Better Interdomain Path Diversity with BGP Path Splicing

Murtaza Motiwala, Nick Feamster, Santosh Vempala
College of Computing, Georgia Tech

1. Introduction

Today’s interdomain routing protocol, Border Gateway Protocol (BGP) [9], scales well but suffers from two significant shortcomings. First, does not provide high availability in the event of failures or changes in network conditions (i.e., it converges slowly). Second, BGP is insecure: it provides little to no protection against malicious parties who can inject incorrect or misleading routing information into the global routing tables. Others have argued that multi-path routing can mitigate these problems, but designing a multipath routing protocol that offers a high degree of path diversity while still scaling well has proven difficult because routers must propagate multiple paths for each destination, both bigger routing tables, as well as more churn when network conditions change. However, BGP routing tables already store many routes to a single destination, which provides an opportunity for diversity without changing the routing protocol, if only end systems could somehow gain access to these alternate paths.

This paper proposes a mechanism called BGP splicing, which simultaneously—and scalably—provides end systems access to multiple end-to-end BGP routes. Even though each BGP router selects only a single best route by default, each BGP-speaking router will typically have in its routing table a number of routes to every destination that is equal to the total number of iBGP and eBGP sessions. In BGP splicing, every border router in the AS installs every route in the routing table into the forwarding tables on the line cards. As in today’s routers, packets can still be forwarded along a default path, but extra bits in the packet header can cause the router to forward packets along a different path by using a different forwarding table entry.

By installing alternate routes in the forwarding table and letting end hosts select among them, BGP splicing trades off more state in router forwarding tables for improved path diversity without introducing any additional protocol complexity. Although BGP splicing could be implemented at all BGP-speaking routers, autonomous systems (ASes) can gain most of the benefits of BGP splicing by deploying splicing at ingress points (whereby BGP splicing can allow end systems to control the path across an AS to a particular egress point) and egress points (whereby end systems can control the next-hop AS along a path to a destination) would be sufficient to achieve the benefits of BGP splicing.

BGP splicing offers many potential benefits, including improved availability (through fast access to backup paths), better security (through the ability to simultaneously compare reachability on multiple paths), and higher throughput (through simultaneous access to multiple paths).

Despite its conceptual appeal, BGP splicing faces many practical challenges. First, line cards must provide direct support for storing multiple routing table entries for a single destination, which not only requires more memory on line cards but also requires using a few extra bits for packet lookups, which might slow packet forwarding. Second, the BGP route selection process must be altered so that end hosts can use splicing bits to gain access to the best set of alternate paths that still comply with an AS’s policy. For example, a router should prefer all customer routes before all provider routes, and, among its set of most preferred routes, and, among the set of most preferred routes, it should rank routes so that spliced paths have high AS-level path diversity. Finally, because BGP splicing can deflect routing traffic onto a different “tree” at any ingress or egress router, it introduces the possibility of longer forwarding paths, and even forwarding loops. We present scenarios where splicing can introduce loops and mechanisms for detecting and preventing loops.

The rest of this paper is organized as follows. Section 2 presents an overview of the general path splicing mechanism, and Section 3 explains path splicing as applied to BGP, highlighting in particular the differences between BGP splicing and path splicing, as well as the required router-level support for BGP path splicing. Section 4 presents safeguards against forwarding loops, Section 5 describes several possible applications of BGP path splicing, Section 6 summarizes related work, and Section 7 concludes.

2. Path Splicing

Path splicing exposes an exponential number of diverse end-to-end paths to end systems with only a modest increase in routing protocol complexity. The original splicing idea (summarized in our previous work [2]) proceeds as follows: First, \( k \) instances of the same routing protocol run over a single network topology; each protocol instance runs with a slightly perturbed version of the configuration (in the case of a shortest paths routing protocol such as OSPF [8], each instance would run with link weights that were slightly perturbed from the original values). Running \( k \) instances of the protocol allows the construction of multiple routing trees over the same fixed topology.

Second, to achieve an exponential gain in the number of available paths, each packet carries routing bits (as shown in Figure 2) that allows that packet to be switched to a different slice at any hop en route to the destination. To send its traffic along a different path to the destination, a host need only change the routing bits in the header.
Our previous work discusses path splicing in the context of intradomain routing, whereby multiple paths can be constructed by running multiple OSPF or IS-IS instances, each time however with a different set of link weights. In that work, we demonstrated that randomly perturbing link weights in each slice can dramatically improve network reliability without prohibitively increasing “stretch” (i.e., latency increase) for each path. Path splicing works end-to-end (each AS’s intradomain path can be spliced as the packet traverses the end-to-end path), but it does not enable splicing of interdomain paths. Fortunately, interdomain routing protocols often already learn multiple disjoint paths to each destination (one over each BGP peering session), but today they only select a single path. In the next section, we explore how splicing can be extended to BGP to allow end hosts to take advantage of the multiple alternate paths already in each router’s routing table.

3. BGP Path Splicing

BGP splicing exploits the fact that each AS learns multiple routes to a prefix from its peers and thus has access to potentially multiple paths to a destination prefix. This section describes BGP splicing. We begin with an overview of BGP splicing, as well as modifications to existing routers and routing protocols that are required to support BGP splicing. We also discuss an incremental deployment scenario for BGP splicing and show that even in such a situation there are significant advantages to be gained.

3.1 Overview

Each router’s routing table may contain multiple entries for each prefix. Some of these routing table entries may correspond to alternate highly disjoint paths in the network, as shown in Figure 1. To allow end systems to use these potentially diverse paths, BGP splicing uses a small, fixed-size header in the packet to give end-users control to change the path followed by their packets at intermediate hops en route to the destination. Splicing thus allows the network to use the information already available from BGP to achieve multi-path diversity.

BGP splicing requires no extra exchange of messages or changes to routing protocols as required by multi-path proposals like MIRO [14], R-BGP [6]. If done by every AS, this provides the same functionality as MIRO without requiring any extra exchange of messages or modifications to BGP messages (as required by R-BGP), at a cost of a linear increase in \( k \) (the number of alternate paths) in router memory for storing the forwarding tables.

3.2 Router Support

BGP splicing requires several changes to today’s routers. First, a router must be able to install forwarding tables on each line card from the RIB. Second, the BGP decision process needs to be modified so that it now selects \( k \) “best” policy-compliant paths instead of just a single best path. Finally, the line cards also need to be modified to read extra routing bits from the packet header. These bits are used to indicate which forwarding table must be used by the router to forward the packet, i.e., which of the \( k \) paths the router should use at any given hop to forwarding the packet. The rest of this section describes these changes in detail.

3.2.1 Routing bits

We introduce extra bits in the packet header (routing bits), as shown in Figure 2, in order to perform the splicing operation. These bits are inserted by the source and are read by both the ingress and egress routers in each AS. Thus, an \( n \)-hop AS path requires \( 2n \cdot \log(k) \) routing bits. The following two cases describe how:

- **Ingress Router**: The ingress router learns multiple paths to a destination prefix from the various border routers using iBGP (either via a “full mesh” iBGP or via its connections to multiple route reflectors) and thus may learn multiple exit points (“egress routers”) from the network for each destination prefix. For each packet, an ingress router reads the rightmost \( \log(k) \) routing bits to determine which egress router should receive the packet.

- **Egress Router**: An egress router learns of multiple paths to a destination prefix from the various border routers of the neighboring ASes using eBGP. It uses the rightmost \( \log(k) \) routing bits to determine which of the \( k \) eBGP-learned routes (i.e., which forwarding table entry) to use.

After reading the corresponding bits, each ingress or egress router removes the rightmost bits from the splicing header to allow the next BGP-splicing router to read the bits corresponding to its hop.

3.2.2 Changes to BGP route selection

Instead of determining only the best route for each prefix, each router must compute the best \( k \) paths. A naïve way to select \( k \) paths for each prefix is to remove the best route from the routing table and apply the BGP decision process again to select the next best path and so on, but this approach could result in a second-best path that is similar to the original best path: for example, if the best path is \( (6530, 321, 2231, 4) \), second-best path might be \( (5432, 321, 2231, 4) \), which has 3 common hops.
BGP splicing would benefit from a selection mechanism that favors AS-disjoint paths over paths with many common AS hops, subject to the constraints that all customer routes are preferred over all peer routes, and so forth, per business constraints [4]. Selecting disjoint paths may also increase the chances of introducing forwarding loops in the "spliced" path, as we discuss further in Section 4. To solve these problems, splicing first attempts to find \( k \) best paths only among the set of customer routes, followed by peer routes and finally provider routes; we explain in the next section why, if routes are preferred subject to these constraints, loops are unlikely. Operators might also tune the amount of preference or policy violations allowed by BGP splicing to allow more disjoint paths to be selected.

### 3.2.3 Multiple forwarding tables in line cards

A router's Routing Information Base (RIB) contains all the routes announced to the AS for every prefix. Splicing requires the BGP route selection process to select the \( k \) best paths from the RIB and insert these entries into forwarding tables on the router line cards, which requires that the line cards provide support for multiple forwarding tables. When a packet arrives at the ingress interface of the card, the routing bits from the header are read and indexed to the corresponding forwarding table, which requires additional logic to read the routing bits from the header and use those to select an entry from the corresponding forwarding table and use that for forwarding the packet. Although these changes may seem substantial we note two features: first, only border routers require these modifications; second, the protocol itself requires no modifications.

### 3.3 Deployment Concerns

BGP splicing can be incrementally deployed in the Internet. Because splicing requires no changes to how BGP messages are exchanged, a splicing-enabled network still interoperate with other networks that have not deployed splicing. Splicing’s routing bits can be stored in a separate shim header, which sits between the IP and transport layers. Routers that do not support BGP splicing have no knowledge of the shim header’s semantics and forward the packet using the default BGP path.

BGP splicing may complicate traffic engineering because network operators like to control the way traffic enters and leaves their networks. Splicing gives the source some control over how its packets are forwarded by the network, which may make traffic engineering more difficult. BGP splicing can also introduce the possibility for forwarding loops; we discuss this possibility and mechanisms for preventing loops in the next section.

### 4. Loop Prevention and Detection

Because BGP splicing constructs a single end-to-end path from multiple routing trees, it introduces the possibility for forwarding loops. This section describes examples where BGP splicing can introduce loops, as well as mechanisms for preventing them.

Loops can occur among peers or in customer-peer-provider relationships, as illustrated in Figure 4. In Figure 4(a), consider the scenario in which AS \( S \) chooses the route through one of its providers to forward the packet to \( D \). When the packet reaches a provider (A, B, or C), even though each AS has a direct path to \( D \), the routing bits may cause the packets to be forwarded among a sequence of peers, resulting in a forwarding loop. Note that such a loops can occur even if standard preference and filtering rules are applied [4], since the actual forwarding graph is an overlay of three separate policy-compliant routing trees. Figure 4(b) shows the possibility of a 3-hop loop between ASes \( S \), \( A \) and \( B \) which can arise if \( S \) selects a path from its provider \( A \), while \( B \) chooses a path through its peer \( S \).

In each of these cases, a packet will not loop forever, because there are a limited number of routing bits in the header. Once the routing bits get exhausted, the packet is forwarded on the default loop-free BGP path. Nevertheless, in the interdomain case, we treat any loop as undesirable, since even a loop involving two ASes may traverse a significant distance. Accordingly, we propose the following two mechanisms to limit the extent and occurrence of forwarding loops.

#### 4.1 Solution 1: Include AS Path in Packet

One approach to detecting loops involves inserting 16-bit hashes of each of the first four hops of the AS path that the
packet traverses in its header. We choose four since most paths in the Internet pass through four ASes or fewer. An AS’s border router can examine these bits and avoid selecting a next-hop AS that has already been visited by the packet (unless there is no other route available). This mechanism does not prevent loops altogether, but it does limit the extent to which packets can be caught in a loop. Of course, the mechanism only prevents short loops (i.e., those less than four AS hops), but the average length of Internet paths and standard policy constraints (i.e., preferring customer routes over peer routes, etc.) make long loops unlikely.

4.2 Solution 2: Deflection Counter

To deal with larger loops, we introduce a 2-bit “counter” in the header. We observe that forwarding loops can only occur when a packet is deflected from a best customer (or peer) route to a peer/provider (or provider) route (i.e., AS-level loops are not likely on spliced valley-free paths, except for the case of an all-peer loop). Accordingly, we introduce a deflection counter to limit the number of times a packet is deflected from its most preferred class of routes: If a router in some AS has a best path to a destination through its customer but it instead chooses a peer or provider path for forwarding a packet (or a provider route instead of a peer route), that router decrements deflection counter of the packet. This mechanism bounds the number of times the packet can be deflected and prevents a packet from being repeatedly forwarded “uphill” (which would be required for a persistent loop). An end system that has no tolerance for loops may set this counter to zero; increasing this counter increases a router’s flexibility in choosing paths that are not policy-compliant, at the cost of increased potential for routing loops.

5. Benefits

In this section, we describe the benefits of BGP splicing, including increasing improving availability, increasing throughput, and making the network control and data planes more secure.

5.1 High Availability

BGP splicing improves availability by allowing end systems to use paths through the network other than the default end-to-end path. When nodes or links fail, and end system can send traffic to a destination via an alternate path as long as some combination of edges in the routing tables allow an end system to construct a path to the destination. Thus, during times of network disruption, end systems can simply select a new path through the network, rather than relying on routing protocols to detect, recover, and converge after a network fault. Whereas BGP convergence can take tens of minutes, BGP splicing allows an end system to react to failures and change to an alternate path as soon as it notices the fault (possibly as fast as a single round-trip time).

5.2 Increased Throughput

An end system could use multiple spliced paths in parallel to improve throughput to a given destination. BGP splicing’s routing bits allow an end host access to multiple paths in parallel. By sending packets in a data stream where different paths have different routing bits, a source can send different packets along different paths. Allowing the source this access to multiple paths thus allows better overall use of network capacity: rather than being constrained by a single bottleneck link, capacity is constrained by the minimum cut of the graph between a source and its intended destination.

5.3 Improved Security

Allowing sources access to multiple paths in parallel can also help make the routing system more robust. In essence, routing bits allow a source to “test” different routes in the routing table to the same destination. Access to multiple paths allows a source to compare different routes for potential inconsistencies (i.e., whether they all go to the same destination); as previously discussed, it also improves availability, which previous work has argued may help improve routing security. Finally, splicing’s ability to redirect traffic “on the fly” may help network operators more easily shunt traffic in the event of denial-of-service (DoS) attacks.

5.3.1 Defending Against Route Hijacks

Attackers can “hijack” (i.e., falsely advertise) BGP routes to divert traffic destined to a different host to their machine. Today’s routing infrastructure does not defend against route hijacking. Although S-BGP proposes to deal with these types of attacks, the security mechanisms pose considerable deployment barriers. Availability-Centric Routing (ACR) [12] argues that instead of augmenting routing protocols with heavyweight cryptographic mechanisms, the routing infrastructure should focus on providing high availability (specifically, ensuring that as long as there exists some path through the network to an endpoint that the routing protocol can expose that path to the end host), leaving the task of authenticating endpoints to end-to-end cryptography. BGP splicing could serve as a mechanism for implementing an ACR-type scheme because it exposes multiple paths to end systems and lets them choose among many possible options.

By allowing end hosts access to multiple disjoint BGP paths in parallel, BGP splicing could allow end hosts to test multiple independent paths to the destination in parallel and check them for consistency. For example, an end system could send a sequence of messages to the same destination address but change the routing bits in the header for each message; by sending a sequence of messages that require responses from the destination that depend on seeing the previous message (e.g., a challenge-response scheme), a source can quickly determine whether multiple routes to a destination are consistent (i.e., whether it is talking to the same endpoint over multiple routes).

5.3.2 Defending Against Denial-of-Service Attacks

Fundamentally, splicing leverages the use of multiple parallel, but slightly perturbed, routing trees to a destination to help achieve path diversity, but not ever set of routing tables necessarily needs to have the same function or amount of routing information. In Mayday, Andersen suggests a general framework whereby a server could be protected by a group of access nodes, which drop traffic for the destination.
achieve high end-to-end availability by using available alternate routes in BGP forwarding tables. Through its use of routing bits, BGP splicing exposes alternate paths to end hosts without using any extra propagation of routing messages between ASes and any without any changes to the BGP protocol. BGP splicing requires modifications to the BGP decision process for selecting the best path and require changes to the line cards in routers to support multiple forwarding tables. The path diversity that splicing provides offers several benefits, including providing more routing security, dealing with Denial-of-Service (DoS) attacks and increasing end-to-end network throughput.

6. Related Work

This section describes proposals for exposing multiple paths to end systems, most of which are motivated by fast failure recovery, as well as related previous work on applications of BGP splicing (specifically, security of control and data planes). Previous work has proposed modifying both routing and forwarding to expose multiple alternate paths to end hosts; BGP splicing is an extension of the path splicing ideas that we have presented in earlier work for intradomain routing [2]. The “routing bits” mechanism that we use in splicing is similar to that proposed by Yang et al., who propose that sources use additional bits in packet headers to deflect packets off of their default forwarding path [15]. Unlike splicing, the routing deflection mechanism does not install alternate forwarding tables but redirects packets on an alternate path.

A recent proposal for multi-topology routing using multiple router configurations suggest precomputing backup topologies based on combinations of failed links and switching to those precomputed topologies when failures occur [7]; MRC does not allow traffic to switch topologies midstream as in BGP splicing. Other proposals have suggested using alternate paths learned through BGP route advertisements to expose multiple paths to end systems, either by precomputing failover paths [6] or by having ASes request alternate paths from the BGP tables “on demand” [14].

As discussed in Section 5.3, BGP splicing’s ability to provide access to multiple paths in parallel makes it a natural mechanism for deploying schemes that seek to achieve routing security by improving availability between endpoints (e.g., ACR [12]), which stand in contrast to routing protocols that rely on lightweight cryptography to verify the authenticity of each routing message [5, 11, 13]. BGP splicing also allows for consistency checks (i.e., to determine whether multiple paths lead to the same end host), which is similar in spirit to the control-plane consistency checks suggested by Listen and Whisper [10].

7. Conclusion

This paper has presented BGP splicing, a method for achieving high end-to-end availability by using available alternate routes in BGP forwarding tables. Through its use of routing bits, BGP splicing exposes alternate paths to end hosts without using any extra propagation of routing messages between ASes and any without any changes to the BGP protocol. BGP splicing requires modifications to the BGP decision process for selecting the best path and require changes to the line cards in routers to support multiple forwarding tables. The path diversity that splicing provides offers several benefits, including providing more routing security, dealing with Denial-of-Service (DoS) attacks and increasing end-to-end network throughput.