Contents lists available at ScienceDirect

Computer Networks

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

Policy-rich interdomain routing with local coordination☆

Xiaozhe Shao, Lixin Gao *

Department of Electrical and Computer Engineering, University of Massachusetts at Amherst, Amherst, MA 01003, United States

A R T I C L E I N F O

Keywords:
Interdomain routing
Routing policy
Routing convergence
BGP

A B S T R A C T

The Internet has evolved from a hierarchical and multi-tiered interconnection network to a meshed network, where autonomous systems (ASes) are interconnected with a dense topology and more and more potential paths can be used to reach a destination. However, routing policies are the key to enable these potential paths and to allow the selection of these paths. While the Gao–Rexford guidelines provide routing policies that derive a safe routing system, those potential paths are not enabled by the Gao–Rexford guidelines. A survey on deployed routing policies has shown that a significant portion of networks or ASes do not follow the Gao–Rexford guidelines completely. In this paper, we propose to broaden routing policies that enable the availability of diverse paths and provide flexibility in selecting these paths. We systematically bring out more and more flexible routing policies to implement various routing requirements. More specifically, we propose FlexibleRC, PeerBoost and hierarchical sibling routing policies and derive sufficient conditions for guaranteeing the routing system derived by these routing policies to be safe. The sufficient conditions can be verified through coordination among neighboring ASes. Thus, local coordination guarantees the routing system to be safe.

1. Introduction

The Internet consists of tens of thousands of autonomous systems (ASes), each of which belongs to an organization such as an Internet Service Provider (ISP), a university, a company, or an Internet Exchange Point (IXP). Traditionally, the interconnections in the Internet have a hierarchical structure, where tier-1 ISPs provide settlement-free services among each other, and provide transit services to regional ISPs or stub networks, and regional ISPs provide transit services to even smaller ISPs or stub networks. As the Internet has evolved, it became a meshed network [2–6], where large content providers and content distribution networks (*e.g.* Google, Facebook and Akamai) interconnect with hundreds or thousands of networks around the world [7–9], and regional ISPs or stub networks interconnect with each other through dedicated links or IXP [6,10,11].

These increasingly interconnected Internet topologies can lead to potential richer routing policies. To illustrate potential routing policies, we use the AS-level topology around University of Vermont (UVM) network and Middlebury College (MC) network as an example.[1] UVM and MC achieve global reachability through commercial ISPs, UVM through Cogent and MC through LEVEL3 respectively. As a member of Internet2, UVM also connects to Internet2 which is well-provisioned and provides high performance connectivity for the participant networks. In addition to provider–customer relationships, regional ISPs and/or stub networks peer with each other to exchange the traffic directly. For example, MC and Education Networks of America (ENA) peer with UVM respectively. Now, we illustrate a potential scenario where the export policy of the peer to peer relationship can be extended to enable more routes. MC can reach the rest of the Internet through its commercial ISP, LEVEL3. Alternatively, MC might want to reach Internet2 through its peer, UVM. If UVM wants to satisfy the routing requirements of MC, UVM should set up its routing policy to announce routes from Internet2 to MC. That is, UVM exports its provider route to its peer. As a result, UVM and MC do not have a conventional peer to peer relationship. Further, MC is not a customer of UVM, since UVM only announces to MC the routes from a specific provider, Internet2, instead of all routes. Not only stub networks but also commercial ISPs and content providers, especially medium-size networks [12], might want to export a provider route to a selected peer.

In addition to announcing more paths to neighbors and allowing more potential paths, routing policies play a critical role on enabling the flexibility of selecting the best path. We use Fig. 2 to illustrate how a network can take advantage of the flexibility of ranking routes. Fig. 2 illustrates the interconnections among the University of Massachusetts

☆ This document is the results of the research project funded by the National Science Foundation Grants CNS-1900866 and CCF-1918187.
* Corresponding author.
E-mail addresses: xiaozheshao@umass.edu (X. Shao), lgao@engin.umass.edu (L. Gao).
[1] For simplicity, Fig. 1 shows only a few neighboring networks of UVM and MC. The interconnections in Fig. 1 are derived from CAIDA AS relationships [1]. Similarly, Figs. 2 and 3 show the interconnections derived from CAIDA AS relationships.

https://doi.org/10.1016/j.comnet.2021.108292
Received 6 January 2021; Received in revised form 3 July 2021; Accepted 5 July 2021

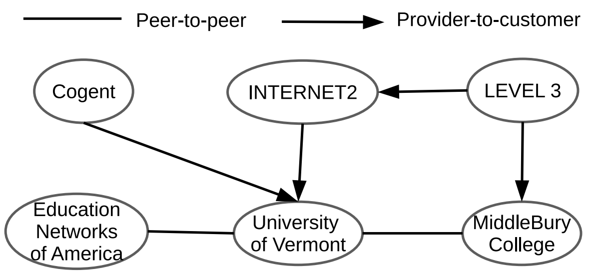

Fig. 1. AS-level topology around the networks of UVM and MC.

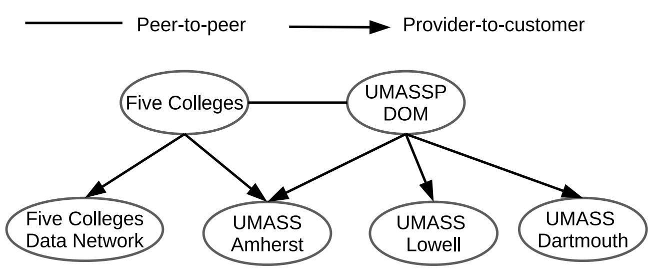

Fig. 2. AS-level topology around the networks of UMass and five colleges.

(UMASS) network and Five Colleges (FC) network. UMASSP DOM provides Internet connections for several UMASS campuses, such as Amherst, Lowell and Dartmouth while FC connects five colleges located in the Connecticut River Pioneer Valley which includes one of UMASS campuses at Amherst. To reach UMASS Amherst, UMASSP DOM can use the directly connected link or the path through FC. UMASSP DOM usually selects the customer link to reach UMASS Amherst. Alternatively, UMASSP DOM might prefer the path through FC, in the case that the link between UMASSP DOM and UMASS Amherst is congested. In order to do so, UMASSP DOM needs to prefer the peer link over the customer link to reach the destinations in UMASS Amherst. It is not uncommon that commercial ISPs need the flexibility of selecting paths as well [4,13].

Although network operators can set up any routing policy for peers to implement their routing requirements, without coordination, the resulting routing policy might lead to routing oscillation. As we illustrate in the AS-level topology of Fig. 2, Five Colleges might prefer peer routes over customer routes. However, if UMASSP DOM also prefers peer routes over customer routes, then the routing system is not safe.

Note that none of the above routing policy examples follows the Gao–Rexford guidelines [14], which provide rules for routing policies to derive a safe routing system. Then, it is not clear what kind of routing policies are guaranteed to be safe. Researchers propose a number of extensions to Gao–Rexford guidelines, in terms of broadening the export routing policies [14,15] or accommodating sibling relationships [16,17]. These rules for routing policies do not enable sufficient flexibility at import routing policies. As for the routing scenario illustrated in Fig. 1, UVM can announce routes from Internet2 to MC by following routing policy rules proposed in [14,15]. However, these routes have to be backup routes. Only if there are no primary routes, these backup routes can be selected as the best routes [15].

In this paper, without compromising the routing stability, we propose to broaden routing policies that enable diverse paths and provide flexibility in selecting these paths. We systematically bring out more and more flexible routing policies and derive the sufficient conditions for guaranteeing the routing systems to be safe. Further, we show that these conditions can be verified through local coordination among neighboring ASes. Specifically, we propose the following three routing policies.

- We propose *FlexibleRC* routing policy that allows ASes to selectively prefer a peer route over one or more customer routes (Section 4).
- We propose *PeerBoost* routing policy to expand the paths beyond what are allowed by the Gao–Rexford guidelines. In particular, an AS can selectively announce its peer routes to another peer and/or provider and selectively announce its provider routes to its peer. In addition, the ranking among these paths is flexible in the sense that the newly allowed paths can be ranked higher than those allowed by the Gao–Rexford guidelines (Section 5).
- We further propose *hierarchical sibling* routing policies to accommodate the sibling contractual agreement (Section 6).

We evaluate the flexibility of the proposed routing policies in Section 7. We discuss related work in Section 8 and conclude in Section 9.

2. Motivation

In this section, we first review the Gao–Rexford guidelines. We then show the motivation of allowing more potential paths through export policies and more flexible ranking among paths through import policies.

2.1. Routing policies under the Gao–Rexford guidelines

Provider–customer agreement and peer to peer agreement are two common agreements in the Internet [18]. Within a provider–customer agreement, an AS as the customer pays its provider to access the rest of the Internet. Within a peer to peer agreement, two connected ASes exchange traffic from their own customers.

The Gao–Rexford guidelines have the following export policy:

- An AS can export its peer routes and provider routes to its customers.
- An AS can export its customer routes to its customers, peers and providers.

A *customer route* is a route in which the first link is a provider–customer link, a *provider route* is a route in which the first link is a customer–provider link, and a *peer route* is a route in which the first link is a peer link.

The Gao–Rexford guidelines have the following import policy:

- ASes prefer customer routes over peer routes and provider routes.

Routing policies that follow the Gao–Rexford guidelines are *safe*, when there are no provider–customer cycles. A routing policy is safe if any routing system with the routing policy always converges to a stable state under any link or node failure [14].

2.2. Motivation to expand routing policies

A number of ASes do not follow the Gao–Rexford guideline to set up their routing policies, according to a survey among network operators [19] and network measurement studies on the inter-domain routing [4,12,13].

A survey on routing policies deployed by network operators shows that 32% networks including transit networks, content providers and stub networks do not follow the Gao–Rexford guidelines completely [19]. More specifically, in 16 out of 97 networks, routes from a peer or transit provider are preferred over customer routes while within 21 out of 97 networks including small/medium/large transit providers routes from peers and providers are announced to peers and providers.

The measurement for the inter-domain routing also implies that the Gao–Rexford guidelines are violated by networks on the Internet [4,12,13]. For example, 34.3% of routing decisions in the Internet routing system cannot be explained by Gao–Rexford model [4] and Qiu et al. find that 11% provider ASes propagate valley announcements by analyzing real-world BGP updates [12].

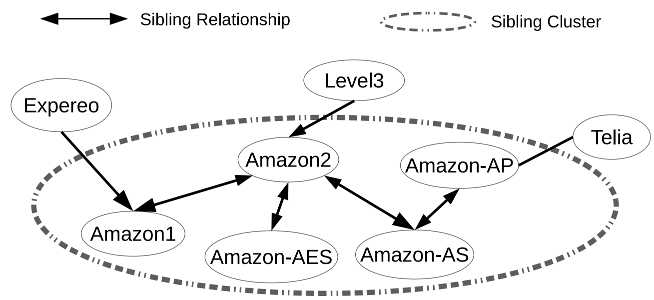

Fig. 3. The topology around ASes of Amazon.

2.3. Export policies that enable potential paths

In the following, we will discuss how to broaden the export policies for peer and accommodate sibling relationships.

2.3.1. Selective export

Peer to peer relationships are established to exchange traffic between customers of two neighboring networks. Following the Gao–Rexford guidelines, an AS only exports customer routes to its peers. In reality, an AS has the need to export its provider routes to its peers and export its peer routes to its peers or providers [19]. We propose to extend the export policy for peer to peer relationships as follows.

Export a provider route to a selected peer. For the example in Fig. 1, MC would like to reach Internet2 through UVM. To do so, UVM has to export its provider route from Internet2 to its peer, MC. To allow ASes to reach its peer's provider, we consider the export policy, where an AS can export a provider route to a selected peer.

Export a peer route to a selected provider. To fetch content back from Internet2, UVM allows traffic from Internet2 to reach MC. That is, an AS might export its peer routes to its providers. To accommodate the routing requirement that an AS reaches its customer's peer, we consider the export policy, where an AS can export a peer route to a selected provider.

Export a peer route to a selected peer. Further, an AS might export one of its peer routes to another peer. In the example of Fig. 1, UVM peers with Education Networks of America (ENA) and MC respectively. MC might want to access the customers of ENA through UVM instead of its commercial ISPs. In that case, UVM needs to announce its peer routes from ENA to MC. That is, an AS might export its peer routes to a selected peer.

2.3.2. Sibling

A pair of neighboring ASes might want to export all its routes to each other. That is the pair of ASes have a *sibling relationship*, which are typically established between ASes that are managed by the same organization or two closely related organizations. For example, in Fig. 3, we illustrate the topology around five ASes that all belong to the same organization, Amazon. The five Amazon ASes can establish the sibling relationship with their neighbors within the sibling cluster. Within the sibling cluster, ASes export all routes to their siblings. For example, all five Amazon ASes can access Telia through the link between Amazon-AP and Telia while all five Amazon ASes can access destinations of Expereo through Amazon1. Formally, when a pair of connected ASes, A and B, establish sibling relationship, A will announce all its routes to B and B will announce all its routes to A.

2.4. Import policies that enable flexible ranking

Given much more available routes, it is also essential to broaden import polices, so that ASes can flexibly select the best one.

Table 1
Peer link statistics of the Internet.

Years	ASes	Links	Peer links	% of Peer links
1998	3638	6728	933	13.87%
2003	15,320	34,720	6635	19.11%
2008	28,411	78,997	24,519	31.04%
2013	44,326	149,476	63,344	42.38%
2018	60,874	300,634	178,608	59.41%

Selectively prefer peer routes over customer routes. An AS might want to balance its traffic between its customer and peer, and therefore prefers a peer over a customer for a destination that be reached through both.

Prefer additional routes from customers over provider routes. When the potential paths are enabled as Section 2.3 shows, these additional routes from customers should be preferred over provider routes. For example, in Fig. 1, UVM can announce the route from MC to Internet2, so that Internet2 can exploit UVM to reach MC. Meanwhile, Internet2 should prefer the additional routes from UVM over its provider routes. Therefore, Internet2 can select the newly-enabled route as the best route.

3. Overview of policy-rich routing

We propose routing policies to enable more potential paths through peer links without compromising the routing convergence. Such policy-rich routing is essential given a meshed Internet, where there are more and more peer links. Table 1 illustrates the number and percentage of the peer links over the last 20 years. We can see that the peer links have increased significantly in both number and percentage. The AS relationships are derived from CAIDA dataset [1]. It has been shown that not all peer links can be identified through measurement [20–22]. We expect that there are even more peer links in reality.

In order to enable more potential routes and allow flexible ranking among routes, we systematically propose a series of routing policies.

As the first step, we relax the constraint that customer routes are more preferred than peer routes. We propose FlexibleRC routing policies that enable flexible ranking between peer and customer routes. Specifically, FlexibleRC allows ASes to prefer peer routes over customer routes.

We then enable more potential routes through selective export. First, we consider *selectively-export-provider-route-to-peer* (P2R) export rule which allows an AS to selectively export provider routes to some of its peers. Second, we consider *selectively-export-peer-route-to-provider* (R2P) export rule which allows an AS to selectively export peer routes to some of its providers. At last, we consider *selectively-export-peer-route-to-peer* (R2R) export rule which allows an AS to selectively export peer routes to some of its peers. We propose PeerBoost routing policies to enable three selective export rules and flexible ranking among peer and customer routes.

Table 2 illustrates the export and import rules of the routing systems enabled by the Gao–Rexford guidelines [14], safe backup routing [15,16] and the proposed routing policies. Safe backup routing enables backup paths to increase the reliability of the networks. FlexibleRC provides more flexible import routing policies than Gao–Rexford guidelines and safe backup routing [15,16] through enabling ASes to prefer peer routes over customer routes. Further, PeerBoost enhances FlexibleRC through enabling P2R, R2P and R2R export rules and preferring these newly-enabled routes over routes enabled by Gao–Rexford guidelines.

Finally, we accommodate sibling relationships in which a pair of ASes export all routes to each other. Table 3 summarizes all proposed routing policies and the respective sections describing these routing policies.

So far, we assume that routing policies are set independent of destination prefixes. Nevertheless, it is possible to address the routing policies in the prefix level. In fact, all of the results in this paper

Table 2
Export and import rules enabled by routing policies.

Export and import rules	Gao–Rexford [14]	Safe backup routing [15,16]	FlexibleRC	PeerBoost
P2R	Not enabled	Enabled	Not enabled	Enabled
R2P	Not enabled	Enabled	Not enabled	Enabled
R2R	Not enabled	Enabled	Not enabled	Enabled
Prefer peers over customers	Not enabled	Not enabled	Enabled	Enabled
Prefer routes enabled by P2R, R2P and R2R	Not enabled	Not enabled	Not enabled	Enabled

Table 3
Summary of proposed routing policies.

Routing policy	Policy description
FlexibleRC (Section 4)	Enable preferring peer over customer routes.
PeerBoost (Section 5)	Enable (1) export rules (P2R, R2P and R2R) (2) preferring routes enabled by the export rules.
Sibling (Section 6)	Accommodate sibling relationship.

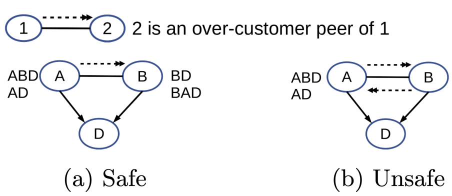

Fig. 4. Routing systems with OC peers.

will apply to routing policies that are set for a specific prefix as well. One issue of per-prefix policies is that the network operators must pay attention to any change of the prefixes with special policies, since the destination network might add or remove the prefix that it owns. For simplicity of exposition, we describe all routing policies independent of destination prefixes throughout this paper.

4. Selectively prefer peers over customers

In this section, we explore the potential flexibility for ranking paths that are permitted by the Gao–Rexford guidelines. More specifically, we propose FlexibleRC routing policies which allow ASes to prefer peer routes over customer routes. We derive a sufficient condition for routing convergence and show that the sufficient condition can be checked through local coordination.

4.1. FlexibleRC routing policy

In FlexibleRC routing policies, the export policy is same with the Gao–Rexford guidelines. To enable flexible ranking among peer routes and customer routes, FlexibleRC has the following import policy:

- An AS prefers customer routes over provider routes.

The import policy of FlexibleRC does not regulate the ranking of peer routes. For example, an AS might prefer a peer route over all customer routes. An AS could also prefer a peer route over some customer routes. Or, an AS might prefer all customer routes over a peer route. In FlexbileRC policies, customers of an AS always have a higher ranking than providers of the AS. If an AS, A, has a peer, B, and A prefers a peer route from B over a customer route, B is an *over-customer peer (OC peer)* of A.

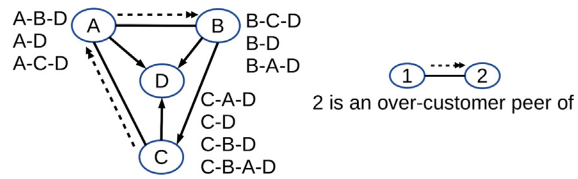

Fig. 5. An unsafe routing system (BAD GADGET) with OC peers.

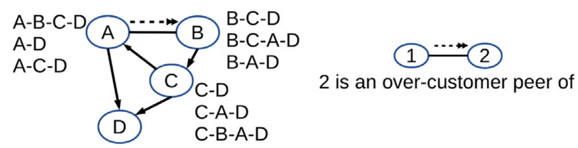

Fig. 6. An example routing system which is safe but contains OC cycles.

4.2. Convergence analysis

FlexibleRC routing policies are not guaranteed to be safe due to the flexible ranking of peer routes. In this section, we explore a sufficient condition for routing convergence.

To show how the OC peers impact on the convergence of a routing system, we show two routing systems with OC peers in Fig. 4. In these figures, we illustrate the topology, all ASes and all potential routes to a destination AS. In the routing system of Fig. 4(a), to reach AS D, AS A prefers the route through B over its customer route and AS B prefers its own customer route over the peer route from AS A. Therefore, AS B is an OC peer of AS A. Clearly, the routing system is safe even though there is an OC peer. However, you might not be able to set up OC peers everywhere in the routing system. For example, in the routing system of Fig. 4(b), to reach AS D, both AS A and AS B prefer the route through each other over their customer routes respectively. Therefore, AS A and AS B treat each other as an OC peer and the OC peer relationship forms a cycle. The routing system is clearly not safe since it is the system DISAGREE in [23]. Fig. 5 illustrates another example routing system which is BAD GADGET [23] and is not safe. In this example, the cycle, A-B-C-A, comprises a provider–customer link and two OC peer links. More generally, when provider–customer links and OC peer links form a cycle, the routing system might not be safe.

We define an *over-customer (OC)* cycle as an AS cycle, $v_n ... v_0$, where $v_0 = v_n$ and $\forall i \in [0, n-1]$, v_i is a customer or an OC peer of v_{i+1}.

Theorem 4.1. *A FlexibleRC routing policy is guaranteed to be safe, if there are no OC cycles.*

We provide the proof in Appendix A.

The absence of OC cycles is a sufficient but not necessary condition for routing convergence. Even if a routing system is safe, there might be an OC cycle. For example, in Fig. 6, we illustrate the whole AS-level topology and all potential routes of a routing system which contains an OC cycle, A-B-C-A. However, the routing system is guaranteed to be safe.

Routing guidelines, such as the Gao–Rexford guidelines [14] and backup routing [15], guarantee the convergence of the routing system without requiring any coordination among neighboring ASes. When the safe condition in Theorem 4.1 is satisfied, FlexibleRC policy enables more flexible routing policies. That is, to establish OC peers without jeopardizing the convergence property of the routing system, we need to ensure that there is no OC cycle.

To guarantee the routing system safe, there are several existing conditions for convergence, such as no Dispute Wheels [24] and the freeness property of the routing system [25,26]. However, to check those conditions, the exact routing polices for all ASes are essential. This is challenging in practice, since ASes are not willing to share their routing policies. FlexibleRC policies enable ASes to prefer peer routes over customer routes while the safe condition needs to be verified without sharing routing policies.

When the network operators of an AS want to change the routing policy, to guarantee the FlexibleRC policy to be safe, the network operators of the AS need to check whether the policy change leads to an OC cycle. OC cycle includes provider–customer links and OC peer links only. Therefore, only when an AS wants to set up a customer or an OC peer, the network operators of the AS need to initiate a coordination process to check whether there will be an OC cycle. During the coordination process, the network operators of an AS ask the network operators of its customers and OC peers whether, through provider–customer links and OC peer links, their network can reach the AS that initiates the coordination process. To answer the question, the network operators of the AS being asked needs to ask the network operators of its customers and OC peers the same question. The process finishes when an AS has no customers nor OC peers or the network operators of an AS finds that the initial AS is a customer or an OC peer. If the answer to the question is yes, then the initial AS should not change routing policy due to the potential routing oscillation.

4.3. Establishing OC peer with local coordination

In practice, special peer routing policies are usually deployed among ASes close to the edge of the Internet. Large ISPs of the high tiers, such as Tier 1 and Tier 2, might not have incentives and routing requirements to set up OC peers. For these ASes close to the edge, the coordination process for establishing OC peers does not have to involve customer ASes. As for the example in Fig. 2, before Five College setting up UMASSP DOM as an OC peer, the network operators of Five College need to coordinate with the network operators of UMASSP DOM. Meanwhile, the network operators of UMASSP DOM know that their customer ASes are all stub networks and cannot discover an OC cycle through coordinating with the network operators of these customer ASes. Thus, the network operators of UMASSP DOM needs to coordinate with the network operators of its OC peers only. If UMASSP DOM does not set up Five College as an OC peer, then there will not be an OC cycle. Otherwise, the network operators of Five College know that UMASSP DOM cannot be set up as an OC peer. The cycle checking can be performed among the network operators of peering ASes within a local scope.

More general, let us consider in what scenario, an AS can set up a peer link as an OC peer link without involving customers in the coordination process. We define a *directed* cycle as an AS cycle, $v_n ... v_0$, where $v_0 = v_n$, that has the following two properties: (a) $\forall i \in [0, n-1]$, v_{i+1} is a provider or peer of v_i; (b) $\exists i \in [0, n-1]$ such that v_{i+1} is a provider of v_i. Namely, the direction of provider–customer links in a directed cycle is always same. A peer link is an *in-loop* peer link, if the peer link is included in a directed cycle.

Corollary 4.1. *If a peer link is not an in-loop peer link, then when an AS sets up the peer link as an OC peer link, coordinating with OC peers only is sufficient to guarantee that the routing policy is safe.*

Proof. If setting the peer link as an OC peer link leads to an OC cycle, then the OC cycle contains peer links only. Otherwise, it will contradict with the condition that the peer link is not an in-loop peer link. Then, to check the potential OC cycle, it is enough to coordinate with the network operators of OC peers only. □

Typically, an AS peers with ASes that have similar size to exchange a comparable volume of traffic. Furthermore, an AS usually provides transit service to an AS with smaller size. Then, an AS is unlikely to peer with one of its (direct or indirect) providers [14]. It is not uncommon that peer links are not in-loop peer links. According to Corollary 4.1, when a peer link is not an in-loop peer link, the network operators of an AS coordinate with the network operators of OC peers only for establishing OC peer.

5. Enable selective export

In this section, we systematically explore the possibility to enable diverse paths and the potential flexibility for ranking these newly-enabled paths. We first propose PeerBoost routing policies that selectively export provider routes to peers, peer routes to providers and peer routes to peers respectively. Then, we derive how PeerBoost policies rank these routes. Instead of being ranked as backup routes, these newly-enabled routes of PeerBoost routing polices can be selected as the best routes. We derive a sufficient condition for routing convergence. When the condition is satisfied, the routing system is guaranteed to be safe.

5.1. Export policy

In addition to paths enabled by the Gao–Rexford guidelines, PeerBoost policies also enable P2R, R2P and R2R export rules. With these P2R, R2P and R2R export rules, more peer routes are enabled by PeerBoost export policy. In the following, we illustrate the routes that can be announced to customers, peers and providers respectively.

We first consider the routes announced to customers. Announcing routes to the customer gives rise to potential income for transit AS. In the Gao–Rexford guidelines, ASes announce all routes to their customers. Same with the Gao–Rexford guidelines, PeerBoost policies enable ASes to announce all routes to their customers.

Now, we consider the routes announced to peers. A settlement-free peering link between two ASes exchanges traffic between these two ASes and their customers. Therefore, customer routes can be announced to peers. Beyond that, PeerBoost policies enable ASes to selectively announce provider routes to peers. Those newly-enabled peer routes are *upstream peer routes*. Formally, if the first non-peer link of a peer route from the source AS to the destination AS is a customer-to-provider link, the peer route is an upstream peer route. In Fig. 7, we illustrate three examples of upstream peer routes.[2] In addition to announcing provider routes to peers, PeerBoost policies also enable ASes to selectively announce peer routes to peers.

Finally, we consider the routes announced to providers. Clearly, the customer routes can be announced to providers. ASes do not announce provider routes to another provider, since ASes do not transit the traffic between two of their providers. Now, let us consider announcing peer routes to providers. There are two kinds of peer routes: upstream peer routes and downstream peer routes. The upstream peer routes have been defined in the above paragraph. The rest of peer routes are downstream peer routes. Namely, if the first non-peer link of a peer route from the source AS to the destination AS is a provider-to-customer link, the peer route is a *downstream peer route*. If all links of a peer route are peer links, we still classify the peer route as a downstream peer route. In Fig. 8, we illustrate three examples of downstream peer routes. ASes can selectively announce downstream peer routes to its providers. We do not consider that an AS exports an upstream peer route to its providers, since a pair of peering ASes does not have economic incentives to transit traffic between their providers through their peer links.

In summary, the export policy have the following three selective export rules.

[2] The legends for peer to peer links and provider to customer links are used throughout the rest of the paper.

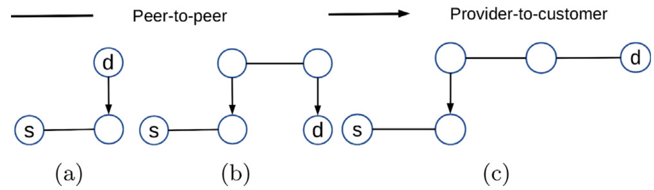

Fig. 7. Upstream peer routes, where s is the source AS and d is the destination AS.

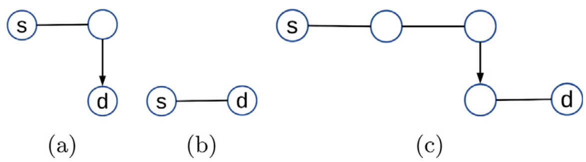

Fig. 8. Downstream peer routes, where s is the source AS and d is the destination AS.

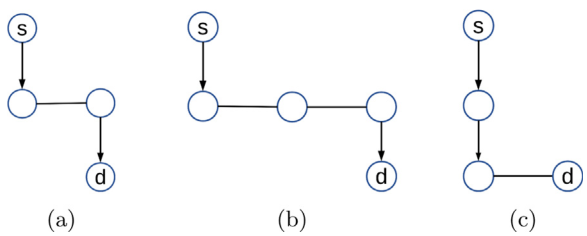

Fig. 9. Customer routes permitted in PeerBoost policies but not permitted under Gao–Rexford guidelines, where s is the source AS and d is the destination AS.

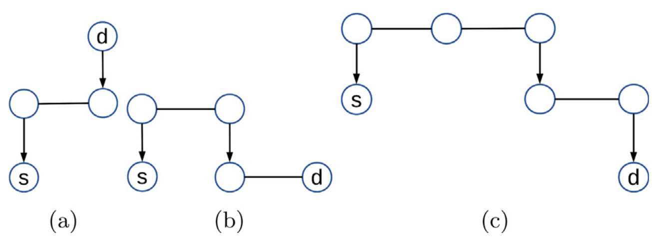

Fig. 10. Three examples of providers routes permitted in PeerBoost policies but not permitted under Gao–Rexford guidelines, where s is the source AS and d is the destination AS.

- An AS can selectively export its peer routes to its peers.
- An AS can selectively export its provider routes to its peers.
- An AS can selectively export its downstream peer routes to its providers.

PeerBoost policies enable almost all possible paths, except the paths indicating that customers transit traffic for their providers. Namely, PeerBoost policies do not enable a route, only if the route is generated through announcing a provider route or an upstream peer route to a provider. PeerBoost policies are still valley-free. In a valid PeerBoost path, all customer–provider links have to come before all provider–customer links, while peer links can be arbitrarily interspersed along the path.

Any route that is permitted under the Gao–Rexford guidelines is permitted in PeerBoost policies. Besides, additional routes are enabled by the P2R, R2R and R2P export rules. Since P2R rules enables an AS to announce a provider route to a peer, upstream peer routes, such as routes in Figs. 7(a) and 7(b) are enabled. R2R and R2P rules enable more downstream peer routes, such as the route in Fig. 8(c). Besides of enabling additional peer routes, more customer and provider routes are permitted in PeerBoost policies. Fig. 9 illustrates three examples of customer routes enabled in PeerBoost policies and Fig. 10 illustrates three examples of provider routes enabled in PeerBoost policies.

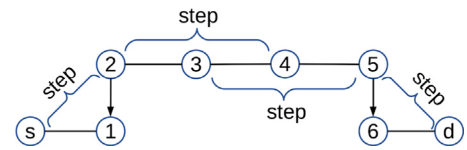

Fig. 11. An example of upstream peer routes with steps permitted in PeerBoost polices but not permitted under the Gao–Rexford guidelines, where s is the source AS and d is the destination AS.

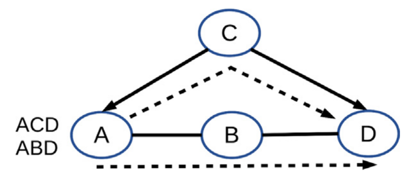

Fig. 12. Prefer peer routes with steps over provider routes.

When an AS exports routes by following the P2R, R2R and R2P export rules, the resulting route starts with a step. A *step* of a route is a two-link segment of the route, if the two links from the source to the destination are one of the following three cases:

- a provider–customer link followed by a peer link.
- a peer link followed by another peer link.
- a peer link followed by a customer–provider link.

We refer to the AS between the two links of a step as the *pivot* of the step. For example, Fig. 11 illustrates an upstream peer route that contains four steps, where AS 1 is the pivot of the step, *s*-1-2 and AS 6 is the pivot of the step, 5-6-d.

5.2. Import policy

In the following, we discuss how to rank the routes enabled in PeerBoost policies.

5.2.1. Ranking routes based on steps

When an AS announces a route by following P2R, R2R and R2P rules, the resulting route starts with a step and the AS is the pivot of the step. The pivot AS might want the resulting route to be used as a preferred route. Alternatively, the pivot AS might want the resulting route to be a backup. To do that, the pivot can determine whether a step is a *preferred step* or a *backup step*. Through determining a step as a backup step of a route, the pivot AS can inform the AS receiving the route announcement that the route should be treated as a backup choice. On the contrary, through determining a step as a preferred step, the pivot AS can inform its neighbor that the route is not treated as a backup choice. For example, in the routing system of Fig. 12, the peer route, A-B-D, with a step can be preferred over the provider route, A-C-D, when the step is a preferred step. Note, the step is also proposed in [15]. However, the routing policy proposed in [15] always prefers a route without a step over routes with steps.

A route might contain multiple backup steps. We refer to the number of backup steps in a route as *backup level* of the route. A route with lower backup level is preferred over the routes with higher backup level.

5.2.2. Ranking routes with same backup level

In the following, we discuss how to rank routes with the same backup level.

Ranking customer and provider routes. Considering only customer routes and provider routes, ASes prefer customer routes over provider routes due to the economic incentives. That is, given two customer routes, the AS can rank those two routes in any order. It is same for two provider routes.

Ranking upstream peer routes. Now, we consider how to rank various upstream peer routes enabled by PeerBoost policies. When an upstream peer route is selected to deliver the traffic, a peer AS might pay its provider for the transit. Thus, customer routes are preferred over upstream peer routes.

Ranking downstream peer routes. When an AS announces a downstream peer route to a peer or a provider without increasing the backup level of the route, we refer to the downstream peer route as *preferred downstream peer route* of the AS. The AS allows its neighbors to prefer the route extended from the preferred downstream peer route. The AS itself should prefer the preferred downstream peer route. Thus, a preferred downstream peer route is preferred over all provider routes and upstream peer routes. The rest of downstream peer routes can be ranked in any order.

5.2.3. PeerBoost import policy

In summary, ASes determine the route ranking by following the import policy rules step by step.

- An AS prefers routes with lower backup level.
- An AS prefers customer routes and preferred downstream peer routes over provider routes and upstream peer routes.

In the following, we describe how to implement the import policy. The key is to let ASes know enough information of each route, so that each AS can rank routes based on the import policy. To do that, we use the community attribute of BGP announcements to convey the necessary information.

To rank routes, an AS needs to know the backup level of each route. However, the AS might not know the relationship between ASes along the route and thus do not know the backup level. We can use the community attribute of BGP announcement to carry the number of backup steps in a route. Namely, when an AS announces a route, the number of backup steps keeps unchanged (for preferred steps) or increases by one (for backup steps).

In addition to the backup level, to rank a peer route, an AS needs to know whether the route is a downstream peer route or an upstream peer route. To do that, we also use the community attribute of BGP announcement to carry the information. More specifically, when an AS announces a provider route or an upstream peer route to a peer, the AS informs the receiving AS that it is an upstream peer route. Alternatively, when an AS announces a customer route or a downstream peer route to a peer, the AS informs the receiving AS that it is a downstream peer route.

The semantics of using BGP communities for PeerBoost policy is only defined and used between neighboring ASes. Thus, when an AS announces a route to its neighbor, the associated BGP communities should be guaranteed to reflect the properties (e.g. the backup level and the type of peer route) of the route. While the BGP communities might influence routing in unintended ways [27], it is widely used to convey various information between neighboring ASes. When ASes assign BGP communities by following the routing policies, the potential disadvantages can be mitigated.

5.3. Convergence analysis

Whether a routing system observing PeerBoost policy is guaranteed to be safe is determined by how the pivot ASes determine the types of these steps. In the following, we will discuss the convergence conditions in terms of the pivot decisions.

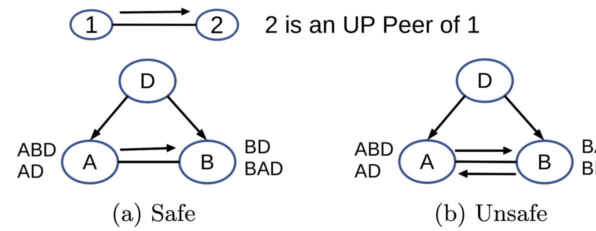

Fig. 13. Routing systems in which provider routes are exported to peers.

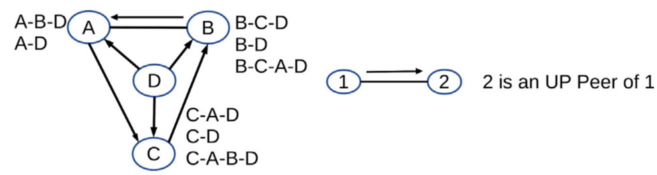

Fig. 14. An unsafe routing system (BAD GADGET) in which provider routes are exported to peers.

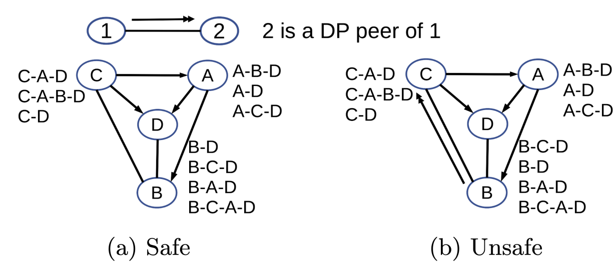

Fig. 15. Routing systems in which downstream peer routes are exported to providers.

5.3.1. Convergence under R2R and P2R rule

Announcing provider routes or upstream peer routes to peers do not imply that the routing system is not safe. For example, in the routing system of Fig. 13(a), AS B announces the provider route B-D to the peer, AS A, without increasing the backup level. However, you might not be able to set up those special arrangement everywhere in the routing system. When those special arrangement forms a cycle, the system might not be safe. When AS A announces a provider or upstream peer route to its peer, AS P, without increasing backup level, AS P is an *uphill preferred peer* (UP Peer) of AS A. In the example of Fig. 13(b), both A and B treat each other as an UP Peer. In addition, the routing system is not safe, since it is a DISAGREE system in [23]. Namely, the UP Peer relationship forms a cycle and it might lead to routing oscillation. More generally, the cycle might contain customer–provider links. In the routing system of Fig. 14, AS B announces the provider route, B-D, to its UP Peer, AS A, and AS A prefers the resulting route, A-B-D, over its provider route, A-D. The routing system is a BAD GADGET and is not safe [23]. We define that a *generic uphill (GU)* cycle is an AS cycle, $v_n ... v_0$, where $v_0 = v_n$ and for each $i \in [0, n-1]$, v_{i+1} is a customer or an UP Peer of v_i .

5.3.2. Convergence under R2R and R2P rules

Announcing downstream peer routes to peer and providers do not imply that the routing system is not safe. For example, in the routing system of Fig. 15(a), although the route B-D is a preferred downstream peer route and is announced to the provider, AS A, the routing system is safe. If AS B changes its route ranking a little bit to prefer B-C-D over

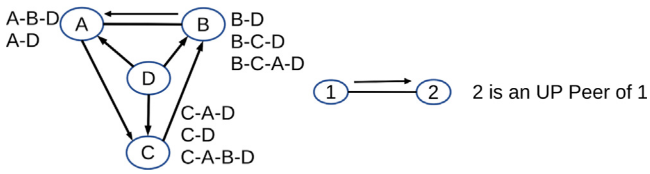

Fig. 16. A safe routing system with GU cycles.

B-D, then the routing system is not safe. That is, the routing system in Fig. 15(b) forms a BAD GADGET system and is not safe. If AS A prefers a downstream peer route from a peer, AS P, over a customer route or a preferred downstream peer route, then AS P is a *downhill preferred peer* (DP Peer) of AS A. For example, AS C is a DP peer of AS B in the routing system of Fig. 15(b). When the DP peer link and the provider–customer link forms a cycle, such as A-B-C-A in Fig. 15(b), the routing system might not be safe. Formally, we define a *generic downhill (GD)* cycle as an AS cycle, $v_n ... v_0$, where $v_0 = v_n$ and for each $i \in [0, n-1]$, v_i is a customer or a DP peer of v_{i+1}.

5.3.3. Convergence condition

Then, we derive the convergence condition for PeerBoost policies.

Theorem 5.1. *A PeerBoost routing policy is safe, if there are neither GD cycles nor GU cycles.*

We provide the proof in Appendix B.

The absence of GD and GU cycles is a sufficient but not necessary condition for routing convergence. A routing system with OC cycles might be guaranteed to be safe. For example, in Fig. 16, the routing system contains a GU cycle, A-B-C-A. However, the routing system is safe. Note, different from the routing system in Fig. 14, AS B prefers the route B-D over all the other routes. Although there is a GU cycle, the routing system always converges to the state, where A chooses A-B-D, B chooses B-D and C chooses C-D. Even if there are link or node failures, the routing system converges. Thus, the routing system is safe.

When the network operators of an AS want to change the routing policies, to guarantee the PeerBoost policy to be safe, the network operators need to check whether the policy change leads to a GD or GU cycle. GU cycle includes provider–customer links and UP peer links only. Therefore, only when an AS wants to set up a customer or an UP peer, the network operators of the AS need to initiate a coordination process to check whether there will be a GU cycle. Similar to the coordination for checking OC cycle, the network operators of the AS ask the network operators of its customers and UP peers whether, through provider–customer links and UP peer links, their networks can reach the AS that initiates the coordination process. GD cycle includes provider–customer links and DP peer links only. Therefore, only when an AS wants to set up a customer or a DP peer, the network operators of the AS need to initiate a coordination process to check whether there will be a GD cycle. Similarly, the network operators of the AS ask the network operators of its customers and DP peers whether, through provider–customer links and DP peer links, their networks can reach the AS that initiates the coordination process.

5.4. Establishing UP peer and DP peer with local coordination

In practice, special peer routing policies are usually deployed among ASes close to the edge of the Internet. In that case, the coordination process for establishing UP peers and DP peers do not have to involve customer ASes. For example, in Fig. 1, if UVM exports the provider routes from Internet2 to MC without increasing backup level, then UVM needs to coordinate with MC only to check whether there is a GU cycle. When UVM wants MC to access Internet2 through itself, UVM might export routes from Internet2 to MC without increasing the backup level. Thus, MC is an UP peer of UVM. To guarantee the routing system to be safe, the network operators of UVM need to coordinate with the network operators of MC and check whether there is a GU cycle. To do that, the network operators of UVM need to coordinate with the network operators of MC only, since MC is a stub network and no more UP peer of MC will be involved in the coordination. The cycle checking can be performed among the network operators of peering ASes within a local scope.

Corollary 5.1. *If a peer link is not an in-loop peer link, then when an AS sets up the peer link as a UP peer link, coordinating with UP peers only can guarantee that the routing policy is safe.*

Corollary 5.2. *If a peer link is not an in-loop peer link, then when an AS sets up the peer link as a DP peer link, coordinating with DP peers only can guarantee that the routing policy is safe.*

According to Corollaries 5.1 and 5.2, when a peer link is not an in-loop peer link, the network operators of an AS coordinate with the network operators of UP peers or DP peers only for setting the peer as an UP peer or a DP peer.

6. Enable sibling

A pair of neighboring ASes might want to export all its routes to each other. In this case, the pair of ASes have a sibling relationship. In this section, we propose routing policies to accommodate sibling relationships.

Clearly, a pair of siblings transit for each other and therefore announce all routes to each other. In the following, we focus on how to rank routes going through sibling links, so that the routing system is safe.

We propose a *hierarchical sibling* routing policy in this section. A cluster of sibling ASes managed by an organization or several closely related organizations can be treated as one entity. Outside of the sibling cluster, we safely apply PeerBoost routing policies to accommodate provider–customer and peer to peer relationships through local coordination. Then, within a cluster of sibling ASes, to set up sibling policies, ASes need to coordinate only with siblings.

6.1. AS hierarchy

A *sibling cluster* is a set of ASes owned by an organization or several closely related organizations, where any AS in the set can reach another AS in the set through sibling links only. We assume that if two ASes in the same sibling cluster are connected through a link, then the link is a sibling link. Note, ASes within a sibling cluster do not have to be fully connected. Namely, even if in a sibling cluster, AS A and AS B are siblings, and AS B and AS C are also siblings, it does not mean that A and C are directly connected.

6.2. Policy rules

The export policy for sibling relationships are clear. That is, ASes announce all routes to its siblings. For the other relationships, we can apply PeerBoost routing policies. In the following, we focus on the import policy of the hierarchical sibling policies.

Since we separate the convergence problem into two levels, let us first partition the routes according to these two levels. In hierarchical sibling routing policies, a route that only contains sibling links is an *internal route*. The other routes are *external routes* which go through multiple sibling clusters. The internal routes are always preferred over external routes. That is, for any destination, the routes that only go through siblings should be preferred. It is analogous to that, to reach a destination within the same domain, a router will no go through routers of another AS.

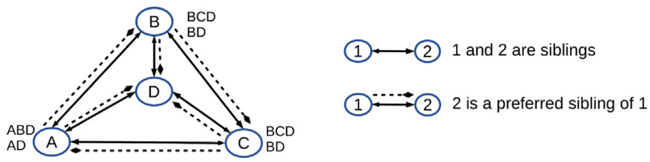

Fig. 17. BAD GADGET with a preferred sibling cycle.

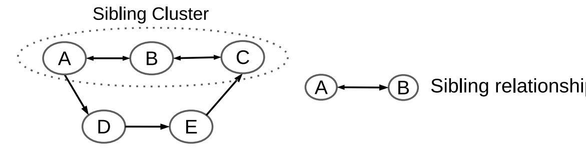

Fig. 18. An example of generic downstream sibling cycles.

Now, we consider how to rank external routes. The path of an external route can be separated into an *internal segment* and an *external segment*. The internal segment is the sequence of sibling links starting from the first AS in the route. The rest of the path is the external segment.

ASes first rank routes based on their external segments and use the internal segments to break tie. More specifically, we use PeerBoost import policy to rank the external segments.

To rank internal segments, each AS will assign a ranking for its siblings. ASes can determine the ranking of their siblings based on their routing requirements. For example, if an AS prefers using a larger sibling to access the rest of the Internet, then an AS can prefer a sibling with larger size over a sibling with smaller size.

Based on the ranking of its siblings, an AS classifies all siblings into *preferred siblings* and *backup siblings*, where an AS prefers the preferred siblings over the backup siblings. Each route uses *sibling backup level* (SBL) to represent backup siblings along the route. When a route is exported from a backup sibling to another backup sibling, SBL is increased by one. Otherwise, SBL keeps unchanged. When a route is exported into a new sibling cluster, SBL of the route is initialized as 0.

In sum, ASes determine the route ranking by following the import policy rules step by step.

- An AS prefers routes with lower backup level.
- An AS prefers internal routes over customer routes and preferred peer routes over provider routes and upstream peer routes.
- An AS prefers routes with lower SBL.
- An AS prefers routes from neighbors that are not siblings over routes from preferred siblings over routes from backup siblings.

6.3. Convergence condition

The hierarchical sibling policies are not guaranteed to be safe. To illustrate with an example, Fig. 17 shows a routing system that follows a hierarchical sibling policy. Clearly, it is a BAD GADGET, which means it is unsafe. Note that in this example, the preferred sibling relationships form a cycle, A-B-C-A. We define that a *preferred sibling (PS)* cycle is an AS cycle, $v_n \dots v_0$, where $v_0 = v_n$ and for each $i \in [0, n-1]$, v_i is a preferred sibling of v_{i+1}. As shown in Fig. 17, a PS cycle might lead to routing oscillation.

In addition to PS cycles, GD or GU cycles should be avoided as shown in Section 5. Here, we extend the definition of UP Peers and DP peers in the hierarchical sibling policy. To do so, we treat all ASes within a sibling cluster as a cluster-AS. When two ASes in different cluster-ASes are connected, the relationship of two cluster-ASes are defined by the relationship of the two ASes. Then, UP Peers can be defined among the cluster-AS. Namely, when a cluster-AS A announces a provider or upstream peer route to its peer, cluster-AS P, without increasing backup level, the cluster-AS P is an uphill preferred peer (UP Peer) of the cluster-AS A. Similarly, the other concepts, such as DP peers, can be extended to cluster-ASes. Accordingly, we can define GD and GU cycles in terms of the hierarchical sibling policy.

Theorem 6.1. *A hierarchical sibling policy is guaranteed to be safe, if there are no PS, GD or GU cycles.*

We provide the proof in Appendix C.

When the network operators of an AS want to change the routing policies, to guarantee the routing policy to be safe, the network operators need to check whether the policy change leads to a PS, GD or GU cycle. Within a sibling cluster, to avoid the PS cycle, ASes coordinate with their siblings only. Fig. 17 illustrates a PS cycle, A-B-C-A. Local coordination among A, B and C can avoid the cycle. For example, A can decrease the ranking for B and set B to be a backup sibling. Note, the routing policy of sibling links within a sibling cluster does not impact the existence of a GD cycle or a GU cycle. Fig. 18 illustrates an example of generic downstream sibling cycle. In this example, as long as A and C are both siblings of B, the AS cycle, A-B-C-E-D-A, is a GD cycle. Outside of the sibling cluster, to avoid the GD and GU cycles, the network operators of ASes in the cluster-ASes need to coordinate with the network operators of ASes in the customer cluster-ASes and the peer cluster-ASes as described in Section 5.

7. Evaluation

In this section, we evaluate the flexibility of the proposed routing policies. To evaluate the flexibility of a routing policy, we consider the number of possible routing configurations that can be applied to an AS when the AS follows the routing policy. If a routing policy enables more possible configurations that an AS can set up, then the routing policy has the greater flexibility. In Section 7.1, we describe how to count the possible configurations. In Section 7.2, we show the number of the possible configurations for ASes under various routing policies.

7.1. Counting possible configurations

To count the possible configurations of an AS, we consider both the export policy and the import policy of the AS. In the following, we propose the models for the export policy and the import policy of an AS respectively. Based on that, we count the routing configurations of an AS.

Routing policies mentioned in this paper, such as Gao–Rexford guidelines, safe backup policy and PeerBoost, all allow different configurations for the neighbors with the same agreement. For example, under these routing policies, an AS can always prefer one customer over another customer. Also, an AS can always prefer a provider over another provider. In order to focus on the difference of these routing policies, when we count the possible policy configurations, we assume that an AS sets up the routing policy for its neighbors based on the agreement established with the neighbors. Then, an AS will use the same routing configuration for its neighbors with the same agreement.

The export policy of an AS determines whether the routes from a neighbor can be announced to another neighbor and whether the backup level of the announced route is increased. Thus, we model the export policy configuration of an AS as a vector of *announcement choices*. We denote each announcement choice as $c_{i,j}$ which indicates the export policy configuration about the routes from neighbor group i to neighbor group j. Each neighbor group is a set of neighbors with the same agreement relationships, such as providers, peers and customers.

There are three possible values for the announcement choice, $c_{i,j}$, as follows.

- $c_{i,j} = 0$: the routes from neighbor group i are not announced to neighbor group j.

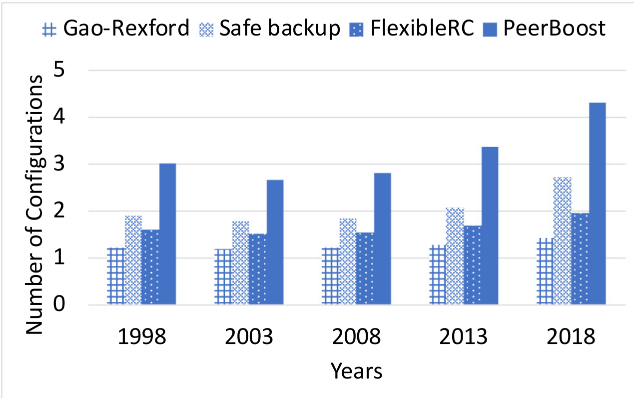

Fig. 19. The average number of possible routing configurations for ASes in various Internet topologies.

- $c_{i,j} = 1$: the routes from neighbor group i can be announced to neighbor group j while the backup level is increased.
- $c_{i,j} = 2$: the routes from neighbor group i can be announced to neighbor group j while the backup level is not increased.

The import policy of an AS determines how the AS ranks the routes from the neighbors. We model the import policy configuration of an AS as a vector of neighbors which indicates the ranking of the routes from different neighbor groups. Namely, if a neighbor group i is the first item in the vector, the routes from neighbor group i are preferred over the routes from the other neighbor groups in the vector.

Let us take MC network in Fig. 1 as an example to illustrate how to count the possible configurations. Assume that MC does not have any other neighbor except LEVEL 3 and UVM. When MC follows Gao–Rexford guidelines, there is only one possible export configuration. Namely, MC does not announce any routers from one neighbor to the other. Similarly, there is only 3 possible import configurations when MC follows Gao–Rexford guidelines. Namely, MC prefers LEVEL 3 over UVM, prefers UVM over LEVEL 3 or assigns the same local preference for LEVEL 3 and UVM. Combining all possible import configurations and all possible export configurations, there are only 3 possible routing configurations for MC. Alternatively, if MC follows PeerBoost policies, then there are 9 possible export configurations. The reason is that MC might apply R2P and P2R export rules, and selects the backup level of the announced routes. Accordingly, there are 3 possible import configurations when MC follows PeerBoost policies. In sum, there are 27 routing configurations of MC.

7.2. Possible configurations on the internet

We use the Internet topologies to evaluate the flexibility of Gao–Rexford guidelines [14], safe backup routing [15,16], FlexibleRC and PeerBoost. We measure the number of possible configurations of ASes in the Internet topologies. The Internet topologies and the AS relationships are derived from CAIDA dataset [1].

We show the number of AS configurations on average in Fig. 19. As Fig. 19 shows, safe backup routing, FlexibleRC and PeerBoost have the greater flexibility than Gao–Rexford guidelines. Safe backup routing, FlexibleRC and PeerBoost enable more possible routing configurations, since they extend Gao–Rexford guidelines in terms of the import policy rules and/or the export policy rules. PeerBoost policies allow the greatest flexibility among these policies. Comparing to Gao–Rexford guidelines, PeerBoost policies increase the number of possible configurations by nearly 2 times.

8. Related work

The evolution of the Internet architecture from a hierarchical to a flat structure has been noticed for years [2–6]. Not only stub networks, but also ISPs are using an open peering strategy to peer with more networks [28]. Dynam-IX [29] is proposed to facilitate operators to speed up peering establishment. Ahmed et al. investigate the performance difference between peering and transit interconnections [30]. For more than 90% ASes, peering paths outperform transit paths in terms of propagation delay, which provides one reason of the evolution.

Network operators and researchers are exploring the flexible usage of these peer links. For example, recently, large content providers, such as Google, Facebook and Akamai, try to exploit the peer links in a more flexible way [7–9]. They focus on a single network or several networks managed by an organization. Given the flat structure of the current Internet, we systematically explore flexible routing policies with convergence guarantee that can take advantage of these peer links in a global perspective.

A number of studies propose routing policy guidelines considering practical contractual commercial agreements between ASes. Gao and Rexford propose a prefer-customer and valley-free routing policy to accommodate provider–customer and peer to peer relationships [14]. After that, additional backup routes are enabled with a lower ranking [15] while sibling relationships are accommodated in [16,17]. In addition, [31] enables peer routes to be preferred over customer routes and keep the stability of the routing system through detecting recurrent routing loops. In this paper, we propose PeerBoost policies to enable both flexible import policy and flexible export policy. In contrast to backup routes enabled in [15], these newly-enabled routes in PeerBoost policies can be assigned with more flexible ranking. For example, the newly-enabled routes can be selected as the most preferred route when all steps are treated as preferred steps. [16] and [17] rank sibling–sibling policies according to the number of sibling links or the AS-PATH length. Sibling policies proposed in this paper, by contrast, enable more flexible route ranking. For example, in the hierarchical sibling policies, the route preference of a route does not decrease when the route is received from a preferred sibling. As a result, routes with more sibling links can be preferred over that with less sibling links.

Wang et al. relax the preferring-customer import policy by allowing each AS to customize the route selection on behalf of each neighbor [32]. Mahajan et al. exploit the negotiation between neighbor ASes to find a best route which minimizes the cost of all ASes along the route [33,34]. They focus on how to select a route flexibly through AS coordination. We explore the local coordination among ASes to enable both diverse paths and flexible route ranking.

To study the safety property of path-vector protocols, researchers proposed a number of abstract methods, including routing algebra [25,26,35,36] and Simple Path Vector Protocols (SPVP) [24,37]. These theoretical tools can be used to analyze whether a routing system is guaranteed to be safe and derive the conditions. We exploit the theoretical tools to derive simplified conditions which can be checked through local coordination without requiring all routing policies.

Since BGP protocol does not guarantee convergence property, there are several existing work [31,38,39] extend BGP protocol to resolve the convergence problem through detecting oscillations in the BGP routing system. We propose routing policies that enable network operators to avoid potential routing oscillations.

The model of inter-domain routing policy is exploited in various research areas, such as inter-domain protocols [40–43], inter-domain security schemes [44–46], inter-domain routing verification [47,48] and Internet path reliability [16,49]. PeerBoost policies proposed in this paper can be used to synthesize routing policies of the Internet for various research purposes. Although PeerBoost provides more choices for routing policies, we do not expect it significantly change on major results in experiments, since our safe conditions limit the extent that flexible routing policies can be applied.

Table A.4
The $\oplus_X$ operator for FlexibleRC routing policy.

$\oplus_X$	ϵ	(o, P)	(c, P)	(r, P)	(p, P)
c	$(c, X \bullet P)$	ϕ	$(c, X \bullet P)$	ϕ	ϕ
o	$(o, X \bullet P)$	ϕ	$(o, X \bullet P)$	ϕ	ϕ
r	$(r, X \bullet P)$	ϕ	$(r, X \bullet P)$	ϕ	ϕ
p	$(p, X \bullet P)$	$(p, X \bullet P)$	$(p, X \bullet P)$	$(p, X \bullet P)$	$(p, X \bullet P)$

A long research thread [10,21,50–57] has aimed to infer the relationship between the ASes from publicly available information, such as BGP routing tables and IXP database. According to the expression of the routing policy, the AS interconnections are classified into three relationship: provider–customer, peer to peer and sibling to sibling. It has long been recognized that peer to peer and sibling relationships are hard to be identified. The flexible routing policies described in this paper could be one of the reasons for the challenge to discover peer or sibling links.

9. Conclusion

We propose policy-rich routing through local coordination among ASes. More specifically, we systematically broaden routing policies to allow ASes to enable these potential routes and take advantage of all routes with great flexibility.

First of all, we propose FlexibleRC policies to provide more flexibility for ranking peer routes. In addition, we propose PeerBoost routing policies to enable selective export for providers and peers. Finally, we propose hierarchical sibling polices to accommodate sibling relationships. We derive convergence conditions for all above policies and propose the cycle-checking scheme to show that the local coordination among neighboring ASes and sibling ASes is sufficient to guarantee the convergence of the routing system. We evaluate the flexibility of the proposed routing policies. As the experimental results show, PeerBoost policy can increase the number of the possible policy configurations of an AS by about 2 times.

CRediT authorship contribution statement

Xiaozhe Shao: Methodology, Investigation, Writing – original draft, Writing – review & editing. **Lixin Gao:** Supervision, Funding acquisition, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Convergence of FlexibleRC policy

We exploit the algebraic theory [25,36] to prove the convergence conditions for policy-rich routing in this paper. When the algebra of a routing system is free, the routing system is guaranteed to converge. In the following proofs, we first model the routing policies through the routing algebra and then prove that the routing policies are guaranteed to converge.

For the FlexibleRC routing policy, we have $L = \{c, o, r, p\}$, $\sum = (L \times P^+) \cup \{\epsilon, \phi\}$, and $W = \{0, 1, 2, +\infty\}$. Links joining providers to customers are called customer links, and have label c; links joining customer to providers are called provider links, and have label p. Links joining two ASes with peer to peer relationships are called peer links. If a peer link joins an AS to its OC peer, the peer link is an OC peer link, and has label o. Otherwise, the peer link is an under-customer peer link, and has label r. The five signatures, ϵ, c, o, r, p for permitted paths represent five classes of paths: trivial path (comprised of a single node); customer paths (in which the first link is a provider–customer link); OC peer paths (in which the first link is an OC peer link); under-customer peer paths (in which the first link is an under-customer peer link) and provider paths (in which the first link is a customer–provider link). P^+ is the set of all possible AS paths.

Each AS modifies the AS path of a route before announcement. Therefore, $\oplus$ of various ASes are different. We use $\oplus_X$ to denote the $\oplus$ of the AS, X. The $\oplus_X$ operation is given in the next chart, where the first operand, a label, appears in the first column and the second operand, a signature, appears in the first row.

For example, $(o, P) \oplus_X o = \phi$ means that an OC peer route will not be exported through an OC peer link, since only customer routes are announced to peers. In Table A.4, $X \bullet P$ indicates appending X into the origin AS path, P.

According to the import policy of FlexibleRC, the function f satisfies the following rule.

$$f((c, P_c)) \prec f((p, P_p)) \tag{A.1}$$

Namely, a customer route is preferred over a provider route.

Definition 1. In routing algebra, a cycle $u_n u_{n-1} \dots u_1 u_0$, with $u_n = u_0$ is free if for every $\alpha_0, \alpha_1, \dots, \alpha_{n-1}, \alpha_n \in \sum - \{\phi\}$, with $\alpha_0 = \alpha_n$, there is an index i, $1 \le i \le n$, such that $f(\alpha_i) \preceq f(L(u_i, u_{i-1}) \oplus \alpha_{i-1})$ [25,36].

According to the freeness property, we use Formula (A.2) to represent that a cycle, $u_n u_{n-1} \dots u_0$ with $u_n = u_0$, is not free.

$$\exists \alpha_0, \alpha_1 \dots \alpha_n, (\alpha_0 \ne \phi) \wedge \dots (\alpha_n \ne \phi) \wedge H(\alpha_0, u_0) \wedge \dots H(\alpha_n, u_n) \wedge \alpha_0 = \alpha_n \wedge$$
$$f\left(L(u_n, u_{n-1}) \oplus \alpha_{n-1}\right) \preceq f(\alpha_n) \wedge \dots \wedge f\left(L(u_1, u_0) \oplus \alpha_0\right) \preceq f(\alpha_1) \tag{A.2}$$

The predicate symbol $H(\alpha, n)$ indicates that the first node of the route α is n.

The freeness property is relevant to the direction of a cycle. Given a cycle $u_n u_{n-1} \dots u_1 u_0$, with $u_n = u_0$, we define its inverse cycle as $u_0 u_1 \dots u_{n-1} u_n$.

To prove the convergence condition, we first prove the following Lemma.

Lemma 1. *If a cycle is not free, then the cycle is an OC cycle or its inverse cycle is an OC cycle.*

Proof. Formally, to prove the lemma, we need to prove the following formula.

$$\forall u_0, \dots, u_n, \neg Free(u_n \dots u_0) \wedge Cycle(u_n \dots u_0) \implies OC(u_n \dots u_0) \wedge OC(u_0 \dots u_n) \tag{A.3}$$

where $Cycle(u_0 \dots u_n)$ is a predicate symbol indicating that $u_0 u_1 \dots u_n$ is a cycle and $Free(u_n \dots u_0)$ is a predicate symbol indicating that the cycle, $u_n \dots u_0$, is free.

We first translate the FlexibleRC policy into the formula rules. The routing system obeys those rules, if the routing system follows FlexibleRC policy. And then, we apply these rules to prove the lemma.

Rule 1: No route is preferred over trivial paths.

$$\forall \alpha, \alpha = \epsilon \wedge H(\alpha, u) \wedge f(L(u, u') \oplus \alpha') \preceq f(\alpha) \implies L(u, u') \oplus \alpha' = \epsilon \implies \bot \tag{A.4}$$

Given any $\alpha = \epsilon$ which is a trivial path, u, if u does not prefer α over another path, $L(u, u') \oplus \alpha'$, then the path, $L(u, u') \oplus \alpha'$, should be a trivial path. Apparently, the path, $L(u, u') \oplus \alpha'$, is not a trivial path. We will use Rule 1 to show that in the freeness property, α cannot be a trivial path.

Rule 2: Customer or OC paths can be preferred over customer routes.

$$\forall \alpha, (\alpha = c \vee \alpha = o) \wedge H(\alpha, u) \wedge f(L(u, u') \oplus \alpha') \preceq f(\alpha)$$
$$\implies \left(L(u, u') = c \vee L(u, u') = o\right) \wedge \alpha' = c \tag{A.5}$$

Given any route, α which is a customer route or an OC peer route of u, if u does not prefer α over another route $L(u,u')\oplus\alpha'$, then $L(u,u')\oplus\alpha'$ has to be a customer route or an OC peer route. Therefore, the link, $L(u,u')$, should be a provider–customer link or a OC peer link.

Rule 3: Under-customer peer and provider paths can be only preferred over under-customer peer and provider paths.

$$\forall\alpha', (\alpha' = r \vee \alpha' = p) \wedge H(\alpha', u') \wedge f(L(u,u') \oplus \alpha') \preceq f(\alpha)$$
$$\wedge\, \alpha \neq \phi \wedge H(\alpha, u) \implies L(u,u') = p \wedge (\alpha = r \vee \alpha = p) \tag{A.6}$$

Given any route, α' which is a provider or peer route of u', if the route is announced to u and u does not prefer its another router α over $L(u,u')\oplus\alpha'$, then u' is a provider of u and α is a peer or provider route.

Then, we will use the aforementioned rules to show that given a cycle, $u_n u_{n-1} \ldots u_0$, is not free in FlexibleRC policy, the cycle is an OC cycle or its inverse cycle is an OC cycle. We will first use Rule 1 to show that the routes on the cycle cannot be a trivial path. Therefore, a route is a provider route, an under-customer peer route, an over-peer route or a customer route. Then, we consider the types of each route from α_n to α_1. If α_n is a customer route or an OC route, we use Rule 2 to show that α_{n-1} should be a customer route. If we apply Rule 2 $(n-1)$ times on $\alpha_{n-1}\ldots\alpha_1$ sequentially, we can show that all routes on the cycle are customer routes or OC routes. Similarly, we can apply Rule 3 for the case that α_n is an under-customer peer route or a provider route. Applying Rule 3 n times, we can show that all routes are under-customer peer routes or provider routes. Finally, for both cases, the cycle is an OC cycle.

Each route, α_i, has five possible values. We can rewrite Formula (A.2) as follows.

$$Formula\ (A.2) \implies Formula\ (A.2) \wedge$$
$$(\alpha_n = \epsilon \vee \alpha_n = c \vee \alpha_n = o \vee \alpha_n = r \vee \alpha_n = p)$$
$$(\alpha_n \textbf{ has five possible values}, \epsilon, c, o, r, p.) \tag{A.7}$$

Then, we can proof the Lemma through applying the aforementioned rules as follows.

$$Formula\ (A.7) \implies \exists\alpha_0, \alpha_1 \ldots \alpha_n, (\alpha_0 = \alpha_n) \wedge (\alpha_0 \neq \phi)$$
$$\wedge \ldots (\alpha_n \neq \phi) \wedge f\big(L(u_n, u_{n-1}) \oplus \alpha_{n-1}\big) \preceq f(\alpha_n) \wedge \ldots$$
$$f\big(L(u_1, u_0) \oplus \alpha_0\big) \preceq f(\alpha_1) \wedge H(\alpha_0, u_0) \wedge \ldots H(\alpha_n, u_n) \wedge$$
$$\big(\alpha_n = c \vee \alpha_n = o \vee \alpha_n = r \vee \alpha_n = p\big)$$
$$(\textbf{Applying Rule 1, where substituting } \alpha \textbf{ for } \alpha_n.)$$
$$\implies \exists\alpha_0, \alpha_1 \ldots \alpha_n, (\alpha_0 = \alpha_n) \wedge (\alpha_0 \neq \phi)$$
$$\wedge \ldots (\alpha_n \neq \phi) \wedge f\big(L(u_n, u_{n-1}) \oplus \alpha_{n-1}\big) \preceq f(\alpha_n) \wedge \ldots$$
$$f\big(L(u_1, u_0) \oplus \alpha_0\big) \preceq f(\alpha_1) \wedge H(\alpha_0, u_0) \wedge \ldots H(\alpha_n, u_n) \wedge$$
$$\Big((\alpha_n = c \vee \alpha_n = o) \wedge \big(L(u_n, u_{n-1}) = c \vee L(u_n, u_{n-1}) = o\big) \wedge \alpha_{n-1} = c\Big)$$
$$\vee \Big((\alpha_n = r \vee \alpha_n = p) \wedge L(u_n, u_{n-1}) \wedge (\alpha_{n-1} = r \vee \alpha_{n-1} = p)\Big)$$
$$(\textbf{Applying Rule 2 and Rule 3 for } \alpha_n\textbf{, wherereplacing } \alpha \textbf{ with } \alpha_n \textbf{ and } \alpha' \textbf{ with } \alpha_{n-1})$$
$$\implies \forall i, 1 \le i \wedge i \le n \wedge$$
$$\bigg(\Big(\alpha_i = c \wedge \big(L(u_i, u_{i-1}) = c \vee L(u_i, u_{i-1}) = o\big)\Big)$$
$$\vee \Big((\alpha_i = r \vee \alpha_i = p) \wedge L(u_i, u_{i-1}) = p\Big)\bigg)$$
$$(\textbf{Applying Rule 2 and Rule 3 on } \alpha_{n-1} \ldots \alpha_1 \textbf{ sequentially.})$$
$$\implies OC(u_0 u_1 \ldots u_n) \vee OC(u_n u_{n-1} \ldots u_0)$$
$$(\textbf{Applying OC cycle definition}) \quad \square \tag{A.8}$$

Theorem 4.1. *A FlexibleRC routing policy is guaranteed to be safe, if there are no OC cycles.*

Proof. If there are no OC cycles, according to Lemma 1 all cycles should be free. Thus, the routing system is guaranteed to converge. □

Appendix B. Convergence of peerboost policy

For PeerBoost policy, we have $L = \{(u_x, u_y), (u_y, u_x) | u_x$ and u_y are connected $\}$ and $\sum = (\{c, p, pdpr, dpr, upr\} \times R_0^+) \cup \{\epsilon, \phi\}$, where $pdpr$ indicates a preferred downstream peer route, dpr indicates a downstream peer route that is not preferred, upr indicates an upstream peer route and R_0^+ is the range for the backup level of a route. According to the first import rule of PeerBoost, $f(\epsilon) \prec f(T_1, N_1) \prec f(T_2, N_2)$, where $T_1, T_2 \in \{c, p, pdpr, dpr, upr\}$, $N_1 and N_2 \in R_0^+$ and $N_1 < N_2$. According to the second import rule of PeerBoost, $f(T_1, N_1) \prec f(T_2, N_1)$, if $T_1 \in \{c, pdpr\}$ and $T_2 \in \{p, dpr, upr\}$.

Lemma 2. *In the algebra for PeerBoost policy, if there is no GD cycles or GU cycles, all cycles are free.*

Proof. To prove the Lemma, we just need to prove that if a cycle is not free, it is a GD cycle or a GU cycle. To do so, we first show that all α_i in Formula (A.2) have the same backup level. Then, we show that each link in the cycle satisfies the definition of GD or GU cycle. To do that, we separate the proof into two steps. In the first step, we want to show that if a cycle in PeerBoost policy is not free, then all routes in the cycle have the same backup level. In the second step, we use the definition of GD and GU cycles, to show that the aforementioned cycle is a GD or GU cycle.

B.1. Equal backup level

We first use logic formulas to represent the rules of PeerBoost policy as follows. Then, we will show that if a cycle is not free, then those routes involved in the cycle have the same backup level.

Rule 4: An AS prefers routes with lower backup level. This Rule is derived from the first import policy rule of PeerBoost.

$$\forall\alpha_{i-1}, \alpha_i, f(\alpha_{i-1}) \preceq f(\alpha_i) \implies B(\alpha_{i-1}) \le B(\alpha_i) \tag{B.1}$$

The function symbol $B(\alpha)$ indicates the backup level of the route α.

Rule 5: Backup level monotonically increases during route announcement. According to the export policy of PeerBoost, the backup level of a route increases or keeps unchanged when the route is announced to another AS.

$$\forall\alpha_i, u_{i+1}, u_i, H(\alpha_i, u_i) \implies B(\alpha_i) \le B(L(u_{i+1}, u_i) \oplus \alpha_i) \tag{B.2}$$

Then, we use those rules for the first step of the proof. We will first use Rule 4 to derive the relationship of the backup level of routes on the cycle. A preferred route has a smaller or equal backup level. Then, we use Rule 5 to show that on a cycle all routes have the same backup level.

$$Formula\ (A.2) \implies B\big(L(u_n, u_{n-1}) \oplus \alpha_{n-1}\big) \le B(\alpha_n)$$
$$\wedge \ldots f\big(L(u_1, u_0) \oplus \alpha_0\big) \preceq f(\alpha_1) \wedge H(\alpha_0, u_0) \wedge \ldots H(\alpha_n, u_n)$$
$$(\textbf{Applying Rule 4 for i = n})$$
$$\implies B\big(L(u_n, u_{n-1}) \oplus \alpha_{n-1}\big) \le B(\alpha_n)$$
$$\wedge \ldots B\big(L(u_1, u_0) \oplus \alpha_0\big) \le B(\alpha_1) \wedge H(\alpha_0, u_0) \wedge \ldots H(\alpha_n, u_n)$$
$$(\textbf{Applying Rule 4 n} - \textbf{1 times})$$
$$\implies B(\alpha_{n-1}) \le B\big(L(u_n, u_{n-1}) \oplus \alpha_{n-1}\big) \wedge \ldots B(\alpha_0) \le B\big(L(u_1, u_0) \oplus \alpha_0\big)$$
$$\wedge B\big(L(u_n, u_{n-1}) \oplus \alpha_{n-1}\big) \le B(\alpha_n) \wedge \ldots B\big(L(u_1, u_0) \oplus \alpha_0\big) \le B(\alpha_1)$$
$$(\textbf{Applying Rule 5 n times for i as 1} \ldots \textbf{n})$$
$$\implies B(\alpha_0) = B(\alpha_1) = \cdots = B(\alpha_n)$$
$$= B\big(L(u_1, u_0) \oplus \alpha_0\big) = \cdots = \big(L(u_n, u_{n-1}) \oplus \alpha_{n-1}\big) \tag{B.3}$$

B.2. GD or GU cycles

Now, we know that all routes that we are considering have the same backup level. In the next step, we will use the properties derived from PeerBoost routing policy to show that those cycles are GD cycles or GU cycles.

In PeerBoost polices, there are only four types of routes: customer routes, downstream peer routes, upstream peer routes and provider routes. We consider two scenarios of the cycles.

- Scenario 1: one of the routes on the cycle, $L(u_{i+1}, u_i) \oplus \alpha_i$, is a customer route or a downstream peer route.
- Scenario 2: one of the routes on the cycle, $L(u_{i+1}, u_i) \oplus \alpha_i$, is an upstream peer route or a provider route.

For Scenario 1, we will show that if the cycle is not free, the cycle is a GU cycle. For Scenario 2, we will show that if the cycle is not free, the cycle is a GD cycle.

B.2.1. Scenario 1: GD cycles

If one of the routes on the non-free cycle, saying $L(u_n, u_{n-1}) \oplus \alpha_{n-1}$, is a customer route or a downstream peer route, then the cycle is a GD cycle. To prove it, we first illustrate two Rules derived from PeerBoost routing policy as follows.

Rule 6: Exporting customer route and preferred downstream peer routes to providers or peers without increasing backup level. According to PeerBoost policies, only a customer route or a preferred downstream peer route can be exported to a peer or a provider without increasing backup level. Therefore, we have the following formula.

$$\begin{aligned}
&\forall B\big(L(u_i,u_{i-1}) \oplus \alpha_{i-1}\big) = B(\alpha_{i-1}) \wedge \\
&\big((L(u_i,u_{i-1}) \oplus \alpha_{i-1}) = c \vee (L(u_i,u_{i-1}) \oplus \alpha_{i-1}) = dpr\big) \\
&\implies \alpha_{i-1} = c \vee \alpha_{i-1} = pdp
\end{aligned} \tag{B.4}$$

$L(u_i, u_{i-1}) \oplus \alpha_{i-1} = dpr$ is a predicate that the route $L(u_i, u_{i-1}) \oplus \alpha_{i-1}$ is a downstream peer route. $\alpha_{i-1} = pdp$ is a predicate that the route α_{i-1} is a preferred downstream peer route.

Rule 7: Only routes from DP peers or customers are preferred over a customer route or a preferred downstream peer route. According to the import policy of PeerBoost, a customer route can be preferred over customer routes or preferred downstream peer routes. Alternatively, according to the definition of DP peer, a route from a DP peer can be preferred over customer routes or preferred downstream peer routes. Then, we have the following formula.

$$\begin{aligned}
&\forall \alpha_i, (\alpha_i = c \vee \alpha_i = pdp) \wedge f\big(L(u_i,u_{i-1}) \oplus \alpha_{i-1}\big) \le f(\alpha_i) \\
&\implies \big(L(u_i,u_{i-1}) = DP \wedge L(u_i,u_{i-1}) \oplus \alpha_{i-1} = dpr\big) \\
&\vee \big(L(u_i,u_{i-1}) = c \wedge L(u_i,u_{i-1}) \oplus \alpha_{i-1} = c\big)
\end{aligned} \tag{B.5}$$

Given Rule 6 and Rule 7, we can prove Lemmas 2 under Scenario 1 as follows.

$$\begin{aligned}
&Formula\ (B.3)\ \wedge \big((L(u_n,u_{n-1}) \oplus \alpha_{n-1} = c) \vee (L(u_n,u_{n-1}) \oplus \alpha_{n-1} = dpr)\big) \\
&\implies Formula\ (B.3)\ \wedge (\alpha_{n-1} = c \vee \alpha_{n-1} = pdp) \\
&(\textbf{Applying Rule 6 for i = n}) \\
&\implies Formula\ (B.3)\ \wedge \Big(\big(L(u_{n-1},u_{n-2}) = DP \wedge L(u_{n-1},u_{n-2}) \oplus \alpha_{i-2} = dpr\big) \\
&\vee \big(L(u_{n-1},u_{n-2}) = c \wedge L(u_{n-1},u_{n-2}) \oplus \alpha_{i-2} = c\big)\Big) \\
&(\textbf{Applying Rule 7 for i = n - 1}) \\
&\implies \forall i, 1 \le i \le n \wedge \big(L(u_i,u_{i-1}) = DP \vee L(u_i,u_{i-1}) = c\big) \\
&(\textbf{Alternatively applying Rule 6 and Rule 7 n - 1 times}) \\
&\implies GD(u_n \ldots u_0) \\
&(\textbf{Definition of GD cycle})
\end{aligned} \tag{B.6}$$

B.2.2. Scenario 2: GU cycles

During the above derivation, we know that if one route on the cycle is a customer route or a downstream peer route, then $\forall i$, $L(u_i, u_{i-1}) \oplus \alpha_{i-1}$ is a downstream peer route or a customer route. Thus, routes of cycles for the rest cases are upstream peer routes or provider routes. Therefore, we can redefine Scenario 2 as all routes on the cycle are upstream peer routes or provider routes. To prove Lemma 2 under Scenario 2, we first illustrate two Rules derived from PeerBoost routing policy as follows.

Rule 8: In Scenario 2, $\forall i, 0 \le i \le n$, α_i is an upstream peer route or a provider route.

$$\begin{aligned}
&\forall \alpha_i, B\big(L(u_i,u_{i-1}) \oplus \alpha_{i-1}\big) = B(\alpha_{i-1}) \\
&\wedge \Big(\big(L(u_i,u_{i-1}) \oplus \alpha_{i-1} = upr\big) \vee \big(L(u_i,u_{i-1}) \oplus \alpha_{i-1} = p\big)\Big) \\
&\wedge B\big(L(u_{i+1},u_i) \oplus \alpha_i\big) = B(\alpha_i) \\
&\wedge \Big(\big(L(u_{i+1},u_i) \oplus \alpha_i = upr\big) \vee \big(L(u_{i+1},u_i) \oplus \alpha_{i-1} = p\big)\Big) \\
&\wedge f\big(L(u_i,u_{i-1}) \oplus \alpha_{i-1}\big) \le f(\alpha_i) \\
&\implies \alpha_i = upr \vee \alpha_i = p
\end{aligned} \tag{B.7}$$

$\alpha_i = upr$ indicates that α_i is an upstream peer route and $\alpha_i = p$ indicates that α_i is a provider route.

Rule 9: If all routes involved in a cycle are upstream peer routes or provider routes, then all links in the cycle are from the provider to the customer or from an AS to its UP peer.

$$\begin{aligned}
&\big(L(u_i,u_{i-1}) \oplus \alpha_{i-1} = upr \vee L(u_i,u_{i-1}) \oplus \alpha_{i-1} = p\big) \\
&\wedge (\alpha_{i-1} = upr \vee \alpha_{i-1} = p) \wedge B(\alpha_{i-1}) = B(L(u_i,u_{i-1}) \oplus \alpha_{i-1}) \\
&\implies L(u_i,u_{i-1}) = UP \vee L(u_i,u_{i-1}) = p
\end{aligned} \tag{B.8}$$

$L(u_i, u_{i-1}) = UP$ indicates that u_{i-1} is an UP peer of u_i.

In the following, we consider Scenario 2. Namely, routes on the cycles are upstream peer routes or provider routes.

$$\begin{aligned}
&Formula\ (B.3)\ \wedge \big((L(u_n,u_{n-1}) \oplus \alpha_{n-1} = upr) \vee (L(u_n,u_{n-1}) \oplus \alpha_{n-1} = p)\big) \\
&\wedge \ldots\big((L(u_1,u_0) \oplus \alpha_0 = upr) \vee (L(u_1,u_0) \oplus \alpha_0 = p)\big) \\
&\implies Formula\ (B.3)\ \wedge \forall i, 1 \le i \le n \wedge (\alpha_i = c \vee \alpha_i = pdp) \\
&(\textbf{Applying Rule 8 for n times}) \\
&\implies \forall i, 1 \le i \le n \wedge \big(L(u_i,u_{i-1}) = UP \vee L(u_i,u_{i-1}) = p\big) \\
&(\textbf{Applying Rule 9 for n times}) \\
&\implies GU(u_n \ldots u_0) \\
&(\textbf{Definition of GU cycle})
\end{aligned} \tag{B.9}$$

Then we prove this lemma. □

Theorem 5.1. *A PeerBoost routing policy is safe, if there are neither GD cycles nor GU cycles.*

Proof. If there are neither GD cycles nor GU cycles, according to Lemma 2, all cycles are free. Then, the routing system is guaranteed to converge. □

Appendix C. Convergence of sibling policies

We derive the algebra for hierarchical sibling policy and show that if the convergence conditions hold, hierarchical sibling policy converges. We have $L = \{(u_x, u_y), (u_y, u_x) | u_x$ and u_y are connected. $\}$ and $\sum = (\{I, c, p, pdpr, dpr, upr\} \times \{E, PS, BS\} \times R_0^+ \times R_0^+ \times A^+ \times P^+) \cup \{\epsilon\}$, where I indicates an internal route, E indicates a route received from a neighbor which is not a sibling, PS indicates a route received from a preferred sibling, and BS indicates a route received from a backup sibling. The first number in the signature is the backup level of the route while the second number is the SBL of the route.

We define a few additional symbols for hierarchical sibling policy. The predicate symbol $IN(\alpha)$ indicates that α is an internal route. The predicate symbols, $E(\alpha)$, $PS(\alpha)$ and $BS(\alpha)$ indicate that α is received from a non-sibling neighbor, a preferred sibling and a backup sibling respectively. The predicate symbol $PL(u, v)$ and $BL(u, v)$ indicates that v is a preferred sibling of u and v is a backup sibling of u respectively. The function symbol $SB(\alpha)$ indicates the SBL of the route α.

Lemma 3. *In the algebra for hierarchical sibling policy, if there is no PS cycles, GD cycles and GU cycles, all cycles are free.*

Proof. To prove the Lemma, we just need to prove that if there is no PS cycles, a cycle that is not free is a GD cycle or a GU cycle. To do that, we consider two possible cases: α_i is an internal route or α_i is not an internal route.

C.1. When a route is an internal route

Considering the hierarchical sibling policy, Rule 4 is satisfied. We also have the following additional formulas.

Internal routes are preferred over all the other routes. When an internal route is exported into a sibling, the SBL monotonically increases.

Rule 10: SBL monotonicity of internal routes

$$\forall \alpha, \alpha', IN(\alpha) \wedge H(\alpha, u) \wedge f\left(L(u, u') \oplus \alpha'\right) \leq f(\alpha) \implies IN(\alpha') \wedge \left(SB(\alpha') < SB(\alpha) \vee SB(\alpha') = SB(\alpha)\right) \tag{C.1}$$

Rule 11: ASes prefer the preferred sibling routes over the backup sibling routes.

$$\forall \alpha, \alpha', SB(\alpha) = SB(\alpha') \wedge f\left(L(u, u') \oplus \alpha'\right) \leq f(\alpha) \wedge IN(\alpha) \wedge IN(\alpha') \wedge H(\alpha, u) \wedge H(\alpha', u') \wedge PS(\alpha) \implies PL(u, u') \wedge BL(u', u) \wedge PS(\alpha') \tag{C.2}$$

Rule 12: The backup sibling routes can be only preferred over the backup sibling routes.

$$\forall \alpha, \alpha', SB(\alpha) = SB(\alpha') \wedge f\left(L(u, u') \oplus \alpha'\right) \leq f(\alpha) \wedge IN(\alpha) \wedge IN(\alpha') \wedge H(\alpha, u) \wedge H(\alpha', u') \wedge BS(\alpha') \implies BL(u, u') \wedge PL(u', u) \wedge BS(\alpha) \tag{C.3}$$

Given any non-free cycle, $u_n ... u_0$, we first assume that α_n is an internal route. For simplicity of expression, we define $Cycle(u_n ... u_0)$ as a predicate symbol indicating that $u_n ... u_0$ is a cycle and $Free(u_n ... u_0)$ as a predicate symbol indicating that the cycle, $u_n ... u_0$, is free. Specifically, we first show that if, in a non-free cycle, one of route, α_n, is an internal route, then α_n is a preferred sibling route or a backup sibling route. Then, we use Rule 10 to show that all routes in the cycle have the same sibling backup level. After that, we can use Rule 11 and Rule 12 to show that the cycle is a PS cycle.

$$\forall u_0, \dots, u_n, \neg Free(u_n ... u_0) \wedge Cycle(u_n ... u_0) \wedge IN(\alpha_n)$$
$$\implies \exists \alpha_0, \alpha_1 ... \alpha_n, (\alpha_0 = \alpha_n) \wedge (\alpha_0 \neq \phi) \wedge ... (\alpha_n \neq \phi) \wedge$$
$$f\left(L(u_n, u_{n-1}) \oplus \alpha_{n-1}\right) \leq f(\alpha_n) \wedge ... f\left(L(u_1, u_0) \oplus \alpha_0\right) \leq f(\alpha_1) \wedge$$
$$H(\alpha_0, u_0) \wedge ... H(\alpha_n, u_n) \wedge IN(\alpha_n) \wedge \left(PS(\alpha_n) \vee BS(\alpha_n)\right)$$

(**Applying Rule 4**)

$$\implies \forall i, 1 \leq i \wedge i \leq n \wedge SB(\alpha_{i-1}) = SB(\alpha_i) \wedge$$
$$SB(\alpha_i) = SB\left(L(u_i, u_{i-1}) \oplus \alpha_{i-1}\right) \wedge SB(\alpha_i) = SB(\alpha_{i-1}) \wedge$$
$$f\left(L(u_i, u_{i-1}) \oplus \alpha_{i-1}\right) \leq f(\alpha_i) \wedge (PS(\alpha_i) \vee BS(\alpha_i))$$

(**Applying Rule 10 n times**)

$$\implies \left(PL(u_n, u_{n-1}) \wedge ... PL(u_1, u_0)\right) \vee \left(PL(u_0, u_1) \wedge PL(u_{i-n}, u_n)\right)$$

(**Applying Rule 11 and 12 n times**)

$$\implies PSC(u_n ... u_0) \text{ (\textbf{Definition of PS cycle})} \tag{C.4}$$

C.2. When a route is not an internal route

Clearly, if one route on a cycle is not an internal route, no route on the cycle is an internal routes. To rank a route, the external segment has the priority. The proof for PeerBoost policies can be applied to the sibling policies, since at sibling cluster level the policy is same with PeerBoost.

Therefore, we show that is a cycle is not free, the cycle is PS, GD or GU cycles. □

Theorem 6.1. *A hierarchical sibling policy is guaranteed to be safe, if there are no PS, GD or GU cycles.*

Proof. If there are no PS cycles, GD cycles or GU cycles, according to Lemma 3, all cycles are free. Then, the routing system is guaranteed to converge. □

References

[1] CAIDA, The CAIDA AS relationships dataset, <05.1998-03.2021>, 2017, http://www.caida.org/data/as-relationships/.

[2] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, F. Jahanian, Internet inter-domain traffic, SIGCOMM Comput. Commun. Rev. 41 (4) (2010) —-.

[3] A. Dhamdhere, C. Dovrolis, The internet is flat: Modeling the transition from a transit hierarchy to a peering mesh, in: Proceedings of the 6th International COnference, in: Co-NEXT '10, ACM, New York, NY, USA, 2010, pp. 21:1—-21:12.

[4] R. Anwar, H. Niaz, D. Choffnes, I. Cunha, P. Gill, E. Katz-Bassett, Investigating interdomain routing policies in the wild, in: Proceedings of the 2015 Internet Measurement Conference, in: IMC '15, ACM, New York, NY, USA, 2015, pp. 71–77.

[5] P. Gill, M. Arlitt, Z. Li, A. Mahanti, The flattening internet topology: Natural evolution, unsightly barnacles or contrived collapse? in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), in: 4979 LNCS, 2008, pp. 1–10.

[6] R. Klöti, B. Ager, V. Kotronis, G. Nomikos, X. Dimitropoulos, A comparative look into public IXP datasets, SIGCOMM Comput. Commun. Rev. 46 (1) (2016) 21–29.

[7] K.-K. Yap, M. Motiwala, J. Rahe, S. Padgett, M. Holliman, G. Baldus, M. Hines, T. Kim, A. Narayanan, A. Jain, V. Lin, C. Rice, B. Rogan, A. Singh, B. Tanaka, M. Verma, P. Sood, M. Tariq, M. Tierney, D. Trumic, V. Valancius, C. Ying, M. Kallahalla, B. Koley, A. Vahdat, Taking the edge off with espresso: Scale, reliability and programmability for global internet peering, in: Proceedings of the Conference of the ACM Special Interest Group on Data Communication, in: SIGCOMM '17, ACM, New York, NY, USA, 2017, pp. 432–445.

[8] B. Schlinker, H. Kim, T. Cui, E. Katz-Bassett, H.V. Madhyastha, I. Cunha, J. Quinn, S. Hasan, P. Lapukhov, H. Zeng, Engineering egress with edge fabric: Steering oceans of content to the world, in: Proceedings of the Conference of the ACM Special Interest Group on Data Communication, in: SIGCOMM '17, ACM, New York, NY, USA, 2017, pp. 418–431.

[9] F. Wohlfart, N. Chatzis, C. Dabanoglu, G. Carle, W. Willinger, Leveraging interconnections for performance: the serving infrastructure of a large CDN, in: Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, ACM, 2018, pp. 206–220.

[10] B. Augustin, B. Krishnamurthy, W. Willinger, IXPs: Mapped? in: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement, in: IMC '09, ACM, New York, NY, USA, 2009, pp. 336–349.

[11] V. Kotronis, R. Klöti, M. Rost, P. Georgopoulos, B. Ager, S. Schmid, X. Dimitropoulos, Stitching inter-domain paths over IXPs, in: Proceedings of the Symposium on SDN Research, in: SOSR '16, ACM, New York, NY, USA, 2016, pp. 17:1–17:12.

[12] S.Y. Qiu, P.D. McDaniel, F. Monrose, Toward valley-free inter-domain routing, in: 2007 IEEE International Conference on Communications, 2007, pp. 2009–2016.

[13] R. Mazloum, M.O. Buob, J. Augè, B. Baynat, D. Rossi, T. Friedman, Violation of interdomain routing assumptions, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), in: 8362 LNCS, 2014, pp. 173–182.

[14] L. Gao, J. Rexford, Stable internet routing without global coordination, IEEE/ACM Trans. Netw. 9 (6) (2001) 681–692.

[15] L. Gao, T.G. Griffin, J. Rexford, Inherently safe backup routing with BGP, in: Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213), Vol. 1, 2001, pp. 547–556.

[16] Y. Liao, L. Gao, R. Guerin, Z.L. Zhang, Safe interdomain routing under diverse commercial agreements, IEEE/ACM Trans. Netw. 18 (6) (2010) 1829–1840.

[17] J.L. Sobrinho, Correctness of routing vector protocols as a property of network cycles, IEEE/ACM Trans. Netw. 25 (1) (2017) 150–163.

[18] G. Huston, Interconnection, peering, and settlements, in: Proc. INET, Vol. 9, 1999.

[19] P. Gill, M. Schapira, S. Goldberg, A survey of interdomain routing policies, SIGCOMM Comput. Commun. Rev. 44 (1) (2013) 28–34.

[20] R. Oliveira, D. Pei, W. Willinger, B. Zhang, L. Zhang, The (in)completeness of the observed internet AS-level structure, IEEE/ACM Trans. Netw. 18 (1) (2010) 109–122.

[21] V. Giotsas, M. Luckie, B. Huffaker, k. claffy, Inferring complex AS relationships, in: Proceedings of the 2014 Conference on Internet Measurement Conference, in: IMC '14, Association for Computing Machinery, New York, NY, USA, 2014, pp. 23–30.

[22] V. Giotsas, S. Zhou, M. Luckie, k. claffy, Inferring multilateral peering, in: Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies, in: CoNEXT '13, ACM, New York, NY, USA, 2013, pp. 247–258.

[23] T.G. Griffin, G. Wilfong, An analysis of BGP convergence properties, in: Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, in: SIGCOMM '99, ACM, New York, NY, USA, 1999, pp. 277–288.

[24] T.G. Griffin, F.B. Shepherd, G. Wilfong, The stable paths problem and interdomain routing, IEEE/ACM Trans. Netw. 10 (2) (2002) 232–243.

[25] J.L. Sobrinho, Network routing with path vector protocols: Theory and applications, in: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, in: SIGCOMM '03, ACM, New York, NY, USA, 2003, pp. 49–60.

[26] T.G. Griffin, J.L. Sobrinho, Metarouting, in: Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM, Philadelphia, Pennsylvania, USA, 2005, pp. 1–12.

[27] F. Streibelt, F. Lichtblau, R. Beverly, A. Feldmann, C. Pelsser, G. Smaragdakis, R. Bush, BGP Communities: Even more Worms in the Routing Can, in: Proceedings of ACM IMC 2018, Boston, MA, 2018.

[28] A. Lodhi, A. Dhamdhere, C. Dovrolis, Open peering by internet transit providers: Peer preference or peer pressure? Proc. IEEE INFOCOM (2014) 2562–2570.

[29] P. Marcos, M. Chiesa, L. Muller, P. Kathiravelu, C. Dietzel, M. Canini, M. Barcellos, Dynam-IX: a dynamic interconnection exchange, Proc. SIGCOMM Posters Demos (2018) 2.

[30] A. Ahmed, Z. Shafiq, H. Bedi, A. Khakpour, Peering vs. transit: Performance comparison of peering and transit interconnections, in: Proceedings - International Conference on Network Protocols, ICNP, Vol. 2017-Octob, 2017.

[31] J.L. Sobrinho, D. Fialho, P. Mateus, Stabilizing BGP through distributed elimination of recurrent routing loops, in: 2017 IEEE 25th International Conference on Network Protocols (ICNP), 2017, pp. 1–10.

[32] Y. Wang, M. Schapira, J. Rexford, Neighbor-specific BGP: More flexible routing policies while improving global stability, in: Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems, in: SIGMETRICS '09, ACM, New York, NY, USA, 2009, pp. 217–228.

[33] R. Mahajan, D. Wetherall, T. Anderson, Negotiation-based routing between neighboring ISPs, in: Proceedings of the 2Nd Conference on Symposium on Networked Systems Design & Implementation - Volume 2, in: NSDI'05, USENIX Association, Berkeley, CA, USA, 2005, pp. 29–42.

[34] R. Mahajan, D. Wetherall, T. Anderson, Mutually controlled routing with independent ISPs, in: 4th USENIX Symposium on Networked Systems Design & Implementation(NSDI'07), 2007, pp. 355–368.

[35] T.G. Griffin, The stratified shortest-paths problem (invited paper), in: 2010 Second International Conference on COMmunication Systems and NETworks (COMSNETS 2010), 2010, pp. 1–10.

[36] J.L. Sobrinho, An algebraic theory of dynamic network routing, IEEE/ACM Trans. Netw. 13 (5) (2005) 1160–1173.

[37] T.G. Griffin, F.B. Shepherd, G. Wilfong, Policy disputes in path-vector protocols, in: Proceedings. Seventh International Conference on Network Protocols, 1999, pp. 21–30.

[38] T.G. Griffin, G. Wilfong, A safe path vector protocol, in: Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), Vol. 2, 2000, pp. 490–499.

[39] C.T. Ee, V. Ramachandran, B.-G. Chun, K. Lakshminarayanan, S. Shenker, Resolving inter-domain policy disputes, SIGCOMM Comput. Commun. Rev. 37 (4) (2007) 157–168.

[40] W. Xu, J. Rexford, MIRO: Multi-path interdomain routing, in: Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, in: SIGCOMM '06, Association for Computing Machinery, New York, NY, USA, 2006, pp. 171–182.

[41] L. Subramanian, V. Roth, I. Stoica, S. Shenker, R.H. Katz, Listen and whisper: Security mechanisms for BGP, in: Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation - Volume 1, in: NSDI'04, USENIX Association, USA, 2004, p. 10.

[42] J.P. John, E. Katz-Bassett, A. Krishnamurthy, T. Anderson, A. Venkataramani, Consensus routing: The Internet as a distributed system, in: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, 2008, pp. 351–364.

[43] S. Goldberg, M. Schapira, P. Hummon, J. Rexford, How secure are secure interdomain routing protocols, SIGCOMM Comput. Commun. Rev. 40 (4) (2010) 87–98.

[44] Z. Duan, X. Yuan, J. Chandrashekar, Controlling IP spoofing through interdomain packet filters, IEEE Trans. Dependable Secure Comput. 5 (1) (2008) 22–36.

[45] P. Gill, M. Schapira, S. Goldberg, Let the market drive deployment: A strategy for transitioning to bgp security, SIGCOMM Comput. Commun. Rev. 41 (4) (2011) 14–25.

[46] R. Lychev, S. Goldberg, M. Schapira, BGP security in partial deployment: Is the juice worth the squeeze? in: Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM, in: SIGCOMM '13, Association for Computing Machinery, New York, NY, USA, 2013, pp. 171–182.

[47] K. Weitz, D. Woos, E. Torlak, M.D. Ernst, A. Krishnamurthy, Z. Tatlock, Scalable verification of border gateway protocol configurations with an SMT solver, in: Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications, in: OOPSLA 2016, Association for Computing Machinery, New York, NY, USA, 2016, pp. 765–780.

[48] X. Shao, L. Gao, Verifying policy-based routing at internet scale, in: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, 2020, pp. 2293–2302.

[49] R. Klöti, V. Kotronis, B. Ager, X. Dimitropoulos, Policy-compliant path diversity and bisection bandwidth, in: 2015 IEEE Conference on Computer Communications (INFOCOM), 2015, pp. 675–683.

[50] L. Gao, On inferring autonomous system relationships in the internet, IEEE/ACM Trans. Netw. 9 (6) (2001) 733–745.

[51] L. Subramanian, S. Agarwal, J. Rexford, R.H. Katz, Characterizing the internet hierarchy from multiple vantage points, Proc. IEEE INFOCOM 2 (2002) 618–627.

[52] T. Erlebach, E. Hall, Classifying Customer-Provider Relationships in the Internet, in: Proceedings of the IASTED International Conference on Communications and Computer Networks, 145, 2002, pp. 538–545.

[53] X. Dimitropoulos, D. Krioukov, B. Huffaker, k.c. Claffy, G. Riley, Inferring AS relationships: Dead end or lively beginning? 2005, arXiv:0507047.

[54] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, k.c. Claffy, G. Riley, AS relationships: Inference and validation, SIGCOMM Comput. Commun. Rev., 37, (1) 2007, pp. 29–40,

[55] G. Di Battista, T. Erlebach, A. Hall, M. Patrignani, M. Pizzonia, T. Schank, Computing the types of the relationships between autonomous systems, IEEE/ACM Trans. Netw. 15 (2) (2007) 267–280.

[56] Y. Jin, C. Scott, A. Dhamdhere, V. Giotsas, A. Krishnamurthy, S. Shenker, Stable and practical AS relationship inference with problink, in: 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19), USENIX Association, 2019, pp. 581–598.

[57] L. Müller, M. Luckie, B. Huffaker, K. Claffy, M. Barcellos, Challenges in inferring spoofed traffic at IXPs, in: Proceedings of the 15th International Conference on Emerging Networking Experiments and Technologies, in: CoNEXT '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 96–109.

Xiaozhe Shao received BE and MS degrees in Computer Science from University of Science and Technology of China in 2009 and 2013 respectively. Since 2014 Spring, he has been working as research assistant in University of Massachusetts at Amherst, where now he is a Ph.D. candidate in the Electrical and Computer Engineering department. His current research interests include inter-domain routing, network verification and graph data anonymization.

Lixin Gao is a University Distinguished Professor of Electrical and Computer Engineering at the University of Massachusetts at Amherst. She received a Ph.D. degree in Computer Science from the University of Massachusetts at Amherst. Her research interests include online social networks, Internet routing, network virtualization and cloud computing. She was an Alfred P. Sloan Fellow between 2003–2005 and received an NSF CAREER Award in 1999. She won the best paper award from IEEE INFOCOM 2010, and the test-of-time award in ACM SIGMETRICS 2010. Her paper in ACM Cloud Computing 2011 was honored with "Paper of Distinction". She is a fellow of IEEE and ACM.