Computer Networks 127 (2017) 13–30

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

EIR: Edge-aware inter-domain routing protocol for the future mobile internet☆

Shreyasee Mukherjee[a,*], Shravan Sriram[a], Tam Vu[b], Dipankar Raychaudhuri[a]

[a] WINLAB, Rutgers University, 671 Route 1 South, North Brunswick, NJ 08902, USA
[b] MNS Laboratory, Department of Computer Science, University of Colorado Boulder, 1111 Engineering Drive, Boulder, CO 80309, USA

ARTICLE INFO

Article history:
Received 12 September 2016
Revised 23 June 2017
Accepted 24 July 2017
Available online 25 July 2017

Keywords:
Inter-domain routing
Future internet architecture (FIA)
Mobility

ABSTRACT

This work describes a clean-slate inter-domain routing protocol designed to meet the needs of the future mobile Internet. In particular, we describe the edge-aware inter-domain routing (EIR) protocol which provides new abstractions, such as aggregated-nodes (aNodes) and virtual-links (vLinks) for expressing network topologies and edge network properties necessary to address mobility related routing scenarios which are inadequately supported by the border gateway protocol (BGP) in use today. Specific use-cases addressed by EIR include emerging mobility service scenarios such as multi-homing across WiFi and cellular, multipath routing over several access networks, and anycast access from mobile devices to replicated cloud services. It is shown that EIR can be used to realize efficient routing strategies for the mobility use-cases under consideration, while also providing support for a range of inter-domain routing policies currently associated with BGP. Simulation results for protocol overhead are presented for a global-scale CAIDA topology, leading to an identification of parameters necessary to obtain a good balance between overhead and routing table convergence time. A Click-based proof-of-concept implementation of EIR on the ORBIT testbed is described and used to validate performance and functionality for selected mobility use-cases, including mobile data services with open WiFi access points and mobile platforms such as buses operating in an urban area.

1. Introduction

The inter-domain routing architecture of the Internet is currently based on the border gateway protocol (BGP) standards [1]. BGP, which was introduced about 25 years ago, represented a major advance in networking because it provided fully distributed, non-hierarchical routing mechanisms between autonomous systems (ASes) at a global scale. More importantly, BGP provides a flexible framework for policy-based routing taking into account local preferences and business relationships [2]. The Internet is currently going through a fundamental change driven by the rapid rise of mobile end-points such as smartphones and embedded Internet-of-Things (IoT) devices [3]. The emerging "mobile Internet" will require new approaches to both intra- and inter-domain routing in order to deal with increased dynamism caused by end-point, network and service mobility. This dynamism can take various forms, ranging from conventional end host mobility and edge network mobility to multi-homing and multi-network access associated with emerging hetnet and 5G cellular scenarios [4]. In addition, mobile edge cloud scenarios [5] involve dynamic cloud service migration across networks, requiring anycast routing capabilities which are not readily supported by current inter-domain protocols. A common thread across all these use-cases is the need for a better visibility of the network connectivity graph and the quality of alternative paths to the mobile end-point in order to be able to make more intelligent and informed routing decisions that takes edge and access network into account.

Emerging Internet requirements such as mobility and content have motivated several clean-slate Internet design projects such as Named Data Network (NDN) [6], XIA [7] and MobilityFirst [8]. Previously published works on these architectures have addressed mobility requirements at the intra-domain level [9,10], but inter-domain routing for the future Internet remains an important open problem. In this paper, we first motivate the need for clean-slate approaches to inter-domain based on several use-cases, and then describe a specific new design called EIR (edge-aware inter-domain routing) intended to meet emerging requirements. The proposed

☆ Research supported by NSF Future Internet Architecture (FIA) grant CNS-134529.
* Corresponding author at: WINLAB, Rutgers University, 671 Route 1 South, North Brunswick, NJ 08902, United States.
E-mail addresses: shreya@winlab.rutgers.edu (S. Mukherjee), sshravan@winlab.rutgers.edu (S. Sriram), tam.vu@colorado.edu (T. Vu), ray@winlab.rutgers.edu (D. Raychaudhuri).

http://dx.doi.org/10.1016/j.comnet.2017.07.013
1389-1286/

protocol provides new abstractions for expressing network topology and edge network properties necessary to support a full range of mobility services such as multi-homing over WiFi and cellular, multipath routing over multiple access networks, and anycast access to cloud services from mobile devices.

The proposed edge-aware inter-domain routing protocol was developed as a part of the MobilityFirst Future Internet Architecture (FIA) project [8] aimed at a clean-slate redesign of the IP protocol architecture. It is noted here that clean-slate research projects like MobilityFirst do recognize the fact that the Internet cannot be changed overnight particularly when dealing with core protocols such as inter-domain routing. However, with the advent of software-based network functionality, it is now increasingly practical to introduce new Internet protocol concepts on a trial basis. In particular, the recently proposed "SDX (software-defined exchange)" concept makes it possible for networks to voluntarily participate in enhanced or new protocol frameworks for inter-domain routing, as discussed by Feamster et al. [11]. For example, a new inter-domain routing protocol like EIR can be implemented by a small number of cooperating ASes as an SDX-hosted function that supplements BGP with the goal of efficiently supporting a specific service such as multi-homing over WiFi and cellular networks. Such an initial deployment can be limited to 10's of networks (content provider, a few transit networks, cellular access network operators, etc.) with the sole purpose of optimizing multi-homed service delivery. As additional networks become aware of the benefits and join these special purpose networks, there could be a critical mass effect leading to broad adoption of a new routing protocol standard. While it is difficult to predict when these large-scale changes in the network will occur, there is no doubt that significant changes to BGP will occur over a ~10 year time horizon, and it is thus timely and important to study inter-domain routing techniques designed to meet future needs.

The main contributions of this work are:

(i) Identifying and reasoning about the new requirements of the future mobile Internet.
(ii) Designing a specific protocol architecture (EIR) which realizes these requirements.
(iii) Presenting a careful evaluation of EIR through simulation, emulation, and implementation to show the feasibility and efficiency of the newly proposed architecture.

2. Emerging network service usecases

In this section, we consider some of the emerging use-cases such as mobility, multipath, edge peering, in further detail and discuss their implications on inter-domain routing.

2.1. Multipath support

A typical mobile hand-held device can see multiple available networks (cellular or WiFi) at the same time. Although the majority of current business models generally restrict a user to a single cellular network provider, with the increasing popularity of "hetnet" mobile services, a mobile device might be soon able to simultaneously connect to a dynamically changing set of cellular and WiFi networks [12,13]. It is possible to consider a variety of service objectives for this scenario, ranging from "most economical" to "highest throughput interface" to "all interfaces". Intermediate solutions to support such connectivity do exist [14–16], but supporting network-wide multi-homing has a very broad architectural implication. Since the cellular and WiFi networks will in general be in different Internet domains, autonomous systems need to support independent paths of connectivity for a single end-to-end flow. Accordingly, having the visibility of the global network graph and some awareness of edge network properties would help the routers to make informed forwarding and/or multicast copy decisions.

2.2. Wireless edge peering

Peering between autonomous domains is one of the most important capabilities of the Internet. ASes employ various types of peering agreements with different number of neighboring ASes and a recent report shows the presence of 75% more peering links than previously known [17]. 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 [18], assuming a sufficient security solution is in place. This wireless peering link would keep the two networks connected even if both the service providers, ISP_1 and ISP_2 are undergoing failures, and can help one network to use the connectivity of the other network in case either one of ISPs has a link failure. We believe that wireless peering will be increasingly important for the future mobile Internet, and requires more flexible and granular policy specifications than currently supported, especially for disaster-recovery (when wired connections to ISPs might fail) and congestion handling (to maintain partial edge-connectivity when the main links become too congested).

2.3. Dynamic network formation and mobility

Another emerging mobility service scenario is that of dynamic network formation along with network mobility. For example, there are opportunities for a network to be formed between groups of vehicles on the highway, and these networks should be able to quickly peer along the edge with different access networks encountered during mobility. As another emerging use-case consider Google's Project Loon, which proposes to beam LTE access in developing countries from a network of aerial balloons [19]. Managing a global scale of unmanned and highly mobile base-stations is challenging, despite the partial point solution that BGP currently provides for airline connectivity [20]. Such techniques cannot scale to a network of hundreds of mobile nodes or respond to changing link quality/capacity at the edge of the network.

2.4. Service anycast

Emerging cloud-based service applications for on-demand computing or storage often require anycast routing for finding the "closest available resource" based on specialized metrics such as latency or bandwidth. Selection of inter-domain paths based on more than just the BGP reachability metric becomes necessary in such cases and is difficult to achieve without setting up of additional overlays [21]. In addition to the support of mobility *as a norm* through the routing plane, we believe that the inter-domain routing protocol should provide means of flexible path selection based on metrics other than the traditional shortest AS hop count.

2.5. Multicast support

With the Internet traffic becoming increasingly content driven [3], support for efficient multicasting becomes crucial. Consider the use-case of multiple mobile users trying to stream a newly released series from a popular content provider, such as Netflix. Not only does it require an anycast *get(content)* request from the users, the content provider can employ multicasting to stream

the content simultaneously to multiple users subscribed to different ISPs. Emerging IoT concepts involving wireless sensor networks (WSNs) also need support for large scale multicasting [22]. Inter-domain multicasting requires fine-grained path-visibility to choose appropriate bifurcation points within the network for data replication as well as efficient group management mechanisms. However multicasting extensions for BGP (MBGP) [23] cannot scale to large groups. The overheads associated with setting up and maintenance of MBGP has also limited its wide-scale deployment.

EIR satisfies the basic inter-domain routing protocol requirements of scalability, robustness, and support for flexible routing policies, in addition to the support for emerging use-cases of network-mobility, multi-homing, multicast and anycast services. Some of these use-cases are currently partially supported through overlay services, such as, Akamai's content delivery system [24], Google's Project Fi [12] for multi-network access, etc. However, given the wide diversity of existing and emerging services, many of these heterogeneous services would benefit from an uniform and intelligent routing plane that provides increased visibility of path and quality metrics. This not only reduces the management complexity of overlay networks per service, but also leverages on the efficiency of not having to infer substrate network topology for each of them.

3. EIR protocol design

In this section we present the key building blocks of EIR. First we describe the design rationale and concepts, followed by in-depth protocol features.

3.1. Design concepts

Our design decisions are directed towards enabling and using (i) information about more links (e.g. internal structure of the AS), and (ii) more metrics about each link (e.g. whether wireless or wired link between networks). Below are the top-level design principles behind EIR.

3.1.1. In-network mapping of names to addresses

The concept of separating names from addresses has been used in several recent proposals (MobilityFirst [8], LISP [25], HIP [26], AIP [27]). As per a recent measurement study [28], this is being increasingly deployed by ASes. The infrastructure for mapping between names and addresses can either be hosted as services external to the network and accessed only by end nodes, or alternatively be implemented in-network and be accessible by both end hosts and routers. We make use of the in-network mapping approach, to ensure delivery of packets in the case of fast end host mobility. All network attached objects (devices, routers, access points, etc.) are assigned unique names and a logically centralized global name resolution service (GNRS) maintains mappings between a name and its routable address(es). Several past works have shown the feasibility of Internet-scale, distributed, in-network mapping infrastructure with extremely small query-response time [29–31].

3.1.2. Propagating network or link properties in inter-domain routing

BGP does not differentiate inter-network links based on link properties (such as wired or wireless links), making it difficult to perform informed routing decisions based on capacity constraints. For example, in an early in-flight WiFi implementation, Boeing associated each flight with an IP address block which was announced into the global routing system from different locations as the plane moved [20]. Other networks receiving such announcements had no idea that the last hop for this path had a ground-to-plane wireless link instead of the usual high-capacity peering-point wired link and thus might have congested the link with excessive traffic. In EIR, coarse-grained link-level information about each inter-network link is propagated through the routing protocol to enable networks to make forwarding decisions based on aggregate edge network properties.

3.1.3. Increased visibility of alternative paths

More often than not, there are multiple routes available between any two networks in the Internet and those routes can entail vastly different properties [32]. In BGP, a network might learn about alternate routes to a destination but can only select and propagate one "best" route to other networks, which leads to a myopic view of the network graph. In order to support the increasingly important use-cases of multipath and multi-network operations, EIR entails network-wide visibility of multiple possible paths between each pair of networks. Note that recent standardization efforts in BGP looks into similar aspects where an AS can advertise multiple paths to the same destination prefix [33]. This requires defining path identifiers to distinguish between the multiple paths announced. This has similarities to EIR where multiple aggregated link information (intra or inter-domain) are advertised in the routing update messages, as explained in detail in Section 3.2.1. However, in EIR, each AS does not advertise specific paths to destinations, but rather exposes a topological graph which can then be utilized by other ASes to compute appropriate paths. In addition, EIR incorporates mechanisms that allow networks to realize policy routing beyond the common routing policies seen today in BGP, as discussed in detail in Section 4.

3.1.4. Flexibility in exposing internal structure

EIR enables flexibility in the amount of internal network structure that a domain announces to other networks. This ensures that networks have the control over the granularity of topology information they want to expose. At the same time, dynamic traffic engineering and differential network services can be realized more easily and efficiently when each network has a more fine-grained view of multiple possible inter-AS and intra-AS paths.

3.1.5. Support for multiple routing policies

EIR enables multiple routing schemes through the propagation of multiple link characteristics in its routing messages. For example, routers can compute routes based on high bandwidth, low latency, high availability and so on. In addition, non-conventional paths based on specific router functionality, such as long-term storage capable routes, fast-path optical network transit routes, "traffic only through customers", etc., can also be computed.

3.2. Protocol building blocks

While BGP is sufficient for basic inter-domain routing with static ASes, it trades flexibility in route selections and the availability of link quality information for a high level of abstraction and scalability. In contrast, we argue for a more balanced architecture that reveals enough internal state of the network so that network entities can make a smarter decision in message delivery, satisfying different requirement of today's services, but also have flexible aggregation capability to make the architecture scalable. We contend that a network entity that wants to deliver a packet to a far away destination does not need to know the most up-to-date state around that destination node until the packet gets closer to the destination. Knowing about the existence of possible alternate paths and the approximate condition of paths connecting the two endpoints is useful to make a smarter routing decision. Following are the key protocol design elements in EIR.

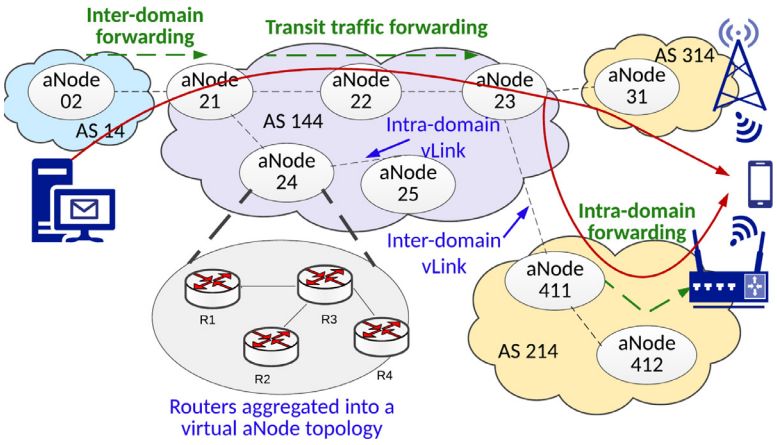


Fig. 1. aNode-vLink topology abstraction for an AS.

3.2.1. Aggregated nodes (aNodes) and virtual links (vLinks)

Each AS has the option of dividing its routers or other network elements (such as access points and base-stations) into one or more than one groups (called aggregated nodes or aNodes) as shown in Fig. 1. Entities belonging to the same aNode typically share some common operational or physical attributes. As examples, possible compositions of aNodes include: the entire AS (similar to the current Internet architecture); group of routers in a geographical area; all routers that support flow-based routing (for example through OpenFlow); wireless routers on bus/train/plane networks; and cell-site routers in flat LTE networks. The network management authority for each AS aggregates routers to aNodes and assigns a unique aNode identifier to each. In this design, an aNode is identified by a "globally unique identifier" (GUID) that is obtained from a trusted naming service which has central visibility of all the allocated names and also manages trust. By convention the routable network address is a hash of the GUID. We refer readers to our prior work for more detailed discussion on GUID creation and mobile naming service [29].

The aNodes are connected via virtual links or vLinks, which are single-hop or multiple-hop connections. The overall architecture is highlighted in Fig. 1. The aNode-vLink abstraction allows a network to partially expose its internal connectivity structure while limiting it to a level of detail that fits its needs. Networks that do not wish to expose internal structure describe themselves as a single aNode. A network state packet (nSP) is used to inform other ASes of the network's internal aNodes and vLinks along with their properties such as bandwidth, latency and availability.

Aggregation techniques have been proposed over the years for hierarchical protocols like PNNI [34] as well as flat OSPF style routing [35] for intra-domain routing, whereas Pathlet [36] proposes similar concepts for inter-domain routing. Understandably, ASes may not be willing to expose internal characteristics globally to other ASes. However, there is a benefit in doing so, namely: (i) the information advertised is aggregated and coarse-grained, therefore, does not expose the intra-domain link state information; (ii) Previous research has shown that BGP route computation often suffers because peering agreements are not available, even though they can be easily inferred through passive monitoring techniques [37,38]; and, (iii) EIR does not necessitate advertisement of internal topology but provides the flexibility of allowing ISPs to expose as much as they want. For example, stub ASes with a single inter-domain link probably has no benefit for exposing internal structure. However, large transit ASes will benefit by exposing multiple ingress-egress points to achieve traffic engineering goals and provide potential value-added services to its customers.

3.2.2. Route dissemination through network state packets

The internal structure of a domain is expressed through a graph of aNodes connected by a set of vLinks. Route update messages consisting of both internal and external properties of a network are periodically disseminated by ASes in the form of network state packets (nSP). The nSP created by each border router contains the aNode-vLink connectivity graph and aggregated state information for aNodes and vLinks.

Fig. 2 highlights the update format with aggregated state of links expressed in the form of a ⟨Bandwidth, Variability, Availability, Latency⟩ tuple. Multiple physical links can be aggregated to a single vLink and as such these parameters can be average of all the links, or the maximum or minimum of them. Such decisions are taken individually by each network and by varying these four parameters, a domain can control traffic patterns that traverses into and inside its network. For example, a vLink connecting a single airplane might have absolute bandwidth equal to the bandwidth that it could deliver to all passengers. In addition, with its fine-grain internal structure exposed to outside networks, a domain can also offer its clients with flexible route selection as a value-added service.

Optional state, capacity, and capability information is expressed through the type-mask and could include type of an aNode or vLink (WiFi enabled aNode, ground-to-satellite vLink, etc.) or enhanced capabilities of an aNode (storage-capable aNode, compute-capable aNode etc), in addition to the policy attributes of vLinks ("peer-to-peer", "customer-provider", etc.) as described in Section 4. The set of generic policies supported by an AS is also part of the state information in the form of service identifier (SID) types, discussed in Section 4. The parameters characterizing the aNodes and the vLinks can and will change over time. They are recomputed by a border router every time it generates an nSP.

As shown in Fig. 2, each nSP is a variable size packet, with the actual size determined by the number of internal vLinks the source AS advertises and the number of neighbors it has. Each intra-network entry consists of 2 aNode IDs (each being a GUID of size 20 bytes [8]), 2 type masks for each aNode and 1 type mask for the vLink connecting the two aNodes, each of size 1 octet, and bandwidth, availability, latency and variability parameters of the vLink, each consisting of 1 octet. Therefore, each intra-network entry totals a size of 47 octets. Similarly, each border entry has a size of 25 octets. Therefore if a source AS has n internal entries and m border entries, its nSP will have a size of $(10 + n \times 47 + m \times 25)$ bytes. For example, assuming an AS exposes a topology with 10 internal vLinks and it has 5 neighbor ASes connected through 5 border vLinks, its nSP will be 605 bytes long.

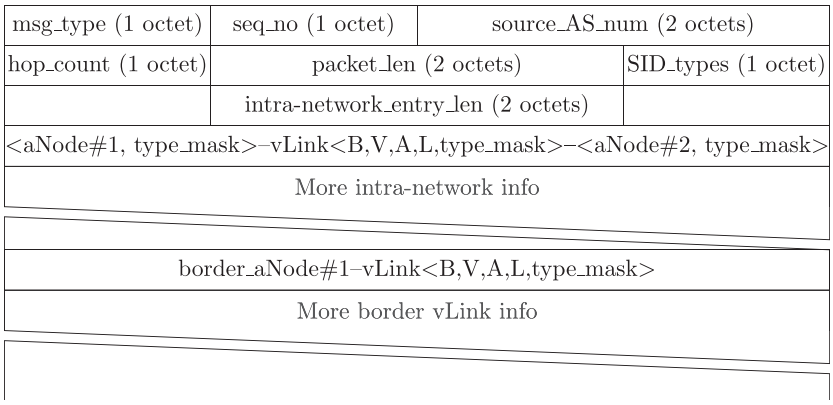


Fig. 2. Inter-AS route update structure exchanged through network state packets (nSPs) between border routers.

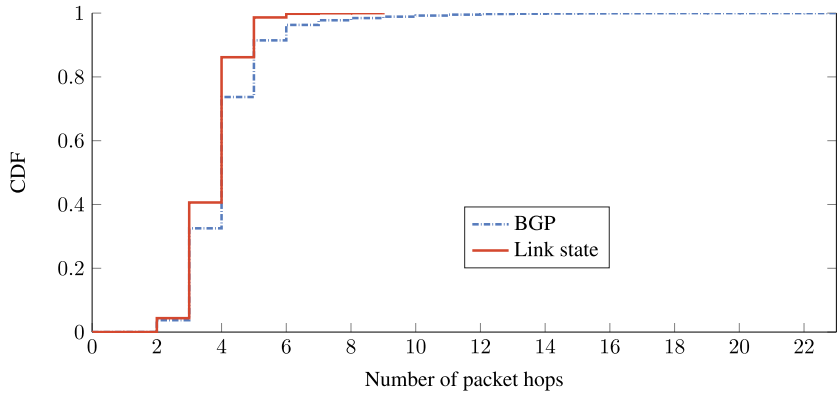


Fig. 3. CDF of AS path length of 2000 randomly chosen ASes in the Internet using BGP; Dijkstra based link state routing highlights the lower bound for the path lengths.

Path computation in EIR is based on link-state routing throughout the whole Internet. Understandably, a major concern for link state routing is its scalability and whether path vector routing such as that employed by BGP is sufficient. In order to answer this question, we performed an analysis of AS path lengths computed by BGP and compared it to a Dijkstra based link state routing on the complete Internet graph. For evaluation purposes we use a publicly available CAIDA dataset of 47,445 ASes and 200,812 inter-AS links [39]. All the BGP decision processes are evaluated in the C-BGP simulator, which is an efficient BGP solver, designed to handle large topologies [40], whereas Dijkstra is run on the same graph in our custom Python based simulator. Fig. 3 plots the cumulative distribution of path lengths computed by 2000 randomly chosen ASes to all other ASes in the graph. As seen from the plot, for the most part, BGP performance is comparable to link state routing (on an average 50% of the destinations are 4 AS hops away for both). However, BGP has a long tail, with some ASes being as far as 23 hops away. Note that the link state routing in this simulation does not take into account any policy based decisions, crucial to the operation of inter-domain routing. However the goal of this exercise is to motivate the fact that, if aggregated connectivity information in a global scale could be distributed across all ASes, there is a benefit in computing *shortest* paths based on different metrics and policies, instead of the conventional approach of advertising a single *best* path for each destination AS.

Note that, traditionally Dijkstra computation was considered an expensive operation. Single source Dijkstra computation on a graph of V vertices and E edges has a complexity of $O(E \log V)$. However, parallel implementations such as the algorithms proposed by Eager or Crauser [41,42] can bring down the complexity to upto $O(V \log V)$ and experimentation with parallel implementation has shown strong scaling properties even for single core processors [43]. Interestingly, EIR does not strictly enforce the use of Dijkstra, but rather provides a routing framework, where alternative routing algorithms can easily be implemented. nSPs provide the global view of the topology, which can be plugged into one or multiple algorithms at each border router to compute forwarding information bases.

A second major concern for flooding of link state messages to all ASes in the Internet is scalability and to address it, EIR uses a telescopic route dissemination mechanism, as described next.

3.2.3. Telescopic flooding of network state

Internal to an AS, routers exchange link information in the form of link state advertisements to build the network graph (refer to our earlier work on generalized storage-aware routing [9]). Border routers, upon receiving the link state advertisements from all the routers inside the AS, construct nSPs by combining the complete view of the internal network and the management enforced aNode topology with export policies of the AS. The nSPs are then announced to neighboring ASes. However, the border routers relay nSPs that originated from other ASes in a *telescopic manner*, which means that the relaying rate of a particular border router is determined by the distance, i.e. AS hop count, between the originator and the relaying border router. As a result, a router will get more frequent (hence up-to-date) routing updates from ASes that are closer to it. The term “telescopic” comes from the analogy of distant nodes seeing each other through the reverse-end of a telescope, i.e. they are visible but less clearly so, similar in concept to fish-eye state routing in ad-hoc networks [44].

Different telescopic functions can be defined by changing the relation between the hold-delay (time for which a border router holds a received nSP before relaying it to other neighbors) and the

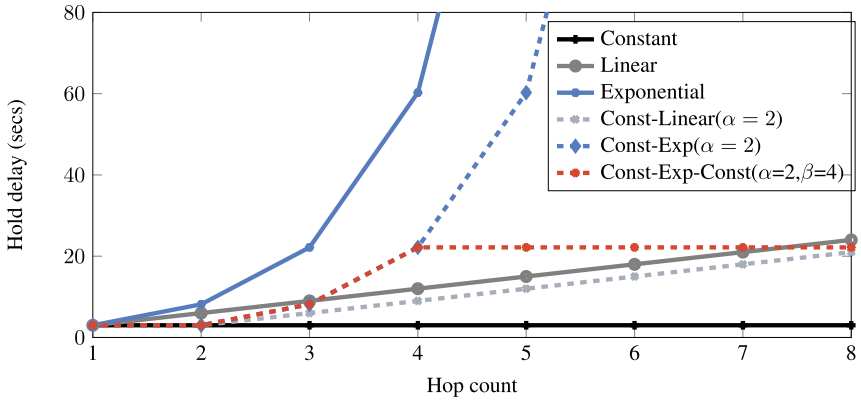


Fig. 4. Shape of telescopic functions of hop count vs. hold delay for $A = 3$.

hop-count. The goal of the function is to increase the hold time as the packet traverses farther and farther from the source, that is, the function should be monotonically increasing. We chose the following equations to characterize the telescopic functions in terms of the relation between hold-delay (denoted by y) and the hop count (denoted by x).

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

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

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

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

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

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

Note that although many other monotonically increasing functions can be defined, we chose a few simple ones, with a good mix of linear and exponential components. Fig. 4 plots each of these functions for a constant value of A. As seen from the plot, the goal is to reduce the effect of flooding of routing updates and slow the process down as you move farther away from the source. In this respect, the steeper the curve, higher the hold-delay of nSPs at each additional hop, and therefore, greater is the reduction in traffic overhead, but it also leads to a corresponding increase in the time taken by far-away nodes to receive an update. For example, constant and linear functions, would have small hold delays at each hop, but considerably higher overhead, whereas, the exponential function will exponentially reduce the overhead at the cost of higher time required for update propagation. We have looked at a range of values of the telescopic function parameters in order to find a reasonable tradeoff, as explained in Section 5.

3.2.4. Late-binding for mobility support

As a side effect of telescopic route update dissemination, network states that a network observed from far away could be obsolete during transit of a data packet and thus result in routing failure. To address this, EIR incorporates the additional design feature of in-network name-to-address binding during the transit of a packet. Late name-to-address binding serves as a fail-safe mechanism that allows routers to actively react to link variations and mobility of end nodes as well as networks. In particular, EIR makes use of a fast in-network name resolution through the GNRS [29] in order to retrieve the current network location of the destination.

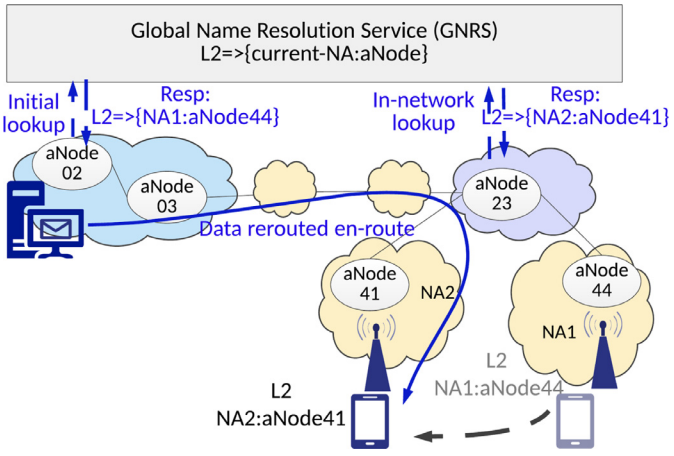


Fig. 5. Late-binding of data to counter stale network state update at a far-away node.

As shown in Fig. 5, network-address mapping of in-transit data can be looked up at an intermediate location within the network to properly route to a new location, without failure in delivery.

3.2.5. Label based path-setup

The EIR protocol has the provision for a border router initiated intra-domain path setup. In this procedure, the border routers compute paths based on bandwidth, link latency or any other local policy and inject forwarding table entries into internal routers along these paths using route-injection messages. Each of these paths are assigned a unique label. Since the labels are relevant only within a domain, management is not a major concern. At the internal routers, the computational complexity is reduced as simple label based switching occurs.

Transit network providers and large ISPs can utilize this label-based fast switching mechanism and set up dedicated routes for transit traffic. Note that the pre-computed paths follow the same set of aNodes exposed by the border routers in their nSPs. This enables any source to infer the end-to-end aNode path a packet would follow through each AS. In order to compute the paths, each border router utilizes the same transit policies and aNode level topology enforced by the network management authority. A pool of unique identifiers, generated by a local trusted naming service can be used by each border router to label the transit paths.

Compared to traditional label distribution protocols (LDPs), such as that employed by MPLS [45], this scheme is much simpler as it leverages on the intra-domain routing information base (RIB) for neighbor discovery. No LDP sessions are required to be main-

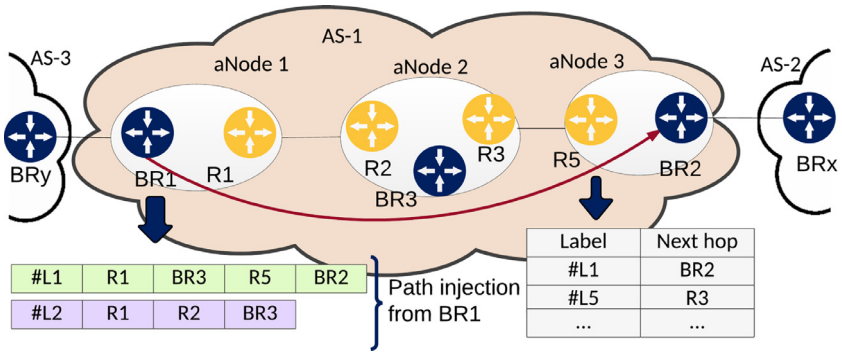


Fig. 6. Border routers generate paths with labels that are injected into the fast path table at the internal routers along the path.

tained across peers as well. As shown in the example scenario in Fig. 6, border router BR_1 chooses a set of paths to reach each of the other border routers based on transit policies. It forwards a route-injection message along the path with the generated label and the path info such that routers along the path can create a fast path entry in the forwarding table (as seen at router R_5). The advantages of this scheme are: (i) Internal routers do not need to perform any inter-domain route processing; and (ii) different types of policies can easily be realized by border routers by creating different paths and assigning labels for each, as explained in detail in Section 4. Similar to a RIB entry, we assume entries in a fast pathtable timeout and the border routers to periodically re-inject the path information for a long-lived transit path. However, scalability is not an issue, since this is a periodic intra-domain message per transit route, forwarded along the route based on the intra-domain forwarding table.

3.3. Supported routing algorithms

Next, we consider representative routing algorithms supported by the EIR framework described in Section 3.2. As mentioned earlier, nSPs include SIDs (service identifiers) which indicate the type of routing algorithms supported by each AS, and this SID is in turn expressed in a data packet to indicate the type of service desired. We assume that the SID space is finite but flexible enough to accommodate future routing policies and algorithms. For implementation purposes, we assigned 1 byte to the SID space, allowing 256 types of SIDs to be realizable. The interpretation of each SID is globally known, however, each AS may only support a subset of them. Note that, this is similar in spirit to classes of services (CoS) and end to end QoS proposed in BGP [46,47].The distinction between them is the way they are propagated. As mentioned earlier, in BGP, even if multiple paths with multiple values of a particular QoS parameter is received at an AS, a single 'best path' per QoS metric would be propagated to its peers. This significantly reduces path diversity and leads to a myopic view of the network.

Similar to EQ-BGP [46], in EIR, routers interpret the SID in the data packet to determine which of the several routing algorithms to use when forwarding. These algorithms can be grouped into 3 main categories:

3.3.1. Shortest path algorithm

EIR computes Dijkstra based global shortest paths using the available vLink parameters as weights. Since multiple coarse-grained parameters are available for each vLink, EIR runs a separate Dijkstra for each of these, resulting in multiple forwarding tables at each border router. On receiving a data packet, the border router looks up the appropriate forwarding table based on the SID expressed in the packet and forwards accordingly.

As shown in Fig. 7, all networks receive the SIDs supported by an AS through telescopic flooding of its nSP. Using this information, for example *NA1* can compute the aNode path and the corresponding ASes that will be traversed when using a certain metric and its corresponding SID. It can accordingly decide to use different paths for different kinds of traffic such as time-critical, reliable or best-effort delivery. This is fundamentally different from the way BGP calculates routes and forwards packets in two main aspects: (i) BGP is path vector based whereas EIR routes are global shortest paths and (ii) BGP routes are computed based on AS hops only, whereas EIR computes multiple routes based on each of the available vLink metrics (including AS hops). In addition route computation in EIR can check the business relationship policy attributes of vLinks in order to ensure that the "valley-free" property of end-to-end route is maintained [38], as explained further in Section 4. It is also potentially possible to define a SID that satisfies multiple forwarding criteria. For example, a SID could be defined for a time-critical emergency application scenario that requires high bandwidth and low latency. This in turn would require an efficient algorithm than then computes the forwarding information base at each router based on both the criteria. While outside the scope of this paper, there are several joint optimization techniques that could be used at each border router for path computation [48,49]. However a key challenge in such cases would be to ensure that the algorithm is fast enough to be run on an Internet-scale topology at every border router.

3.3.2. AS-level path computation

In addition to global shortest path routing, EIR also provides the functionality of using AS hop-counts for path computation. This allows network operators to realize traditional "hot-potato" or early-exit routing [50] where a transit network operator wants to reduce network resource usage by sending traffic out of its network through the "nearest" egress border router. As shown in Fig. 7, *NA2, NA3, NA5* broadcast their support for such routing which is leveraged by *NA1* for best-effort delivery.

3.3.3. Default routes

Finally, network operators have the option of falling back to a default routing table which is based on the inter-domain link delay or estimated time of transmission (ETT). This happens when data arrives at an ISP which is either not able to interpret the SID or does not support routes for that particular SID. This ensures that even if the route information is out-of-date due to en-route link or router outages or stale SID information from telescopic flooding, networks have a mechanism to route packets towards the destination, through a default path. Note that although an AS has the flexibility to aggregate, it should atleast broadcast the ETT of its inter-domain links, in order to compute routes for the default SID.

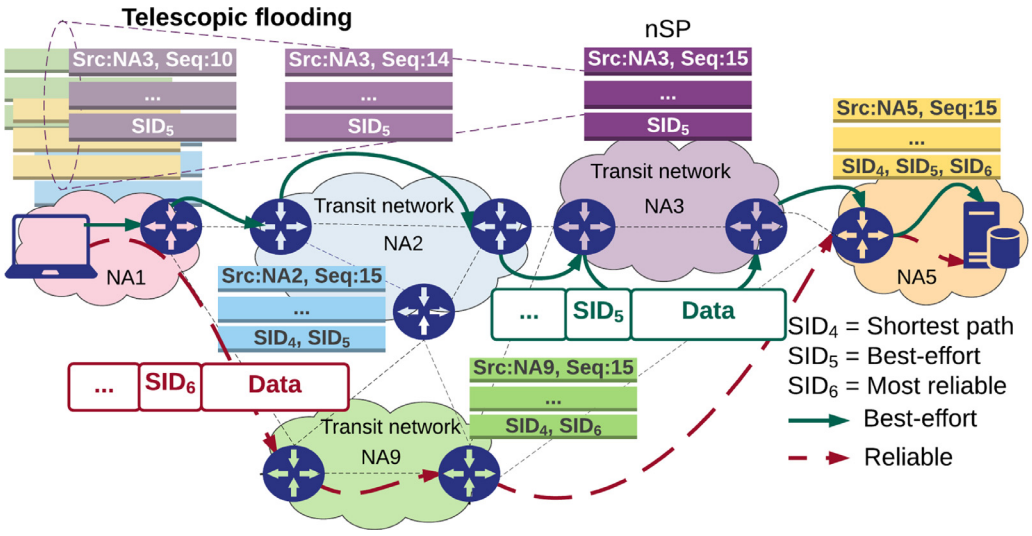


Fig. 7. Telescopic flooding and support for multiple shortest paths based on SIDs.

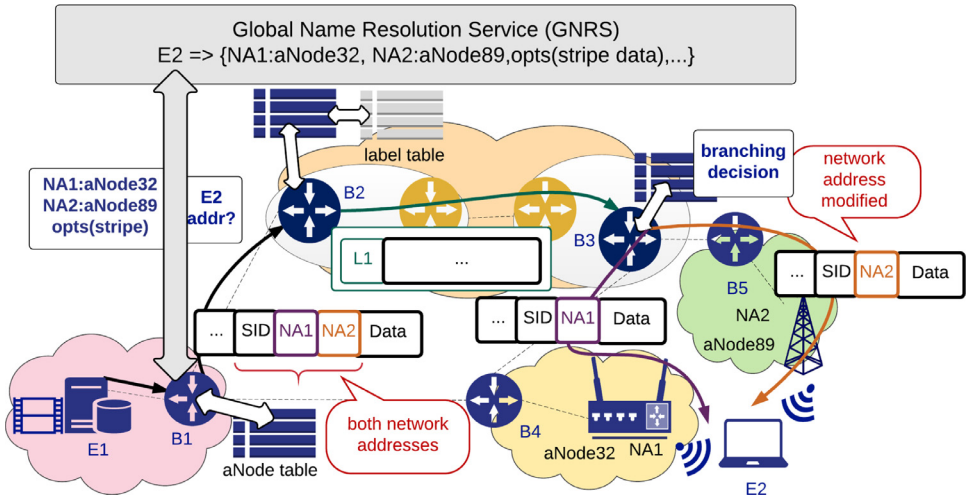


Fig. 8. A multi-homing scenario highlighting data delivery to client *E2* through two interfaces.

3.4. EIR routing examples

Bringing all of the discussed features of EIR together, in this section we walk through two examples to show how the features of EIR can be effectively utilized for client multi-homing and delay tolerant delivery.

3.4.1. Multi-homing scenario

Fig. 8 highlights a multi-homing scenario where a device with GUID *E2*, is connected to two different networks at the same time through WiFi and LTE and wishes to receive data across both the interfaces. The GNRS stores the up-to-date mapping of *E2*'s GUID to network addresses. Sender *E1* simply sends the data into the network with destination *E2*, where border router, *B1* does a GNRS lookup. It binds the data to *E2*'s current network addresses (*NA1* and *NA2)* and the appropriate SID (based on *E2*'s preference) as shown. Every border router looks at *NA1* and *NA2* and takes an independent decision based on their aNode forwarding table whether or not to bifurcate the data stream. As shown, *B1* decides to defer bifurcating to downstream routers. Data is forwarded internally through the transit network using label based forwarding. *B3* decides to bifurcate and accordingly modifies the packet header of the data sent across each network with the appropriate network address. The algorithm used to decide on branching could be a simple one such as "longest common path" in which only a single packet is forwarded as long as both *NA1* and *NA2* are on the shortest path. Note that mechanisms for multihoming also require a reliable transport for flow control as explained in our previous works [51,52].

3.4.2. Delay tolerant delivery

Next consider the scenario, where *E2* is mobile and would prefer to receive delay tolerant data as shown in Fig. 9. In this case, *B1* chooses an appropriate late binding point to temporarily store the data and rebind it to its new location whenever available. Accordingly, the network address is set to *NA5* of the late binding router, *B5* and the SID is set to late bind. The choice of late-binding node is an interesting problem, and would depend on several factors, including mobility rate, frequency of disconnection, type of data, storage availability at the late binding point, etc. The choice can be further improved if probabilistic information regarding *E2*'s future point of connection is available. As shown later in Section 5.2, one possible choice of late binding is to use the aNode with the highest degree along the path to the previously known location, thus providing multiple paths to nearby networks where the end-point

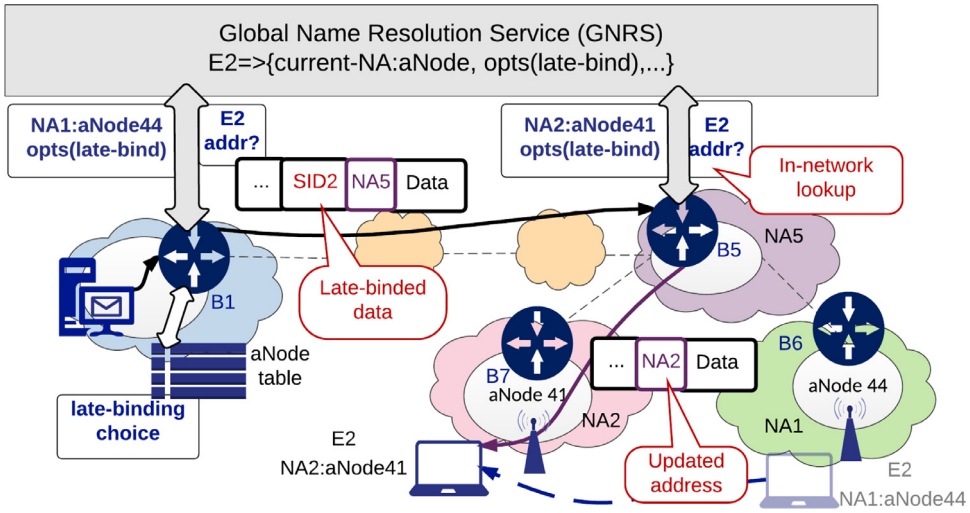


Fig. 9. Delay tolerant delivery to mobile client *E2* using late-binding.

may potentially reappear. In this case, *B5* queries the GNRS again to re-route data, as shown.

As seen from the above two examples, an in-network name-resolution service helps in supporting mobility, multihoming and other emerging network services. However, deploying a resolution service by itself is not sufficient, since all of these services require path diversity and path quality information, which is not provided by the current inter-domain routing. For example, best interface routing for multihoming and anycast requires knowledge of all the paths available. This in turn necessitates network states to be distributed globally, with the state including additional path quality information. Similarly routing to an intermediate router for late binding and subsequent routing to an end node requires path information from the intermediate ASes, so as to find an appropriate point in the network to send and rebind.

4. Policy specifications

Policy support is an integral part of any inter-domain routing protocol as network operators need to control the traffic flowing through their networks in a flexible manner that is consistent with business and performance objectives. In this section we discuss the range of existing inter-domain polices as well as a few of the emerging policy requirements that can be supported through the EIR framework.

4.1. Generic policy support using SIDs

EIR leverages on the SID space to define a set of user and network driven policies that can be supported at each AS. Examples of such policies are use “high-bandwidth path” or “low-latency path” or “most-reliable path”. Each of these SID intents would lead to the use of a different forwarding table at each AS, based on the corresponding vLink parameters. The SID space can be used to express more complicated policies such as “use high-bandwidth path if available, else, switch to the most-reliable path”. Note that we assume that the mapping of the “meaning” of a policy to its corresponding SID is known at every border router. In addition, the subset of SIDs supported at each AS could be different and this is expressed in the nSPs, as shown earlier in Fig. 7. This ensures that when a particular source or network expresses an intent, it has a means of verifying that an end-to-end path supporting that intent exists. As a fallback mechanism, the absence of an SID or the failure to support a specific SID in an incoming packet at a border router, would lead to the use of the default SID which corresponds to the ETT based shortest path through that AS.

The intent to use a particular forwarding metric could be both user-driven as well as network-driven. For example, edge networks could mark a certain subset of packets (based on metrics like host mobility, user-type, policy agreements with individual users) on behalf of the end hosts. Networks could also mark data with an SID indicating “least resource usage” that leads to the use of the AS-hop count based forwarding table at each hop. We realize that this brings up the fundamental question of “who controls the path?” For example, consider the scenario where an end host expresses an intent that conflicts with the networks operator’s traffic engineering policy. Unfortunately, this is out of scope of our current work and we assume that on conflict of SID’s, the ultimate decision is left upto individual network operators on how to route the packet. Failure to support an SID at a network will always lead to falling back to the default route through that network.

4.2. Support for business relationships

The nSPs convey coarse-grained information about the internal organization of the ASes as well as the inter-domain link quality between neighboring ASes. As mentioned earlier, the vLinks that represent inter-domain links between ASes are tagged with four main business-relationship indicators, namely, “customer-to-provider”, “provider-to-customer”, “peer-to-peer” or “backup”. Note that BGP does not expose business relationships globally, which leads to convergence loops as pointed out in [53]. Further, workarounds have been proposed [37,38] to infer such relationships crucial for route convergence. Using the business relationship information of vLinks, in EIR, the route computation algorithm is a modified-Dijkstra, to ensure that the shortest path computed is “valley-free”, that is, it does not violate the universal economic best-practices [38].

4.3. Dynamic traffic engineering

EIR also provides the flexibility for ASes to perform dynamic traffic engineering. Standard “hot-potato” style traffic engineering [50] can be easily reflected using the AS-hop count based routing table. Networks can tag packets with an SID expressing “least-resource” in order to indicate the use of the AS-hop count based forwarding. This in turn will result in the packet exiting an AS to a neighboring AS as quickly as possible. Note that the default

Table 1
Comparative analysis of policy support in EIR, Pathlet and BGP.

Type	Policy	BGP	Pathlet	EIR	Note for EIR
Business relationship	Local Pref	✓	✓	✓	Bias vLink metrics
	Community attribute	✓	✓	✓	Tag vLinks with relationship
Traffic engineering	Hot potato routing	✓	X	✓	Use AS hop count forwarding
	Load balancing	✓	✓	✓	Route injection
	AS path inflation	✓	Not reqd	Not reqd	Global view of end-to-end paths
Scalability	Prefix aggregation	✓	✓	X	nSP aggregation not supported
	Default routes	✓	✓	✓	Use of ETT forwarding table
	Route flap damping	✓	?	✓	Modify telescopic flooding
Others	User-initiated	X	✓	✓	Use of SIDs
	Network-initiated	X	✓	✓	Use of SIDs
	Global roaming	X	X	✓	Use of GNRS
	Blacklisting	X	✓	✓	Stitching of inter-domain tunnels

ETT based route computation would lead to "cold-potato" routing, where data would always egress an AS through the ETT-based shortest path. In addition, EIR also allows ASes to dynamically change their aNode level topology to achieve real-time traffic engineering. For example, routers from a congested part of the network could be excluded from the aNode graph formation and not broadcasted in the nSP. Similarly, link failures could be reflected in the change of vLink parameters or exclusion of certain set of vLinks.

4.4. GNRS-assisted global roaming agreements

Supporting global roaming for end hosts in BGP is challenging as this requires not only initial policy agreements among the participating ASes, but also a means of tracking and verifying users subscribed to each. There are partial and limited deployments of such polices, such as Eduroam [54] and Google's Fi Project [12]. In EIR, ASes can easily enter into global roaming agreements with each other and form an AS roaming group, which is then assigned a unique GUID. The mapping of the group GUID to the participating ASes is maintained in the GNRS. When a participating domain's client migrates to and associates with the another participating domain, the AS 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 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 domain's client.

4.5. Inter-AS agreements for tunnel setup

We have also explored the concept of extension of intra-domain path setup across multiple domains through a GNRS-assisted tunnel maintenance [55]. This is useful for enforcing policies such as "blacklisting". If an AS does not want its traffic to flow through a subset of ASes, it can explicitly do so by stitching up multiple tunnels across ASes in its "whitelist". This is also helpful for emerging content delivery network use-cases, such as Netflix OpenConnect CDN [56], where a content delivery network would want to enter into agreements with multiple ASes along the path to maintain QoS guarantees and thereby stitch a dedicated end-to-end transit path for traffic flowing between its data-centers and its customers.

Table 1 provides a summary of comparison of policies currently supported in BGP and the ability of EIR to emulate them (refer to [2] for detailed description of BGP supported policies). In addition, since Pathlet [36] evaluates itself with contemporary routing protocols [25,57,58] and supports a wide variety of routing policies, we highlight the key distinguishing policy support features between EIR and Pathlet. As seen from the table, most of the existing as well as emerging policy based control can be supported through EIR. We note that baseline EIR does not support aggregation of nSPs from different neighbors, the counter-part of BGPs prefix aggregation. However aggregation of nSPs is not crucial for the protocol performance and overhead studies using EIR indicate that the global overhead of nSP propagation is negligible compared to the total Internet traffic, as explained in detail in Section 5. Also note that since EIR and Pathlet follow the similar principle of representing network connectivity in terms of aggregated topology abstractions, it can use a combination of Local-Transit style and BGP style policies to emulate many of the contemporary routing protocols.

5. Evaluation

In this section, we evaluate the EIR protocol in terms of scalability and mobility service performance through a large-scale Click software router based prototype evaluation and an Internet-scale simulation study. Section 5.1 describes the setup and insights from an Internet scale simulation effort, which is followed by Section 5.2, that describes the implementation details. Finally, we also describe the results from our in-depth mobility study experiments based on the prototype implementation.

5.1. Overhead and scalability studies

One of the main challenges of propagating link state routing information throughout the Internet is scalability. We have performed extensive simulations in network-simulator (NS3) and our custom Python-based simulator to analyze the overhead and settling time for different telescopic function and parameters (refer to Section 3.2 for details) that provide good performance tradeoffs between overhead and scalability. Since it is infeasible to have a packet-level simulation of the complete AS-level graph of the current Internet in NS3, we used a scaled-down topology of 200 nodes, which mimics the AS-level structure of the Internet. We first extract the degree distribution and the latency distribution of the measured AS-level graph from the DIMES database [59]. Next, we build a Jellyfish topology [60] consisting of 200 nodes by matching the distribution of ASes in each layer and the proportion of links between layers, to the values ascertained from the DIMES dataset. Siganos et al. [60] show that the jellyfish topology can be used as an accurate conceptual model for the internet topology and is able to capture most of its graphical properties. Real-world measured latency values from DIMES are then used to assign link delays in our topology in a manner that preserves the latency distribution. Fig. 10 compares the CDF of the latency values used in our topology with that of the complete AS-level graph obtained from DIMES.

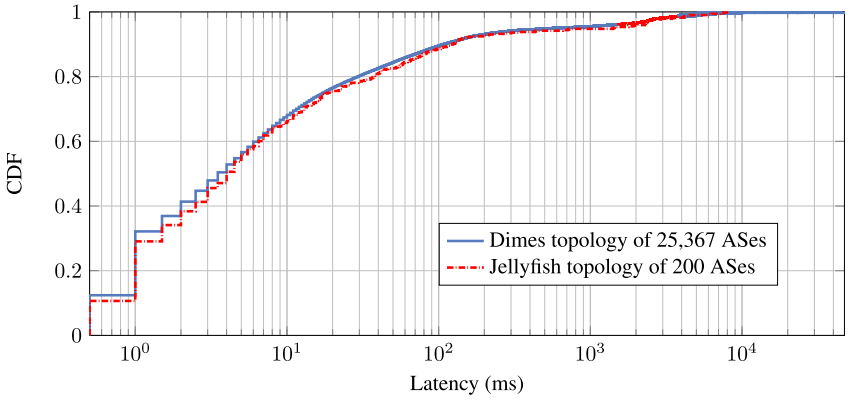


Fig. 10. CDF of inter-AS latency in the Dimes topology and in our 200 node synthetic topology.

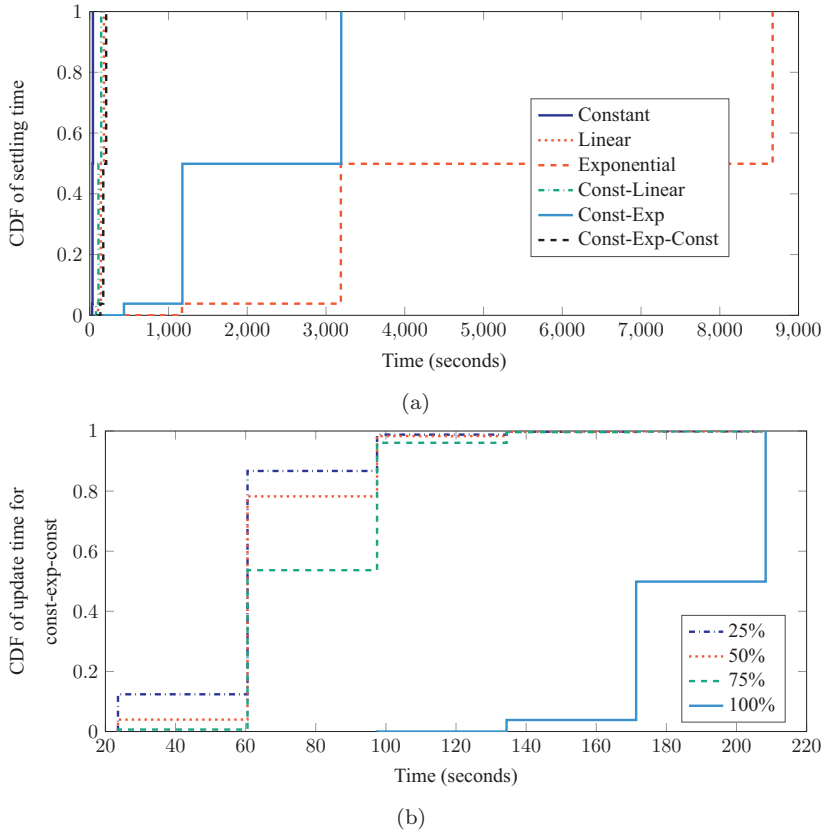


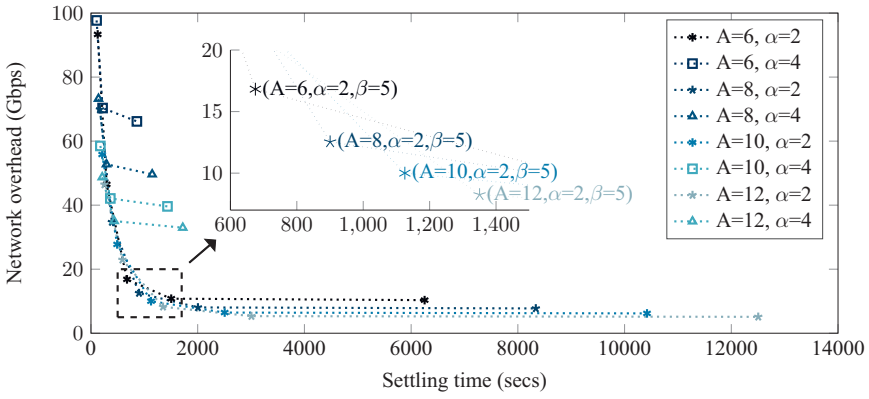
Fig. 11. (a) CDF of receiving an update at each AS for different types of telescopic functions, and, (b) that with different percentile of recipients for const-exp-const telescopic function.

5.1.1. Worst case update time

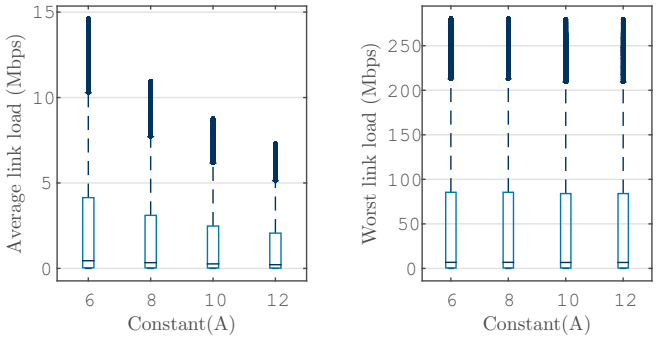
The six different monotonically increasing telescopic functions defined earlier, were simulated in NS-3 and the settling time for each was analyzed in order to choose a telescopic function suited for an Internet-scale topology. Settling time is defined as the time required for an update to propagate throughout the network. Fig. 11(a) shows the cumulative distribution of the time at which each AS receives an update following its generation. As seen from the plot, other than constant-exponential and exponential functions, the others converge in less than 250 seconds for $< A = 5, \alpha = 2, \beta = 4 >$ in equations defined in Section 3.2.3. Note that the exact convergence time would vary based on the parameters *A*, α, β and the nsP generation periodicity, however, this plot shows us the trend of settling times for representative values of the parameters. It also highlights the fact that the constant-exponential-constant telescopic function which has a comparatively lower overhead than a constant or linear telescopic function, provides *reasonable* settling time. Fig. 11(b) further shows that even though the worst case settling time for constant-exponential-constant function is about 220 seconds, 75% of the nodes received the update in less than 150 seconds. Note that since EIR uses link-state instead of path vector, there will be no path divergence issues as possibly found in BGP.

5.1.2. Internet-scale overhead

In order to analyze the tradeoff between routing overhead and settling time further, we simulated one of the complete AS-level datasets from the year 2013 available at CAIDA [39] in our custom simulator. This dataset is composed of 47,445 ASes and 200,812 inter-AS links using which, we simulate the generation and prop-



(a)



(b)

Fig. 12. (a) Overhead vs. settling time for different parameters of the constant-exponential-constant telescopic function, and, (b) Average and worst case load on links for values that provide a good tradeoff.

agation of nSPs across the network. Fig. 12(a) shows the global routing overhead vs. settling time for different values of the parameters of the constant-exponential-constant telescopic function. Each curve is for a fixed A and α, as shown in the legend, with $\beta \in \{3 \rightarrow 8\}$, $\beta \neq \alpha$. As seen from the figure, there are a subset of values ($\alpha = 2, \beta = 5$ and $A \in \{6, 8, 10, 12\}$) that have low overhead as well as low settling time which can be used for setting the telescopic function parameters. Notice, that the worst case network overhead is about 100 Gbps, assuming 1000 byte nSPs. This is a negligible fraction of the total Internet traffic of $\sim$182 Tbps as of 2014 [3]. Fig. 12(b) further plots the average and worst case link load for these subset of parameter values. As seen from the plot, the worst case load on a link was about 300 Mbps, but on average link load was less than 15 Mbps. Note that although the average link load reduces with increasing in periodicity of nSP, the worst case link load is almost constant, as the latter is based on the instantaneous link load, which is not affected by the periodicity.

5.1.3. Link failure analysis

A key concern for any routing protocol is handling transient conditions, either due to failures of routers and links or link flapping. To understand the transient behaviour of EIR, we simulated a vLink failure on a small topology in NS-3, as highlighted in the bottom-right of Fig. 13. In this simulation, a client connected to *NA*6 is downloading a large file from a back-end server in *NA*1. All physical links were set at 1Gbps with 10 millisecond latency, and each vLink was assumed to be a direct mapping of the underlying physical link. Each of the *NA*s shown, are representative of an aNode in the topology.

Data delivery starts at 30 seconds (assuming the aNode forwarding tables have converged) and the path followed is $< server \rightarrow NA1 \rightarrow NA2 \rightarrow NA3 \rightarrow NA5 \rightarrow NA6 \rightarrow client >$. At the 35th second, we simulate failure of the vLink $NA5 \rightarrow NA6$. We plot the throughput at the client per second in Mbps, and as shown in Fig. 13, throughput immediately goes to zero. nSPs are propagated periodically and aNode forwarding tables are recomputed every time when a new nSP is received. In this experiment, nSPs are advertised every 5 seconds and therefore, until the 40th second, the information of the link failure is not propagated in the control plane. MobilityFirst uses a hop-by-hop reliable transport in the link layer [61], which is particularly beneficial in this case, since data continues to be pushed towards the destination and gets temporarily stored at *NA*5. At the 40th second, *NA*5 and *NA*6 both generate a nSP with the updated vLink information and forward them to their neighbors. On receiving this updated nSP, *NA*4 is now able to compute an alternate route, whereas, *NA*5 also computes the same alternate route, based on the nSP it receives from *NA*3. Data delivery therefore resumes around the 40th second, even though routing tables at *NA*1 and *NA*2 have not converged. Stored data gets rerouted through the alternate path, resulting in a temporary increase in the client throughput.

This experiment, although simple in essence, highlights two key features of EIR: (i) Path diversity helps in transient conditions. If EIR followed a traditional BGP style dissemination approach, the alternate path information would have taken much longer to be available at the nodes undergoing the link failure; and,(ii) It is not necessary for every routing table to converge in order to resume data flow, due to the hop-by-hop store-and-forward delivery ap-

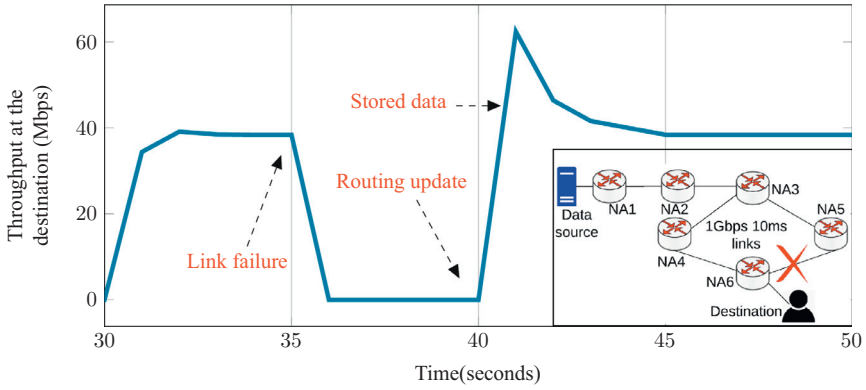


Fig. 13. Data delivery to an end-host, with core link failure in EIR.

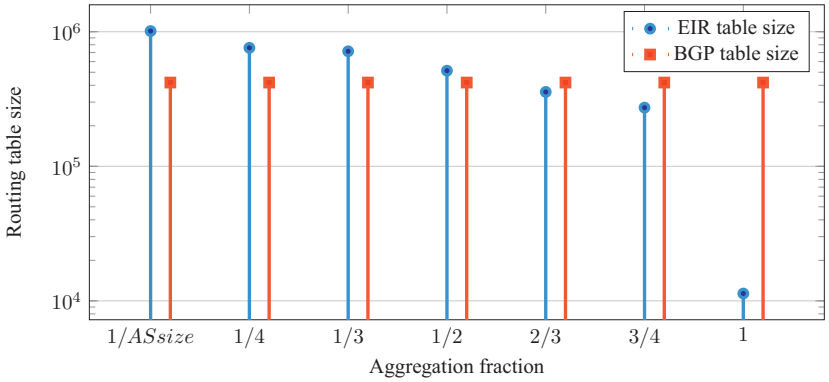


Fig. 14. Inter-domain table size at each border router for different levels of aggregation.

proach of MobilityFirst. This is further highlighted in our network mobility experiment, described later in Section 5.2.

5.1.4. Routing table size

Maintenance of a global aNode based topology at each border router would imply that the inter-domain forwarding table size to 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 point-of-presence (PoP) topology of $\sim$ 22,000 ASes. After parsing for intra and inter-domain links and removing clusters from graph that were not connected (due to incomplete dataset), we evaluated global routing table size for a graph of 11,340 ASes with their connected PoP topology. As explained earlier, EIR allows a flexible aggregation scheme, wherein each AS can independently decide on the number, types and properties of aNodes they wish to publish in their nSP. We define aggregation as a fraction varying between 1/*size* and 1, where *size* is the number of PoPs belonging to that AS. A value of 1/*size* indicates, every PoP in an AS is advertised as a separate aNode, whereas a fraction of 1 indicates an entire AS is a single aNode. A simple case to evaluate would be to consider all ASes to aggregate uniformly, that is, every AS chooses the same aggregation fraction, which is shown in Fig. 14. In the figure, the blue lines plotted in log scale, show the inter-domain table size in terms of the number of entries at each border router with varying levels of aggregation. The red lines show the average BGP table size as reported by CIDR [62] for the same month and year. 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, which is much larger than the total number of ASes in the internet. As seen from the plot, even though EIR maintains a global view of the network, aNode table sizes are comparable to current BGP table sizes, for moderate levels of aggregation.

In a realistic scenario, we expect ASes to not follow a uniform aggregation scheme and therefore the table sizes would vary, depending on how many aNodes each AS advertises. However, if we assume each AS to randomly choose an aggregation fraction, in the above experiment, on an average, the aggregation fraction would be close to 1/2 and therefore the aNode table sizes will be close to 510K, which is slightly larger than the corresponding BGP table size. In reality, however, we expect most ISPs to choose a relatively high aggregation factor and the global table sizes to lie towards the right of the plot.

5.1.5. Memory requirements

EIR also requires each border router to store the latest copy of the network state packet received from all the other ASes in the network. As explained earlier in Section 3.2.2, each nSP packet size is different, based on the aggregation policies of the source AS and the number of inter-domain neighbors it has. However, assuming a maximum packet length of 4096 bytes (same as the maximum BGP packet size [1]) and considering the total number of ASes in the network to be 57,840 (as published by CIDR for June 2017 [62]), this would require a memory size of $4096 \times 57,840 = 237$ MB. Similarly, BGP update packet sizes are also variable and each peer needs to generate a separate update packet for every unique path. For example, for 672,522 destination prefixes (as of June 2017 [62]), there could be as many as 50,000 unique paths from a peer. While it is difficult to calculate the exact memory requirements, as Cisco points out, a minimum memory size of 512 MB is recommended for each BGP router [63] which should also be sufficient for the deployment of EIR.

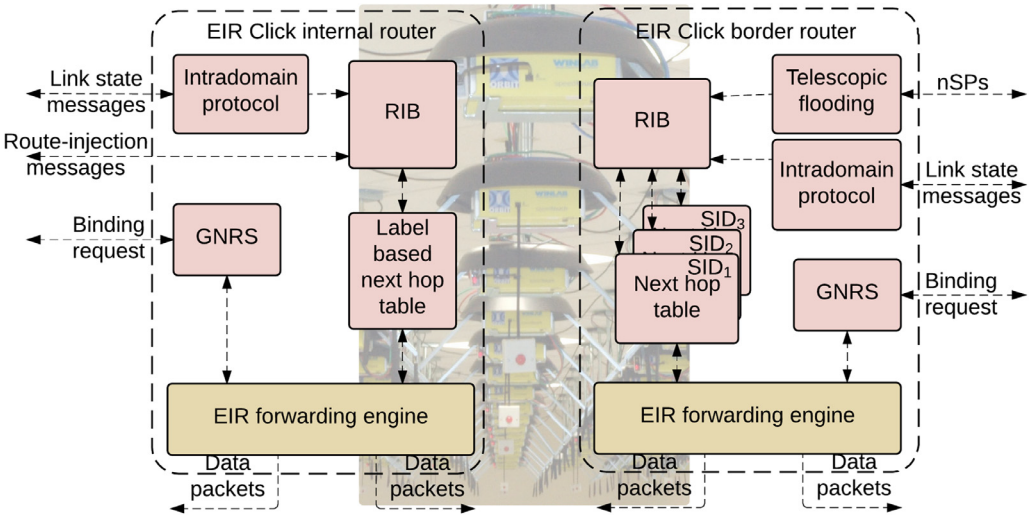


Fig. 15. Overview of the Click router prototype for border and internal routers.

5.2. Prototype evaluation

To measure the performance and implementation feasibility of EIR, we have built a prototype router (based on the Click modular router design [64]) and evaluated it using the ORBIT testbed [65]. Our router consists of two components: control plane and data plane. As shown in Fig. 15, border routers send and forward nSPs as per the specifications outlined in Section 3.2 through the control plane, whereas internal routers simply use label based paths set up by the border routers. In addition, all routers exchange intra-domain link probes and link state updates through the control plane to build up the intra-domain topology using MobilityFirst's generalized storage-aware routing (GSTAR) [9]. GSTAR is a link state routing protocol, where link state messages carry the estimated time of transmission (ETTs) of intra-domain links. This in turn is utilized by the border routers in EIR to build aggregated vLinks. Our current prototype only computes the latency metric of vLinks. However, in future, we plan to augment GSTAR with additional parameters in order to compute bandwidth, availability and variability of vLinks.

One of the key aspects of EIR is its support for mobility, both for individual devices as well as for networks as a whole. In order to evaluate such scenarios, we used a realistic inter-domain topology and a probabilistic mobility transition matrix which is briefly described below. This was used with the Click software prototype implementation on the ORBIT testbed for end-user and edge-network mobility evaluations.

5.2.1. Topology generation and probabilistic mobility

We start with the previously described CAIDA dataset from 2012 with PoP-level topologies, 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. a user 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. Since AS tier information was not available in the dataset, a random choice ensures that we get a good mix of ASes from different tiers. Given the PoP-level topology, a corresponding aNode topology is developed for each of the participating ASes based on geographical proximity, that is, PoPs belonging to the same AS and located close to each other are clustered to the same aNode. This lead to a final inter-domain EIR topology of 53 aNodes.

Table 2
Probabilistic transition for user mobility.

Basic parameters:	
Z	avg number of network transitions/s
K	total number of network transitions
T	granularity of transition (s)
r	avg distance to neighbors (m)
s	avg speed of mobility (m/s)
$w = s/r$	average transition rate/s
α	probability of transition to a network
Transition probability from node N_j:	
$\alpha(wT)/N_j$	to each of N_j's neighbors
$(1-\alpha)(ZT)/K$	to each of K non-neighbors

In order to realistically model inter-domain mobility our transition probability matrix takes into account the following factors: (i) Local mobility within a certain radius (denoted as r), with equal probability of transition to all aNodes within the "local boundary"; (ii) biased transitions between aNodes belonging to the same AS within the local boundary, 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; and, (iii) biased transitions (determined by α) to a random, k number of "macro mobility" points based on the average number of networks visited by a user per day [66]. Table 2 explains the transition probability computations.

5.2.2. Mobility support through late binding

Based on the San Francisco topology and a mobility matrix generated for a typical mobile user, we analyzed the path stretch that is incurred with and without late binding. Path stretch is defined as the number of hops traversed by a packet to the number of hops across the shortest path between the source and the destination. Note that without late binding, failure in delivery would result in rebinding through a GNRS re-lookup at the previous point of attachment. On the other hand, late-binding would re-bind the network address at an intermediate router, as explained in Section 3.4. The late-binding algorithm for this evaluation chooses the aNode with the highest degree along the path as the late-binding point. The intuition behind this logic is that a highly connected node would have shorter path stretch to the next point of association for the user.

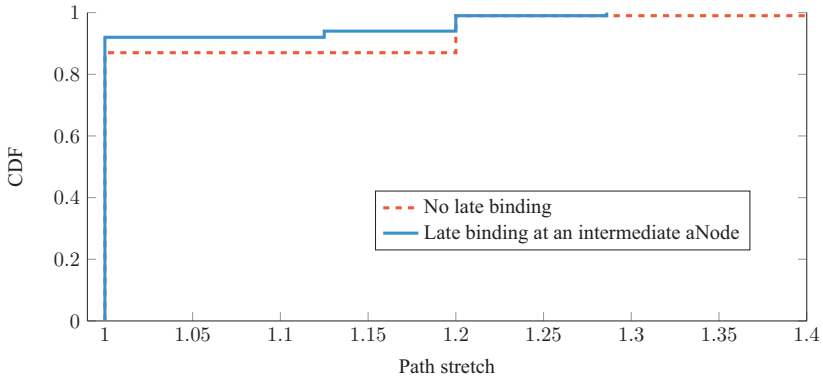


Fig. 16. CDF of path stretch with and without late binding for end-user mobility.

Fig. 16 highlights the improvement in path-stretch when packets are late binded along the way. Notice that the solid blue and the dotted red curves are fairly close since once a mobile node 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. Also note that MobilityFirst data packets carry both the GUID and the network address in its header [8]. Therefore, lookups do not need to be done at every aNode. Once a lookup is done, the up-to-date address is reflected in the packet to reduce further lookups. GNRS responses can also be temporarily cached at a router, such that subsequent packets do need a lookup. However, in our experiment, every packet incurred a GNRS server (located 1 hop away) lookup roundtrip delay. Previous works have looked at how to distribute GNRS servers in order to further reduce this lookup latency [29,30].

In future evaluations, we 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.

5.2.3. Network mobility

Based on the same topology, we evaluate an use-case of network mobility, where the evaluation scenario consists of a mobile aNode connecting to different ASes as it moves and a source in a distant AS trying to deliver data to the mobile network. To realistically model network mobility, we use actual bus traces from San Francisco Municipal Transit system [67]. We measure the data delivery failure rate for different routing update rates, where failure rate is defined as the ratio of number of packets not received to the number of packets sent. Since through rebinding and delay tolerant delivery supported by the MobilityFirst architecture, packets will eventually be delivered at any mobile node, for the purpose of this experiment, we calculate failure at the previous point of association, before they are re-routed to the next.

Fig. 17 shows the delivery rate at mobile buses on 9 randomly picked routes, for different update intervals of the telescopic function. Similar to our previous experiments, the values of α and β were kept constant at 2 and 5 respectively as they provided *reasonable* overhead and settling time, based on our Internet-scale simulation. We also looked at the number of AS transitions for each trace which determines the failure rate and observed that 2 hops AS transition tend to dominate these mobility events. Of the 9 randomly picked traces, trace 1 resulted in a scenario that had primarily 1-hop transitions and hence the data delivery rate is almost similar for different telescopic hold time. Whereas in the other traces, there are a few transitions to ASes that are multiple AS-hops away. Consequently, the failure rate increases with A as the reachability to the mobile aNode is not known for a longer period of time due to the telescopic hold function of the nSPs.

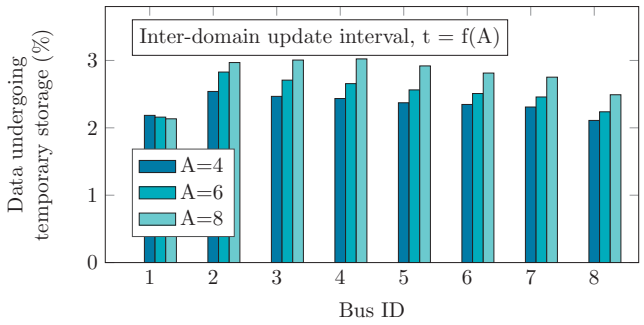


Fig. 17. Data delivery failure rate for different telescopic update intervals for network mobility.

6. 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.

6.1. Extensions to BGP

Proposals such as path splicing [68] and route-deflections [69] 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. Yang and Wetherall [69] provide a limited choice of paths, whereas Motiwala et al. [68] provide path diversity, however does not address the issue of scalability. MIRO [58] 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. Wang and Gao [70] propose similar fail-over path set-up techniques in order to reduce disconnectivities on link failures. In contrast, Verkaik et al. [71] propose a service plane with active servers that act in conjunction with BGP in order to perform dynamic route control, which requires additional resources and results in increased control overhead.

Recent works on BGP, have introduced the mechanism of advertising multiple paths for the same address prefix [72] as well as defining and advertising multiple classes of services (CoS) [46]. Multipath BGP [72] is akin to the EIR design choice of advertising multiple paths in each AS. However, EIR goes one step farther by providing path diversity in the form of aggregated intra-domain routes as well. The concept of advertising multiple CoS in EQ-BGP [46] is also similar in spirit to EIR's SID space. In this de-

sign however, there is no concept of multiple paths being advertised for each CoS. The authors propose to extend BGP reachability messages to include CoS information of each AS. Q-BGP [73] proposes the same ideas as EQ-BGP but by defining new update messages to disseminate QoS classes. In EIR, we have adopted the former design choice, in order to keep the routing control overhead tractable. In this respect, EIR design is conceptually a union of EQ-BGP (multiple service classes) and MP-BGP (path diversity).

6.2. 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 split approach (LISP) [25], tunnels are set up between egress points in an AS, similar to MPLS [45], 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 [36]. Recent standardization efforts have looked into segment routing [74] which also proposes abstracting the network into segments and then choosing appropriate segments at the source to build and end-to-end path. However, our path-selection approach is quite different from that of Pathlet and segment routing, both of which perform loose source routing. Instead, EIR provides the flexibility to choose end-to-end routes to both end-hosts as well as intermediate ASes through the use of SIDs. EIR also utilizes the name resolution service (GNRS) of MobilityFirst to perform dynamic re-routing of in-transit packets during mobility and changing network conditions, which is difficult to do in source-based routing. HLP [75] uses a hybrid link-state and path-vector approach where provider-customer sub-graphs use link-state routing for path-diversity and peers 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. NIRA [57] 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.

7. Conclusions

In this paper, we have proposed the edge-aware inter-domain (EIR) routing protocol as a potential solution for inter-network routing in the future mobile Internet. The proposed architecture has been shown to provide improved support and flexibility for routing to wireless devices, network-assisted multipath routing, routing to multiple interfaces (multi-homing) and service anycast. Our results show that even with increased expressiveness of network structure and node/link properties, the protocol can be designed to have reasonably small overhead via telescopic dissemination of the nSPs. Further, prototype evaluations using Click software routers on the ORBIT testbed show proof-of-concept level feasibility. Experimental results for selected use-cases show good service-level performance can be achieved in highly mobile scenarios. For further work, we plan to deploy EIR on the GENI large scale testbed to evaluate service capabilities and performance in more realistic global network scenarios.

References

[1] Y. Rekhter, T. Li, S. Hares, A border gateway protocol 4 (bgp-4), IETF RFC 4271, 2005.
[2] M. Caesar, J. Rexford, BGP routing policies in ISP networks, IEEE Netw. 19 (6) (2005) 5–11.
[3] C.V.N. Index, Global Mobile Data Traffic Forecast Update, 2014–2019, 2015. White Paper, February.
[4] Ericsson, 5G Radio Access-Research and Vision, 2013. White Paper, June 2013.
[5] M. Satyanarayanan, Mobile computing: the next decade, ACM SIGMOBILE Mob. Comput. Commun. Rev. 15 (2) (2011) 2–10.
[6] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, et al., Named data networking, ACM SIGCOMM Comput. Commun. Rev. 44 (3) (2014) 66–73.
[7] D. Han, A. Anand, F.R. Dogar, B. Li, H. Lim, M. Machado, A. Mukundan, W. Wu, A. Akella, D.G. Andersen, et al., Xia: efficient support for evolvable internetworking., NSDI, 12, 2012. 23–23.
[8] D. Raychaudhuri, K. Nagaraja, A. Venkataramani, Mobilityfirst: a robust and trustworthy mobility-centric architecture for the future internet, ACM SIGMOBILE Mob. Comput. Commun. Rev. 16 (3) (2012) 2–13.
[9] S.C. Nelson, G. Bhanage, D. Raychaudhuri, Gstar: generalized storage-aware routing for mobilityfirst in the future mobile internet, in: Proceedings of the Sixth International Workshop on MobiArch, ACM, 2011, pp. 19–24.
[10] A. Hoque, S.O. Amin, A. Alyyan, B. Zhang, L. Zhang, L. Wang, Nlsr: named-data link state routing protocol, in: Proceedings of the 3rd ACM SIGCOMM Workshop on Information-Centric Networking, ACM, 2013, pp. 15–20.
[11] A. Gupta, L. Vanbever, M. Shahbaz, S.P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, E. Katz-Bassett, Sdx: a software defined internet exchange, ACM SIGCOMM Comput. Commun. Rev. 44 (4) (2015) 551–562.
[12] Project Fi, (https://fi.google.com) Accessed: 2017-06-23.
[13] Republic Wireless, (https://republicwireless.com/) Accessed: 2017-06-23.
[14] P. Rodriguez, R. Chakravorty, J. Chesterfield, I. Pratt, S. Banerjee, Mar: a commuter router infrastructure for the mobile internet, in: Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services, ACM, 2004, pp. 217–230.
[15] D.S. Phatak, 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, vol. 2, IEEE, 2002, pp. 773–781.
[16] J.R. Iyengar, P.D. Amer, R. Stewart, Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths, Netw. IEEE/ACM Trans. 14 (5) (2006) 951–964.
[17] B. Augustin, B. Krishnamurthy, W. Willinger, Ixps: mapped? in: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference, ACM, 2009, pp. 336–349.
[18] Long-Range 802.11n (5GHz) Wi-Fi Backhaul, (https://www.ruckuswireless.com/products/access-points/zoneflex-outdoor) Accessed: 2017-06-23.
[19] Project Loon, (https://x.company/loon/) Accessed: 2017-06-23.
[20] B. Abarbanel, Implementing global network mobility using bgp, NANOG Presentation, 2004.
[21] H. Ballani, P. Francis, Towards a global IP anycast service, ACM SIGCOMM Comput. Commun. Rev. 35 (2005) 301–312. ACM.
[22] J.S. Silva, T. Camilo, A. Rodrigues, M. Silva, F. Gaudencio, F. Boavida, Multicast in wireless sensor networks the next step, Proceeding of IEEE ISWPC 2007, IEEE, 2007.
[23] C.R.K.D. Bates, T, Y. Rekhter, Multiprotocol Extensions for BGP-4, RFC 4760(2007).
[24] S. Saroiu, K.P. Gummadi, R.J. Dunn, S.D. Gribble, H.M. Levy, An analysis of internet content delivery systems, ACM SIGOPS Ope. Syst. Rev. 36 (SI) (2002) 315–327.
[25] D. Meyer, D. Lewis, The Locator/ID Separation Protocol (LISP), Technical Report, IETF RFC, 2013.
[26] R. Moskowitz, P. Nikander, P. Jokela, T. Henderson, Host Identity Protocol, RFC5201, 2008.
[27] D.G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, S. Shenker, Accountable internet protocol (AIP), ACM SIGCOMM Comput. Commun. Rev. 38 (2008) 339–350. ACM.
[28] D. Saucez, L. Iannone, B. Donnet, A first measurement look at the deployment and evolution of the locator/id separation protocol, ACM SIGCOMM Comput. Commun. Rev. 43 (2) (2013) 37–43.
[29] T. Vu, et al., DMap: a shared hosting scheme for dynamic identifier to locator mappings in the global internet, in: Proceedings of ICDCS '12, 2012.
[30] A. Sharma, X. Tie, H. Uppal, A. Venkataramani, D. Westbrook, A. Yadav, A global name service for a highly mobile internetwork, ACM SIGCOMM Comput. Commun. Rev. 44 (2014) 247–258. ACM.
[31] C. Kim, M. Caesar, J. Rexford, Floodless in seattle: a scalable ethernet architecture for large enterprises, ACM SIGCOMM Comput. Commun. Rev. 38 (2008) 3–14. ACM.
[32] Y. Wang, I. Avramopoulos, J. Rexford, Design for configurability: rethinking interdomain routing policies from the ground up, Sel. Areas Commun. IEEE J. 27 (3) (2009) 336–348.
[33] D. Walton, A. Retana, E. Chen, J. Scudder, Advertisement of multiple paths in BGP, IETF RFC 7911, 2016.
[34] A.F.T. Committee, Private Network-Network Interface Specification, Version 1.0 (PNNI), 1996.
[35] J. Moy, OSPF Version 2, IETF RFC 2328, 1998.

[36] P. Godfrey, I. Ganichev, S. Shenker, I. Stoica, Pathlet routing, ACM SIGCOMM Comput. Commun. Rev. 39 (2009) 111–122. ACM.
[37] L. Subramanian, S. Agarwal, J. Rexford, R.H. Katz, Characterizing the internet hierarchy from multiple vantage points, INFOCOM 2002, IEEE, 2002.
[38] L. Gao, On inferring autonomous system relationships in the internet, IEEE/ACM Trans. Netw. (ToN) 9 (6) (2001) 733–745.
[39] The CAIDA UCSD Internet Topology Data Kit, (http://www.caida.org/data/internet-topology-data-kit) Accessed: 2017-06-23.
[40] B. Quoitin, S. Uhlig, Modeling the routing of an autonomous system with c-bgp, Netw. IEEE 19 (6) (2005) 12–19.
[41] U. Meyer, P. Sanders, δ-stepping: a parallel single source shortest path algorithm, Algorithms-ESA'98, Springer, 1998.
[42] A. Crauser, K. Mehlhorn, U. Meyer, P. Sanders, A parallelization of Dijkstra's shortest path algorithm, Mathematical Foundations of Computer Science 1998, Springer, 1998.
[43] N. Edmonds, A. Breuer, D. Gregor, A. Lumsdaine, Single-source shortest paths with the parallel boost graph library, The Ninth DIMACS Implementation Challenge: The Shortest Path Problem, 2006.
[44] P. Guangyu, G. Mario, C. Tsu-Wei, Fisheye state routing in mobile ad hoc networks, in: Proc. of ICC, 2000.
[45] B.S. Davie, Y. Rekhter, MPLS: Technology and Applications, Morgan Kaufmann Publishers Inc., 2000.
[46] A. Beben, EQ-BGP: an efficient inter-domain QoS routing protocol, in: Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on, vol. 2, IEEE, 2006, pp. 5–pp.
[47] T. Braun, M. Diaz, J.E. Gabeiras, T. Staub, End-to-End Quality of Service Over Heterogeneous Networks, Springer Science & Business Media, 2008.
[48] Q. Li, J. Beaver, A. Amer, P.K. Chrysanthis, A. Labrinidis, G. Santhanakrishnan, Multi-criteria routing in wireless sensor-based pervasive environments, Int. J. Pervasive Comput. Commun. 1 (4) (2005) 313–326.
[49] X. Chen, H. Cai, T. Wolf, Multi-criteria routing in networks with path choices, in: Network Protocols (ICNP), 2015 IEEE 23rd International Conference on, IEEE, 2015, pp. 334–344.
[50] R. Teixeira, A. Shaikh, T. Griffin, J. Rexford, Dynamics of hot-potato routing in ip networks, ACM SIGMETRICS Perform. Eval. Rev. 32 (1) (2004) 307–319.
[51] K. Su, F. Bronzino, K. Ramakrishnan, D. Raychaudhur, Mftp: A clean-slate transport protocol for the information centric mobilityfirst network, in: Proceedings of ACM ICN 2015, ACM, 2015.
[52] S. Mukherjee, A. Baid, I. Seskar, D. Raychaudhur, Network-assisted multihoming for emerging heterogeneous wireless access scenarios, in: Proceedings of IEEE PIMRC 2014, IEEE, 2014.
[53] S.Y. Qiu, P.D. McDaniel, F. Monrose, Toward valley-free inter-domain routing, IEEE ICC, IEEE, 2007.
[54] Education Roaming (eduroam), (https://www.eduroam.org/). Accessed: 2017-06-23.
[55] A. Lara, S. Mukherjee, B. Ramamurthy, D. Raychaudhuri, K. Ramakrishnan, Inter-domain routing with cut-through switching for the mobilityfirst future internet architecture, in: Communications (ICC), 2016 IEEE International Conference on, IEEE, 2016, pp. 1–6.
[56] Netflix Open Connect, (https://openconnect.netflix.com/) Accessed: 2017-06-23.
[57] X. Yang, D. Clark, A.W. Berger, Nira: a new inter-domain routing architecture, IEEE/ACM Trans. Netw. (ToN) 15 (4) (2007) 775–788.
[58] W. Xu, J. Rexford, MIRO: Multi-Path Interdomain Routing, 36, ACM, 2006.
[59] Y. Shavitt, E. Shir, Dimes: let the internet measure itself, ACM SIGCOMM Comput. Commun. Rev. 35 (5) (2005) 71–74.
[60] G. Siganos, S.L. Tauro, M. Faloutsos, Jellyfish: a conceptual model for the as internet topology, J. Commun. Networks 8 (3) (2006) 339–350.
[61] M. Li, D. Agrawal, D. Ganesan, A. Venkataramani, H. Agrawal, Block-switched networks: a new paradigm for wireless transport., in: NSDI, 9, 2009, pp. 423–436.
[62] CIDR-Report, (http://www.cidr-report.org/as2.0/) Accessed: 2017-06-23.
[63] How Much Memory Should I have in My Router to Receive the Complete BGP Routing Table from my ISP?, (http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/5816-bgpfaq-5816.html#anc20). Accessed: 2017-06-23.
[64] E. Kohler, R. Morris, B. Chen, J. Jannotti, M.F. Kaashoek, The click modular router, ACM Trans. Comput. Syst. (TOCS) 18 (3) (2000) 263–297.
[65] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, 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, vol. 3, IEEE, 2005, pp. 1664–1669.
[66] Z. Gao, A. Venkataramani, J.F. Kurose, S. Heimlicher, Towards a quantitative comparison of location-independent network architectures, ACM SIGCOMM Comput. Commun. Rev. 44 (4) (2015) 259–270.
[67] SFMTA Municipal Transport Agency, (https://www.sfmta.com/). Accessed: 2017-06-23.
[68] M. Motiwala, M. Elmore, N. Feamster, S. Vempala, Path splicing, ACM SIGCOMM Comput. Commun. Rev. 38 (2008) 27–38. ACM.
[69] X. Yang, D. Wetherall, Source selectable path diversity via routing deflections, ACM SIGCOMM Comput. Commun. Rev. 36 (2006) 159–170. ACM.
[70] F. Wang, L. Gao, Path diversity aware interdomain routing, in: INFOCOM 2009, IEEE, IEEE, 2009, pp. 307–315.
[71] P. Verkaik, D. Pei, T. Scholl, A. Shaikh, A.C. Snoeren, J.E. Van Der Merwe, Wresting control from BGP: Scalable fine-grained route control., in: USENIX Annual Technical Conference, 2007, pp. 295–308.
[72] D. Walton, A. Retana, E. Chen, J. Scudder, Advertisement of multiple paths in bgp, IETF RFC 7911, 2016.
[73] M.P. Howarth, M. Boucadair, P. Flegkas, N. Wang, G. Pavlou, P. Morand, T. Coadic, D. Griffin, A. Asgari, P. Georgatsos, End-to-end quality of service provisioning through inter-provider traffic engineering, Comput. Commun. 29 (6) (2006) 683–702.
[74] C. Filsfils, N.K. Nainar, C. Pignataro, J.C. Cardona, P. Francois, The segment routing architecture, in: Global Communications Conference (GLOBECOM), 2015 IEEE, IEEE, 2015, pp. 1–6.
[75] L. Subramanian, M. Caesar, C.T. Ee, M. Handley, M. Mao, S. Shenker, I. Stoica, Hlp: a next generation inter-domain routing protocol, ACM SIGCOMM Comput. Commun. Rev. 35 (2005) 13–24. ACM.

Shreyasee Mukherjee received her B.Tech in electrical engineering from Bengal Engineering and Science University (BESU) Shibpur, India in 2011. She completed her M.S. in electrical & computer engineering from Rutgers University, New Jersey, USA in 2013 and is currently pursuing a Ph.D. From the same. Her research interests include clean slate future internet architectures, analysis of routing protocols and mobility support for emerging 5G networks.

Shravan Sriram received his B.Tech and M.S. in computer engineering from Amrita Vishwa Vidyapeetham, India in 2013 and Rutgers University in 2015 respectively. Currently he is a software engineer at Turbonomic Inc. where he works on the decision making engine of the product. His areas of interest are adaptive routing schemes for large-scale networks, future internet architecture for wireless /mobile cloud networking, resource allocation in virtualized systems and use of pricing models for QoS adherence and cloud migration.

Tam Vu received the B.S in computer science from Hanoi University of Technology, Vietnam in 2006, and the Ph.D. in computer science from WINLAB, Department of Computer Science, Rutgers University, New Jersey, USA, in 2013. He is currently an assistant professor and Director of the Mobile and Networked Systems Lab at the Department of Computer Science, University of Colorado Boulder. His research interest is mobile healthcare, mobile centric network, mobile communication, and mobile security. He is the recipient of CRC Interdisciplinary Fellowship at UC Denver 2015. He received Google Faculty Research Award in 2014 for his work in Chrome browser authentication. He received best paper award for inventing new form of communication, called Capacitive Touch Communication, in ACM MobiCom 2012. He was also a recipient of ACM MobiCom 2011 best paper award for his work on driver phone use detection. His research also received wide press coverage including CNN TV, NY Times, The Wall Street Journal, National Public Radio, MIT Technology Review, Yahoo News, among other venues.

Dipankar Raychaudhuri is Distinguished Professor, Electrical & Computer Engineering and Director, WINLAB (Wireless Information Network Lab) at Rutgers University. As WINLAB's Director, he is responsible for an internationally recognized industry-university research center specializing in wireless technology. He is also PI for several large U.S. National Science Foundation funded projects including the "ORBIT" wireless testbed and the MobilityFirst future Internet architecture. Dr. Raychaudhuri has previously held corporate R& D positions including: Chief Scientist, Iospan Wireless (2000–2001), AGM & Dept Head, NEC Laboratories (1993–1999) and Head, Broadband Communications, Sarnoff Corp (1990–1992). He obtained the B.Tech (Hons) from IIT Kharagpur in 1976 and the M.S. and Ph.D degrees from SUNY, Stony Brook in 1978 and 1979. He is a Fellow of the IEEE.