Dynamic Partition Healing in Internet Routing

Alan Keith
Colgate University
Hamilton, NY, USA
akeith@colgate.edu

Vijay Ramachandran
Colgate University
Hamilton, NY, USA
vramachandran@colgate.edu

Colgate University Technical Report
TR–CS–2011–01

Abstract

IP-routing protocols assume that the subnetworks they connect are always contiguous; however, in next-generation networks (including mobile or wireless settings) where this assumption may be violated, networks may be unnecessarily partitioned into disconnected zones, even though viable router-level paths exist to maintain connectivity. In this paper, we propose alternate loop-detection and protocol-configuration paradigms for Internet routing protocols that enable dynamic healing of network partitions in an incrementally deployable manner. We use our two proposals to discuss performance tradeoffs inherent in partition-healing methods, including effects on routing-table size, control-plane overhead, and path efficiency.

I. Introduction

Today's Internet design suffers from the *network partitioning* problem [1], in which connectivity is lost when a routing domain is split into isolated parts, even though there exist data-forwarding routes to maintain connectivity. This is because the computation of routes is divided hierarchically into inter- and intradomain routing (with intradomain routing sometimes further divided by using multiple protocols or areas), and protocols at each level of the hierarchy lack the required visibility of the global, router-level topology to circumvent a network partition. Although this division of route computation has worked well in the current Internet, yielding benefits such as scalability, autonomy, and curtailed control-plane overhead, the underlying assumption that subnetworks at a given level of the hierarchy are contiguous is less true in next-generation networks, *e.g.*: (1) a routing domain consisting of geographically separate subnetworks or containing fragile cut edges; (2) a military tactical subnetwork in which intra-network links are unreliable, but connectivity can be provided through adjacent networks; (3) networks in which devices are mobile and cannot be reconfigured each time they move to a different region. Resilience to network partitions is increasingly important as policy-routing protocols are considered for deployment in networks with characteristics that are different than the traditional wired Internet [2]. Current methods to combat network partitions take for granted protocols' contiguity assumptions and amount to some combination of the following approaches:

Redundancy: Additional physical or virtual links, or the clever use of additional routing-protocol instances, ensure that routing domains are always connected through a designated fault-tolerant "backbone" subnetwork, thus maintaining the availability of paths that, as perceived by routing protocols, are free of loops [3], [4]. Unfortunately, this requires some advance thought to identify backbone components and the susceptible connections which require redundancy; in addition, this approach also incurs a cost (in materials or amount of configuration) to create and maintain the additional links.

Outsourcing: Networks unable to maintain the required level of redundancy can cooperate with service-provider networks who, through supplemental protocols, protocol extensions, or alternate forwarding techniques, can maintain connectivity among network components subject to partitioning (or are already partitioned due to, *e.g.*, geography) [5]. This approach cannot be deployed unilaterally by a network and may also come at an additional cost.

Manual intervention: Rather than maintain a resilient configuration, change the configuration once a partition is detected, *e.g.*, reassign network components to different routing areas or create a temporary virtual link [6]. Unfortunately, connectivity will be lost until a repair can be made, and any repairs may cause unforeseen side effects.

Interestingly, these "solutions" seem to direct route computation away from routing protocols, instead using more complex configuration techniques and/or human intervention to achieve what a routing system should ostensibly be able to do. In this paper, we propose a simpler approach involving changes to routing protocols' configurations and their loop-detection mechanisms to permit dynamic reconnection of network partitions. Our scheme is incrementally deployable by networks wishing to improve resilience and suggest principles useful in the design of routing protocols for, or the application of current routing protocols to, next-generation networks.

A. BGP and AS splits

The Border Gateway Protocol (BGP) [7] connects autonomous systems (ASes), assuming that ASes themselves are contiguous networks. Its standards document [7] states: "If the AS_PATH attribute of a BGP route contains an AS loop, the BGP route should be excluded.... AS loop detection is done by scanning the full AS path (as specified in the AS_PATH attribute), and checking that the autonomous system number of the local system does not appear in the AS path." Unfortunately, this loop-detection mechanism can lead to unnecessary network partitions. Consider the network fragment shown in Fig. 1, containing five ASes. Only the internal view of AS 2 has been sketched. Assume that the border routers A and B run and IGP to learn about internal destinations, external BGP (eBGP) with routers in other ASes to learn about external destinations, and internal BGP (iBGP) to share eBGP-learned information with each other. Assume that AS 2 is configured to advertise the address space 2.0.0.0/16; both A and B may separately do so via eBGP.

Suppose the dashed internal path in AS 2 fails, isolating A and B from each other; they no longer share information via an IGP or iBGP. Although AS 3 and AS 4 have viable paths to AS 1 and AS 5, respectively, A rejects the path to AS 5, and B rejects the path to AS 1, because the corresponding AS-level paths contain loops (*e.g.*, 2(A)–3–4–2(B)–5). Furthermore, if aggregation is used to advertise AS 2's address space, both A and B will continue to

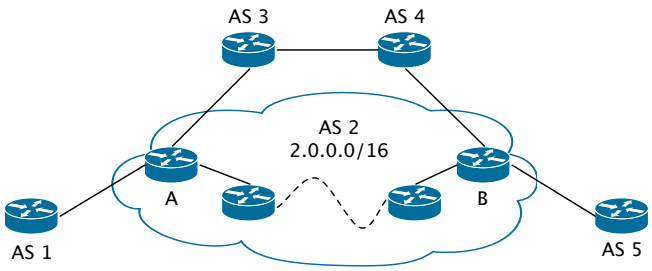

Fig. 1. Example network; links represent direct physical connections while the dotted line represents an intradomain path.

advertise the entire /16 subnet: This is because advertising an aggregate requires just one of the subnet's components to be present in the forwarding table. This is problematic, as AS 3 and AS 4 will send all traffic for AS 2 to just one of the border routers, regardless of whether the destination can be reached from there, even though viable paths to all destinations in AS 2 exist from these ASes.

B. Multi-area OSPF and backbone splits

OSPF [8] is one of the most commonly used Interior Gateway Protocols (IGPs) to establish intradomain connectivity. It is a link-state protocol: Routers broadcast update messages containing link weights to enable shortest-path computation, giving naturally loop-free routes (excising a loop from a path yields a shorter one). However, due to control-plane overhead, it is recommended to split the broadcast region among multiple OSPF areas, limiting the number of routers in each to 50 [3]. Link-state updates are broadcast within a single area only; connectivity among different areas is established by summary updates exchanged by area border routers (ABRs) belonging to more than one area. These summary updates essentially "advertise" destinations present in one area to another without sharing topology details; this likens inter-area route distribution to distance-vector routing, requiring loop detection [6].

This is achieved by designating a "zero area" to serve as the OSPF backbone. ABRs then exchange summary updates in a restricted manner [8], [9]: Destinations from nonzero areas are made available to the backbone, and backbone destinations are made available within (but not beyond) nonzero areas. In particular, summary updates do not propagate between nonzero areas. Reachability between routers in different areas is permitted only when there exists an inter-area route transiting the backbone; non-backbone inter-area paths can only be used to improve the efficiency of an existing backbone inter-area paths. This precludes forwarding loops by imposing a two-level hierarchy on the OSPF-area topology but assumes the backbone never splits: Once partitioned, isolated parts of the backbone cannot use nonzero areas to establish connectivity, because nonzero areas cannot rebroadcast summary updates.[1]

[1]Certain router vendors permit ABRs without a backbone connection to reach adjacent areas [10]; but, our emulations showed that ABRs connected to multiple backbone partitions are unable to establish connectivity to any of those partitions, in turn losing connectivity to other areas. Thus, backbone partitions are still a problem with these OSPF implementations.

C. Our Results

Unnecessary network partitions occur when protocols take for granted connectivity within the subnetworks they connect, and, when that assumption fails, they do not offer the ability to repair a connection using alternate routes. In this paper, we focus on the general case of using available interdomain paths to reconnect separated partitions of an AS. This cross-protocol approach is novel and addresses many failure scenarios. (The use of multiple protocols or routing processes for resiliency using only intradomain paths has been studied, *e.g.* as in [4], but the use of interdomain routes less so; furthermore, the former often involves complex configuration but is limited in the types of solutions possible.) We present two such configuration paradigms that exhibit a tradeoff among performance characteristics, such as path efficiency and routing-table size, inherent in implementing a partition-healing scheme.

II. Assumptions and Analysis Criteria

Routing inter-partition traffic across interdomain paths requires a viable router-level path between the partitions; our approach will not work on ASes with a single interdomain link. It also requires at least one unique subnet advertisement from somewhere in each partition; otherwise, nowhere in the interdomain-routing system can the partitions be distinguished as separate routable entities. Therefore, we require that aggregation of prefixes is limited only in the following way: There must be a nonempty set of contiguous ASes adjacent to the partitions that relay partitions' advertisements to each other without aggregation. This set can be small: *e.g.*, if there is a single AS adjacent to the partitions, that AS can aggregate announcements to other ASes, as long as the partitions receive route announcements identifying each other through the adjacent AS. Thus, any overall growth in routing-table size can still be mitigated by route aggregation. We also require that BGP speakers peering with partitions do not practice sender-side loop detection (SSLD); otherwise, routes required for partition healing will not be learned because they would contain an AS-path loop when announced to the partitioned AS. Fortunately, SSLD is not widely used and is disabled in major vendors' implementations by default.

Analogously, if an IGP is configured to aggregate subnets, dynamic de-aggregation must be possible in the event that the aggregated address space is partitioned; otherwise, the partitions will not appear as distinct routable entities. We also assume that the IGP provides connectivity from each interior router to a border router, so that an interdomain partition-healing path may eventually be used; however, it does not matter whether the network administrator accomplishes this through redistribution of healing routes into the IGP or through default routes and gateways.

We insist that any partition-healing scheme does not change computed routes when networks are not partitioned; in particular, we ensure that intradomain paths are used to carry traffic whenever they are present and would have been used in the absence of any partition-healing interdomain paths. Most importantly, we ensure that our approaches do not introduce any forwarding loops: Although loop detection can be turned off in many vendors' BGP implementations (yielding visibility of interdomain partition-healing paths), doing so without additional precautions may cause forwarding loops [1] or exhibit count-to-infinity behavior as seen in RIP [11].

The next two sections present two different approaches to healing network partitions with interdomain paths. The approaches mainly differ on the metric of *path efficiency*, which we take to mean the increase in path length from the best router-level partition-healing path (as would be determined by IGPs, BGP, and policy) to the path induced by our approach. The first approach achieves good path efficiency by increasing visibility of the partitioned network components at the AS level. Unfortunately, there are tradeoffs among path efficiency and other desirable metrics such as impacts on routing-table size and control-plane overhead (in particular, potential for BGP update churn). Our second approach sacrifices path efficiency for reduced impacts on table size and overhead. To our knowledge, our work is the first to define relevant metrics for partition-healing approaches and to propose general solutions while examining the tradeoffs among them.

III. Partition-level Loop Detection

At the core of our first configuration paradigm is augmentation of BGP's loop-detection mechanism so that disconnected partitions are treated as separate subnetworks at the AS level.

We propose that BGP-speaking routers tag all advertised routes within an optional, transitive BGP attribute. The tag (ASN, ID) consists of the AS number of the router and an identifier recognizable by that router. To perform loop detection, routers check this attribute for an identical (ASN, ID) tag instead of checking a route's AS-path attribute for a router's own AS number. Furthermore, one of the following two actions must be taken: (1) for all $k > 0$, routes containing k occurrences of tags from the same AS number must have lower preference than routes containing fewer than k tags;[2] or (2) if identifiers are recognizable by other routers in the same AS, routes tagged by routers that are reachable via a wholly intradomain path should be filtered (although this option requires additional hardware support). The granularity of loop detection grows with the number of identifiers used: Using a single identifier for all routers makes this loop-detection scheme equivalent to BGP's existing AS loop filtering; using a distinct identifier for each router permits maximum flexibility in partition healing. (An ideal balance is one identifier per partition, but this would of course be hard to pre-configure without knowing where splits might occur.) Note that the presence of identifiers reveals little about the internal structure of a network: (1) identifiers only need to be recognized by the router that placed the tag, so their values need not have public meaning; (2) the assignment of identifiers to routers can be adjusted to reveal more or less about the mapping of routes to ingress points; and (3) only BGP-speaking routers place tags, thus the number of identifiers is limited and internal-router information is not included.

The AS-path attribute should still be modified in the usual way to ensure compatibility with routers and ASes not adopting the augmented loop-filtering scheme. Furthermore, note that this scheme is incrementally deployable: the only changes needed to make partition-healing paths available are those made to routers in the partitioned AS itself.

[2]This divides routes into preference classes determined by number of tags from the router's own AS; however, any other route preferences dictated by policy can be applied between routes in the same class.

Example: Regarding the network in Fig. 1, suppose AS 2 implements partition-level loop detection: any routes advertised by A or B contain tags (2,A) or (2,B). Consider the physical path from source AS 5 to destination AS 1 via AS 4 and AS 3, which contains an AS-level loop and would be filtered by BGP. When announcement of this path reaches router B, it will contain the tag (2,A); because this does not match router B, it will be accepted. If AS 2 is not split, then this route with egress point B will be de-preferenced at B so that it has lower rank than the route with egress point A (presumably learned at B through an iBGP session), which prevents a forwarding loop. If AS 2 is split, this route restores connectivity between B or AS 5 and AS 1.

Two issues remain. First, if partitions of an AS continue to advertise the same address space, other ASes may send data to the wrong partition. Thus, we require that only subnets present in a contiguous partition of an AS be advertised by border routers in that partition; unfortunately, this has the potential to increase BGP table size without dynamically adjusting advertisements (which would cause update churn). However, there is a tradeoff between table size and granularity of partition healing. At one extreme is redistribution of IGP routes into BGP, automatically advertising subnets present in each partition; however, this would significantly increase routing-table size and control-plane overhead. At the other extreme is assigning subnets to be advertised by a particular border router ahead of time, but such a solution is not fully dynamic (although it is no less so than current approaches to multi-homed traffic engineering); however, if part of a subnet is stranded in the partition of a non-designated border router, connectivity to that component will be lost because its advertised prefix will originate elsewhere. The best solution from a table-size perspective is to advertise an aggregate prefix unless a partition occurs, when border routers start advertising more specific subnets. This causes negligible impact on the routing table when the AS is contiguous, but results in update churn when a split occurs; furthermore, because the more specific routes need to propagate through BGP, partition-healing time may be long. Such an approach would require support for a special type of conditional advertisement. Although some versions of Cisco IOS support this today, some complexity is needed to achieve the correct trigger conditions, in particular, a combination of route redistribution, export filters, and advertisement route maps.

Secondly, routers may have to decide among routes to internal destinations learned from different protocols; prior to our loop-detection change, only IGP routes would have been available. Routers should prefer IGP-learned routes to internal destinations over BGP-learned ones; this ensures that intradomain traffic only uses interdomain routes when necessary. This can be achieved in several ways: (1) the administrative distances of protocols can be set to ensure this ordering; (2) subnets in the IGP can have longer prefixes (packets are forwarded to the most specific subnet); (3) configuring backdoors or using a type of conditional advertisement to prevent BGP routes from being used or shared if the destination subnet is already in the forwarding table.

IV. Interdomain-routable Tunnel Overlays

Our second partition-healing scheme does not require an additional BGP attribute, but sacrifices path efficiency. The main idea is an overlay network of tunnels among BGP-speaking border routers in an AS and use modified loop-detection so that tunnels can reroute over interdomain paths in the event of a network partition. Configuration

of tunnels may seem complex, but many ASes already use tunnels to avoid injecting iBGP routes into the IGP: By using tunnels to direct interdomain traffic to the appropriate egress router, consistent forwarding-table entries for interdomain destinations are not required on all routers in the AS.

We propose that each border router originates BGP announcements for the AS as usual, along with a specific prefix associated with its tunnel loopback. This limits the increase in the BGP table size to linear in the number of border routers, which is linear in the number of supported partitions. We acknowledge that advertisement of the tunnel loopback address into BGP may violate current "best practices," which discourage advertisement of long prefixes into BGP (*e.g.*, subnets more specific than /24) for security and scalability reasons. However, we note that the modest increase in table size should allow announcements for this purpose (which could be specially tagged as such), and that an implementation in a next-generation network may facilitate these types of announcements. Furthermore, particular scenarios may make this approach viable: Segments of the network owned by a service provider may permit a contracted set of tunnel-endpoint announcements as a service to customers, supporting dynamic partition healing; alternately, military- or government- owned sections of the network that highly value resiliency may be able to deploy this technique across selected subnetworks.

The BGP announcement of the tunnel loopback permits the endpoint to be reached via an interdomain path, assuming that BGP loop detection is modified to permit consideration of the tunnel-loopback announcements. In particular, we can implement the following rules: (1) Disable BGP loop detection, allowing paths with a router's own AS number; but (2) implement an additional route filter as a substitute that discards any routes containing the router's own AS number unless the destination address matches a tunnel loopback other than the one advertised by the router itself. Such routes, if accepted, should never be re-advertised to any BGP peers.

Information on network reachability will need to be shared through the tunnel. The specific tunnel configuration can be tailored to suit the needs of a given network topology, but there are some general guidelines that all configurations share. If administrators desire that traffic traverses the tunnel even when the network is not split (*e.g.*, to make interdomain destinations available without injecting forwarding-table entries for them), each tunnel's endpoints must appear in the IGP via a more specific subnet or a more preferred route—because of administrative distance or a backdoor configuration—than in BGP. Otherwise, if BGP's routes or prefixes take precedence, the tunnel will always travel outside the AS, regardless of the presence of an internal path. The higher-priority route must not be advertised through the tunnel, otherwise any wrapped packets will continuously be sent back to the tunnel interface in a misguided attempt to reach the tunnel destination address, a problem known as *recursive routing*. Although any preference adjustments are straightforward using mechanisms available on routers today, we note that if the tunnel overlay is only used for partition healing, fewer changes are needed: If the tunnel link is weighted high in the IGP, other intradomain routes, if present, will take precedence.

Administrators can choose how border routers share intradomain routes through the tunnel, depending on the chosen IGP(s). Protocols such as RIP [11] can prevent the tunnel loopback prefixes from being advertised through the tunnel interface; in such cases, the IGP can simply be run through the tunnel. OSPF's backbone dependency and lack of intra-area route filtering make it a less than ideal choice to extend through the tunnel; though not

impossible, it may be more beneficial (but resource-intensive) to run a different protocol through the tunnels and redistribute desired routes into it.

Example: Consider again the network in Fig. 1; to implement the tunnel overlay, we configure a point-to-point tunnel between routers A and B, and advertise the endpoints and the aggregate AS prefix (2.0.0.0/16) via BGP. If AS 2 splits, routers A and B continue to advertise the aggregate, thus no update churn occurs. However, disappearance of the intradomain path between A and B causes the tunnel to reroute over the available BGP path through AS 3 and AS 4; running RIP through the tunnel makes available any destinations in both partitions to each other. Any traffic that AS 3 wishes to send AS 2 will traverse its original path to AS 2; if the destination is actually in partition B, it will be routed there over the tunnel—even though the physical path traversed by such traffic crosses the AS 3–router A link twice, tunnel encapsulation prevents any forwarding loops.

Providing transit service when an AS is split may or may not be beneficial, based on the economic implications of using interdomain paths for the tunneled transit traffic. It is easy to shut down transit traffic during a partition, however, by keeping the iBGP session between partitions from being restored through the tunnel, *e.g.*, by filtering out the corresponding iBGP-peering interfaces in the tunnel (just as we do with the tunnel source and destination prefixes).

Point-to-point tunnels can be used in networks with a small number of border routers; in larger networks, multi-point tunnels can provide scalability. Multi-point tunnels require only one tunnel interface per node in the overlay (*i.e.*, per border router) and thus have significantly less configuration overhead then developing an equivalent overlay with point-to-point tunnels. Next-hop resolution protocol (NHRP) [12] is run with multi-point tunnels to allow NHRP clients to dynamically discover the tunnel destination endpoints.

This scheme is incrementally deployable using today's protocols; the only changes necessary are the creation of the tunnel overlay and changes to BGP loop-detection internally by an AS. With the tunnel approach, few changes are observed by other ASes except for the additional announcement of tunnel-endpoint subnets into the routing table.

V. Discussion and Analysis

A. Forwarding Loops

Given that we modify loop detection, we first argue that our configuration paradigms are free of forwarding loops.

Proposition 1: Partition-level loop detection (as described in Sec. III) does not introduce loops.

Proof: In the case of equal local-preference settings, BGP's decision process prefers routes with shorter AS paths; thus, any partition-level loop will be less preferred than a loop-free path. Because routes transiting one's own AS are either filtered when the AS is not split or are de-preferenced based on the number of tags, AS-loop paths will only be used if non-loop paths are not available. ■

Proposition 2: The modified loop-detection rules in the tunnel-overlay paradigm (Sec. IV) do not introduce loops.

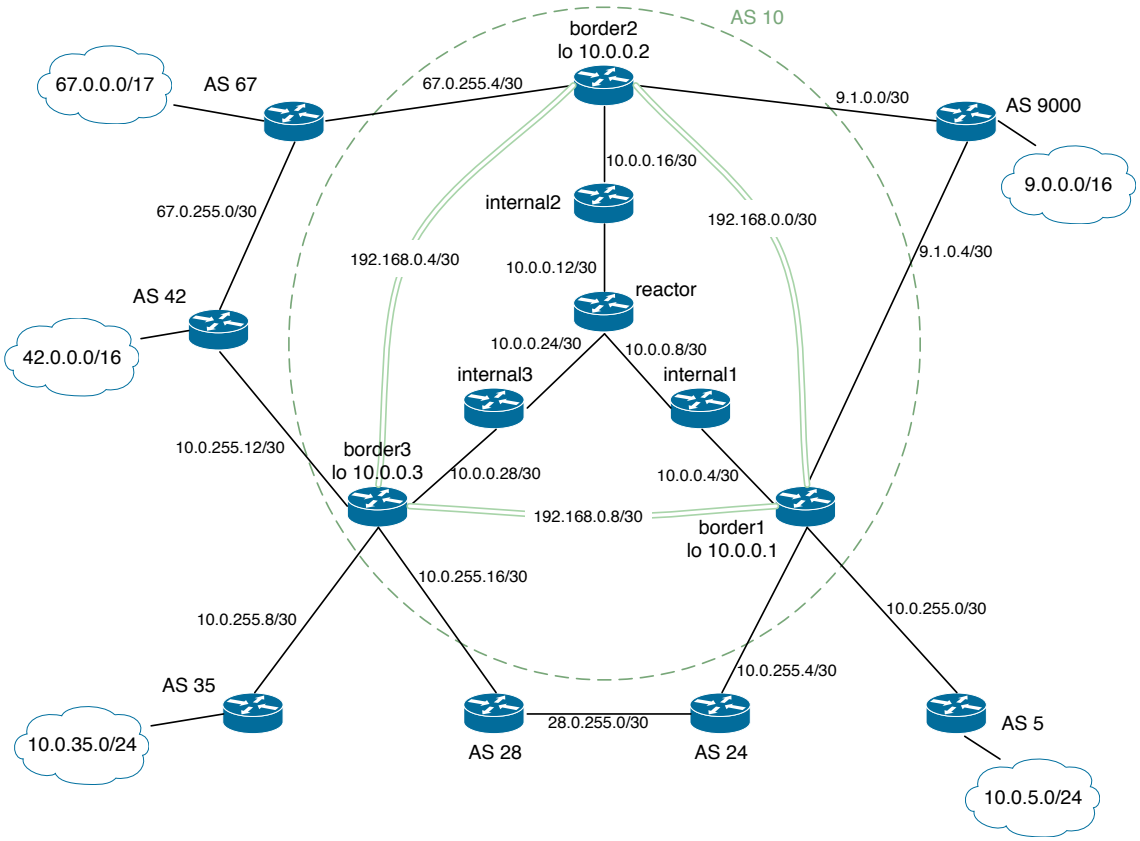

Fig. 2. Emulation topology.

Proof: The only additional BGP routes introduced are those to tunnel loopback addresses; these are never re-advertised. Each border router has a unique address, which is not accepted as a destination at that router. Thus, no BGP-level forwarding loops can exist. The tunnels appear as links in the IGP, but our precautions against recursive routing and (unmodified and existing) loop-free guarantees of the IGP will ensure no IGP-level forwarding loops. ■

B. Emulation Methodology

We used GNS3 [13] to emulate the hardware for Cisco 7200 series routers. An emulator aims to reproduce the underlying hardware of a system, where as a simulator reproduces the observed functionality. Thus, by using an emulator we can observe unusual behavior which might not be implemented in a simulator, but occurs in the physical hardware.

The general topology was designed to be easily partitioned for ease of testing purposes; see Fig. 2 for a diagram. OSPF is being run as the IGP, with all the routers for AS 10 in area 0. The border routers advertise a default route to the internal routers in the network (using the command `default-information originate always`).

Point-to-point tunnels are configured between the three border routers for split healing and regular transit. Through these tunnels is a second OSPF process, which contains a copy of the original IGP, minus the tunnel sources. This is achieved via redistribution with filtering:

```
router ospf 2
...
redistribute ospf 1 subnets route-map OSPF_RED

route-map OSPF_RED permit 5
match ip address prefix-list NO_LOOPBACK

ip prefix-list NO_LOOPBACK seq 5 deny 10.0.0.41/32
ip prefix-list NO_LOOPBACK seq 6 deny 10.0.0.45/32
ip prefix-list NO_LOOPBACK seq 7 deny 10.0.0.49/32
ip prefix-list NO_LOOPBACK seq 8 permit 0.0.0.0/0 le 32
```

As can be seen above, ospf process 1 is redistributed into ospf process 2 according to the route-map `OSPF_RED`. The route-map matches prefixes based on the `NO_LOOPBACK` prefix-list, which denies the /32s used as tunnel source IP addresses but permits all other addresses. This prevents the recursive routing problem from occurring, when the tunnel endpoint is reached through the tunnel itself.

The difference between the tagging and AS-path configurations lies in BGP. In both cases, `allow-as in` is configured for each eBGP neighbor, and an aggregate address is used with an empty suppress-map (to prevent the default elimination of less specific routes). Backdoors are set up for the three tunnel loopbacks, overriding Cisco's default route-selection policy; this is one of the options discussed earlier for ensuring that internal routes are preferred whenever present.

In the modified AS path filtering scheme, two route-maps are utilized for every eBGP neighbor: `ASLOOP` and `ASLOOPout`. Here are the relevant configurations:

```
ip as-path access-list 1 deny _10_.*_10_
ip as-path access-list 1 permit _10$
ip as-path access-list 1 deny _10_
ip as-path access-list 1 permit .*

route-map ASLOOP permit 10
 match ip address prefix-list prune
 match as-path 1
```

```
ip as-path access-list 2 deny _10$
ip as-path access-list 2 permit .*

route-map ASLOOPout permit 10
 match as-path 2

ip prefix-list prune seq 1 deny 10.0.0.0/24
ip prefix-list prune seq 2 deny 10.0.0.41/32
ip prefix-list prune seq 3 permit 0.0.0.0/0 le 32
```

The route-map `ASLOOP` matches according to the prefix-list `prune`, which has been configured for border1 in this particular case. It rejects the prefixes it has already been configured to announce, mainly the /24 aggregate and the /32 tunnel loopback. Everything else proceeds to the next round of condition checking, as all prefixes match seq 3. AS-path matching is then utilized. First, routes that have one's own ASN, in this case 10, more than once are denied. Then prefixes whose AS path end in 10 are accepted, while those who have 10 elsewhere are rejected. Finally, all other AS-paths are accepted. `ASLOOP` is used to filter out incoming routes. `ASLOOPout` is simpler: It directly checks the AS-path before advertising and, if 10 appears as the origin, suppresses the route.

Tag filtering is slightly different. Two route-maps are used once again, `TAG_CHECK` and `TAGME`. Here are the relevant configurations:

```
access-list 2 permit 0.0.0.0 255.255.255.255

route-map TAGME
match ip address 2
set community 1 additive

route-map TAG_CHECK deny 10
match community 1

route-map TAG_CHECK permit 20
match community 2

ip community-list 1 permit 1
ip community-list 2 permit internet
```

`TAGME` is used on outgoing routes. To simulate the function of an additional BGP attribute, we used community attributes. In order for the configuration to work, we needed every BGP speaking router to use the

`send-community` command on their neighbors so the tags would propagate. If the tagging system were ever deployed, such configuration would not be necessary as the attribute would be transitive. The access-list is configured to accept everything, so all outgoing routes are tagged.

`TAG_CHECK` is used on incoming routes. This particular configuration is for border 1. The route-map's first command denies routes, meaning that matched prefixes will be excluded from the RIB. Thus, `TAG_CHECK` first denies routes tagged with 1, the tag assigned to border router 1. The second route-map matching command is a permit, and it is used to accept all other routes.

C. Comparison Between Partition-Healing Schemes

Both partition-healing schemes were able to establish alternate interdomain routes after AS splits 100% of the time in testing, without forwarding loops.

The main tradeoffs we observed were, as expected, among path efficiency, routing-table size, and control-plane overhead (described in Sec. II). Partition-level loop detection has different impacts on table size based on the granularity of partition healing as described in Sec. III, but its major benefit is path efficiency: Exposing the partition-level path allows BGP to compute best paths to each partition individually rather than for the AS as a whole. The impact of the additional attribute on BGP route announcements is minimal; they contain at most four tags per contiguous AS partition traversed by that route. Tags are introduced by BGP speakers only; if there is more than one tag for a partition, the two border routers in that partition must communicate via iBGP. In the case of a full-mesh configuration, the partition will generate at most two tags (for the ingress and egress router); in the case of route reflection, at most four (up to two route reflectors in addition).

Failure recovery was slower for partition-level loop detection than for the tunnel-overlay scheme, because BGP updates were required to propagate between partitions. Tunnels quicken recovery: Once the tunnel destination in another partition disappears from the border router's IGP, external paths are used and connectivity is restored. The impact on table size is minimal: linear in the number of partitions—the smallest impact possible when healing splits with BGP, as each partition needs a unique subnet associated with it. Also, the use of tunnels hides the existence of partitions from other ASes and shares only the existence of tunnel endpoints.

There are, however, downsides to using tunnels. External-to-internal and transit traffic may suffer from path inefficiency. As in the above example, packets being sent to a partitioned AS will use the best BGP-computed path to an advertised prefix, which is perhaps an aggregate that does not help distinguish which partition contains the destination. Once at a border router of the partitioned AS, the packet is sent into a tunnel and redirected to the correct partition. Inefficiency occurs when the tunnel traverses the same path that packets may have used to get to the initial border router. Furthermore, there is bandwidth and CPU overhead associated with transmitting packets and running an IGP through tunnels.

VI. Related Work

Stewart [1] describes that BGP's loop detection can result in partitions but discourages disabling it, as that may introduce forwarding loops or give rise to the count-to-infinity problem seen in distance-vector protocols like

RIP [11]. A Virtual Private Network (VPN) is the standard for connecting components of a split AS, but the goal is quite different than the one we study here. VPN configurations do partly suppress BGP's loop detection, but often couple it with other configuration changes (such as modifying AS numbers) that do not apply to dynamic partition healing [14]. IP tunnels [15] or MPLS [5] can provide virtual intra-domain connectivity but are used when the split topology is known in advance.

The recommended solution for maintaining connectivity among OSPF areas is to use virtual links [3], [6], [9] which consist of a designated intra-area path connecting two ABRs in separate partitions of the backbone. While this creates the appearance of a contiguous backbone, there are several downsides to this approach. First, the virtual link must be manually configured; if the intent is to supplement backbone connectivity with virtual links, the operator must determine the set of virtual link endpoints in advance. Second, because virtual links can connect ABRs through only one additional area, the set of topologies for which virtual links could help heal backbone partitions is limited. Third, OSPF virtual links cannot be rerouted dynamically though different areas based on need, requiring advance setup of all possible partition-healing paths. Zhang [16] suggests that ABRs should calculate a spanning tree of virtual links, on-demand, when a disconnection from the backbone area occurs. Although such a technique could help heal OSPF partitions, its scope is limited: a backbone split with interdomain connectivity but no OSPF paths could not be healed with dynamic virtual links.

Work by Le *et al.* [4], [17] documents how route redistribution among multiple routing processes can be used for partition healing (also called "domain backup" in their work). Their work is based on uses of the technique by operators and gives analysis of what constitutes safe implementation of route redistribution. However, route redistribution alone does not solve the network-partitioning problem. First, their analysis shows that route redistribution can easily be implemented in an unsafe manner, resulting in routing-system instability. Second, redistribution among intradomain protocols alone does not make available interdomain partition-healing paths, which may be necessary depending on the topology of the split. Third, while redistribution in some ways is simpler than configuration of BGP, many in the operator community feel the opposite: route redistribution exposes more routes to more routers in the network than, *e.g.*, using tunnels or iBGP and default routes, resulting in increased routing-table size, increased exposure to control-plane churn, and increased visibility of a network's internal structure to the rest of the Internet.

We used IP tunnels [15] as part of one dynamic partition-healing solution. The use of tunnels in complex network configuration is prevalent; however, rather than partition healing, use cases include supporting deployment of VPNs [5], fast-reroute mechanisms [18]–[20], and non-routing problems such as transitioning to IPv6 and securing virtual links. However, to our knowledge, tunnels have not been used in a dynamic fashion over interdomain routes for partition healing.

Alternate routing architectures can often provide clean solutions to network partitioning. For example, Pathlet routing [21] can take advantage of its flexible topology representation using virtual nodes to heal partitions [22]. While these clean-slate designs provide nice properties without the baggage of existing protocols, our paper shows that configuration support for partition healing is possible even with current protocols.

VII. Conclusion

This paper has investigated how to configure networks to support dynamic partition healing. We proposed two different paradigms for doing so and evaluated their effects on the routing system, including impacts on table size, path efficiency, and control-plane overhead. Our methods are independently and incrementally deployable, not requiring extensive coordination with provider networks, unlike existing methods for connecting partitioned subnetworks. Our work demonstrates that simple, efficient changes to route configuration can overcome the lack of resilience against partitioning events when today's protocols are applied to next-generation networks.

References

[1] J. W. Stewart III, *BGP4: Inter-Domain Routing in the Internet*. Addison-Wesley Professional, Dec. 1998.

[2] C. E. Fossa and T. G. Macdonald, "Internetworking tactical MANETs," in *Proc. Mil. Comm. Conf. (MILCOM) 2010*, Nov. 2010, pp. 611–616.

[3] T. M. Thomas II, "The fundamentals of OSPF routing and design," in *OSPF Network Design Solutions*. Cisco Press, 1998, ch. 5, pp. 203–266.

[4] F. Le, G. G. Xie, D. Pei, J. Wang, and H. Zhang, "Shedding light on the glue logic of the internet routing architecture," in *Proc. ACM SIGCOMM'08*, Aug. 2008, pp. 39–50.

[5] E. Rosen and Y. Rekhter, "BGP/MPLS IP virtual private networks (VPNs)," RFC 4364, IETF, Feb. 2006.

[6] A. Zinn, "Link-state routing protocols," in *Cisco IP Routing: Packet Forwarding and Intra-domain Routing Protocols*. Addison-Wesley, 2002, ch. 9, pp. 405–550.

[7] Y. Rekhter, T. Li, and S. Hares, "A border gateway protocol 4 (BGP-4)," RFC 4271, IETF, Jan. 2006.

[8] J. Moy, "OSPF version 2," RFC 2328, IETF, Apr. 1998.

[9] J. T. Moy, *OSPF: Anatomy of an Internet Routing Protocol*. Addison-Wesley Professional, Feb. 1998.

[10] A. Zinin, A. Lindem, and D. Yeung, "Alternative implementations of OSPF area border routers," RFC 3509, IETF, Apr.l 2003.

[11] G. Malkin, "RIP version 2," RFC 2453, STD 56, IETF, Nov. 1998.

[12] J. Luciani, D. Katz, D. Piscitello, B. Cole, and N. Doraswamy, "NBMA next hop resolution protocol (NHRP)," RFC 2332, IETF, Apr. 1998.

[13] "GNS3: Graphical network simulator," software version 0.7.4, Apr. 2011. [Online]. Available: www.gns3.net

[14] Cisco Systems, Inc., "Allowas-in feature in BGP configuration example," Document ID: 112236, Nov. 2010. [Online]. Available: http://www.cisco.com/image/gif/paws/112236/allowas-in-bgp-config-example.pdf

[15] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, "Generic routing encapsulation (GRE)," RFC 2784, IETF, Mar. 2000.

[16] Z. Zhang, "Fixing backbone partition with dynamic virtual links," Internet Draft (work in progress), IETF, Nov. 1997, draft-zhang-ospf-dvl-01.

[17] F. Le and G. G. Xie, "On guidelines for safe route redistributions," in *Proc. ACM SIGCOMM Internet Management Workshop (INM'07)*, Aug. 2007, pp. 274–279.

[18] S. Bryant, C. Filsfils, S. Previdi, and M. Shand, "IP fast reroute using tunnels," Internet Draft (work in progress), IETF, Nov. 2007, draft-bryant-ipfrr-tunnels-03.

[19] Y. Yang, M. Xu, and Q. Li, "A light-weight IP fast reroute algorithm with tunneling," in *Proc. IEEE Int'l Conf. Commun. (ICC'10)*, May 2010, pp. 1–5.

[20] Q. Li, M. Xu, J. Wu, X. Shi, D. M. Chiu, and P. P. C. Lee, "Achieving unified protection for IP routing," in *Proc. IEEE Int'l Conf. Comp. Comm. Net. (ICCCN'10)*, Aug. 2010, pp. 1–6.

[21] P. B. Godfrey, I. Ganichev, S. Shenker, and I. Stoica, "Pathlet routing," in *Proc. ACM SIGCOMM'09*, Aug. 2009, pp. 111–122.

[22] P. B. Godfrey, "The AS partitioning problem in pathlet routing," Informational slides (unpublished), Jul. 2009. [Online]. Available: http://www.cs.uiuc.edu/homes/pbg/pathlets/partitioning.pdf