Bootstrapping evolvability for inter-domain routing with D-BGP

Raja R. Sambasivan
Boston University

David Tran-Lam
University of Wisconsin-Madison

Aditya Akella
University of Wisconsin-Madison

Peter Steenkiste
Carnegie Mellon University

ABSTRACT

The Internet's inter-domain routing infrastructure, provided today by BGP, is extremely rigid and does not facilitate the introduction of new inter-domain routing protocols. This rigidity has made it incredibly difficult to widely deploy critical fixes to BGP. It has also depressed ASes' ability to sell value-added services or replace BGP entirely with a more sophisticated protocol. Even if operators undertook the significant effort needed to fix or replace BGP, it is likely the next protocol will be just as difficult to change or evolve. To help, this paper identifies two features needed in the routing infrastructure (i.e., within any inter-domain routing protocol) to facilitate evolution to new protocols. To understand their utility, it presents D-BGP, a version of BGP that incorporates them.

CCS CONCEPTS

• **Networks** → **Network design principles**; **Network protocol design**; **Routing protocols**; **Public Internet**;

KEYWORDS

BGP; Control plane; Extensibility; Evolvability; Routing

ACM Reference format:
Raja R. Sambasivan, David Tran-Lam, Aditya Akella, and Peter Steenkiste. 2017. Bootstrapping evolvability for inter-domain routing with D-BGP. In *Proceedings of SIGCOMM '17, Los Angeles, CA, USA, August 21–25, 2017,* 14 pages.
https://doi.org/10.1145/3098822.3098857

1 INTRODUCTION

The Internet's inter-domain routing infrastructure is a critical component of its architecture. The routing paths it computes and disseminates in the control plane allow us to access all of the services and content that we hold dear. Today, this infrastructure is provided by a single inter-domain routing protocol—the Border Gateway Protocol (BGP)—which is plagued with problems. It does not provide domains sufficient influence to limit the amount of traffic they receive [18]; its paths are slow to converge and prone to oscillations [33]; it indiscriminately chooses a single best-effort path per router, robbing other domains of paths they may prefer more [63]; and it is prone to numerous attacks, including prefix hijacking and traffic interception [47]. Worst of all, BGP is architecturally rigid [20]—it requires direct neighbors to use the same inter-domain routing protocol—and thus cannot facilitate the deployment of new inter-domain routing protocols.

This rigidity has made widespread adoption of the numerous critical fixes to BGP proposed by the operator and research communities incredibly difficult. Examples include adding awareness of path costs to limit incoming traffic to domains [32], adding backup paths to reduce convergence times [29], and adding secure path announcements via BGPSec [8] to prevent prefix hijacking. It has also prevented BGP from being replaced altogether with more sophisticated protocols that are more suited for today's Internet. Examples include path-based routing [61, 63] to offer source domains more control over path selection and multi-hop routing to additionally allow for rich policies [19, 21]. Finally, it has depressed ISPs' ability to sell value-added services, such as differentiated QoS [36] or alternate paths [60], to combat their ever-increasing commoditization.

BGP's rigidity stifles innovation because it makes it more difficult for the Internet's routing infrastructure to change or evolve. It mandates that new protocols be deployed within isolated islands (groups of contiguous domains) that cannot discover one another, disseminate new protocols' information to one another, or benefit from using the new protocol to route traffic amongst themselves. One workaround that can be used to circumvent BGP's rigidity is to use an overlay to forcibly (i.e., without support from BGP) route traffic to domains that have deployed a desired new protocol [36, 49, 60]. But, this approach has significant drawbacks. Most notably, the tunnels an overlay uses to hide traffic's true destinations from domains that have not yet deployed the new protocol interfere with those domains' routing decisions and thus can significantly increase their operating costs. This can disincentivize them from supporting evolution. Even if operators undertook the massive effort necessary to upgrade or replace BGP, it is likely that the new protocol would be as difficult to upgrade.

The goal of this paper is to identify what features are needed in *any* inter-domain routing protocol (i.e., within the routing infrastructure) to bootstrap evolution to new inter-domain routing protocols—i.e., facilitate their deployment across non-contiguous domains and, if desired, gradually replace itself in favor of one of them. By incorporating the evolvability features, these new protocols would themselves be able to bootstrap evolvability to further new protocols. We present Darwin's BGP (D-BGP), a version of BGP extended with these features to understand the difficulty of incorporating them in a specific existing inter-domain routing protocol. We use D-BGP to illustrate our evolvability features' utility in enabling evolution. D-BGP's extensions could also be added to a secure protocol (e.g., BGPSec [8]).

SIGCOMM '17, August 21–25, 2017, Los Angeles, CA, USA

ACM ISBN 978-1-4503-4653-5/17/08…$15.00
https://doi.org/10.1145/3098822.3098857

The evolvability features we identify—*pass-through support* within routers and *multi-protocol advertisements*—reduce rigidity by cleanly separating the information contained in protocols' connectivity advertisements from that used by their path-selection algorithms. This allows protocols' advertisements to become containers that can disseminate multiple other protocols' control information and facilitate discovery across islands. Loop freeness is guaranteed by requiring all protocols to use the same loop avoidance mechanism. Our evolvability features do not require overlays, elevating whether they are used to be a protocol-specific consideration.

Our experiences indicate it is easy to implement D-BGP and deploy new protocols using it. Though we cannot claim D-BGP can facilitate the introduction of *all* new protocols, it is *sufficient* to bootstrap evolvability to a wide range of critical BGP fixes, value-added services, and multi-hop-based or path-based replacements (see Section 2). It can also foster innovation by facilitating a rich Internet comprised of multiple disparate protocols (e.g., multi-hop-based and path-based ones). Simulations show that, compared to BGP, D-BGP incentivizes adoption of new protocols by accelerating the rate at which adopters see those protocols' benefits. The benefits afforded at any adoption level vary depending on the type of protocol. We leave a thorough discussion of how D-BGP itself could be deployed to future work.

We present the following contributions:

1) Based on an analysis of 14 recently proposed inter-domain routing protocols, we identify evolvability features any inter-domain routing protocol should provide to bootstrap evolvability to various critical fixes to it or entirely new inter-domain routing protocols.
2) We describe the design of D-BGP, a version of BGPv4 [44] that incorporates the needed evolvability features. We describe our experiences implementing a D-BGP proof-of-concept, called Beagle, in an open-source router [39].
3) Via MiniNeXT-based [50] experiments using Beagle, we show that deploying two new protocols (Wiser [32] and Pathlet Routing [21]) using D-BGP requires only 255-293 lines of per-protocol code modifications.
4) Via simulation, we analyze how well D-BGP accelerates benefits for different types of new protocols. We show that D-BGP's control-plane overheads are modest (between 1.3x and 2.5x) even when supporting hundreds of critical BGP fixes and sophisticated replacements. This is because many critical BGP fixes can share protocol-specific control information with BGP.

2 TOWARD AN EVOLVABLE INTERNET

The evolvable Internet we envision will use multiple inter-domain routing protocols, many of which will only be partially deployed. This is because domains or ASes will naturally update to new protocols at different timescales and because different domains may want to use different protocols (e.g., to provide different value-added services). To ensure global connectivity, we assume all ASes will share one inter-domain routing protocol, called the *baseline*. The baseline could be a one-way protocol that disseminates its control information in path advertisements upstream from destinations to sources. Alternatively, it could be a two-way protocol that sends additional control information downstream from destinations to sources (e.g., to refine path selection). We assume the baseline (and all future baselines) will be based on path vector to allow ASes the flexibility to make independent routing decisions. Today's baseline is BGP, a one-way, path-vector protocol.

In this evolvable Internet, *islands* name a cluster of one or more contiguous ASes that support the same protocol. *Gulfs* name the set of ASes separating two islands. These ASes do not support the same protocol as the islands they separate, but do support the baseline and possibly others. ASes may use distributed control (i.e., individual routers advertise and choose paths) or centralized control (i.e., single entities within each AS advertise and choose paths). When discussing the deployment of a new routing protocol, we will sometimes refer to islands and ASes in gulfs as *upgraded islands* and *gulf ASes* for clarity.

This Internet may use multiple *network protocols* in the data plane to forward packets, either because it is evolving its address format (e.g., transitioning from IPv4 to IPv6) or because different network protocols are used by different routing protocols (e.g., path-based forwarding, used by SCION, and hop-based forwarding, used by BGP). As a result, traffic that crosses gulfs may need to be encapsulated with multiple network protocols' headers in the data plane. We call such headers *multi-network-protocol headers*.

To understand what evolvability features are needed in the routing infrastructure (i.e., within any inter-domain routing protocol) to facilitate this evolvable Internet, we analyzed 14 recently-proposed protocols from the research and operator communities [1, 6, 8, 17, 19, 21, 29, 32, 36, 53, 59–61, 63]. Our goal was to systematize what support they would need to be deployed across gulfs. We found that we could map the protocols into three distinct *evolvability scenarios* that differ in their goals, needed control-plane and data-plane support, and operators' incentives for supporting them (see Table 1 for a summary). These differences informed scenario-specific requirements that any features for enabling evolvability must satisfy.

In this section, we first discuss an important data-plane issue that can arise in an evolvable Internet and then introduce the evolvability scenarios. Since our scenarios focus on recently-proposed protocols, we assume the baseline is BGP when discussing them.

2.1 Routing compliance

Figure 1 illustrates a simple version of our evolvable Internet. In this figure, a source S wants to communicate with destination D using a new inter-domain protocol that is only partially deployed (ignore Wiser-specific information, which will be discussed later). Arrows show the direction of path advertisements. Note that traffic from S to D can use routing paths that traverse islands that support the new protocol (shown in grey) and ASes in gulfs that do not support it (shown in white). In an evolvable Internet, *routing compliance* refers to the degree to which such paths are compliant with the goals and policies of the new, but only partially deployed, protocol.

Reduced routing compliance can limit or obviate the benefits of using a new protocol to deliver traffic. Protocols that aim to expose extra within-island information, such as extra intra-island paths, are least sensitive to reduced compliance. In contrast, protocols that aim to optimize some global objective function, such as bottleneck bandwidth, security, or latency will be more sensitive because the limiting variable (e.g., the bottleneck) may be within gulf ASes. The extent of a protocol's sensitivity to reduced compliance depends on the objective function it aims to optimize.

Protocol	Summary	Details
Baseline → critical fix		
BGPSec [8]	Prevents path hijacking	⋆ Path attestations
EQ-BGP [6]	Adds end-to-end QoS	⋆ QoS metrics
Xiao et al. [59]	”	”
LISP [17]	Supports mobility	⋆ Dest. ingress IDs
R-BGP [29]	Enables quick failover	⋆ Extra backup paths
Wiser [32]	Limits ingress traffic	⋆ Path costs
Baseline → custom protocol		
MIRO [60]	Exposes alt. paths	⋆ Service's existence ◇ Tunnels
Arrow [36]	” + intra-island QoS	”
RON [1]	Creates low-latency paths	”
Baseline → replacement protocol		
NIRA [61]	Path-based routing	⋆ Multiple paths ◇ Fwd w/custom hdrs ◇ multi-network-proto hdrs
SCION [63]	”	”
Pathlets [21]	Multi-hop routing	⋆ Pathlets ◇ Fwd w/custom hdrs
YAMR [19] ”	”	◇ multi-network-proto hdrs
HLP [53]	Hybrid PV/LS	⋆ Path costs

Table 1: Protocols we analyzed. Protocols are grouped according to the evolvability scenarios most suited to them. Any protocol could be deployed as a custom protocol. Extra information that must be disseminated in the control plane is denoted by ⋆ and support needed in the data plane by ◇. A *PV* indicates path vector and *LS* indicates link state. A ” indicates that a column inherits from the corresponding entry in the previous row. HLP's link-state functionality can only be deployed within islands.

Routing compliance can be increased by tunneling traffic to force it to travel through a specific sequence of islands. But, tunnels come with significant drawbacks—they interfere with gulf ASes' routing decisions and use up extra addresses. Therefore, their use should be optional and depend on how sensitive a given protocol is to reduced compliance.

2.2 Baseline → baseline with critical fix

The goal of this scenario is to deploy modified versions of the baseline that incorporate various critical fixes. Critical fixes extend the baseline by disseminating extra control information to improve path selection or the protocol itself. Examples of critical fixes to today's baseline, BGP, include Wiser [32] for fixing BGP's lack of support for limiting ingress traffic at ASes [18], EQ-BGP [6] for adding generic QoS information to routing paths, and BGPSec [8], for fixing BGP's susceptibility to route hijacking [47].

Data-plane support: Tunneling traffic to increase routing compliance is optional and depends on the needs of the protocol. Multi-network-protocol headers are not needed because critical fixes use the same network protocol as the baseline.

Example: Figure 1 illustrates an Internet in which some islands have deployed Wiser [32]. Wiser fixes BGP's inability to help ASes limit ingress traffic by disseminating an extra path cost in advertisements, which influences path selection. Upgraded ASes add their

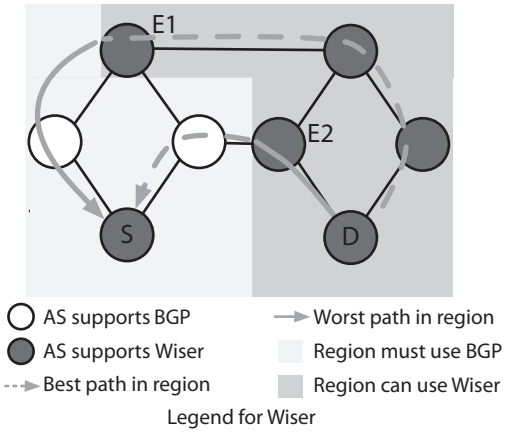

Figure 1: S can't see path costs, so it will choose the highest-cost one.

internal costs of routing traffic to the path costs they receive before selecting the one with the lowest cost. Cheating ASes can add abnormally high internal costs to prevent paths that include them from being selected. Wiser prevents this by using two-way communication to periodically exchange the total costs of paths neighbors receive from each other. It uses this data to scale the path costs an AS receives from a neighbor to be comparable to the path costs it advertises to that neighbor. It then adds the AS's internal costs and selects paths.

Since BGP requires neighbors to use the same routing protocol to create paths, the two ASes at the edge of the large Wiser island, E1 and E2, must use BGP to advertise paths to a destination (D) to ASes in the BGP gulf. Lines show paths advertised and arrows show the direction of the advertisement. This creates two problems. First, a potential source (S), which supports Wiser, cannot see Wiser's path costs. Second, because S, E1, and E2 are now separated by a gulf, they cannot exchange the cost of paths they receive from one another to compute scaling factors (assume S is also advertising paths, which we do not show). As a result of these issues, S must use BGP to select paths, which means it will choose the shortest path (due to BGP's path-selection algorithm), which has the highest path cost.

Incentives for deployment: ASes will be incentivized to deploy a critical fix if the benefits afforded by that protocol can be realized quickly. These benefits will increase incrementally as a function of the protocol and the number of islands that can route traffic among themselves using it.

Gulf ASes that do not plan to upgrade soon will have no incentive not to support this scenario if doing so does not interfere with their routing decisions and/or increase their costs. Islands can mitigate potential for interference by avoiding using tunnels. They could also give gulf ASes additional visibility and control by exposing critical fixes' control information to them. For example, with this knowledge, gulf ASes' operators could filter paths that use problematic protocols. They might also be able to use knowledge of what protocols are used on paths to debug problems.

Requirements: As the example above shows, without establishing overlays and tunneling traffic, islands that deploy critical fixes cannot discover one another. They also cannot disseminate critical fixes'

control information among themselves (this would also enable discovery). Overlays and tunnels can be made optional for evolvability by evolvability features that meet the following requirement:

CF-R1 *Disseminate critical fixes' control information across gulfs.*

Critical fixes' control information could be disseminated across gulfs in two ways. It could be disseminated *in-band* of the baseline by including it within baseline advertisements. This would expose new protocols' control information to gulf ASes and allow the baseline to be seamlessly updated. But, it would also result in larger advertisement sizes.

Critical fixes' control information could also be disseminated *out-of-band* of the baseline's advertisements via existing baseline paths. This requires establishing a minimal correspondence between critical fixes' control information relevant to a path and the baseline advertisement for that path. This approach would not inflate advertisement sizes, but would hide information from gulf ASes. Also, compared to the in-band approach, it will suffer from an additional constant performance penalty due to the overhead of external accesses on the critical path of advertisement processing.

To allow the baseline to be seamlessly updated and to avoid disincentivizing gulf ASes, we require disseminating critical fixes' control information in-band whenever possible. Therefore, we introduce the following requirement for our evolvability features:

CF-R2 *Disseminate critical fixes' control information in-band of the baseline's advertisements.*

2.3 Baseline → baseline // custom protocol

The goal of this scenario is to allow islands to deploy new protocols in parallel (//) with the baseline. These new protocols are used to route selected traffic, while the baseline is used for the rest. New protocols can be custom-made by the island and there is no expectation that they will be globally adopted. This scenario enables islands to sell value-added services to other customers (e.g., other islands or end users). Examples include selling alternate paths from BGP's single path [1, 36, 60], providing a VPN service, and selling access to some non-baseline protocol (e.g., any of the ones considered in this paper).

Data-plane support: Routing compliance is needed because custom protocols' traffic must reach specific islands (e.g., an island whose value-added service(s) a customer has purchased). Multi-network-protocol headers may be needed if islands use different network protocols than the baseline.

Example: Figure 2 shows a transit island (marked T) that wishes to avoid the single poorly performing path advertised by BGP (the dashed path) to a destination (D). An island that supports MIRO [60] (marked M) offers alternate paths for payment. But, BGP does not facilitate discovery of islands' custom services or how they must coordinate out-of-band to exchange control information. For MIRO, this information includes the alternate paths offered, the payment required to use a given path, and the tunnel address that must be used to route traffic along the chosen path. Island T remains ignorant of Island M. Though the example shown is about off-path discovery (i.e., Island M is not on routing paths to the destination (D) advertised to Island T), BGP also does not support on-path discovery (i.e., Island M is on routing paths to the destination (D) advertised to Island T).

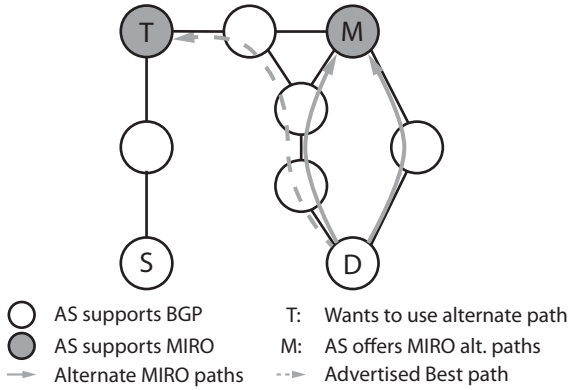

Figure 2: T cannot discover the MIRO service.

This limits Island M's potential customers to only its direct neighbors. Island M could use a web site for discovery, but it may go unnoticed.

Incentives for deployment: Islands that deploy custom protocols do so to sell value-added services. Gulf ASes may be incentivized to support custom-protocol deployments because they themselves may wish to sell such services in the future.

Requirements: To support this scenario, islands supporting custom protocols must be able to discover each other using the evolvability features. Thus, we have:

CP-R3 *Facilitate across-gulf discovery of islands running custom protocols and how to negotiate use of their services.*

2.4 Baseline → replacement protocol

The goal of this evolvability scenario is to allow new protocols that are radically different from the baseline to completely replace it within islands or Internet wide. Unlike the previous scenario, a given new protocol is used to forward *all traffic* within its islands. This is a very aggressive scenario and likely to be only attractive if there are strong incentives or requirements that are impossible to meet with the baseline or its critical fixes.

Protocols apt for this scenario are ones that use different network protocols compared to the baseline. One set of examples include path-based routing protocols, such as SCION [63], which expose multiple paths to sources and allow them to encode which ones they want to use in packet headers. Another set of examples include multi-hop protocols, such as Pathlet routing [21], which allow islands to expose intra-island path fragments or pathlets. These pathlets can be combined by other islands into larger pathlets or end-to-end paths. Sources can pick which pathlets they want to use by encoding them in packet headers.

Data-plane support: Tunneling traffic to increase routing compliance is optional and depends on the needs of the protocol. Multi-network-protocol headers are needed to cross gulfs.

Example: Figure 3 illustrates a scenario in which BGP is being replaced by SCION [63], a path-based protocol. In this case, the rightmost SCION island in the diagram exposes two paths to a destination (D). To provide basic connectivity, it redistributes [30] one SCION path into BGP. However, the second path cannot be redistributed and is lost because BGP is designed to operate a less-advanced network

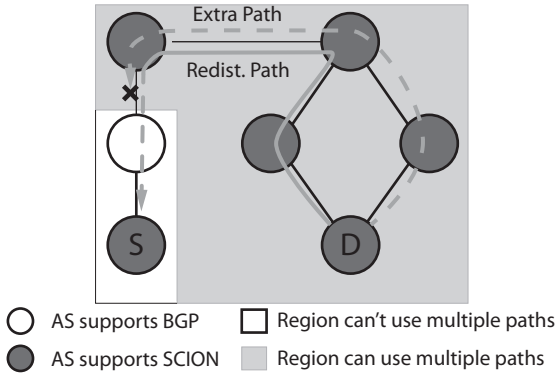

Figure 3: S cannot be advertised both paths to D.

protocol than SCION and thus supports advertising only one path per router.

Incentives for deployment: Identical to critical fixes.

Requirements: Identical to critical fixes.

2.5 Global evolvability requirements

In addition to the scenario-specific evolvability requirements above, features that aim to support evolvability in an Internet that runs many diverse routing protocols must satisfy two global requirements.

G-R4 *Inform islands and gulf ASes of what protocols are used on routing paths.*

G-R5 *Avoid loops across all protocols used on routing paths.*

This first is needed to allow islands to make informed routing decisions, give gulf ASes more visibility, and to inform sources how to create multi-network-protocol headers. The reasoning for the second is obvious.

2.6 Needed evolvability features

All of the requirements above can be satisfied by two complementary features. The first is *pass-through support*, which allows routers or ASes to pass through control information for protocols they do not support to adjacent ones. This allows protocols' control information to cross gulfs (*CF-R1*).

The second feature is a *multi-protocol* data structure that encodes what protocols are used by islands on routing paths (*G-R4*). It provides the additional building block necessary to facilitate discovery of islands running custom protocols (*CP-R3*). (See Section 3.4 to see how this data structure can enable off-path discovery).

Encoding the multi-protocol data structure within the baseline's advertisements enables in-band dissemination (*CF-R2*). It also allows all protocols, including the baseline, to use a common loop-detection mechanism (*G-R5*).[1]

Comparison to BGP: Pass-through support is provided in BGP via its optional transitive attributes [44], but it is not presently used to deploy new routing protocols. BGP does not provide a systematized multi-protocol data structure or loop detection mechanism that allows multiple, diverse inter-domain protocols to co-exist.

[1]With additional support from islands, loops can be avoided even if multi-protocol support is provided out-of-band.

3 DESIGN OF D-BGP

This section and the following ones seek to understand how our evolvability features can be incorporated into an existing inter-domain routing protocol and the difficulty/utility of doing so. We incorporated the features into BGPv4 [44], today's baseline, because it is a logical starting point for any evolvability efforts related to inter-domain routing. This section describes the design of Darwin's BGP (D-BGP), a version of BGPv4 minimally extended with pass-through support and a multi-protocol data structure in advertisements. D-BGP can be used by ASes with distributed control (i.e., those that use individual routers as BGP speakers) or centralized control (i.e., those that use centralized BGP speakers [15, 46]). It requires data-plane support, similar to that used by MPLS [3] and Arrow [36], to support multi-network-protocol headers.

Incorporating the evolvability features make D-BGP's advertisements a *shared container* that can carry multiple inter-domain routing protocols' control information, including BGP. These advertisements are carried across gulfs by the path-selection choices of the ASes within them. Because D-BGP's advertisements are now shared among multiple inter-domain routing protocols, we need to redefine the term critical fixes to specifically refer to changes to BGP's path-selection algorithm and control information. To demonstrate D-BGP's utility in bootstrapping evolution to new protocols, we assume it is the baseline and that IPv4 is the baseline address format in this section. Later sections discuss challenges involved in implementing D-BGP and the evolvability benefits it can provide when it is the baseline.

Our experiences indicate that using BGPv4 as the starting point to incorporate our evolvability features provides significant benefits. But, because our features rely on an existing inter-domain routing protocol to bootstrap deployment of new protocols, the former's limitations can limit the latter's benefits. At the end of this section, we discuss how BGP's limitations can reduce new protocols' benefits and how these issues can be mitigated by using a more advanced protocol as the starting point.

3.1 Assumptions

To prevent conflicts, we assume that all new protocols will be assigned unique IDs by a governing body, such as the IETF [26] or ARIN [2]. These governing bodies could also assign islands unique IDs. Alternatively, islands could create island IDs themselves by hashing the AS numbers of their border ASes. For clarity, we use protocol names for protocol IDs, letters for island IDs, and numbers for existing AS numbers. We use singleton islands' existing AS numbers as their island IDs. To prevent a proliferation of (potentially buggy) new protocols that aim to succeed the baseline, we assume that critical fixes and replacement protocols that aim to do so will be ratified by the governing body.

We assume that all new protocols will use path vectors when communicating across islands, as our evolvability features require that they use the same loop-detection mechanism as the existing baseline. Islands could use non-path-vector protocols internally. For example, they could use custom protocols that are based on link-state. Alternatively, they could use a replacement protocol that uses link-state for intra-island communication and path vector for inter-island communication [53].

3.2 Integrated advertisements

Integrated advertisements (IAs) extend existing BGP advertisements to provide multi-protocol support. Each IA compactly describes a path that can be used to reach a destination address, named using the baseline address format. Figure 4 shows an example IA for a path, which includes a Wiser [32] island (AS 3), a SCION [63] island (Island A), a MIRO [60] island (Island G), an AS in a gulf (AS 4000), and a BGPSec [8] island (Island K). Only fields relevant to multi-protocol support are shown, so standard BGP fields, such as withdrawn prefixes have been omitted.

The *path-vector* field states the path. It is the common denominator that all protocols on the path must use to avoid loops. We allow island IDs or AS numbers to be listed and delegate the choice to individual islands. Islands that list their IDs abstract away their intra-island paths from D-BGP's loop-detection mechanism. Doing so is necessary for islands running certain replacement protocols whose within-island paths cannot be expressed in a path vector (e.g., hybrid path-vector / link-state ones [53]). Islands may also choose to list their IDs because it simplifies effort needed to deploy a new replacement protocol using D-BGP or because they wish to hide intra-island paths for competitive reasons. Islands that list their IDs reduce path diversity for member ASes because this forces loop detection to work at the granularity of entire islands. Paths that enter and leave the island multiple times without causing AS-level loops will be thrown out.

In the figure, Island A, which runs SCION, has chosen to list only its island ID in the path-vector field, perhaps due to competitive concerns. If it wishes to mitigate reduction in path diversity, it would need to list every AS on the multiple within-island paths it offers in the path-vector field to prevent loops. To prevent gulf ASes from thinking the IA presents an overly long path, the island could optionally list these AS numbers within an AS SET relationship. (BGP's decision process prefers shorter AS-level paths.) This is similar to what BGP does today when aggregating multiple advertisements for contiguous destination addresses at proxies (i.e., at transit or tier-1 ASes) [44].

Island G, which runs BGP in parallel with MIRO, a custom protocol, exposes only its AS-level BGP path to the destination in the path-vector field. It does not expose details of its alternate paths because their use is coordinated out-of-band of BGP.

The *island ID* field allows membership of ASes listed in the path vector to be identified. It is necessary to tell routers how to layer headers when encapsulating data packets to cross BGP gulfs and use custom or replacement protocols on routing paths. In the figure, AS 4000 does not have an island ID since it is part of a BGP gulf. AS 3 uses its AS number as its island ID as it is a singleton island.

Path descriptors fields bear resemblance to BGP's path attributes, but are explicitly structured for multi-protocol support. They describe per-protocol attributes of the entire path. Critical fixes use them to encode their control information. The example shown in Figure 4 includes Wiser's scaled path cost [32] and BGPSec's attestations [8]. Other potential path descriptors include Xiao et al.'s [59] and EQ-BGP's [6] QoS metrics. Since BGPSec requires an unbroken chain of participation, starting from the destination, to provide benefits, island K could optionally drop the attestation before sending it to insecure islands. We include it here for illustrative purposes.

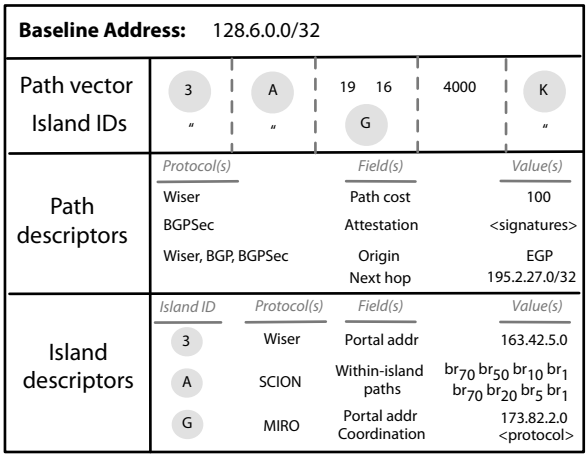

Figure 4: An example D-BGP integrated advertisement.

To our knowledge, BGP does not have an explicit analog to *island descriptors*. These fields encode attributes specific to individual islands. Islands that support custom protocols use them to facilitate discovery. Those that use replacement protocols use them to encode their control information. Those that support two-way protocols use them to identify how upstream islands can exchange control information with downstream islands. Note, this exchange must be done out-of-band of D-BGP using existing (baseline-only) paths because BGP is a one-way protocol. Stub islands that use a different address format than the baseline (e.g., IPv6 or content names [62]) can use island descriptors to help establish a mapping between the two formats. For example, an island could originate an IA for a gateway and include an island descriptor with the address of a lookup service. The service would contain within-island addresses that can be reached via the gateway. This would let islands route traffic among themselves using the new format.

The example IA shown in Figure 4 shows a Wiser island's descriptor. It includes the address of a portal downstream neighbor islands can contact to periodically send the path costs of paths they receive from this island. The IA also includes a descriptor for the SCION island, which lists two within-island paths that can be used to reach the destination. They are specified at the granularity of border routers, which have been assigned random IDs (e.g., br_1). The descriptor for the MIRO island enables on-path discovery. It includes the IP address of a portal that customers can contact to exchange relevant control information.

Limiting IA sizes: Critical fixes listed in IAs can share control information that is identical across them and BGP. This can drastically reduce IA sizes, as critical fixes often disseminate the same control information as BGP except for one or two extra protocol-specific fields (e.g., BGPSec [8] disseminates only one extra field: a path attestation). In Figure 4, BGP, BGPSec [8], and Wiser [32] all share control information.

Custom and replacement protocols' contribution to IA size will be small because they do not need to disseminate much control information outside their islands. For example, a SCION island that disseminates five within-island paths, each consisting of five intra-island hops, will only need to disseminate about 200 bytes of control

information to describe them (this assumes 4-byte border router IDs). IAs can be compressed to further reduce their size.

3.3 IA processing

D-BGP modifies BGP's advertisement processing to support IAs, provide pass-through support, and give gulf ASes some visibility and control. For clarity, we first describe how D-BGP's advertisement processing enables evolvability for critical fixes and custom protocols. We then expand our discussion to include replacement protocols. We briefly describe the role of D-BGP's various import/export filters in this section, but a detailed discussion of how they might be used to implement the new types of policies and AS relationships possible in an Internet running multiple inter-domain protocols is out of the scope of this paper.

Figure 5 shows D-BGP's advertisement processing, which could be running either within ASes' border routers or within ASes' centralized BGP speakers. It also shows the steps required to receive an incoming IA, select best paths, and create a new IA. This IA processing module belongs to an island that has recently deployed a critical fix to BGP (shown in purple or dark grey). Previously, it was using BGP (shown in blue or light grey). The island may support custom protocols, but they are not shown in the figure as they do not affect processing of incoming IAs or best-path selection. D-BGP's processing includes the following novel components needed to support IAs and pass-through control information.

First, it includes *global import/export filters*, which allow operators to enforce policies common to all protocols, but perhaps specific to individual neighbors (e.g., valley-free routing [38]). Gulf ASes' operators can use global filters to assert a limited amount of control over what protocols can be used on their paths. For example, they could remove control information for protocols known to be problematic (they would only need to know the protocol ID to do so).

Loop detection is implemented at the global filtering stage, either at the ingresses (input filters) or egresses (output filters) of ASes. This is similar to BGP's processing, but is now extended to all protocols contained in IAs. At the egresses of islands, output filters are used to state island membership or abstract away intra-island details. The former is accomplished by a filter that adds an island ID to the island ID field and states which set of contiguous AS entries in the path vector correspond to member ASes. The latter is accomplished by a filter that removes the most recent set of contiguous path-vector entries that correspond to member ASes and replacing them with a single island ID.

Second, it includes multiple *decision modules*, corresponding to BGP and the critical fixes it supports. Only one protocol and hence decision module is active at a time for a given address range. Having multiple protocols be active for the same range would add little value because only a single protocol's path choice can be installed in a single IP forwarding table. The processing module shown in Figure 5 includes a decision module for BGP (shown in blue/light grey) and one for a critical fix (shown in purple/dark grey). Only the critical fix's module is active because operators have chosen to route all traffic through it.

Each decision module encapsulates the data structures (e.g., RIB and path-selection algorithm) a given protocol uses to select best paths. Decision modules can optionally include import and export

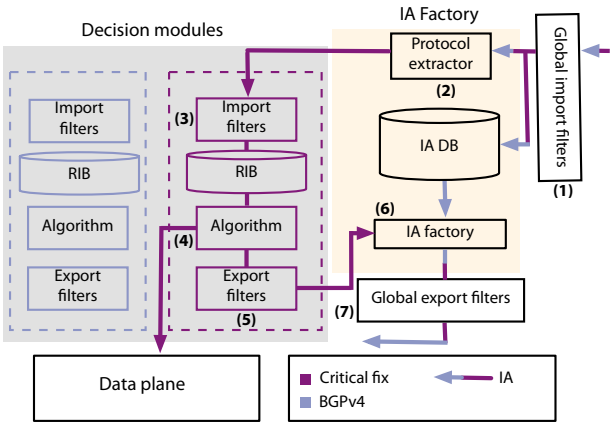

Figure 5: D-BGP's IA processing. *(1)*: Global import filters are applied to the IA and it is forwarded to the protocol extractor and the IA DB. The AS number could be added at this stage or at the output filter. *(2)*: The protocol extractor determines the active protocol for the prefix and extracts control information specific to it. It also extracts shared control information, such as the path vector, neighbor that sent the IA, and prefix to allow them to be also used during path selection. It forwards all of this information to the appropriate decision module. *(3)*: Protocol-specific import filters are applied and control information is stored in the decision module's RIB. *(4)*: The decision module's path-selection algorithm selects a new best path for the IA's prefix. It forwards the new best path to the data plane and the its control information to the export filter. *(5)*: Protocol-specific export filters are applied and the best path's control information is sent to the IA factory. Control information could have been modified at any stage within the decision module. *(6)* The IA factory creates a new IA for the prefix by including the modified control information. It provides pass-through functionality by copying over unused protocols' control information from the incoming IA for the best-path selected into the new one. *(7)*: Global export filters are applied to the new IA and the IA is sent to neighbors. These filters may replace within-island AS numbers with a single island ID depending on the island's policy.

filters specific to the relevant protocol. At a minimum, these filters will modify per-protocol control information. Protocols that are guaranteed not to be active at the same time may be able to share some data structures, such as RIBs (not shown).

Third, it includes an *IA factory*, which replaces similar functionality for BGP's advertisements. The IA factory is responsible for receiving IAs and communicating the per-protocol and shared information contained in them to active decision module(s). It is also responsible for creating new IAs for selected best paths. The IA factory is agnostic to per-protocol information; it only needs to know the active protocols' IDs to do its job.

The IA factory provides pass-through functionality when creating new advertisements for selected best paths. Specifically, when creating new IAs for a selected best paths, it indexes into a database of received IAs to retrieve the incoming IA for the chosen best path. It then copies over all control information for protocols in the incoming IA that were not used for best-path selection to the new one. This is (somewhat) similar to the pass-through functionality provided by BGP's optional transitive attributes today [44].

Supporting islands running replacement protocols: In this case, D-BGP is only used at islands' borders. Each replacement protocol

must provide a decision module. It must also additionally provide a redistribution module, an ingress translation module, and an egress translation module. The *redistribution module* interposes between the data plane and protocols' decision modules to redistribute routes into BGP. *Ingress and egress translation modules* encapsulate the protocol's decision module to map between IAs and the replacement protocol's advertisement format. The ingress module is responsible for preserving D-BGP path vectors and the egress module is responsible for encoding within-island paths into D-BGP path vectors. The global filter may replace these within-island paths with a single island ID. If the replacement protocol does not provide its own multi-protocol and pass-through support (or is based on link-state [53]), that island's border ASes must coordinate to exchange unused protocols' information (and the ingress D-BGP path vector if it is not directly compatible with the replacement protocol).

3.4 Example usage

Evolvability for critical fixes: D-BGP allows E1 and E2 from the example in Section 2.2 to include Wiser's path costs in IAs they advertise to ASes in the gulfs. They also include the address of a service portal that neighbors on the other side of the BGP gulf can contact to periodically send the costs of paths those neighbors receive from them. This allows path costs to be scaled before path selection (the scaling value must be guessed to initially select paths). Path costs and the portal address are passed through the gulf so that the source AS (S) is able to see them and use them to select the lower cost, longer path.

Off-path discovery for custom protocols: D-BGP allows Island T in the example from Section 2.3 to discover and use one of the MIRO island's alternate paths as follows. First, the MIRO island (M) uses IAs to advertise a path to a service portal it provides. It includes an island descriptor within the IA similar to the one described in Section 3.2. Second, Island T receives the IA, along with the island descriptor, which has been passed through the BGP gulf. Third, Island T contacts the service portal to negotiate use of the alternate path and the tunnel address to use. Fourth, Island T tunnels its traffic destined for the destination (D) to Island M.

Evolvability for replacement protocols: D-BGP enables the rightmost SCION island in the example from Section 2.4 to advertise both of its within-island paths to the SCION island (S). To do so, the leftmost border router in the rightmost SCION island creates an IA for the prefix advertised that includes control information for a SCION path that has been redistributed into BGP. It also includes both of its within-island paths within an island descriptor. When the SCION island (S) receives the IA, it extracts the SCION-specific control information, chooses a within-island path, and encodes it in a SCION header, which it attaches to data packets. It encapsulates the packet with an IPv4 header so that the packet can cross the gulf.

3.5 Limitations & discussion

Limitations of evolvability features: D-BGP is subject to the limitations of the evolvability features it incorporates. Most notably, the benefits afforded to new protocols can be curtailed by *indiscriminate path choices within gulfs* and *ill-informed path choices within upgraded islands*. Both will result in routing paths that are less compliant with the desired protocols' goals than other possible paths that could have been selected. The former occurs because ASes in gulfs will not take into account new protocols' goals when selecting paths. The latter occurs because upgraded islands will need to select paths while unaware of (potentially) important information within gulf ASes.

Protocols that aim to optimize some global objective function are affected by both limitations. To help protocols of this category that are especially sensitive to decreased routing compliance, ASes could employ the following techniques to increase it. They could wait until a minimum threshold number of ASes install a given new protocol before using it to select paths (i.e., before deploying it). Alternatively, they could deploy a version of the new protocol that is capable of switching between either the baseline's path-selection algorithm or the new protocol's algorithm depending on the number of domains on individual routing paths that have themselves deployed it. Either in addition to or instead of the above approaches, they could use tunnels. Protocols that only aim to expose information local to islands are only affected by the first limitation. They are less sensitive to reduced compliance, but could also use the above approaches to increase it.

Limitations of using BGP as a starting point: There are three key limitations. First, D-BGP cannot accelerate incremental benefits for critical fixes that add secure path advertisements (e.g., BGPSec [8]) because attackers could send spoofed advertisements to the first gulf AS on routing paths regardless of how many other islands are included [12, 31]. However, D-BGP's extensions could be deployed alongside these protocols. Second, D-BGP's advertisements are hampered by BGP's single best-path limitation, which prevents them from disseminating multiple *inter-island* paths. This means islands running path-based (e.g., SCION [63]) or multi-hop protocols (e.g., Pathlet Routing [21]) must choose a single best inter-island path for a prefix at their borders. Using a path-based protocol, multi-hop protocol, or a version of BGP that supports advertising multiple paths per destination [57] as the starting point for integrating our evolvability features would eliminate this limitation. Third, because BGP only supports one-way advertisements (from destinations to sources), D-BGP cannot naturally facilitate the deployment of two-way protocols, such as Wiser [32] or R-BGP [29]. Control information sent from sources to destinations must occur out-of-band of D-BGP.

Potential concerns with D-BGP: We do not anticipate D-BGP's evolvable Internet will result in increased convergence times because BGP already gives ASes significant flexibility when making routing decisions and in choosing the rate at which to disseminate advertisements. Since D-BGP's IAs will be larger than BGP's advertisements (see Section 6.2), D-BGP may increase convergence times when a large number of them must be transferred at the same time (i.e., after session resets between D-BGP peers). Incorporating fault-tolerance mechanisms within D-BGP speakers could mitigate the need for such transfers [51]. Islands may cause convergence issues if they switch to using new protocols very often—e.g., at the same rate as link failures (about 172 per day over a two-month period as measured from four Internet vantage points [28]). But, we anticipate that islands will deploy new protocols more slowly than the rate at which link failures occur and will use planned rollouts to minimize disruptions.

Our D-BGP design does not allow for prefix aggregation by proxies, which may negatively impact control-message sizes and processing overhead. Our initial D-BGP design incorporated proxy-aggregation support, but we removed it because we found that proxy aggregation is barely used today (only 0.1% paths are aggregated [9]) and many

of the protocols we analyzed have limited potential to use it. For example, BGPSec's attestations cannot be aggregated [9] and it is not clear how to aggregate Wiser's path costs.

Deployment of D-BGP itself: This section and the following ones assume that D-BGP is the baseline protocol. But, getting to this point requires a transitional phase during which D-BGP is incrementally being deployed across contiguous domains. During this phase, D-BGP speakers could simply drop IAs' extra fields before sending advertisements to legacy ones that have not deployed it. They could translate between D-BGP's path vector and BGP's path vector (which only allows 2-bytes per entry) using techniques similar to how 4-byte-per-entry path vectors are being deployed today within BGP [56]. We leave a thorough discussion of how D-BGP could be incrementally deployed to future work.

4 D-BGP'S EVOLVABLE INTERNET

To show D-BGP's utility, Figure 6 illustrates the type of rich and evolvable Internet with many routing options it could enable. This rich Internet is composed of several different types of protocols, including BGP, different critical fixes to it, different types of replacement protocols (path-based, multi-hop), and custom protocols. This Internet could either converge to using a single critical fix or replacement protocol Internet-wide or could continue to exist in its current heterogeneous state. Our example uses protocols already discussed in this paper, but other types of critical fixes [6, 59], custom protocols [1, 36], and replacement protocols [19, 61] could also be used.

Figure 7 shows one possible IA in this rich, evolvable Internet. It is the one disseminated by Island G to Island 8 for the 131.4.0.0/24 prefix. This IA illustrates how Pathlet-Routing islands could use IAs, how Wiser islands could exchange path costs using them, and how SCION islands could use them. The path shown in the IA traverses Island G (Pathlet Routing), Island 11 (Wiser // MIRO), Island F (SCION), AS 14 (BGP), and Island D (Pathlet Routing). An alternate path to this prefix would have been through AS 10 (BGP). But, because Island F has only one peering point with Island 11, it had to pick one of the two inter-island paths due to BGP's single best-path limitation.

Path descriptors for this IA include one for Wiser, which describes Island 11's contribution to the path cost, and various ones shared by Wiser and BGP. This IA contains many island descriptors. Island D has included island descriptors for Pathlet Routing. These descriptors contain forwarding IDs (e.g., 1) and the within-island pathlets that correspond to them (e.g., (dr_1, dr_2)). This means that traffic received by Island D's border router dr_1 whose headers contain a forwarding ID of 1 will be forwarded to some within-island router dr_2. dr_2 will examine the packet header and forward traffic as per the next ID listed. Island D has exposed pathlets that can be combined into two distinct within-island paths to the destination.

Island F has included one island descriptor describing the within-island paths that can be used within its island. These paths are specified at the level of member ASes' border routers (fr). Island 11 has inserted an island descriptor stating the IP address of a portal Wiser islands on the other side of gulfs can contact to exchange the path costs they advertise to each other. For this advertisement, Island 11's across-gulf neighbors are Islands 8 and B. Island 11's descriptor also includes information about its MIRO service.

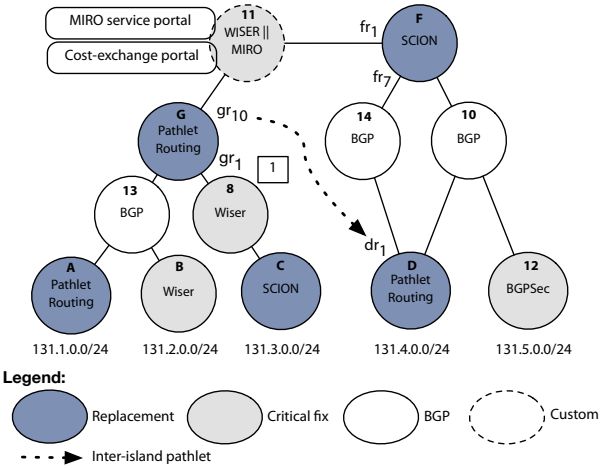

Figure 6: A rich & evolvable Internet facilitated by D-BGP.

Baseline Address: 131.4.0.0/24

Path vector Island IDs	G	11	F	14	D
	"	"	"		"

Path descriptors	*Protocol(s)*	*Field(s)*	*Value(s)*
	Wiser	Path cost	75
	Wiser, BGP	Origin	EGP
		Next hop	195.2.27.0/32

Island descriptors	*Island ID*	*Protocol(s)*	*Field(s)*	*Value(s)*
	G	Pathlet Routing	Within-island pathlets	1: (gr_1, gr_4), 3:(gr_4, gr_{10}), 6: (gr_1, gr_3), 7:(gr_3, gr_{10}), 8:(gr_{10},dr_1)
	11	Wiser	Portal addr	154.63.23.1
		MIRO	Portal addr Coordination	154.63.23.2 <protocol>
	F	SCION	Within-island paths	fr_1 fr_9 fr_{11} fr_7 fr_1 fr_2 fr_3 fr_7
	D	Pathlet Routing	Within-island pathlets	1: (dr_1, dr_2), 5: (dr_2, dr_4), 3: (dr_1, dr_3), 4: (dr_3, dr_4), 9: (dr4, 131.1.4.0/24)

Figure 7: IA at point 1 of the rich-world topology.

Island G has exposed a set of pathlets that provide connectivity between its island border routers gr_1 and gr_{10}. It has also exposed an *inter-island multi-hop* pathlet to Island D's border router (shown by (gr_{10}, dr_1) in Figure 7 and the dotted line in Figure 6). Doing so allows Pathlet Routing's decision modules to work without knowledge of gulfs when creating end-to-end paths out of the pathlets they receive.

5 BEAGLE: A D-BGP PROTOTYPE

To understand challenges involved in implementing our D-BGP design, we extended Quagga 0.99.24.1's BGP daemon to support IAs and IA processing. Quagga is a software-based, open-source router that is used in datacenters [40]. It supports IPv4- and IPv6-based network protocols. Therefore, *Beagle*, the name of our prototype, can support inter-domain routing protocols that use IPv4 or IPv6, either

natively (e.g., BGP and its critical fixes), or by using within-island tunnels to mimic needed additional data-plane features (e.g., as needed for multi-hop protocols and path-based ones). Modifying Quagga to create Beagle involved adding or modifying 769 lines of code. We do not include the automatically generated protocol-buffer [37] code we used to serialize IAs in this number.

Beagle works by interposing on Quagga's BGP advertisement processing for IPv4. It adds APIs that allow protocols to define their own decision modules and to allow replacement protocols to define translation modules and redistribution modules. It modifies Quagga's processing to extract the active protocol's control information (and any shared information) from IAs and send it to the relevant decision module. Decision modules share Quagga's IPv4 RIB.

Overall, we found it easy to add support for IAs and IA processing to create Beagle. The main difficulties were as follows. First, we had to re-architect Quagga slightly to support multiple inter-domain routing protocols' decision modules; currently, the assumption that only BGP's path-selection algorithm will be used is baked into the code. Second, we found Quagga's hand-rolled routines for serializing advertisements and memory management brittle and complicated. Thus, for the purposes of experimentation, Beagle disseminates IAs out-of-band by storing them in a lookup service. We would need to implement more extensive modifications to extend Beagle to support natively network protocols that are not based on IPv4 or IPv6. For example, we would need to break Quagga's (and BGP's) requirement that direct neighbors support the same set of network protocols.

Overhead of Beagle's code modifications: We ran a stress test on Beagle and Quagga to understand the overhead of Beagle's serialization and processing additions. The test used multiple peers to send advertisements to the router under test (either Beagle or Quagga). Each peer sent 150,000 advertisements (advertisements were collected from RIPE [11]). We used BGP's path-selection algorithm to select paths when benchmarking both routers to isolate the overhead of our evolvability extensions. The test was run on a single machine with two Intel E5-2640 CPUs (16 cores each), with one core assigned to the router under test. Six concurrent peers were used to saturate the router's CPU. Our results, which are the average of three runs of the stress test, show that Beagle's processing overhead is negligible compared to Quagga (40,700 prefixes/s vs 40,900 prefixes/s for BGP-only advertisements). When IAs are additionally exchanged, Beagle's performance decreases with IA size due to extra serialization cost (e.g., 7073 prefixes/s for 32KB IAs and 926 prefixes/s for 256KB IAs).

6 EVALUATION

In this evaluation, we seek to answer the following questions. First, how much effort is required to deploy new protocols across gulfs using D-BGP? Second, what is the control-plane overhead of using D-BGP to facilitate an evolvable Internet? Third, for different protocol types, how much does D-BGP accelerate incremental benefits?

6.1 Deploying protocols

Effort required: To gain insight into this question, we implemented basic versions of Pathlet Routing [21] (a replacement protocol) and Wiser [32] (a critical fix) and then modified them to be deployed across gulfs. We implemented and modified both protocols within Beagle. Wiser simply extends Beagle's existing BGP decision module. We chose to implement Pathlet Routing within Beagle to demonstrate one way this replacement protocol could incorporate our evolvability features.

Our basic Pathlet-Routing implementation (without modifications to support being deployed across gulfs) totals 509 lines of code. It names pathlets by assigning unique IP addresses to them. It uses IAs that carry individual pathlets as its advertisement format. Our basic Wiser implementation only required 109 lines of code as it is very similar to BGP.

We found that it was straightforward to modify both protocols to be deployed across gulfs. For Pathlet Routing, 293 lines of additional code was required. We had to create a module to redistribute a set of pathlets that could be used to reach within-island destinations or islands' egress points into BGP. We also had to create translation modules to translate between within-island advertisements (which only carry single pathlets) and IAs that cross gulfs (which can carry many). Our translation-module implementation was simplified somewhat because we only added island IDs to IAs and omitted within-island ASes.

For Wiser, only 255 lines of additional code was needed. We only needed to create a cost-exchange service to allow downstream islands to exchange path costs with upstream ones. In a purely contiguous deployment, this need for downstream communication could be averted if upgraded ASes' border routers coordinate to sum the cost of paths they receive from direct neighbors [32].

Testing our modifications: To verify that our modifications work, we deployed Wiser and Pathlet Routing across a gulf using the topology shown in Figure 8. D-BGP, as implemented by Beagle, is the baseline. The lookup service is used to exchange IAs out-of-band. We verified that the source AS S in Island B could see important per-protocol control information for paths to the destination AS D in Island A. The islands shown use either Wiser or Pathlet Routing depending on the protocol we are testing. In Wiser's case, the lookup service is also used as cost-exchange portals for both islands.

To test our Wiser modifications, we set up path costs so that the longer path to AS D has a higher cost than the shorter one. We verified that AS D saw these path costs. For Pathlet Routing, we disseminated four one-hop pathlets to AS D within island A using its advertisement format (shown by the single dotted arrows). We configured Border AS A2 to create a two-hop pathlet out of two of the one-hop pathlets it receives. It translates this two-hop pathlet and

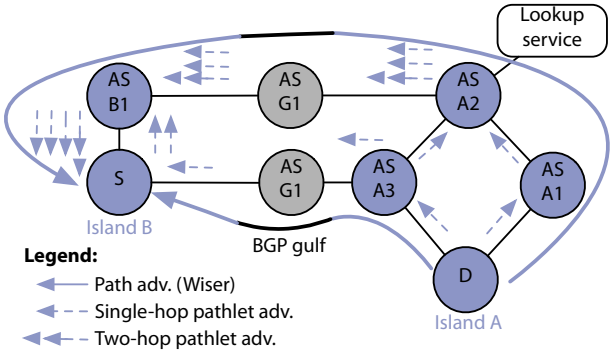

Figure 8: Topology used to deploy Wiser and Pathlet Routing.

Parameter	Variable	Ranges considered	Rationale
		General Internet topology	
# of prefixes	P	600,000 - 1,000,000	600K prefixes in tier-1 ASes' tables today; allow room for growth
# of prefixes in D-BGP's Internet	P_d	625,000 - 1,050,000	Allow for more prefixes to allow for off-path discovery
Avg. BGP path length	PL	3 - 5	Derived from analysis of routing tables [7]
		Critical fixes (CFs)	
# of critical fixes	CFs	10 - 100	Assume governing body will limit total number
Critical fixes / path	CFs / $path$	3-5	Assume one critical fix (or BGP) per hop on path
Control info / critical fix	CI / CF	4 KB - 256 KB	4 KB is max size for BGP [44]; up to 256KB for future protocols
Unique control info / critical fix	CFu	0.1 - 0.3	Most critical fixes share majority of control info w/each other
		Custom or replacement protocols (CRs)	
# of custom or replacements	CRs	10 - 1,000	Many possible because large fraction need not be regulated
Custom or replacements / path	CR / $path$	3 - 5	Assume one custom/replacement per hop on path
Ctrl info / custom or replacement	CI / CR	100 B - 10 KB	Not much info needs to be disseminated outside islands

Table 2: Parameters and ranges considered for analyzing D-BGP's control-plane overhead.

	Contribution to IA size by....			
Name	**CFs**	**CRs**	**# of advertisements**	**Total overhead**
Basic	$CFs \cdot \frac{CI}{CF}$	$CRs \cdot \frac{CI}{CR}$	P_d	
	40 KB - 25 MB	1 KB - 9.8 MB	625,000 - 1,050,000	24 GB - 36,000 GB
+ Avg. path lengths	$\frac{CFs}{path} \cdot \frac{CI}{CF}$	$\frac{CRs}{path} \cdot \frac{CI}{CR}$	"	
	12 KB - 1.3 MB	0.3 KB - 50 KB	"	7 GB - 1,300 GB
+ Sharing	$\frac{CFs}{path} \cdot \frac{CI}{CF} \cdot (CFu) + \frac{CI}{CF} \cdot (1 - CFu)$	"	"	
	4.8 KB - 0.56 MB	"	"	3 GB - 610 GB
Single protocol	$\frac{CI}{CF}$	0	P	
	4 KB - 256 KB	0	600,000 - 1,000,000	2.3 GB - 240 GB

Table 3: Control-plane overhead of D-BGP. This table shows estimated IA sizes and number of IAs that would be received at a tier-1 AS as a function of various parameters. Equations or values identical to the previous corresponding entry are marked with a ".

its remaining two one-hop pathlets into an IA and sends it across the gulf. It also redistributes the two-hop pathlet into BGP so that potential sources within gulf ASes can route traffic to the destination. AS A3 translates the single one-hop pathlet it receives into an IA and also redistributes it into BGP. Border routers at island B translate IAs they receive into Pathlet Routing's advertisement format. We verified that AS S saw all five pathlets that should be advertised to it.

6.2 Control-plane overhead

Methodology: We evaluated control-plane overhead by estimating properties of IAs that would be received at a tier-1 AS in an Internet that is using D-BGP to run multiple inter-domain routing protocols. Tier-1 ASes reside at the top of the Internet hierarchy, so they will see the highest overheads. We analyzed three types of overhead: the size of individual IAs that are received (indicative of per-IA CPU cost due to serialization at a tier-1 AS), the number of IAs that are received (also reflective of CPU cost), and aggregate size of all IAs received (reflective of total overhead and the amount of state that must be kept at the tier-1 AS). Our analysis does not account for the fact that tier-1 ASes will see multiple IAs for the same prefix. Incorporating this would inflate our calculated overheads by a constant amount.

To derive estimated IAs sizes and their number, we used characterizations of the Internet topology and protocols' expected control-information sizes culled from recent research and RFCs [5, 7, 44]. Table 2 lists key parameters, the ranges of values we consider, and our reasoning behind our choices for these ranges. Whenever possible, we chose ranges based on estimates in the literature. For parameters whose values are more uncertain (e.g., number of critical fixes), we consider a broad range of possible values to allow for future protocols' as-yet undetermined needs. We do not consider proxy aggregation because of its limited use today [9].

Results: Table 3 shows our results. IA sizes are further broken down into contribution by protocol type (critical fix or custom / replacement). For each overhead type estimated, we list a range of minimum and maximum values, derived from the equations, parameters, and values discussed in Table 2.

We find that a basic analysis that assumes individual IAs received at a tier-1 will contain information for all protocols yields very large aggregate overheads. *+ Avg. path lengths* improves this analysis by

accounting for the fact that IA size is a function of the number of protocols on a routing path, not the total number. This reduces our estimate of maximum aggregate overhead by an order of magnitude. + *Sharing* improves our analysis by accounting for the fact that many critical fixes can share the majority of their control information. This yields significant savings and reduces both our minimum and maximum estimates by an additional order of magnitude.

We also compare D-BGP's overheads with multiple protocols to the case where only a single protocol is running, which should be similar to the overheads seen today with BGP (4KB control information) or a large critical fix (256KB control information). Despite our generous assumption of 3-5 critical fixes and 3-5 custom or replacement protocols on routing paths, we find that D-BGP only adds a factor of 1.3x overhead for our minimum estimates and 2.5x for our maximum estimates. This is largely a result of the savings due to sharing of critical fixes' control information.

6.3 Incremental benefits

Methodology: We evaluated D-BGP's ability to provide incremental benefits for protocol archetypes that have different aims. To measure the benefits D-BGP can natively provide, we do not assume tunnels will be used. We do not consider benefits for protocols that require an unbroken chain of participation (e.g., secure protocols) as D-BGP cannot accelerate incremental benefits for them.

To measure incremental benefits, we simulated various archetypes' path choices on an AS-level topology in which an increasing fraction of ASes have adopted (i.e., deployed) the archetype and the rest use BGP to select paths (i.e., are in gulfs). We plotted the benefits afforded to upgraded ASes at each adoption level (the slope of which corresponds to incremental benefits). We compare two basic cases: an Internet in which BGP is the baseline and an Internet in which D-BGP is the baseline. In the former case, new protocols' control information must be dropped before sending advertisements across gulfs. In the latter case, it can be passed through.

Our topology, which is annotated with customer/provider relationships, but not peering ones, is generated by BRITE [34], configured to generate 1,000 ASes using a Waxman model (with $\alpha = 0.15$ and $\beta = 0.25$) [14, 16, 25]. We simulated various archetypes' decisions and that of BGP using a version of the simulator used by Jon et al. and Peter et al. [27, 36]. Protocols' path choices are always valley-free [38]. ASes that have not been upgraded choose paths with the shortest path length [10]. This is BGP's second path-selection criteria, ranked only below ASes' local preferences, which are opaque to us.

We consider two archetypes. The *extra-paths archetype* corresponds to protocols, such as SCION [63], NIRA [61], and Pathlet Routing [21], that only aim to expose extra information within islands. Given a set of inter-island paths to a destination, this archetype chooses the one containing the greatest number of total paths to it. Our implementation only allows each inter-island path to carry a maximum of ten paths to a destination.

The *bottleneck-bandwidth archetype* corresponds to protocols, such as EQ-BGP [6], that aim to optimize a global objective function. Given a set of paths to a destination, this archetype chooses the one with the greatest minimum per-AS bandwidth (only upgraded ASes expose their bandwidth). We chose this archetype because it is one of the most difficult objective functions with which to see incremental benefits. Its benefits depend on a single AS's bandwidth, which may be in a gulf until deployment rates are high. In contrast, some other protocols that aim to optimize a global objective, such as end-to-end latency, would see higher rates of incremental benefits.

For our experiments, upgraded ASes are chosen randomly, reflecting the ideal case of providing ASes the flexibility to deploy a new protocol independently of their neighbors. Our results reflect the average of nine trials, each with different random seeds. Benefits are plotted at adoption increments of 10% and error bars indicate 95% confidence intervals. For the bandwidth experiment, ASes' bandwidths are values on their traffic ingress links. They are uniformly distributed between a range of 10 and 1024 (the range does not affect our results much).

Results: Figure 9 shows the benefits offered to upgraded ASes by the *extra-paths archetype* as a function of the adoption level. We measure benefit as the number extra paths available to destinations at upgraded stubs. At 10% adoption, there is no difference between BGP and D-BGP baseline because there are not enough ASes that have adopted the protocol (this is a limitation of our AS-level topology, which does not expose intra-AS paths). Benefits at 20% adoption are greater than with the BGP baseline in many runs of our experiment. In cases where they are equal, there are not enough contiguous ASes to form extra paths.

We additionally see that incremental benefit (the slope of the lines) with the D-BGP baseline is greater than the BGP baseline at low adoption rates (i.e., between when 10 and 40% of ASes have upgraded). Incremental benefit for the BGP baseline exceeds D-BGP at higher adoption rates because, at this point, large upgraded islands start to connect and see massive benefits as a result of doing so. However, total benefits with the D-BGP baseline is always greater (or equal to) than the BGP baseline at all adoption levels.

Figure 10 shows the benefits afforded to upgraded ASes by the *bottleneck-bandwidth archetype*. We measure benefit as the average bottleneck bandwidth associated with the best paths chosen at each upgraded AS. The status quo line represents the average bottleneck bandwidth associated with ASes' best path choices at 0% adoption. With both the D-BGP baseline and the BGP baseline, benefits initially decline at low adoption rates compared to 0% adoption. This is because upgraded ASes are making ill-informed path choices with

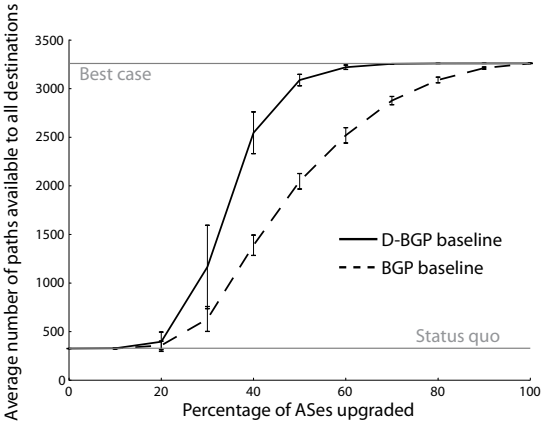

Figure 9: Incremental benefits for extra-paths archetype.

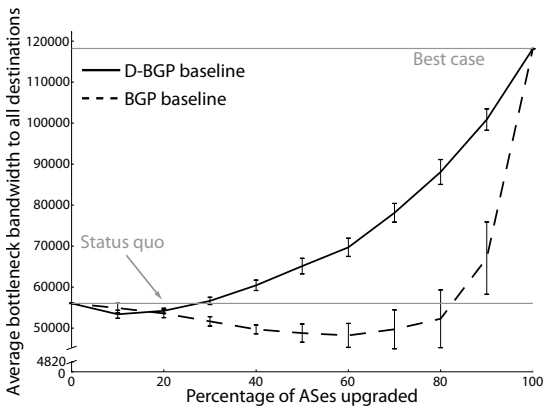

Figure 10: Incremental benefits for bottleneck-bandwidth archetype.

very little information. In contrast, our simulator's shortest-path preference (at 0% adoption) reduces the probability of encountering low-bandwidth ASes.

We see that the D-BGP baseline starts to provide benefit over the status quo at 30% adoption. This cross-over point could represent the minimum amount of participation needed to before deploying protocols as challenging as this archetype. Alternatively, with a D-BGP baseline, this protocol could disseminate extra control information that allows upgraded ASes to ignore bandwidth information until a certain number of ASes on routing paths have upgraded. Without D-BGP benefits remain lower than the status quo until 90% of ASes have upgraded. Comparing incremental benefits (the slopes of the lines), we see that the D-BGP baseline's incremental benefits are greater than that of the BGP baseline at adoption levels below 80%.

7 RELATED WORK

There has been much interest in evolving network architecture and and enhancing routing functionality. We survey them below.

Evolving network architecture: 4D [22] advocates a clean separation of network architecture along four key axes (discovery, decision, control, and data) to enable innovation. Our evolvability features enable this clean separation for inter-domain routing. Omega [42] and SDIA [41] advocate for a clean separation between intra-domain and inter-domain routing (e.g., as enabled by SDNs). They use this separation to specify a minimal interface for inter-domain routing that gives neighboring ASes the ability to run any protocol of their choosing. Incorporating our evolvability features into this interface would allow non-contiguous islands running new protocols to discover each other and avoid tunnels when routing traffic among themselves.

Ratnasamy et al. [43] describe what requirements a network architecture must satisfy to incentivize ASes to deploy new versions of IP. The requirements they identify are compatible with those identified in this paper. Several research efforts focus on evolvability for the data plane [24, 54, 55, 58, 62]. Our research complements them by focusing on the control plane.

Adding functionality externally to BGP: Due to the difficulty of modifying BGP, many systems add routing functionality externally to it. Most are operated by single domains. For example, Akamai's DNS black magic [52] monitors the performance of BGP paths from Akamai's CDN clusters to customers' DNS servers. It uses this information to select which paths to use when routing traffic to different customers. Google's Espresso [35] monitors paths to customers at peering points and interposes on BGP's forwarding decisions to route traffic on performant ones. In contrast, D-BGP and our evolvability features allow all domains to enjoy the benefits of a new routing protocol.

Software-defined exchanges [23] allow third parties (e.g., content or application providers) to interpose on BGP's forwarding decisions at IXPs. Interposition is a different technique than passing-through new protocols' control information. More research is needed to understand which method provides more benefits or whether both provide distinct ones. Many approaches, such as Arrow [36], rely on tunnels to direct traffic to domains that have deployed new protocols.

Adding functionality to BGP: Optional transitive attributes provide a form of pass-through support within BGP. They have been used to deploy limited amounts of new functionality across non-contiguous domains (e.g., a path vector capable of admitting 4-byte AS numbers). Expanding their role to include deploying new routing protocols presents a promising avenue for deploying D-BGP or selected new protocols.

Most other attempts to add functionality to BGP are targeted toward single or contiguous domains. Multi-protocol extensions to BGP [4] allow direct neighbors to name the same physical destination using different address formats (e.g., IPv4 and IPv6). BGP's community attributes [13, 48] are tags that ASes attach to paths. They allow ASes to meet internal policies (e.g., filter paths received from a specific border router at other border routers) or communicate policies with their providers (e.g., state which paths are backups). MPLS-over-BGP [45] allows customers sites connected to the same provider to exchange internal routes in-band of BGP.

8 SUMMARY

BGP cannot easily be evolved. Based on requirements identified by an analysis of key evolvability scenarios, we identified features needed to support evolvability and modified BGP to include them. Our modified version of BGP can support evolution to a wide range of critical fixes to BGP, sophisticated BGP replacements, and protocols that run in parallel with BGP to provide functionality it doesn't.

ACKNOWLEDGEMENTS & CODE LOCATION

We thank our shepherd, Barath Raghavan, the anonymous SIGCOMM reviewers, Ilari Shafer, David Naylor, Michelle Mazurek, Brent Stephens, Lily Sturmann, and Jethro Sun for their insightful feedback and comments. We thank Harshal Sheth and Andrew Sun for help with experiments. This research was funded in part by NSF under award numbers CNS-1345305, CNS-1565277, and CNS-1636563. Code used for the experiments in this paper can be found at http://www.darwinsbgp.com.

REFERENCES

[1] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient Overlay Networks. In *Proc. SOSP*, 2001.

[2] American Registry of Internet Numbers. http://www.arin.net.

[3] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. RSVP-TE: Extensions to RSVP for LSP Tunnels. RFC 3209, IETF, Dec. 2001.

[4] T. Bates, R. Chandra, D. Katz, and Y. Rekhter. Multiprotocol Extensions for BGP-4. RFC 4760, IETF, Jan. 2007. http://www.rfc-editor.org/rfc/rfc4760.txt.

[5] T. Bates, P. Smith, and G. Huston. CIDR Report. https://tools.ietf.org/html/draft-sriram-bgpsec-design-choices-01. Accessed: 2017-06-19.

[6] A. Beben. EQ-BGP: An Efficient Inter-Domain QoS Routing Protocol. *Networking and Applications*, 2:5 pp., Apr. 2006.

[7] BGP Routing Table Analysis Reports. http://bgp.potaroo.net/as6447/.

[8] BGPSec Protocol Specification Version 18. https://tools.ietf.org/html/draft-ietf-sidr-bgpsec-protocol-18f.

[9] BGPSec Design Chocies and Summary of Supporting Discussions, version 11. https://tools.ietf.org/html/draft-sriram-bgpsec-design-choices-01.

[10] M. Caesar and J. Rexford. BGP routing policies in ISP networks. *IEEE Network*, 19(6):5–11, Nov. 2005.

[11] R. N. C. Center. RIS Raw Data, 2016.

[12] H. Chan, D. Dash, A. Perrig, and H. Zhang. Modeling Adoptability of Secure BGP Protocols. In *Proc. SIGCOMM*, 2006.

[13] R. Chandra, P. Traina, and T. Li. BGP Communities Attribute. RFC 1997, IETF, Aug. 1996. https://tools.ietf.org/html/rfc1997.

[14] W. Chen, C. Sommer, S.-H. Teng, and Y. Wang. A Compact Routing Scheme and Approximate Distance Oracle for Power-Law Graphs. *ACM Transactions on Algorithms*, 9(1):4:1–4:26, Dec. 2012.

[15] Corsa SDN BGP Gateway. https://www.corsa.com/solutions/sdn-wan-bgp-gateway/.

[16] Q. Duan, E. Al-Shaer, and H. Jafarian. Efficient Random Route Mutation Considering Flow and Network Constraints. In *Proc. IEEE Conference on Communications and Network Security*, 2013.

[17] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis. The Locator/ID Separation Protocol (LISP). RFC 6830, IETF, Jan. 2013. http://www.rfc-editor.org/rfc/rfc6830.txt.

[18] N. Feamster, H. Balakrishnan, and J. Rexford. Some Foundational Problems in Interdomain Routing. In *Proc. HotNets*, 2004.

[19] I. Ganichev, B. Dai, P. B. Godfrey, and S. Shenker. YAMR: Yet Another Multipath Routing Protocol. *ACM SIGCOMM Computer Communication Review*, 40(5):13–19, Oct. 2010.

[20] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox. Intelligent Design Enables Architectural Evolution. In *Proc. HotNets*, 2011.

[21] P. B. Godfrey, I. Ganichev, S. Shenker, and I. Stoica. Pathlet Routing. In *Proc. SIGCOMM*, 2009.

[22] A. Greenberg, G. Hjálmtýsson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang. A Clean Slate 4D Approach to Network Control and Management. *ACM SIGCOMM Computer Communication Review*, 35(5):41–54, Oct. 2005.

[23] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett. SDX: A Software Defined Internet Exchange. In *Proc. SIGCOMM*, 2014.

[24] D. Han, A. Anand, F. Dogar, B. Li, H. Lim, M. Machado, A. Mukundan, W. Wu, A. Akella, D. G. Andersen, J. W. Byers, S. Seshan, and P. Steenkiste. XIA: Efficient Support for Evolvable Internetworking. In *Proc. NSDI*, 2012.

[25] Y.-Y. Huang, M.-W. Lee, T.-Y. Fan-Chiang, X. Huang, and C.-H. Hsu. Minimizing Flow Initialization Latency in Software Defined Networks. In *Proc. Network Operations and Management Symposium*, 2015.

[26] Internet Engineering Task Force. http://www.ietf.org.

[27] J. P. John, E. Katz-Bassett, A. Krishnamurthy, T. Anderson, and A. Venkataramani. Consensus Routing: The Internet as a Distributed System. In *Proc. NSDI*, 2008.

[28] E. Katz-Bassett, C. Scott, D. R. Choffnes, I. Cunha, V. Valancius, N. Feamster, H. V. Madhyastha, T. Anderson, and A. Krishnamurthy. LIFEGUARD: Practical Repair of Persistent Route Failures. In *Proc. SIGCOMM*, 2012.

[29] N. Kushman, S. Kandula, D. Katabi, and B. M. Maggs. R-BGP: Staying Connected In a Connected World. In *Proc. NSDI*, 2007.

[30] F. Le, G. G. Xie, and H. Zhang. Understanding Route Redistribution. In *Proc. ICNP*, 2007.

[31] R. Lychev, S. Goldberg, M. Schapira, and R. Lychev. BGP Security in Partial Deployment: Is the Juice Worth the Squeeze? *ACM SIGCOMM Computer Communication Review*, 43(4):171–182, Aug. 2013.

[32] R. Mahajan, D. Wetherall, and T. Anderson. Mutually Controlled Routing With Independent ISPs. In *Proc. NSDI*, 2007.

[33] Z. M. Mao, R. Govindan, G. Varghese, and R. H. Katz. Route Flap Damping Exacerbates Internet Routing Convergence. In *Proc. SIGCOMM*, 2002.

[34] A. Medina, A. Lakhina, B. Matta, and J. Byers. BRITE: An Approach to Universal Topology Generation. In *Proc. MASCOTS*, 2001.

[35] M. Motiwala, K. K. Yap, A. Vahdat, B. Koley, S. Padgett, M. Kallahalla, A. Singh, A. Narayanan, B. Tanaka, B. Rogan, C. Rice, C. Ying, D. Trumic, G. Baldus, M. Hines, A. Jain, M. Verma, M. Holliman, M. Tariq, P. Sood, J. Rahe, T. Kim, M. Tierney, V. Valancius, and V. Lin. Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering. In *Proc. SIGCOMM*, 2017.

[36] S. Peter, U. Javed, Q. Zhang, D. Woos, T. Anderson, and A. Krishnamurthy. One Tunnel is (Often) Enough. In *Proc. SIGCOMM*, 2014.

[37] Google Protocol Buffers. http://code.google.com/apis/protocolbuffers/.

[38] S. Y. Qiu, P. D. McDaniel, and F. Monrose. Toward Valley-Free Inter-domain Routing. In *Proc. IEEE ICC*, 2007.

[39] Quagga Routing Suite. http://www.nongnu.org/quagga/.

[40] Quagga: A Success, and Yet a Failure, of Open-Source in Networking? https://www.sdxcentral.com/articles/interview/quagga-project-martin-winter-interview/2014/02/.

[41] B. Raghavan, M. Casado, T. Koponen, S. Ratnasamy, A. Ghodsi, and S. Shenker. Software-defined Internet Architecture: Decoupling Architecture from Infrastructure. In *Proc. HotNets*, 2012.

[42] B. Raghavan, T. Koponen, A. Ghodsi, V. Brajkovic, and S. Shenker. Making the Internet More Evolvable. Technical report, International Computer Science Institute, Oct. 2012.

[43] S. Ratnasamy, S. Shenker, and S. McCanne. Towards an Evolvable Internet Architecture. In *Proc. SIGCOMM*, 2005.

[44] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). RFC 4271, IETF, Jan. 2006. http://www.rfc-editor.org/rfc/rfc1771.txt.

[45] E. Rosen and Y. Rekhtar. BGP/MPLS IP Virtual Private Networks (VPNs). RFC 4364, IETF, Feb. 2006. https://tools.ietf.org/html/rfc4364.

[46] C. E. Rothenberg, M. R. Nascimento, M. R. Salvador, C. N. A. Corrêa, S. Cunha de Lucena, and R. Raszuk. Revisiting Routing Control Platforms with the Eyes and Muscles of Software-defined Networking. In *Proc. HotSDN*, 2012.

[47] Chinese ISP Hijacks Internet. http://www.bgpmon.net/chinese-isp-hijacked-10-of-the-internet/.

[48] S. Sangli and D. Tappan. BGP Extended Communities Attribute. RFC 4360, IETF, Feb. 2006. https://tools.ietf.org/rfc/rfc4360.txt.

[49] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan. Detour: Informed Internet Routing and Transport. *IEEE Micro*, 19(1):50–59, Jan. 1999.

[50] B. Schlinker, K. Zarifis, I. Cunha, N. Feamster, and E. Katz-Bassett. PEERING: An AS for Us. In *Proc. HotNets*, 2014.

[51] J. Sherry, P. X. Gao, S. Basu, A. Panda, A. Krishnamurthy, C. Maciocco, M. Manesh, J. a. Martins, S. Ratnasamy, L. Rizzo, and S. Shenker. Rollback-Recovery for Middleboxes. In *Proc. SIGCOMM*, 2015.

[52] A.-J. Su, D. R. Choffnes, A. Kuzmanovic, and F. E. Bustamante. Drafting Behind Akamai (Travelocity-Based Detouring). In *Proc. SIGCOMM*, 2006.

[53] L. Subramanian, M. Caesar, C. T. Ee, M. Handley, M. Mao, S. Shenker, and I. Stoica. HLP: A Next Generation Inter-domain Routing Protocol. In *Proc. SIGCOMM*, 2005.

[54] D. L. Tennenhouse and D. J. Wetherall. Towards an Active Network Architecture. *ACM SIGCOMM Computer Communication Review*, 26(2):5–17, Apr. 1996.

[55] A. Venkataramani, J. F. Kurose, D. Raychaudhuri, K. Nagaraja, M. Mao, and S. Banerjee. MobilityFirst: a Mobility-Centric and Trustworthy Internet Architecture. *ACM SIGCOMM Computer Communication Review*, 44(3), July 2014.

[56] Q. Vohra and E. Chen. BGP Support for Four-Octet Autonomous System (AS) Number Space. Internet Request for Comments, Dec. 2012. https://tools.ietf.org/html/rfc6793.

[57] D. Walton, A. Retana, and J. Scudder. Advertisement of Multiple Paths in BGP. RFC 7911, IETF, July 2006. https://tools.ietf.org/html/rfc7911.

[58] T. Wolf, J. Griffioen, K. L. Calvert, R. Dutta, G. N. Rouskas, I. Baldin, and A. Nagurney. ChoiceNet: Toward an Economy Plane for the Internet. *ACM SIGCOMM Computer Communication Review*, 44(3), July 2014.

[59] L. Xiao, J. Wang, K.-S. Lui, and K. Nahrstedt. Advertising Interdomain QoS Routing Information. *IEEE Journal on Selected Areas in Communications*, 22(10):1949–1964, Dec. 2004.

[60] W. Xu and J. Rexford. MIRO: Multi-path Interdomain ROuting. In *Proc. SIGCOMM*, 2006.

[61] X. Yang, D. Clark, and A. W. Berger. NIRA: A New Inter-Domain Routing Architecture. *IEEE/ACM Transactions on Networking*, 15(4):775–788, Aug. 2007.

[62] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang. Named Data Networking. *ACM SIGCOMM Computer Communication Review*, 44(3), July 2014.

[63] X. Zhang, H.-C. Hsiao, G. Hasker, H. Chan, A. Perrig, and D. G. Andersen. SCION: Scalability, Control, and Isolation on Next-Generation Networks. In *Proc. IEEE Symposium on Security and Privacy*, 2011.