to make store/forward decisions at storage-capable routers.

Link State Advertisement handler:

This component primarily announces the state of a routers quality of connection(ETTs) to all of its neighbours. Also a router on receiving this packet logs the information and rebroadcasts the packet to all neighbours except the sender of the LSA. This information is used for computing its routing tables using Dijkstras shortest path algorithm.

Network State Packet handler:

This is the first of the elements discussed whose performance differs based on the router type. This module performs the clustering of routers into aNodes and vLinks by running a user defined algorithm and forms the nSPs that contains the aNode level graph of the AS. These are then broadcasted across the network. If a border router receives an nSP, it logs in the topology information of the source AS and holds the packet for a period of time before rebroadcasting if the router that forwarded this does not belong to the same AS. However, if the router is not a border router, it just forwards the packet without inspecting it further. Border routers use the information in these nSPs to later compute the aNode forwarding tables.

Path-Injection Packet handler:

This element generates path injection packets when run at the border routers. While generating a packet, this element uses a hash based on the source(self) and destination GUID pair to generate unique labels. Then, the path to reach this destination that has been precomputed based on factors like link latency or bandwidth or other policies is written on the packet marked with the label and sent out. Non-border routers along the path, on receiving these packets update their corresponding fast-path tables for the label received.

5.1.2 Data Plane

The elements of the data-plane work in harmony to ensure successful delivery of data chunks. There are four major requirements that the elements try to address.

Chunking-up and aggregation of data blocks:

The segmentor and aggregator perform this chunking and aggregation of data blocks. These elements implement Hop [18], the transport protocol used in MobilityFirst. In this protocol, the elements exchange synchronization messages to ensure that the receiver is ready before exchanging messages. At any router, the segmentor issues a $CSYN$

message before sending out the segmented data block. On receiving this, the aggregator of the receiving router responds with a *CSYN-ACK*. The sender starts sending the data only on receiving the acknowledgement for the *CSYN* sent. Finally, the aggregator on receiving the chunks pieces them back together asynchronously. The routers employ a hop-by-hop backpressure through an ack-withholding mechanism in order to provide more robustness and better utilization of the resources.

Address resolution:

The network binder is the one that communicates with the GNRS service in order to obtain the latest binding for a GUID. In this module, responses for GNRS queries are temporarily cached in order to provide quicker responses to subsequent queries. The network binder is accessed at the late-binding points to get a fresher address resolution. Stored packets also trigger communication with the network-binder for rebinding.

Route Lookup:

All the information obtained in the control plane are used to compute a set of forwarding tables at every router. Fig.5.2 has a sample topology with a list of all the routers at every node. Where the notations AN, G and FP denote aNode, GSTAR and fast-path forwarding tables respectively. As can be seen, all the border routers of the network maintain an extra forwarding table known as the aNode forwarding table. This maintains the aNode level topology graph and hence the next-hop anode for any destination. The aNode table is used while making any inter-domain routing decisions that includes finding the ingress and egress border routers. There is a special case in the example discussed, Router R5 that is part of AS2 does not belong to any aNode and hence is not used for routing transit traffic. Thus the only table it contains is the Gstar forwarding table. However, all other routers have two other tables in common, these are the GSTAR and the fast-path forwarding tables. The routers consult the GSTAR forwarding tables for intra-domain traffic (destination is in the same AS as the router). However, if the router is just forwarding traffic towards a destination in a different AS, a pre-set fast-path is followed. The standard routing metric used is the

ETT. The number of such tables at a router increases if various routing metrics are employed. If a source prefers a particular metric to be used while being routed, it can specify that through a Service Identifier(SID). In such an instance, the forwarding table with the corresponding metric will be used.

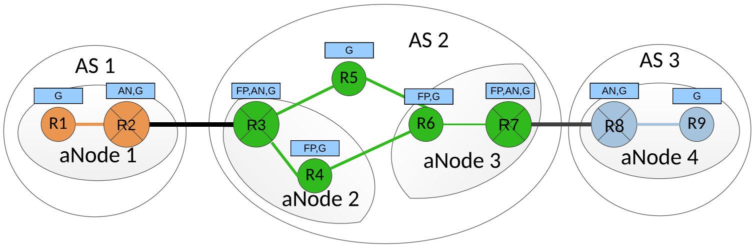


Figure 5.2: Routing tables at the routers in a sample topology

Handling On-Demand Queries:

This is a very useful feature that is used while making inter-domain routing decisions. A consequence of the telescopic-flooding of nSPs is the possession of outdated information about the organization of an AS. Though the late-binding support of GNRS provides a way for obtaining a more recent attachment point of nodes, the information obtained though useful, is minimal. The on-demand query support helps us to obtain the current link quality of the last hop, quality of the path to the destination and list of the on-path late-binding/storage-capable nodes. In this mechanism, the requesting router is responded to with an acknowledgement by the destination. Routing decisions can be made based on the acknowledgement received.

5.2 Router Decision Process

The routing decisions made at a router are affected primarily based on whether the router is a core or a border router. It is the router type that decides the number of forwarding tables that exists at a router. Also the position of a router in any given topology plays an important role in the router's decision process, a border router can be a Closest Border Router(CBR) or an Egress Border Router(EBR) or an Ingress Border

Router(IBR) and the decision that ensues is dependant on this type. This can be seen in Fig.5.3.

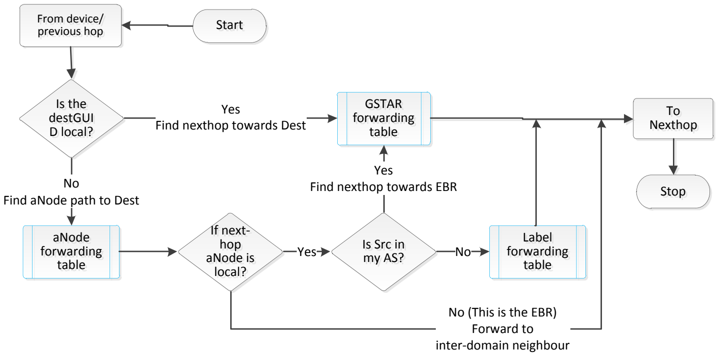


Figure 5.3: Routing decisions in a border router

In this system, if the destination for a chunk is not in the same AS as the core router sending it, the chunk is routed to the CBR as can be seen in Fig.5.4. The CBR inturn consults the aNode forwarding table to find the appropriate EBR to route the traffic to while also setting the transit flag. Along the path towards the EBR, intermediate routers use the fast-path table and perform label-based forwarding of the chunk. On reaching the EBR, the aNode forwarding table is consulted and the chunk is routed to the appropriate IBR.

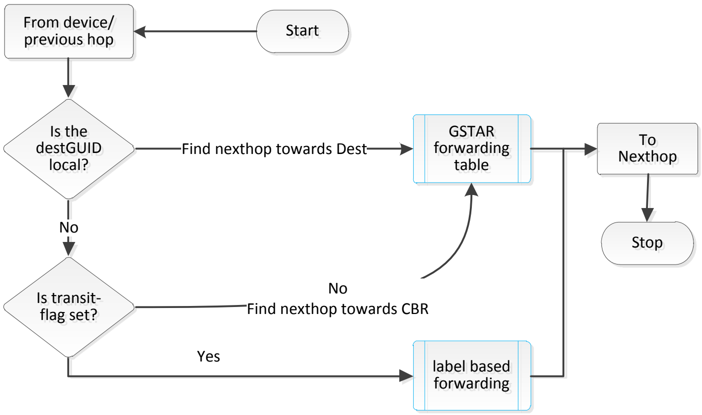


Figure 5.4: Routing decisions in a core router

All along during the forwarding process, the packets were stored or forwarded based on how the SETT varied from the LETT. Forwarding decisions can be made based on how the last-hop SETT is at the instant the chunk is forwarded. This can be done by using the On-Demand Query feature. The query is responded to with an acknowledgement that contains the requested SETT, and path quality. Forwarding decisions can be made based on weather the metrics are within permissible limits.

Chapter 6

Evaluation

In this section, we evaluate the EIR protocol in terms of scalability and mobility support capability through a large-scale prototype evaluation and an Internet-scale simulation study. Sec. 6.0.1 describes the setup and insights from an Internet scale simulation effort, and Sec. 6.0.2 describes the setup details and the results for our mobility study experiments.

6.0.1 Overhead and scalability studies

One of the foremost challenges of allowing routing information to be propagated throughout the Internet is scalability in terms of routing overhead. Also, analysis of any inter-domain protocol is incomplete without describing the convergence quantitatively. In this section, we evaluate the routing overhead and convergence of the tables in routers employing EIR for various telescopic functions for medium to large scale topologies.

Overhead vs. settling time tradeoff

We explore the tradeoff between scalability and over-head by varying the parameters α, β and A for the constant-exponential-constant telescopic function on an AS-level topology containing 47,445 ASes and 200,812 inter-domain links, evaluated through a python script that emulated the telescopic hold operation of the routers. Fig. 6.1 illustrates this trade-off between overhead and settling time during the formation of the inter-domain forwarding tables. Settling time refers to the maximum time taken for the forwarding tables of all the routers to converge following the occurrence of a change in the anode level topology. While overhead is the overall traffic across the internet due to the flooding of the nSPs.

Notice, that the worst case network overhead is about 100Gbps, assuming 1000 byte nSPs. This is a negligible fraction of the total Internet traffic of ˜182 Tbps as of 2014 [14]. As such, other values of telescopic parameters could be chosen as well, that provides much faster convergence.

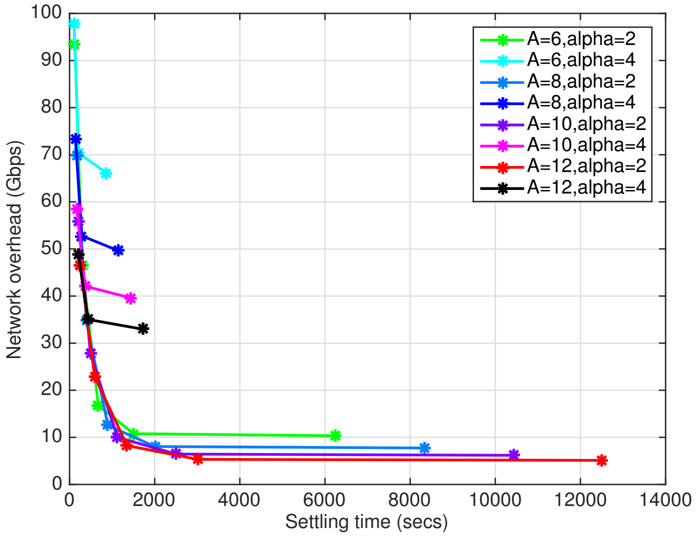


Figure 6.1: Overhead vs. settling time for constant-exponential-constant telescopic function for varying values of A, α and β

Global routing overhead

In order to analyze routing overhead for an Internet scale topology, we consider an AS level dataset available at Caida [3]. Monthly data for the year 2013 is used for this purpose which is a consolidation of data collected by the Route Views(RV) project and RIPEs Routing Information Service (RIS), and consists of an AS-level topology of 47,445 ASes and 200,812 inter-AS links. Using the above dataset, we simulate the generation and propagation of nSPs across the network. Since, analysing packet flow from each of the 47,445 nodes was not computationally feasible, we choose a random subset of ASes which generate nSPs(sources in Fig. 6.2) and the corresponding average outbound overhead at each of the other nodes. As seen from the figure, the outbound traffic at each router is of the order of 3-5 Mbps which can be easily handled by a border router. In addition, depending on the degree of each router, the outbound overhead per link is much lower and the corresponding plot is not included here for brevity.

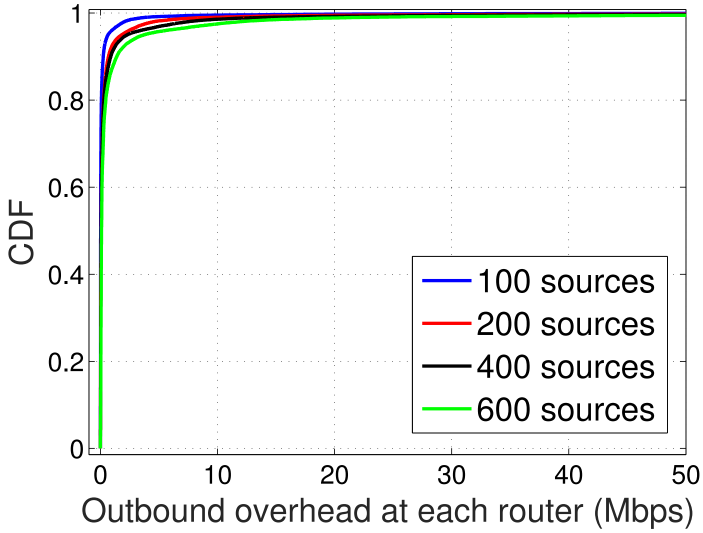


Figure 6.2: Outbound overhead at each router for the 2013 Internet topology due to nSP generation by 100-600 random ASes

Worst case update time

Next using the same topology we analyse the time delay in receiving a routing update across all ASes. Fig. 6.3 shows the CDF of the settling time or in other words, the worst case time required for every AS in the graph to get a single update generated by any AS. For this experiment, the values $5, 2, 4$ were chosen for the telescopic parameters A, α and β respectively. The plot highlights that when using the constant functions, all ASes receive the update in a short time. However, the exponential and the constant exponential functions have longer update times.

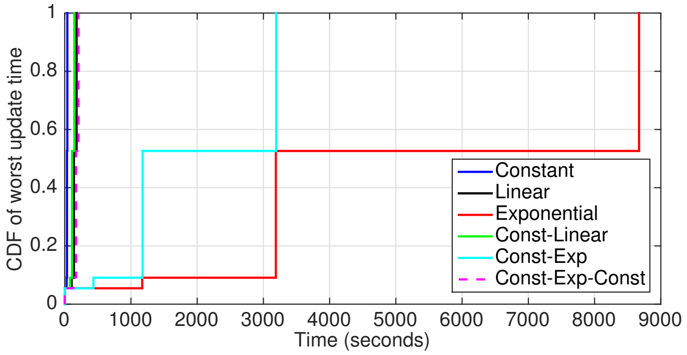


Figure 6.3: Longest routing event update time in the global routing topology for different telescopic functions

Routing table size

Maintenance of a global aNode based topology at each border router would imply that the inter-domain forwarding table size would be equal to the total number of aNodes in the topology. In order to investigate the scalability in terms of routing table entries, we look at a July, 2012 Caida dataset that provides the intra-domain topology of about 22,000 ASes. As explained earlier, EIR allows a flexible aggregation scheme, wherein each AS could independently decide on the amount of information they wish to disclose about their internal topology and accordingly the number, types and properties of aNodes they wish to publish in their nSP. However for the sake of simplicity in the following evaluation, we consider all ASes to aggregate uniformly based on the fraction of aggregation, varying from 0 to 1. A value of 0 indicates, there is no aggregation at all, or in other words, every physical router within an AS is a separate aNode. On the other extreme, a value of 1 indicates that all the routers belonging to an AS are aggregated to a single aNode. As shown in Fig. 6.4, the blue curve indicates the inter-domain table size in terms of the number of entries at each border router with ASes employing varying levels of aggregation. In a realistic scenario, we expect ASes to not follow a uniform aggregation scheme, but on average have a fairly large aggregation fraction, and the global table size to lie somewhere along the blue line. The red curve in Fig. 6.4 shows the average BGP table size as reported daily by CIDR [1] for the month of July in 2012. Note that although BGP does not provide any intra-domain topology information, it needs to maintain an entry for every aggregated address prefix announced in the Internet. As seen from the plot, even though EIR maintains a global view of the network, aNode table sizes are comparable to the current BGP tables, for moderate levels of aggregation.

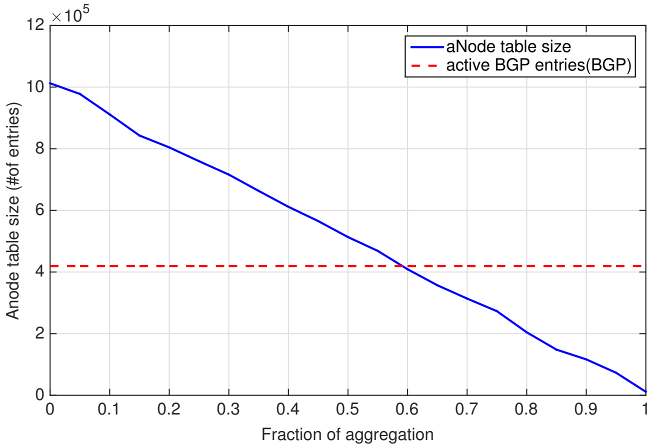


Figure 6.4: Inter-domain table size at each border router for different levels of router to aNode aggregation

6.0.2 Mobility evaluations

To measure the performance of the different use-cases of EIR, we built a prototype router (based on the Click modular router design [16]) and deployed it on ORBIT [24]. One of the key aspects of EIR is its support for mobility, both for individual devices as well as networks as a whole. In order to evaluate such scenarios, we design a realistic inter-domain topology and a probabilistic mobility transition matrix which is briefly described below.

Topology generation and probabilistic mobility

We start with a Caida dataset from 2012 [1] that provides router level topologies of 22,000 ASes, and parse the dataset based on cities. Specifically we focus on San Francisco, which has a point of presence of about 326 ASes. We consider a cooperative scheme where a multitude of ASes agree to share coverage and connectivity among their customers, i.e. an User X can decide to switch from one network provider to another when moving, provided the latter provides a better coverage in the region. Out of all the available ASes in the dataset, we choose 15 random ASes to participate in this cooperative scheme. Given the router level topology, a corresponding aNode topology is developed for each of the participating ASes based on geographical proximity, which leads to a topology of 53 aNodes. Fig. 6.5(a) shows the geographical distribution of the aNodes, where each AS is denoted by a different color, and Fig. 6.5(b) shows the connectivity graph of the aNodes. In order to realistically model inter-domain mobility our transition probability matrix takes into account the following factors:

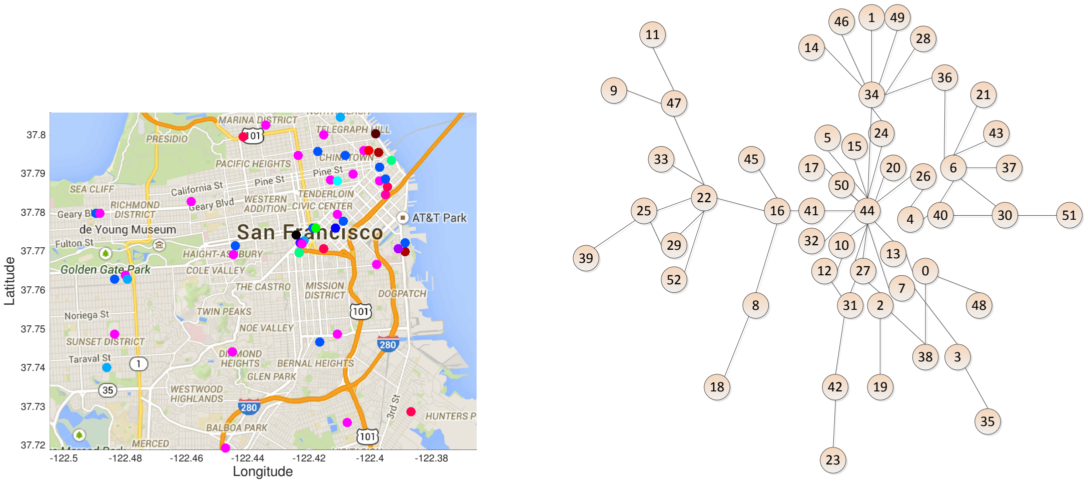
- Local mobility is considered within a certain radius (denoted as r), with equal probability of transition to all aNodes within the 'local boundary' based on the average mobility speed of the user

- Transition between aNodes belonging to the same AS within the local boundary are favoured, as users tend to remain connected to the same network provider as they move, unless no connectivity by the current provider is available at the new location

- A random, k number of 'macro mobility' transitions based on the average number of networks visited by a user per day [12] are assigned non-zero probability (determined by α) of transition

The transition probability computations considers the following variables:

$$\begin{aligned}
Z &= \text{avg number of network transitions /sec} \\
K &= \text{total number of network transitions} \\
T &= \text{granularity of transition (in sec)} \\
r &= \text{avg distance to neighbors (in meters)} \\
s &= \text{avg speed of mobility (in m/sec)} \\
w &= \text{average transition rate/sec} = s/r
\end{aligned}$$



(a)

(b)

Figure 6.5: End-host mobility topology:(a) aNode distribution in the evaluation topology, (b) Connectivity graph of the aNodes

$$\text{Transition probability to each of the } N_j \text{ neighbors from } node_j = \alpha(wT)/N_j$$

$$\text{Transition probability to each of the } K \text{ non-neighbors from } node_j = (1-\alpha)(ZT)/K$$

End host mobility support

Based on the San Francisco topology and the mobility matrix generated for a typical mobile user, we looked at the path stretch that is incurred with and without late binding. Note that without late binding, failure in delivery to the end-host at an attachment point would lead to rebinding of the packet through a GNRS lookup to the new point of attachment of the user. On the other hand, the packet could be made to late bind somewhere closer to the edge, where the routers have a "fresher view" of the network. In addition, delaying the binding of a packet to its ultimate network address helps in the case of mobility, as the end-user might already move from its point of attachment while the packet is in transit. An observation that was made in the end-host mobility scenario is that, during a mobility event, only the chunks in-transit undergo rebinding due to a delivery failure. These account to less than 1% of the total chunks transmitted.

Fig. 6.6 highlights the improvement in path-stretch when packets are late binded along the way at an aNode with a high degree (denoted as the junction aNode). Note that the plots are pretty close since once a mobile moves, only the packets in transit are rerouted and suffer a path stretch, whereas newer packets are automatically sent to the new destination, from the source, following a GNRS lookup. Future evaluations plan to look at different late binding techniques, so as to minimize path-stretch and improve latency of data delivery across a broad range of mobility scenarios.

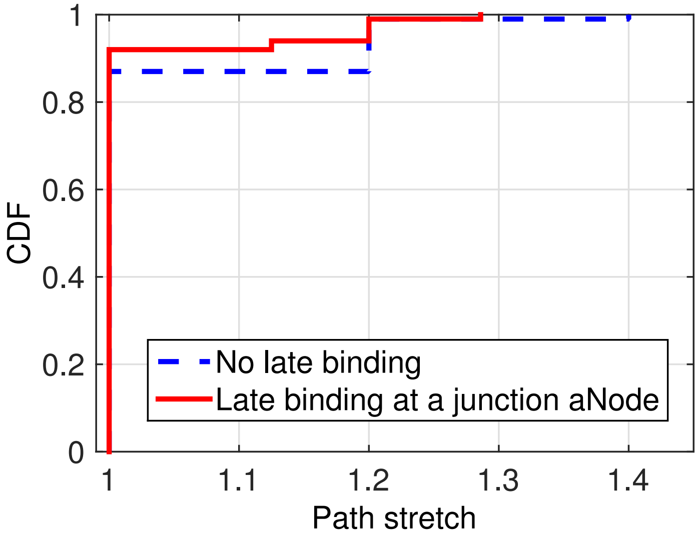
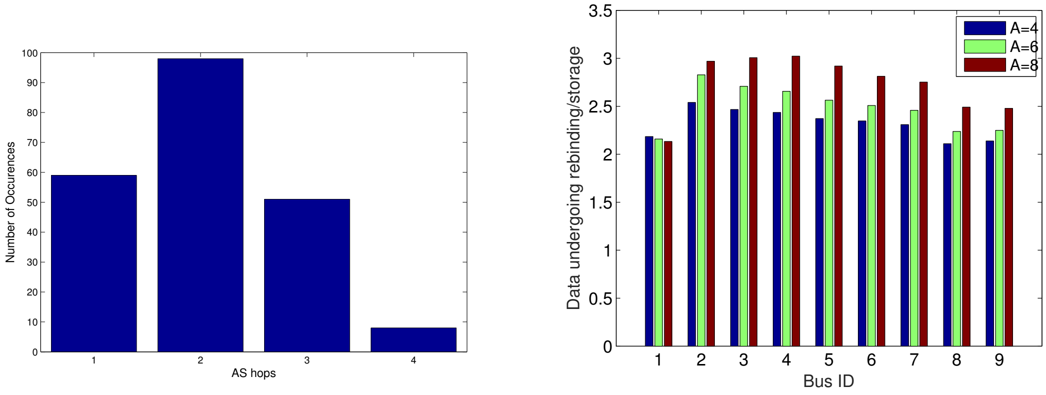


Figure 6.6: CDF of path stretch with and without late binding

Network mobility support through nSP propagation

EIR advertises any changes in the network organisation through the propagation of the abstracted topology in terms of aNodes and vLinks. Consider the scenario where, the mobile phones in a bus are being serviced by the Wi-Fi support provided within the bus. In this scenario, there are a bunch of mobile nodes that are moving while remaining connected to the network through one or more links. This can be visualised as a mobile network-entity such as a mobile aNode(bus) connected through vLinks. Thus, in this experiment a bus containing mobile devices(aNode) was considered to follow a trace within San Francisco. As the bus moved it connected to the nearest available aNode through a vLink. Also, while moving, the bus preferred staying in the same AS and this placed a restriction on the number of aNodes that the mobile node can connect to. For this experiment, data was transmitted from a far off source (source that is



(a)

(b)

Figure 6.8: Network mobility:(a) Distribution of the number of AS transitions for network mobility events, (b) percentage of packets that undergoes storage or rebinding along several bus traces in SFO

about 4 AS-hops away from San Francisco). Then, the delivery rate is measured at the destination. We define delivery rate as the ratio of number of chunks received to the number of chunks sent. For this evaluation, we did not consider the rebinding or DTN capability on the intermediate nodes.

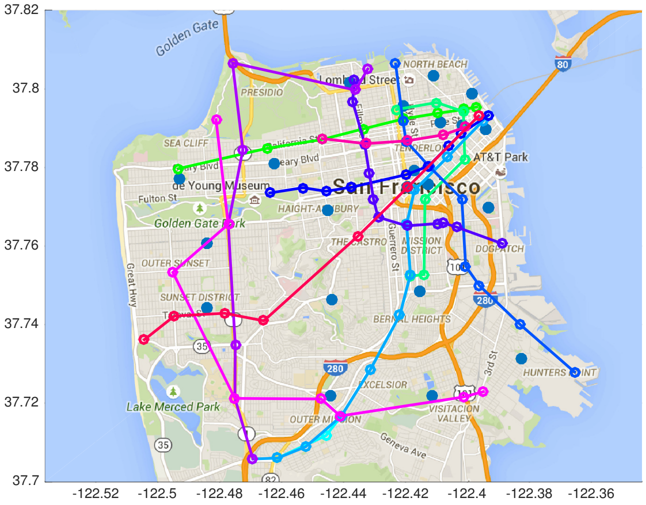


Figure 6.7: Taxi traces in SFO considered for evaluating network mobility

The various traces considered for this evaluation are shown in Fig. 6.7. Fig. 6.8(b) shows how the delivery rate varies on 9 randomly picked routes. The distribution of the number of AS-hops a device traverses when moving from one attachment point to another is illustrated in Fig. 6.8(a). It is clearly evident that there is a domination of transitions involving 2 AS hops that need to be traversed during these mobility events. Of the 9 randomly picked traces, trace 1 resulted in a scenario that had a domination of 1-hop transitions i.e, most of the mobility events resulted in the transition to a neighbouring AS the is 1 AS-hop away from the current AS. In this case, the data delivery rate is almost similar in all 3 cases(A=4,6,8) because the packets get stored for a period of time until the first nSP from the AS to which the mobile node attached to is received. Whereas in the other traces, there are a few transitions to ASes that are multiple AS-hops away. There is a lower data-delivery rate because more packets in transit get stored as the reachability to the aNode in motion is not known for a longer time due to the telescopic hold of the nSPs.

Chapter 7

Future Work and Conclusion

7.1 Future Work

Future work on EIR includes:

- **GNRS Assisted service realisation:** Extending current work to utilize GNRS support (GNRS assisted EIR) for implementing some of the services

- **Policy setup:** We have analysed how to setup local policies by agreeing upon business relationships between neighbouring ASes. We should however further evaluate other possible policies(such as roaming agreements and algorithms for dynamic traffic-engineering) that can be setup between ASes.

- **Multi-homing/Anycast use-case:** This work focussed on the use-cases associated with mobility. However, there are scenarios that focus on the performance of the system under the presence of multiple access-technologies or diverse paths. The performance enhancement observed by using these choices must be evaluated.

- **Comparison with existing protocols:** This work proves its feasibility and benefits through several experiments. However, more credibility can be achieved if we compare the performance of a protocol like BGP under several of the use-cases discussed.

7.2 Conclusion

This work presents the feasibility and evaluates the benefits of having a clean-slate inter-domain routing protocol called EIR(edge-aware inter-domain routing) as the Internet is fast approaching a phase where mobile devices would outnumber the fixed end-hosts.

The EIR protocol is based on telescopic flooding of an abstracted internal network topology and state of ASes in "Network State Packets (nSPs)" along with late binding of end-point names to locators at intermediate routers in the network. Dissemination of nSPs across the Internet provides improved visibility of the network topology as a whole. These mechanisms enable routers to make informed routing decisions for mobility-oriented services such as roaming with intermittent disconnection, multi-homing and edge-peering. The EIR protocol has been extensively validated through both large-scale simulations and Orbit testbed emulations of networks running the Click software implementation of routers. The results showcase that the telescopic-flooding keeps the overhead within bounds while making it possible to have reasonable convergence times and the aggregation of the internal topology into aNodes results in inter-domain tables that are comparable in size to the current BGP table instances. Also, the mobility experiments with frequent migration of clients and networks across domains highlight the performance enhancements that can be achieved by exposing the network state through the EIR protocol.

References

[1] CIDR-Report. `http://www.cidr-report.org/as2.0/`.

[2] Long-Range 802.11n (5GHz) Wi-Fi Backhaul. `http://www.ruckuswireless.com/products/zoneflex-outdoor/7731`.

[3] The CAIDA UCSD Internet Topology Data Kit. `http://www.caida.org/data/internet-topology-data-kit`.

[4] Wi-Fi First: Knocking some sense into the smartphone. `http://www.wififirst.org/wififirst-ebook.pdf`.

[5] B. Abarbanel. Implementing global network mobility using bgp. In *NANOG Presentation, http://www. nanog. org/meetings/nanog31/abstracts. php*, 2004.

[6] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker. Accountable internet protocol (aip). In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 339–350. ACM, 2008.

[7] B. Augustin, B. Krishnamurthy, and W. Willinger. Ixps: mapped? In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 336–349. ACM, 2009.

[8] M. Caesar and J. Rexford. Bgp routing policies in isp networks. *Network, IEEE*, 19(6):5–11, 2005.

[9] Cisco White Paper. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011-2016, Feb. 2012.

[10] D. Farinacci, D. Lewis, D. Meyer, and V. Fuller. The locator/id separation protocol (lisp). 2013.

[11] L. Gao. On inferring autonomous system relationships in the internet. *IEEE/ACM Transactions on Networking*, 9:733–745, 2000.

[12] Z. Gao, A. Venkataramani, J. F. Kurose, and S. Heimlicher. Towards a quantitative comparison of location-independent network architectures. In *Proceedings of the 2014 ACM conference on SIGCOMM*, pages 259–270. ACM, 2014.

[13] P. Godfrey, I. Ganichev, S. Shenker, and I. Stoica. Pathlet routing. In *ACM SIGCOMM Computer Communication Review*, volume 39, pages 111–122. ACM, 2009.

[14] C. V. N. Index. Global mobile data traffic forecast update, 2014-2019. *White Paper, February*, 2015.

[15] J. R. Iyengar, P. D. Amer, and R. Stewart. Concurrent multipath transfer using sctp multihoming over independent end-to-end paths. *Networking, IEEE/ACM Transactions on*, 14(5):951–964, 2006.

[16] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The click modular router. *ACM Transactions on Computer Systems (TOCS)*, 18(3):263–297, 2000.

[17] N. Kushman, S. Kandula, D. Katabi, and B. M. Maggs. R-bgp: Staying connected in a connected world. USENIX, 2007.

[18] M. Li, D. Agarwal, and V. A. Block-switched networks: A new paradigm for wireless transport.

[19] M. Motiwala, M. Elmore, N. Feamster, and S. Vempala. Path splicing. In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 27–38. ACM, 2008.

[20] S. C. Nelson, G. Bhanage, and D. Raychaudhuri. GSTAR: Generalized storage-aware routing for MobilityFirst in the future mobile Internet. In *Proceedings of MobiArch'11*, pages 19–24, 2011.

[21] W. B. Norton. Internet service providers and peering, 2000.

[22] D. S. Phatak and T. Goff. A novel mechanism for data streaming across multiple ip links for improving throughput and reliability in mobile environments. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 773–781. IEEE, 2002.

[23] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani. Mobilityfirst: a robust and trustworthy mobility-centric architecture for the future internet. *ACM SIGMOBILE Mobile Computing and Communications Review*, 16(3):2–13, 2012.

[24] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh. Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 3, pages 1664–1669. IEEE, 2005.

[25] P. Rodriguez, R. Chakravorty, J. Chesterfield, I. Pratt, and S. Banerjee. Mar: A commuter router infrastructure for the mobile internet. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 217–230. ACM, 2004.

[26] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture, rfc 3031. 2001.

[27] I. Seskar, K. Nagaraja, S. Nelson, and D. Raychaudhuri. MobilityFirst Future Internet Architecture Project. In *MobilityFirst Project, Proc. ACM AINTec 2011.*

[28] I. Seskar, K. Nagaraja, S. Nelson, and D. Raychaudhuri. Mobilityfirst future internet architecture project. In *Proceedings of the 7th Asian Internet Engineering Conference*, pages 1–3. ACM, 2011.

[29] N. Somani, A. Chanda, S. C. Nelson, and D. Raychaudhuri. Storage-Aware Routing for Robust and Efficient Services in the Future Mobile Internet. In *Proceedings of ICC FutureNet V, 2012.*

[30] L. Subramanian, M. Caesar, C. T. Ee, M. Handley, M. Mao, S. Shenker, and I. Stoica. *HLP: a next generation inter-domain routing protocol*, volume 35. ACM, 2005.

[31] T. Vu et al. DMap: A Shared Hosting Scheme for Dynamic Identifier to Locator Mappings in the Global Internet. In *Proceedings of ICDCS '12*, pages 698–707, 2012.

[32] S. Van Dongen. Graph Clustering Via a Discrete Uncoupling Process. *SIAM Journal on Matrix Analysis and Applications*, 30(1):121–141, 2008.

[33] F. Wang and L. Gao. Path diversity aware interdomain routing. In *INFOCOM 2009, IEEE*, pages 307–315. IEEE, 2009.

[34] W. Xu and J. Rexford. *MIRO: multi-path interdomain routing*, volume 36. ACM, 2006.

[35] X. Yang, D. Clark, and A. W. Berger. Nira: A new inter-domain routing architecture. *IEEE/ACM TRANSACTIONS ON NETWORKING*, 2007.

[36] X. Yang and D. Wetherall. Source selectable path diversity via routing deflections. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 159–170. ACM, 2006.