5-4-2010

Safe Inter-domain Routing under Diverse Commercial Agreements

Yong Liao
University of Massachusetts - Amherst

Lixin Gao
University of Massachusetts - Amherst

Roch A. Guérin
University of Pennsylvania, guerin@acm.org

Zhi-Li Zhang
University of Minnesota - Twin Cities

Accepted for publication in IEEE/ACM Transactions on Networking.

Safe Inter-domain Routing under Diverse Commercial Agreements

Abstract

Commercial agreements drive the routing policies used in today's Internet. The two most extensively studied commercial agreements are transit and peering; however, they are only two of many diverse and continuously evolving commercial agreements that ISPs enter into. So far, the only known practical safe and robust routing policy is Gao and Rexford's policy guideline, which is applicable to transit and peering agreements only. It is, therefore, of importance to identify routing policies that are safe and robust and at the same time capable of accommodating the diverse commercial agreements existing in the Internet. In particular, this paper investigates the extent to which routing policies can be devised to accommodate complex mutual transit agreements. We propose a series of policy guidelines that allow mutual transit agreements with progressively broader semantics to be established. Those policy guidelines guarantee routing safety and robustness as long as the AS graph satisfies a corresponding set of precise topological constraints. An experimental evaluation of the proposed policy guidelines demonstrates the benefits they would likely afford in terms of routing reliability, if adopted in the current Internet.

Keywords

Inter-domain routing, policy, safeness, BGP

Disciplines

Digital Communications and Networking | OS and Networks

Comments

Accepted for publication in IEEE/ACM Transactions on Networking.

Safe Inter-domain Routing under Diverse Commercial Agreements

Yong Liao, Lixin Gao, *Fellow, IEEE,* Roch Guerin, *Fellow, IEEE,* and Zhi-Li Zhang

***Abstract*—Commercial agreements drive the routing policies used in today's Internet. The two most extensively studied commercial agreements are *transit* and *peering*; however, they are only two of many diverse and continuously evolving commercial agreements that ISPs enter into. So far, the only known *practical* safe and robust routing policy is Gao and Rexford's policy guideline, which is applicable to transit and peering agreements only. It is, therefore, of importance to identify routing policies that are safe and robust and at the same time capable of accommodating the diverse commercial agreements existing in the Internet. In particular, this paper investigates the extent to which routing policies can be devised to accommodate complex mutual transit agreements. We propose a series of policy guidelines that allow mutual transit agreements with progressively broader semantics to be established. Those policy guidelines guarantee routing safety and robustness as long as the AS graph satisfies a corresponding set of precise topological constraints. An experimental evaluation of the proposed policy guidelines demonstrates the benefits they would likely afford in terms of routing reliability, if adopted in the current Internet.**

I. Introduction

The Internet consists of a large number of inter-connected autonomous systems (ASes). Each AS enters into certain commercial agreements with a few other ASes so as to attain global reachability across the Internet. These commercial agreements determine how and what traffic the ASes exchange and thereby dictate their inter-domain routing policies. Two typical commercial agreements are transit and peering agreements. Commercial agreements between ASes are, however, continuously evolving and commonly take many forms beyond the above two agreements. Their existence and evolution are driven by the business interests of ISPs and other players, the competitive marketplace, and the constantly changing Internet structure.

For example, one ISP may acquire or merge with another ISP. Since it is often not economically feasible to physically merge two existing networks, the relationship between the two ASes needs to be redefined: they may want to use each others' providers to reach certain destinations (i.e., the two ASes now provide transit to each other). As another example, an AS might establish a private transit agreement for a particular customer with one of its neighbors (an instance of *selective transit*), while establishing a peering agreement with that neighbor for the rest of its customers. Similarly, two physically co-located enterprise networks might establish a mutual backup agreement, where one provides transit service to the other only when the other's link to its own provider fails or is in maintenance. By entering into various forms of diverse commercial agreements, ASes can not only achieve cost savings, they can also enhance service reliability and availability to their customers. Furthermore, the economic structure of the Internet is likely to evolve in many directions [1]–[3], and this in itself will translate into a broader set of commercial agreements.

Yet, broadening the set of commercial agreements that can be accommodated in inter-domain routing is easier said than done. Commercial agreements dictate the routing policies adopted in each AS, and it is well known that the use of "arbitrary" routing policies can lead to routing oscillations [4]. So far, the only known *practical* safe and robust routing policy is Gao and Rexford's policy guideline [5], which is applicable only to transit and peering agreements, with extension to the backup agreement [6]. Arbitrary agreements, such as an AS transiting traffic between any two other ASes, have been shown to possibly cause persistent routing oscillations [7]. Clearly, some caution is in order when contemplating more general agreements.

The possible agreements between ASes can take many different forms. This paper studies routing policies that guarantee routing safety and robustness while accommodating a set of commercial agreements that offer additional diversity. We focus on the cases where two ASes are willing to provide connectivity to each other to reach the rest of the Internet, i.e., they transit traffic for each other, and therefore establish one of the so-called *mutual transit agreements* [8]. As we will see later in the paper, such mutual transit agreements cover many possible forms of complex agreements among ISPs. Some of these agreements already exist in the Internet, but how to safely accommodate them is not yet fully understood. More importantly, as the Internet's diversity continues to grow, more ASes are expected to enter into various complex agreements such as mutual transit agreements. To provide guidelines on how to handle the mutual transit agreements, we introduce routing polices that expose increasingly larger sets of paths. We show that those paths are indeed needed to accommodate the diverse mutual transit agreements. The policies are provably safe and robust, as long as the Internet AS-level topology satisfies certain constraints. We also perform a representative set of experiments to show that allowing ASes to enter into mutual transit agreements can substantially improve Internet routing resiliency to certain failures.

Yong Liao and Lixin Gao are with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01002 (email: {yliao, lgao}@ecs.umass.edu). Roch Guerin is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 (email: guerin@ee.upenn.edu). Zhi-Li Zhang is with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55416 (email: zhzhang@cs.umn.edu).

The rest of the paper is organized as follows. Section II gives some background on inter-domain routing policies, motivations for accommodating more diverse commercial agreements, and a brief overview of the paper. Section III details the admissible path sets produced by mutual transit agreements. Section IV specifies how to rank those paths to avoid policy disputes. Section V presents the routing policies considered in the paper and formally establishes their safety and robustness properties. The practical implications of the proposed routing policies are discussed in section VI. Section VII presents experiments aimed at evaluating the potential fault-tolerance benefits when some ASes extend the agreements they engage into to include mutual transit agreements. Section VIII concludes the paper.

II. Background, Motivation and Overview

In this section, we first provide some background on inter-domain routing policies and how they relate to routing safety and robustness. We then discuss AS business relations (or commercial agreements) that dictate routing policies, and outline the Gao-Rexford policy guideline. We argue that in practice there exist more diverse and complex commercial agreements, but how to safely accommodate those agreements is not yet clear. Therefore, studying this problem is both valuable in theory and needed in practice.

A. Routing Policies, Routing Safety and Robustness

In essence, routing policies specify two things: (i) the paths that are exposed or announced to neighbors, via *export* policies, and (ii) preferences or ranking of the paths learned from neighbors, via *import* policies. It is well known that without any restriction on policies, so-called "policy disputes" may arise and lead to routing oscillation [9,10]. To avoid such a situation, certain limitations must be applied to routing policies. Griffin *et al.* introduce the notions of *routing safety and robustness* [4,10]. Informally, a set of routing policies are said to be *safe* if the resulting routing system always converges to a unique stable state. Such routing policies are *robust* if they are safe under any topology changes (e.g., link failures). Furthermore, a sufficient condition for routing safety and robustness is identified in [10]: if a set of routing policies do not lead to a *dispute wheel*, they are safe and robust (see Appendix A for the definition of dispute wheel). The problem of safety and robustness in policy routing is further investigated in [7]. The authors show that if ASes are allowed to arbitrarily filter their routes, a safe and robust routing has to constrain the path ranking to be selecting the path with the shortest weighted path length.

The *safe path vector protocol* is proposed in [11], which includes a mechanism to dynamically detect oscillations induced by policy disputes. This is further extended in [12], which resolves routing oscillations by letting an AS select a less preferred but more stable route when that AS detects that it is itself involved in a policy dispute. Jaggard *et al.* study the routing safeness problem in class based path vector systems in [13]. Sobrinho studies the convergence of path vector routing protocol using the *routing algebra* framework in [14,15]. Based on the routing algebra framework, a meta routing language is proposed in [16], which can be used to describe and construct safe routing protocols.

B. Practical Routing Policy Guidelines Accommodating Transit and Peering Agreements

In practice, the routing policies adopted by ASes are often dictated by the commercial agreements they have with other ASes and their own business interests. The most common agreements are *transit* where the provider AS provides service to the customer AS in connecting to the Internet, and *peering* where two ASes agree to swap traffic between their respective customers without monetary settlement [17]. Taking these two common business relations into account, Gao and Rexford present the *prefer customer* and *no valley path* policy guideline, which guarantees routing safety and robustness if the AS topology does not contain any provider-customer cycle [5]. The "prefer customer" guideline constrains the configuration of import policies to assign higher preference to paths learned from customers than to paths learned from peers and providers[1]. The "no valley path" guideline specifies that the export policies of ASes should not allow *valleys* to appear in any AS paths. A *valley path* arises when an AS announces a path learned from a peer or provider to another peer or provider. The AS graph topological constraint needed to ensure the safety and robustness of the Gao-Rexford policy guideline is fairly mild, because an AS usually chooses other ASes of bigger size or coverage than itself as its providers [5][2].

C. Diverse Commercial Agreements

As just alluded to, while *transit* and *peering* agreements are the most common ones, far more diverse and complex commercial agreements exist in practice. A well-known and easy to understand example is the *sibling* relation [8,17], where two ASes provide transit service to each other. This relation could be established because: an ISP owns two ASes in two geographical regions, or an AS merges with or acquires another AS. At first glance, it would seem that a sibling relation could be treated as two separate "provider-customer" relations, to which the Gao-Rexford policy guideline could be applied. Such a treatment, however, would lead to a major technical problem: it violates the mild topological constraint under which the Gao-Rexford policy guideline is proved to be safe and robust. We use a realistic example in Fig. 1 to illustrate the potential issues. In the middle of 2007, Tiscali (AS3257) acquired Pipex Broadband (AS5413) [18]. Both Tiscali and Pipex bought their transit service from TeliaSonera (AS1299), which is a tier-1 ISP [19]. Before their merging, Tiscali and Pipex used TeliaSonera to reach some destination prefix p. However, if they treat each other as customers, Tiscali would prefer Pipex's route to p and Pipex would prefer

[1] The actual policies applied in reality could be quite complicated. There are cases where some large ISP prefers peer paths over customer paths for certain destinations.

[2] The size of an AS could be quantified by its traffic volume or degree in the AS graph. The coverage of an AS is usually the geographical area that AS covers.

Tiscali's route too. This is basically a DISAGREE scenario described in [10]. Routing oscillation may occur because no unique stable state exists in a DISAGREE scenario. As there is no systematic guideline for handling sibling relation yet, when two ASes merge, they usually have to treat each other as peers. This is a conservative treatment that under-utilizes the connections between them, as they only use those connections to reach each other's customers.

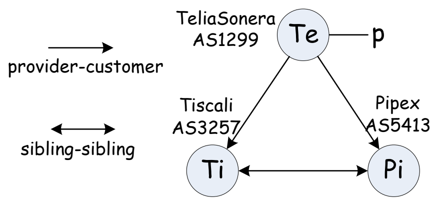


Fig. 1. Example of sibling relation established between merging ASes.

Besides the sibling relation, another example of diverse agreements is two peering ASes with special agreements for certain destinations, where they provide transit to each other but only for those destinations. For other destinations, they exchange customer traffic as per the standard peering agreement.

Except for the backup agreement studied in [6], it has *until now* not been clear what practical policy guidelines are needed to accommodate more diverse commercial agreements, e.g., the sibling relation, the case of peering relation with special mutual transit arrangement, and so forth, while ensuring the safety and robustness of the global inter-domain routing system. In practice, ASes or ISPs commonly use a few local tweaks to better meet their own business interests, with little concern or respect for the safety and robustness of the global routing system. Hence, it is important to understand how one can accommodate more diverse agreements in a safe and robust manner. Our paper is devoted to this problem.

D. Accommodating Mutual Transit Agreements: An Overview

We focus primarily on how to safely accommodate a family of what we term mutual transit agreements. In general, *a mutual transit agreement between two ASes means that they are willing to provide each other with connectivity to reach the rest of the Internet* [8]. For example, the sibling relation discussed above is one type of mutual transit agreement. In practice, mutual transit agreements can have a wide-range of semantics regarding what paths the ASes entering into those agreements expose to each other. We first study the mutual transit agreement where two ASes expose to each other their provider, customer, and peer paths, which is most likely what happens in the current Internet when two ASes are merging. Next, we expand the semantic of mutual transit, so that an AS can also announce certain paths learned from its own mutual transit neighbors to other neighbors with which it has mutual transit agreements. Finally, we consider the most general form of mutual transit, i.e., two ASes entering into an agreement where they announce all their paths to each other.

In section III, we study what type of paths should be exposed to support the various mutual transit agreements we have just identified. How to setup the preference of those admissible paths to avoid potential policy disputes is discussed in section IV. In section V, we present a series of policy guidelines that allow progressively larger sets of admissible paths, and can therefore, accommodate mutual transit agreements with progressively broader meanings. We show that those guidelines can be provably safe and robust.

In the rest of the paper, we say that two ASes have an *MTran agreement* or they are *MTran neighbors*, if they have entered into a mutual transit agreement. The link between two MTran neighbors is called an *MTran link*. The routes learned from an MTran neighbor are referred to as *MTran routes* or *MTran paths*.

III. Admissible Paths for Accommodating Mutual Transit Agreement

In this section, we first introduce an abstract AS graph model that captures the complex nature of mutual transit agreements. Next, we introduce the concept of admissible path set. The admissible paths essentially specify the export policy of the policy guidelines required to make mutual transit agreements safe.

A. AS Graph Model

We model the Internet AS-level topology as a graph $G = (V, E)$, where the nodes are ASes and edges represent agreements between ASes. An edge in G can be undirected, directed, or bi-directed. An undirected edge $(u{-}v)$ indicates a peering agreement between u and v; a directed edge $(u{\rightarrow}v)$ represents a transit agreement where u is the provider of v; and a bi-directed edge $(u{\leftrightarrow}v)$ represents a mutual transit agreement between u and v. Let $\overline{E}$ denote the set of undirected edges, $\overrightarrow{E}$ the set of directed edges, and $\overleftrightarrow{E}$ the set of bi-directed edges. Obviously, $E = \overline{E} \cup \overrightarrow{E} \cup \overleftrightarrow{E}$.

B. AS Paths, Steps, and AS Paths with Steps

A *path* P in graph $G = (V, E)$ is an ordered sequence of distinct nodes, i.e., $P = u_0 u_1 \dots u_m$, where $u_i \neq u_j, \forall i \neq j$. If m=0, we say P is a *trivial* path; otherwise P is a *non-trivial* path. P is a *downhill path* if P is a trivial path; or all edges in P are directed edges and any node (except the first one) is a customer of its previous node in P. That is, P is a downhill path if m=0; or $(u_i{\rightarrow}u_{i+1}) \in \overrightarrow{E}$, $\forall i \in [0, m-1]$. P is an *uphill path* if all edges in P are directed edges and any node (except the first one) is a provider of its previous node. That is, P is an uphill path if $(u_{i+1}{\rightarrow}u_i) \in \overrightarrow{E}$, $\forall i \in [0, m-1]$.

We say that P is a *step* if all edges in P are bi-directed edges, i.e., $(u_i{\leftrightarrow}u_{i+1}) \in \overleftrightarrow{E}$, $\forall i \in [0, m-1]$. In particular, step P is referred to as a k-step if it contains k bi-directed edges. We also refer to k as the *step width* of a k-step.

Path P is referred to as a *downhill path with steps* if no segment of P is an uphill path and it contains at least one bi-directed edge, i.e., $\nexists i \in [0, m-1]$, $(u_{i+1}{\rightarrow}u_i) \in \overrightarrow{E}$ and $\exists j \in [0, m-1]$, $(u_j{\leftrightarrow}u_{j+1}) \in \overleftrightarrow{E}$.[3] P is referred to as an *uphill*

[3]Note that a path with only bi-directed edges is a downhill path with steps.

path with steps if no segment of P is a non-trivial downhill path, and P has at least one directed edge and one bi-directed edge. That is, P is an uphill path with steps if $\nexists f \in [0, m-1]$, $(u_f \rightarrow u_{f+1}) \in \overrightarrow{E}$, and $\exists i, j \in [0, m-1]$, $(u_{i+1} \rightarrow u_i) \in \overrightarrow{E}$, $(u_j \leftrightarrow u_{j+1}) \in \overleftrightarrow{E}$.

When P is a downhill path with steps and the widest step in P is a k-step, P is referred to as a *downhill path with k-steps*. Uphill path with k-steps can be similarly defined. See Fig. 2 for an illustration of uphill/downhill paths (with steps).

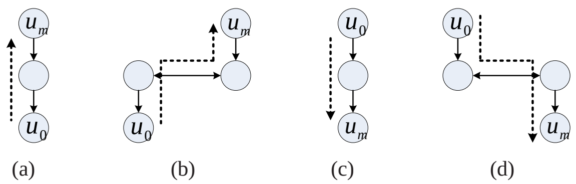


Fig. 2. Examples of uphill/downhill paths (with and without steps). The solid arrows represent AS relationships. The dashed arrows represent AS paths. (a) is an uphill path; (b) is an uphill path with step; (c) is a downhill path; (d) is a downhill paths with step.

C. Admissible Path Set

Next we illustrate the kind of paths that should be permitted to accommodate the mutual transit agreements.

1) Not allowing valley paths: In general, no *valley paths* should be allowed. Allowing valley paths essentially asks ASes to transit traffic for their providers. Given that customers must pay their providers for all traffic going to or coming from themselves, such a practice does not make economic sense. The "valley paths" considered in this paper have a broader meaning than those in the Gao-Rexford policy guideline due to the introduction of mutual transit agreements. We say a path P has a *valley* if P contains a *downhill* segment (with or without steps) followed by an *uphill* segment (with or without steps); or it contains a *downhill* segment (with or without steps), followed by an undirected edge, maybe then an *uphill* segment (with or without steps). A path that contains a valley is a *valley path*. Fig. 3 shows several examples of valley paths.

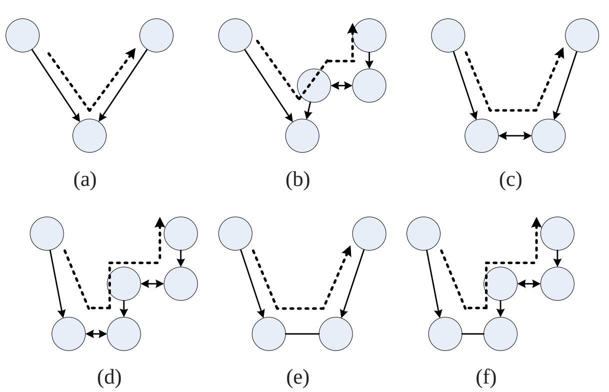


Fig. 3. Examples of valley paths. In (a) and (b), an AS transits traffic for its two providers; in (c) and (d), ASes with mutual transit agreements transit traffic for their providers; in (e) and (f), two peering ASes transit traffic for their providers.

2) Allowing valley-free paths with steps: It is necessary to permit valley-free paths with steps in order to accommodate mutual transit agreements. When two MTran neighbors, ASes u and v, announce to each other their provider routes, customer routes, and peer routes, the result is that all valley-free AS paths including u and v have at least a 1-step, i.e., edge $(u \leftrightarrow v)$. Further, if u and v have mutual transit agreements with other ASes and they also announce the routes learned from those ASes to each other, we will see valley-free paths including steps wider than one. In general, we define the *set of admissible paths* $\mathcal{P}_k$ in Definition III.1, which includes all valley-free paths with steps not wider than some number k. Fig. 4 provides some examples of valley-free paths in $\mathcal{P}_1$.

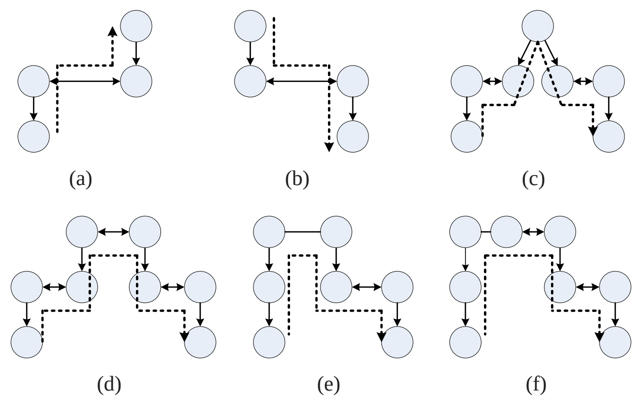


Fig. 4. Example paths in set $\mathcal{P}_1$. The dashed arrows represent AS paths.

Definition III.1 ($\mathcal{P}_k$) *The set of admissible paths, $\mathcal{P}_k$, includes:* **(i)** *uphill paths with steps of width at most k,* **(ii)** *downhill paths with steps of width at most k,* **(iii)** *paths consisting of an uphill segment followed by a downhill segment and with no steps wider than k,* **(iv)** *paths consisting of an uphill segment followed first by an undirected edge, and next by a downhill segment, and with no steps wider than k.*

Clearly, $\mathcal{P}_{k+1} \supset \mathcal{P}_k$, and in particular, $\mathcal{P}_k \supset \mathcal{P}_0$, where $\mathcal{P}_0$ is the collection of admissible paths under the Gao-Rexford policy guideline, which covers only the transit and peering agreements. As mentioned, an AS path with only bi-directed edges is a downhill path with steps, therefore, an m-step path where $m \leq k$, is an admissible path in $\mathcal{P}_k$.

Here we provide some motivations for our definition of admissible path sets $\mathcal{P}_k$. First, by allowing valley-free paths with 1-step, i.e., those paths in $\mathcal{P}_1 \cap \overline{\mathcal{P}_0}$ ($\overline{\mathcal{P}_0}$ is the complement of $\mathcal{P}_0$), two ASes can establish a mutual transit agreement where they announce to each other all paths except the paths learned from other MTran neighbors. If two ASes have a mutual transit agreement where they also announce to each other certain paths learned from other MTran neighbors, it is necessary to expand the admissible path set to $\mathcal{P}_k$ where $k > 1$. Further, if two MTran neighbors announce to each other all their paths, the admissible path set should be $\mathcal{P}_\infty$.

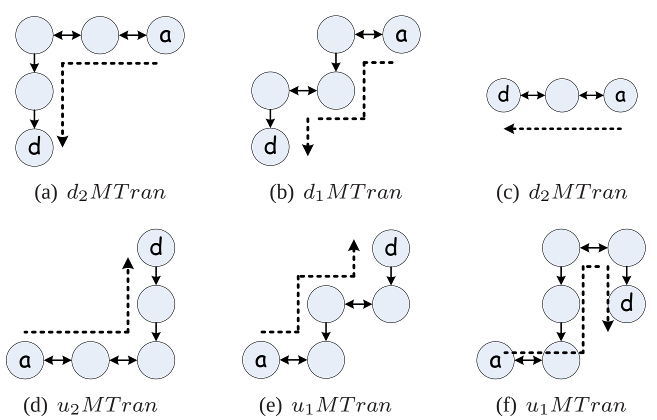
IV. Classes of Paths and Ranking of the Paths

We have seen that the mutual transit agreements give rise to admissible path sets including valley-free paths with steps. The next natural question would be how to rank these paths so as to setup their preferences. Appropriate path ranking is important, otherwise "*policy disputes*" may arise. In this section, we first classify paths in the admissible path sets, and then we study how to rank the paths based on their classes.

A. Classes of Paths in the Admissible Path Set

In set $\mathcal{P}_k$, we still have provider paths, customer paths, and peer paths, which come from the transit and peering agreements. If AS a_0 learns path P from a provider (resp., customer, peer) and $P \in \mathcal{P}_k$, we say P is a provider (resp., customer, peer) path of a_0. Besides those three types of paths, in set $\mathcal{P}_k$ where $k>0$, there are also paths learned from mutual transit neighbors.

For two MTran neighbors a_0 and a_1, we further distinguish the paths that a_1 exports to a_0 into those going downhill and those going uphill in the AS hierarchy. Given an AS graph $G = (V, E)$, a path $P = a_0 a_1 ... a_m Q$ ($m \geq 1$) learned by a_0 from its MTran neighbor a_1 is called a $d_m MTran$ path if $(a_i \leftrightarrow a_{i+1}) \in \overleftrightarrow{E}$, $\forall i \in [0, m\text{-}1]$ and Q is a customer path of a_m. In other words, a $d_m MTran$ path has an m-step at the *beginning*, which is followed by a segment going downhill in the AS hierarchy. Likewise, we say P is a $u_m MTran$ path of a_0 if Q is a provider path or peer path of a_m, i.e., Q is a segment going uphill in the AS hierarchy (may be followed by a downhill segment). When the context is clear, we sometimes drop the index m, and use the terms $dMTran$ and $uMTran$ paths to refer to any $d_m MTran$ and $u_m MTran$ paths in $\mathcal{P}_k$ ($m \leq k$), respectively. Note that a route to a prefix owned by the AS itself is considered to be a customer route of that AS, so a path consisting of only bi-directed edges is a $dMTran$ path, i.e., P is a $dMTran$ path if $Q = null$. Fig. 5 shows some examples of $dMTran$ and $uMTran$ paths.



(a) d_2MTran (b) d_1MTran (c) d_2MTran

(d) u_2MTran (e) u_1MTran (f) u_1MTran

Fig. 5. Examples of $dMTran$ paths and $uMTran$ paths. The dashed arrows represent AS paths. AS a in (a) has a d_2MTran path to AS d. The path in (b) is a d_1MTran path because it has one MTran link in the beginning. Fig. (c) shows a path with only MTran links and it is a d_2MTran path. Fig. (d) and (e) depict examples of u_2MTran path and u_1MTran path, respectively. An $uMTran$ path can have a downhill segment, as Fig. (f) shows.

Having classified paths in $\mathcal{P}_k$ into provider, customer, peer, $dMTran$, and $uMTran$ paths, next we proceed to rank them. As in the Gao-Rexford policy guideline, we prefer customer paths over peer paths and provider paths; no preference is needed between peer and provider paths. The remaining unspecified cases are how to rank between MTran paths and other types of paths, and how to rank MTran paths among themselves. Section IV-B considers ranking $dMTran$ paths, while section IV-C studies the ranking of $uMTran$ paths. Section IV-D summarizes the ranking rules.

B. Ranking $dMTran$ Paths

In discussing each ranking rule, we use an example to show that a dispute wheel will arise if the ranking does not follow the rule. Dispute wheel related terms, such as *pivot node*, *spoke path*, and *rim path*, will be used in the discussion. Their definitions can be found in Appendix A.

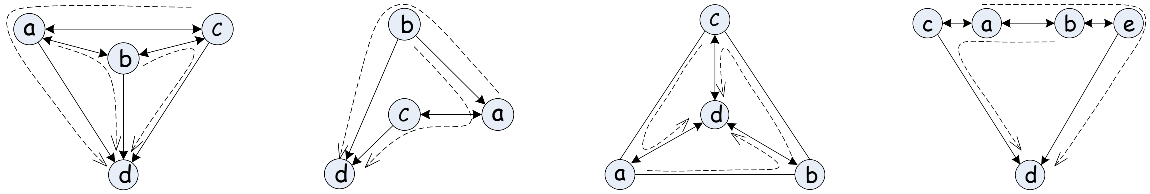
1) Customer path and $dMTran$ path: We use the example of Fig. 6(a) to show that a customer path should be preferred over a $dMTran$ path to avoid policy disputes. ASes a, b, and c in Fig. 6(a) are MTran neighbors and d is their customer. ASes a, b, and c have direct customer paths to d and they announce their customer paths to each other, so that they also have $dMTran$ paths to d. If $dMTran$ paths are preferred over customer paths, Fig. 6(a) has a dispute wheel. That is, a, b, and c are the pivot nodes; their customer paths are the spoke paths; and their $dMTran$ paths are the rim paths. Preferring customer path over $dMTran$ path breaks the dispute wheel, because the pivot nodes will prefer their spoke paths over rim paths.

Preferring customer paths over $dMTran$ paths not only solves the potential routing oscillation, it also makes economic sense. Because customers always pay for the traffic transited by their providers, customer paths should always be preferred.

2) Provider path and $dMTran$ path: Next we study how to rank between provider paths and $dMTran$ paths. In Fig. 6(b), ASes a and c are MTran neighbors; b is the provider of a and d; c is a provider of d. AS b has two customer paths to d, one is the direct path and the other is via a. AS a learns a provider path from b and a $dMTran$ path from c. If b prefers the customer path via a and a prefers its provider path over its $dMTran$ path, there is a dispute wheel. That is, the pivot nodes are a and b; the spoke paths are a:c:d and b:d; and the rim paths are a:b:d and b:a:c:d. The policy dispute in Fig. 6(b) can be resolved if a prefers its spoke path a:c:d over its rim path a:b:d. Hence, we should prefer $dMTran$ paths over provider paths.

There is also an economic justification for this ranking rule. Sending traffic to providers always increases one's cost. However, using $dMTran$ path will not cost more, because two MTran neighbors usually do not charge each other (e.g., two merging ASes). Besides, preferring $dMTran$ path over provider path can benefit the MTran neighbor, because it will send the traffic to a customer and charge that customer.

3) Peer path and $dMTran$ path: $dMTran$ paths should be preferred over peer paths; otherwise a dispute wheel as shown in Fig. 6(c) can occur. Here a, b, and c are peers and they are MTran neighbors of d. ASes a, b, and c learn their $dMTran$ paths from d; they also have peer paths to d once they announce their $dMTran$ paths to each other. If peer paths are preferred over $dMTran$ paths, Fig. 6(c) has a dispute



(a) customer&$dMTran$ (b) provider&$dMTran$ (c) peer&$dMTran$ (d) $dMTran$&$dMTran$

Fig. 6. Examples showing the potential policy disputes when $dMTran$ paths are not properly ranked. (a) shows a policy dispute if $dMTran$ paths are preferred over customer paths; (b) shows a policy dispute if provider paths are preferred over $dMTran$ paths; (c) shows a policy dispute if peer paths are preferred over $dMTran$ paths; (d) shows a policy dispute if d_1MTran paths are preferred over d_2MTran paths. The dashed arrows are the preferred paths to destination d in those policy disputes. The policy disputes in (a) and (c) are examples of the BADGADGET scenario discussed in [10]; (b) and (d) are DISAGREE scenarios [10].

wheel. That is, a, b, and c are the pivot nodes; their $dMTran$ paths are the spoke paths; and their peer paths are the rim paths. This dispute can be resolved by preferring $dMTran$ paths over peer paths.

Again, such a ranking makes economic sense: Two ASes having a mutual transit agreement usually belong to the same ISP (such as merging ASes). Since a $dMTran$ path goes through a customer of the MTran neighbor, sending the traffic through an MTran neighbor will benefit that neighbor, as its customers always pay.

4) Between $dMTran$ paths: Given a d_iMTran path and a d_jMTran path, if $i<j$, we should prefer the d_iMTran path over the d_jMTran path. In other words, the $dMTran$ path with less MTran links at its beginning should be preferred. Violating this ranking rule would result in policy disputes like the one in Fig. 6(d). Here d is a customer of c and e. AS a and AS b have d_1MTran paths a:c:d and b:e:d to d, respectively. ASes a and b announce their d_1MTran paths to each other so that they also have d_2MTran paths to d. If d_2MTran paths are preferred over d_1MTran paths, there is a policy dispute between a and b.

It also makes sense economically to prefer the $dMTran$ path with less steps at its beginning. As the traffic will eventually be sent to some AS that is not an MTran neighbor, it is better to shift the traffic "off-the-net" as soon as possible.

C. Ranking $uMTran$ Paths

Similar to the discussions in section IV-B, in this section we also use examples to illustrate the ranking rules needed to avoid policy disputes.

1) Customer path and $uMTran$ path: We use Fig. 7(a) to show that customer paths should be preferred over $uMTran$ paths to avoid policy disputes. In Fig. 7(a), a and b are MTran neighbors; c is a provider of b and d; a is also a provider of d. AS b has a provider path and a $dMTran$ path to d. AS a has a direct customer path and a $uMTran$ path to d. We already know that b prefers its $dMTran$ path b:a:d to d. If a prefers its $uMTran$ path over its customer path, there is a dispute wheel in Fig. 7(a). That is, the pivot nodes are a and b; the spoke paths are a:d and b:c:d; and the rim paths are a:b:c:d and b:a:d. Hence, we should prefer customer paths over $uMTran$ paths.

2) Provider path and $uMTran$ path: Between provider paths and $uMTran$ paths, provider paths should be preferred; otherwise the network of Fig. 7(b) will have a dispute wheel. In Fig. 7(b), a, b, and c are MTran neighbors and they are customers of d. ASes a, b, and c have both direct provider paths and $uMTran$ paths to destination d. If $uMTran$ paths are preferred, there is a dispute wheel in Fig. 7(b), where a, b, and c are the pivot nodes; their direct provider paths are the spoke paths; and their $uMTran$ paths are the rim paths.

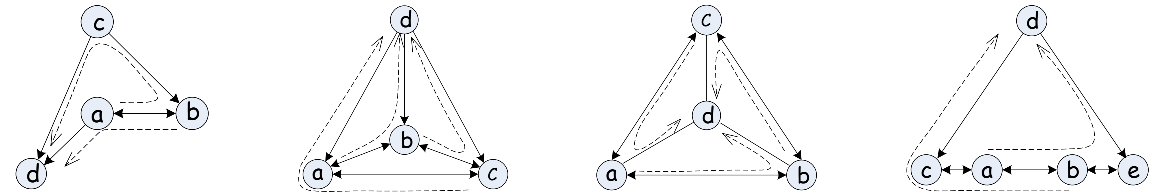
Preferring provider paths over $uMTran$ paths also has economic justifications. Consider the case where an AS has both a provider path and a $uMTran$ path, the latter one goes through a provider of an MTran neighbor. If the two ASes belong to a single (merged) ISP, it is better to shift the traffic "off-the-net" as soon as possible, rather than carrying it "on-the-net" between the two ASes, as eventually the ISP needs to pay a provider to transit the traffic. Even if the two ASes are separately owned MTran neighbors, using $uMTran$ paths instead of provider paths would not benefit either of them, because one of them must pay a provider to transit the traffic.

3) Peer path and $uMTran$ path: We use Fig. 7(c) to show that peer paths should be preferred over $uMTran$ paths to avoid potential policy disputes. In Fig. 7(c), a, b, and c are MTran neighbors and they have d as a peer. Hence, a, b, and c have both peer paths and $uMTran$ paths to d. If $uMTran$ paths are preferred over peer paths, Fig. 7(c) has a dispute wheel, i.e., a, b, and c are the pivot nodes, their peer paths are the spoke paths, and their $uMTran$ paths are the rim paths. Preferring peer paths over $uMTran$ paths breaks this dispute wheel because the pivot nodes will use their spoke paths.

4) Between $uMTran$ paths: For two $uMTran$ paths, the one prefixed by fewer MTran links should be preferred to avoid the policy dispute of Fig. 7(d). Fig. 7(d) is similar to Fig. 6(d) except that destination d is a provider of c and e. If a and b prefer their u_2MTran paths over their u_1MTran paths, there is a policy dispute between a and b. To avoid such a policy dispute, we should prefer u_iMTran paths over u_jMTran paths if $i<j$.

D. Summary of Path Ranking Rules

Based on the above discussions, our path ranking rules can be uniquely determined. Let $P_1 \succ P_2$ denote preferring path P_1 over P_2. We have $customer \succ dMTran \succ provider \succ uMTran$, and $customer \succ dMTran \succ peer \succ uMTran$; between multiple $dMTran$ paths, the one prefixed by the least number of MTran links should be preferred; between



(a) customer&$uMTran$ (b) provider&$uMTran$ (c) peer&$uMTran$ (d) $uMTran$&$uMTran$

Fig. 7. Examples showing the potential policy disputes when $uMTran$ paths are not properly ranked. (a) shows a policy dispute if $uMTran$ paths are preferred over customer paths; (b) shows a policy dispute if $uMTran$ paths are preferred over provider paths; (c) shows a policy dispute if $uMTran$ paths are preferred over peer paths; (d) shows a policy dispute if u_2MTran paths are preferred over u_1MTran paths. The dashed arrows are the preferred paths to destination d in those policy disputes. The policy disputes in (a) and (d) are DISAGREE scenarios [10]; (b) and (c) are BADGADGET scenarios [10].

multiple $uMTran$ paths, the one prefixed by the least number of MTran links should be preferred.

V. Policy Guidelines for Accommodating Mutual Transit Agreements

We are now in a position to formally and completely specify the generalized policy guidelines needed to accommodate a range of mutual transit agreements. The safety and robustness properties of those guidelines will also be formally established.

A. Policy Guidelines

We present three instances of policy guidelines, which accommodate mutual transit agreements with progressively broader meanings. Policy V.1 accommodates the agreement where two MTran neighbors announce to each other their provider, customer, and peer paths. Policy V.2 further allows certain MTran paths to be announced to MTran neighbors. Finally, Policy V.3 accommodates the mutual transit agreement where two MTran neighbors can announce any paths to each other.

Policy V.1 (1-step policy)

Export Policy
• To Customer: *announce all routes*
• To Peer: *announce customer and* d_1MTran *routes*
• To MTran: *announce customer, peer, and provider routes*
• To Provider: *announce customer and* d_1MTran *routes*
Import Policy
• *customer* $\succ d_1MTran \succ$ *provider* $\succ u_1MTran$
• *customer* $\succ d_1MTran \succ$ *peer* $\succ u_1MTran$

1) 1-step policy: Policy V.1, denoted as the *1-step* policy, accommodates a basic mutual transit agreement where two MTran neighbors announce to each other all their paths except MTran paths. Because MTran paths are not announced to MTran neighbors, consecutive MTran links will not appear in any AS paths. If this policy is adopted, the valid AS paths include all valley-free paths and valley-free paths with 1-steps. In other words, the admissible path set of Policy V.1 is $\mathcal{P}_1$.

We believe that the valley-free paths with steps allowed by the *1-step* policy are most likely what are used in practice by some ISPs today. Since an AS usually has only one MTran neighbor, no consecutive bi-directed edges will appear in any AS paths.

2) k-step policy: For a fixed $k{>}1$, Policy V.2 further extends the admissible path set to $\mathcal{P}_k$, i.e., any valley-free paths with steps not wider than k. We call Policy V.2 the *k-step* policy. The *k-step* policy allows an AS to announce certain MTran paths to its MTran neighbors, i.e., announcing those paths prefixed by less than k MTran links to MTran neighbors.

Policy V.2 (k-step policy)

Export Policy
• To Customer: *announce all routes*
• To Peer: *announce customer and* d_iMTran *routes* $\forall i \leq k$
• To MTran: *announce customer and provider routes; announce* d_iMTran *and* u_iMTran *routes* $\forall i < k$
• To Provider: *announce customer and* d_iMTran *routes* $\forall i \leq k$
Import Policy
• *customer* $\succ d_iMTran \succ d_jMTran$ $(\forall j > i) \succ$ *provider* $\succ u_iMTran \succ u_jMTran$ $(\forall j > i)$
• *customer* $\succ d_iMTran \succ d_jMTran$ $(\forall j > i) \succ$ *peer* $\succ u_iMTran \succ u_jMTran$ $(\forall j > i)$

3) any-step policy: Lastly, Policy V.3, named the *any-step* policy, allows valley-free paths with steps of any width. In other words, the admissible path set is $\mathcal{P}_\infty$. In a sense, Policy V.3 allows announcing the maximal set of paths in accommodating mutual transit agreements, i.e., it allows any paths to be announced to any MTran neighbors.

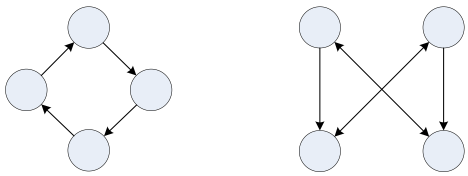
Policy V.3 (any-step policy)

Export Policy
• To Customer: *announce all routes*
• To Peer: *announce customer and* $dMTran$ *routes*
• To MTran: *announce all routes*
• To Provider: *announce customer and* $dMTran$ *routes*
Import Policy
• *customer* $\succ d_iMTran \succ d_jMTran$ $(\forall j > i) \succ$ *provider* $\succ u_iMTran \succ u_jMTran$ $(\forall j > i)$
• *customer* $\succ d_iMTran \succ d_jMTran$ $(\forall j > i) \succ$ *peer* $\succ u_iMTran \succ u_jMTran$ $(\forall j > i)$

B. Safety and Robustness of the Policy Guidelines

The safety and robustness of the policy guidelines presented in section V-A can be guaranteed when AS graph G has certain topological properties. Remember that the Gao-Rexford policy guideline guarantees routing safety and robustness when AS graph G is acyclic, i.e., the directed edges in graph G do not form any cycles. When ASes enter into mutual transit agreements so that bi-directed edges are present in AS graph G, we need to re-establish the topological properties that guarantee routing safety and robustness.

We say that an ordered sequence of nodes, $C = u_0 \dots u_{m+1}$ where $m > 1$ and $u_{m+1} = u_0$, is a *cycle with steps* if all directed edges in C point in the same direction, and C has at least one directed edge and one bi-directed edge. Further, if the widest step in C is a k-step, C is referred to as a *cycle with* k*-steps*, or an s_kCycle. For example, we refer to a directed cycle (without steps) as an s_0Cycle. Fig. 8 shows examples of s_0Cycle and s_1Cycle.



(a) s_0Cycle (b) s_1Cycle

Fig. 8. Examples of s_0Cycle and s_1Cycle.

To capture the AS graph topological properties that will guarantee the safety and robustness of our policy guidelines, we introduce the definition of AS graph family $\mathcal{ASG}_k$ as follows.

Definition V.1 ($\mathcal{ASG}_k$) *An graph G is s_kCycle-free if it contains no $s_hCycles$, where $0 \le h \le k$. The collection of all s_kCycle-free graphs is denoted as $\mathcal{ASG}_k$.*

Note that there may be an s_hCycle $(h > k)$ in $G \in \mathcal{ASG}_k$. Hence, we have $\mathcal{ASG}_{k+1} \subset \mathcal{ASG}_k$. In particular, $\mathcal{ASG}_0$ is the family of acyclic AS graphs, which have no cycle in the provider-customer relationships. The Gao-Rexford policy guideline is safe and robust for $G \in \mathcal{ASG}_0$.

The *k-step* policy guarantees routing safety and robustness as long as AS graph G has no $s_kCycles$, i.e., $G \in \mathcal{ASG}_k$, as stated in Theorem V.1.

Theorem V.1 *For any AS graph $G \in \mathcal{ASG}_k$, the k-step policy is safe and robust.*

One intuitive but rather informal way to understand Theorem V.1 is as follows. If the AS graph $G \in \mathcal{ASG}_0$, i.e., provider-customer relationships in G do not have any cycles, Theorem V.1 essentially restates that the Gao-Rexford policy is safe and robust. With the presence of mutual transit agreements in AS graph G, we can consider that a provider-customer relationship indicates two ASes in different “tiers” of G and a mutual transit relationship indicates two ASes in the same “tier”. Hence, if AS graph $G \in \mathcal{ASG}_k$ for $k > 0$, G is still hierarchical and the *k-step* policy guarantees routing safety and robustness. To formally prove Theorem V.1, we first introduce Lemma V.2.

Lemma V.2 *For any AS graph $G \in \mathcal{ASG}_k$, if there is a dispute wheel $W = (\mathcal{U}, \mathcal{Q}, \mathcal{R})$ by adopting the* k-step *policy, the rim of W cannot have only MTran links.*

Proof: For an AS graph $G \in \mathcal{ASG}_k$ where the *k-step* policy is adopted, we first assume that a dispute wheel $W = (\mathcal{U}, \mathcal{Q}, \mathcal{R})$ of size m exists, where R_i has only MTran links, $\forall i \in [0, m-1]$. Obviously, because u_i prefers R_iQ_{i+1} over Q_i, $\forall i \in [0, m-1]$, Q_i cannot be a customer route of u_i; Q_i cannot be a provider route or peer route of u_i either. Therefore, $\forall i \in [0, m-1]$, Q_i must be an MTran path of u_i. Besides, all Q_is are $uMTran$ routes of u_i, or all Q_is are $dMTran$ routes of u_i.

Case 1: If $\forall i \in [0, m-1]$, Q_i is u_i's $uMTran$ path, let $\mathcal{H}(R)$ be the step width at the beginning of path R, we have

$$\begin{cases} \mathcal{H}(R_0) + \mathcal{H}(Q_1) \le \mathcal{H}(Q_0) \\ \mathcal{H}(R_1) + \mathcal{H}(Q_2) \le \mathcal{H}(Q_1) \\ \dots \\ \mathcal{H}(R_{k-1}) + \mathcal{H}(Q_0) \le \mathcal{H}(Q_{k-1}) \end{cases}$$

From the above inequations, we can have $\sum_{i=0}^{k-1} \mathcal{H}(R_i) \le 0$, which is impossible because $\min(\mathcal{H}(R_i)) = 1$.

Case 2: If $\forall i \in [0, m-1]$, Q_i is a $dMTran$ route of u_i, we can similarly derive a contradiction.

Hence, the rim of W cannot have only MTran links. ■

With Lemma V.2, we further prove that if the *k-step* policy is adopted and there is a dispute wheel W, the rim of W must be an s_hCycle where $h \le k$.

Lemma V.3 *If a dispute wheel $W = (\mathcal{U}, \mathcal{Q}, \mathcal{R})$ exists in a routing system adopting the* k-step *policy, the rim of W must be an s_hCycle where $h \le k$.*

Proof: Without loss of generality, we first consider the case where Q_0 is a customer route of u_0.

If Q_0 is u_0's customer path, R_0Q_1 must be a customer route of u_0 too. Hence, R_0 is a downhill path from u_0 to u_1. Because no valley is allowed, Q_1 is a customer path or a $dMTran$ path of u_1. For either case, R_1Q_2 must be either a customer path or a $dMTran$ path of u_1, so that u_1 can prefer R_1Q_2 over Q_1. Therefore, R_1 is a downhill path from u_1 to u_2. By repeating this, we have $R_0R_1...R_{m-1}$ is a downhill path from u_0 to itself. According to Lemma V.2, the rim of W cannot be all MTran links, so it is an $sCycle$.

Next we show that $R_0R_1...R_{m-1}$ cannot have a segment with more than k consecutive MTran links. Assuming the rim of W has such a segment, it must be located at the concatenation point of R_i and $R_{(i+1)\%k}$. Let $\mathcal{H}(R)$ and $\mathcal{T}(R)$ represent the width of the step at the beginning and at the end of path R, respectively. Without loss of generality, we assume

$$\mathcal{T}(R_{m-1}) + \mathcal{H}(R_0) > k \quad (1)$$

This also implies R_0Q_1 is an MTran path of u_0. We consider the following two cases:

Case 1: If R_0Q_1 is a $uMTran$ path u_0, Q_0 must also be a $uMTran$ path of u_0. Because u_0 prefers R_0Q_1, we have

$$\mathcal{H}(R_0Q_1) \le \mathcal{H}(Q_0) \quad (2)$$

Also because $R_{m-1}Q_0$ is a valid path of u_{m-1}, it should not have steps wider than k, i.e.,

$$\mathcal{T}(R_{m-1}) + \mathcal{H}(Q_0) \le k \quad (3)$$

From (2) and (3), we can derive $\mathcal{T}(R_{m-1}) + \mathcal{H}(R_0Q_1) \le k$. This contradicts (1) because $\mathcal{H}(R_0Q_1) \ge \mathcal{H}(R_0)$.

Case 2: If R_0Q_1 is a $dMTran$ path of u_0, Q_0 can be a $dMTran$ path, a peer path, a provider path, or a $uMTran$ path of u_0. ***Case 2.1:*** If Q_0 is a $dMTran$ of u_0, we can derive a contradiction similar to case 1. ***Case 2.2:*** If Q_0 is a provider path, a peer path, or a $uMTran$ path of u_0, $R_{m-1}Q_0$ must be a $uMTran$ path or a provider path of u_{m-1}. Because u_{m-1} prefers $R_{m-1}Q_0$ over Q_{m-1}, Q_{m-1} must a $uMTran$ path or a provider path of u_{m-1}. Hence, $R_{m-2}Q_{m-1}$ is a $uMTran$ path or a provider path of u_{m-2}. By keeping doing this, we can derive that R_0Q_1 is a $uMTran$ path or a provider path of u_0, this contradicts with the assumption that R_0Q_1 is a $dMTran$ path of u_0.

Since inequation (1) does not hold for case 1 or case 2, the rim of W is an s_hCycle where $h \leq k$.

For other cases where Q_0 is a provider path, a peer path, a $dMTran$ path, or a $uMTran$ path of u_0, we can similarly derive the same conclusion, i.e., $R_0R_1...R_{m-1}$ is an s_hCycle where $h \leq k$. ∎

With Lemma V.2 and Lemma V.3, now we can prove Theorem V.1.

Proof: When the *k-step* policy is adopted and a dispute wheel exists, Lemma V.3 tells us that the rim of the dispute wheel must be an s_hCycle where $h \leq k$. This contradicts the fact that the AS graph $G \in \mathcal{ASG}_k$. Therefore, the dispute wheel does not exist and the *k-step* guarantees routing safety and robustness. ∎

As a special case of Theorem V.1, we have Corollary V.4, which establishes the safety and robustness of the *1-step* policy. The *1-step* policy accommodates the mutual transit agreements where all paths except MTran paths can be announced to MTran neighbors. Therefore, among the three policy guidelines presented in this paper, the safety and robustness of the *1-step* policy require the least restrictions to AS graph G, i.e., $G \in \mathcal{ASG}_1$.

Corollary V.4 *For any AS graph $G \in \mathcal{ASG}_1$, the 1-step policy is safe and robust.*

Finally, if AS graph G is $sCycle$-free ($G \in \mathcal{ASG}_\infty$), the *any-step* policy is safe and robust. This fact is formally stated in Corollary V.5. The *any-step* policy has the least constraints on what paths can be announced to MTran neighbors. However, to guarantee routing safety and robustness, we have to place the most restrictive assumptions on AS graph G, namely, G contains no $s_iCycles$ for any i (thus G is strictly hierarchical).

Corollary V.5 *For any AS graph $G \in \mathcal{ASG}_\infty$, the any-step policy is safe and robust.*

VI. Practical Implications

After presenting the policies and studying their safety and robustness properties, in this section we discuss some practical implications of our policy guidelines. We show how these policies can be realized in BGP without significant configuration effort. Other practical issues are also discussed, such as which ASes can safely establish mutual transit agreements, and how to handle selective mutual transit.

A. Realizing the Policy Guidelines in BGP

Realizing the policies put forth in section V does not require significantly more configuration efforts beyond what are required for BGP today, and the extra configuration efforts are only imposed on those ASes having mutual transit agreements. In realizing the *1-step* policy, the only extra care required is to distinguish between d_1MTran and u_1MTran routes. For the *k-step* policy and the *any-step* policy, we also need the initial step width index i in d_iMTran and u_iMTran routes to rank them. In the following, we provide an example implementation of how such information can be incorporated in the BGP community attribute.

Recall that the 4-octet community attribute is typically represented as x:y (an AS:VALUE pair), where the first two octets x denote the AS number and the second two octets y denote the value. We define the two octets y in such a matter that the first octet y_1 in y=y_1:y_2 represents the type of routes: $customer$, $dMTran$, $peer$, $provider$, or $uMTran$ routes. For $dMTran$ and $uMTran$ routes, the second octet y_2 represents the initial step width. When an AS imports a route from a customer, peer or provider, it sets octet y_1 to $customer$, $peer$ or $provider$ accordingly[4], and sets octet $y_2 = 0$. Before exporting a customer route to an MTran neighbor, it sets the two octets in y to y_1=$dMTran$ and y_2=1. Likewise, before exporting a provider or peer route to an MTran neighbor, it sets y_1=$uMTran$ and y_2=1. Hence, when an AS imports a route from an MTran neighbor, the y_1:y_2 value can indicate whether it is a $dMTran$ or $uMTran$ route and the initial step width. If an AS needs to further export an MTran route to another MTran neighbor, it simply increments y_2 by one before exporting it. On the other hand, if this AS exports a $dMTran$ or $uMTran$ route to a customer, peer or provider, it sets y_2=0, y_1=$customer$, $peer$, or $provider$ before exporting the route.

B. Safely Establishing Mutual Transit Agreements

Certain care must be taken when establishing mutual transit agreements between ASes, because the safety and robustness of the policy guidelines presented in this paper hinge on certain AS graph topological properties. However, given that the provider-customer relationships are usually acyclic, it immediately implies that any two tier-1 ASes can establish a mutual transit agreement where they expose to each other all their paths, and the AS graph still has no $sCycles$. Similarly, any two stub ASes can also safely establish a mutual transit agreement where they announce to each other all their paths, and the resulting AS graph remains to be $sCycle$-free. Stub ASes can safely establish mutual transit agreements is a particularly useful insight, because the majority of ASes in the Internet are stubs.

In general, for ASes other than stub ASes and tier-1 ASes, one can ensure that the resulting AS graph is free of any $sCycles$ and the policies guidelines presented in section V-A guarantee safe and robust routing, as long as mutual transit agreements are established only between ASes of similar size and coverage. Note that it is to an AS's own advantage to establish mutual transit agreements only with ASes of similar

[4]Depending on the arrangement between neighboring ASes, the community attribute may in fact be set by the neighboring AS before the route is exported.

size and coverage. Otherwise, the larger AS would rather be a provider of the smaller AS to generate higher revenue.

C. Handling Selective Mutual Transit Agreements

In previous discussion, we assumed that a mutual transit agreement between two ASes was in effect for all prefixes, i.e., an MTran link has a unique meaning. In practice, however, mutual transit can be applied selectively so that the semantics of a link vary for different sets of prefixes. A realistic example could be two peering ASes agreeing to use their peering link to do mutual transit only for certain destinations. Ideally, we could configure different policies for different prefixes. However, configuring policies for each prefix is difficult in practice because of the large number of prefixes in the Internet. Doing policy configuration on a per-neighbor manner is more practical. We show such an example in Fig. 9, which is similar to Fig. 1. Here Tiscali and Pipex can have a selective mutual transit agreement where Tiscali is willing to transit traffic for Pipex's customer c and Pipex is willing to transit traffic for Tiscali's customer a. As before, the BGP community attribute can be used to realize this per-neighbor based mutual transit configuration. Tiscali and Pipex can locally agree on some community number to indicate mutual transit agreement for certain prefixes. When Tiscali imports routes from customer a, Tiscali uses import filters to assign a community number to those routes. That community number should be preserved when Tiscali announces those routes to Pipex, so that Pipex can know the mutual transit semantic of those routes.

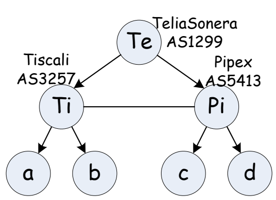


Fig. 9. Per-neighbor based selective mutual transit agreement.

VII. Potential Benefits of Mutual Transit Agreements

In this section, we provide some quantifications of the potential benefits if ASes enter into mutual transit agreements. We study the benefits of tolerating several types of failures, when two peering ASes can safely include mutual transit in their agreement by following the policy guidelines presented in section V (assuming they are willing to do so). Peering ASes are the most natural candidates to enter into mutual transit agreements, because peering relationships are typically established between ASes of similar size and coverage.

A. Experiment Setting

We carry out our investigation by performing a number of experiments on an AS graph derived from the Routeviews BGP tables [20]. We use 160 BGP table snapshots archived in January 2008 as our data set. The AS relationships are inferred using the algorithm in [8]. To speed up our experiments, all stub ASes are removed and only transit ASes are included in the AS graph [21].

Note that the actual benefit of extending peering agreements into mutual transit agreements can be more significant than indicated by the experimental results presented in this section. First, because Routeviews does not have complete BGP tables, our AS graph derived from Routeviews BGP tables misses a large set of peering links [22]. If more peering links are present in the AS graph, more ASes can potentially benefit from extending their peering agreements to mutual transit agreements. Second, the AS relationships are inferred by a heuristic algorithm, which can misclassify some links. Most of the inaccuracy is in misclassifying peering links into provider-customer links [8,23]. Again, if those links are correctly classified so that the AS graph has more peering links, more ASes will be able to benefit from extending mutual transit agreements to their peering links.

B. Fault Tolerance Benefits

We are interested in a few common failure scenarios and how mutual transit agreements can help better tolerate those failures. In our experiments, we compare the Gao-Rexford policy guideline (which accommodates only the transit and peering agreements) to the *1-step* policy and the *any-step* policy. For each failure scenario, we count the number of reachable AS pairs before and after the failure. If AS u can reach AS v and AS v can reach AS u using paths permitted by the corresponding routing policy, we say u and v are a *reachable* AS pair. If u and v are reachable and they become unreachable after the failure, we say u and v are a *disconnected* AS pair.

1) Access link failures: Access links are the links connecting an AS to its providers. An AS with a peer neighbor can tolerate access link failures by expanding its peer agreement into a mutual transit agreement. That is, if all access links of an AS fail, the peering neighbor can transit its traffic. We ran 50 instances of failure experiments. In each instance, one AS among all the ASes that can safely convert one of their peer agreements into mutual transit agreements is selected, and all its access links are failed. We count the number of disconnected AS pairs in each experiment instance. The results of disconnected AS pairs are presented in Fig. 10. As we can see, a significant number of AS pairs become disconnected when using the Gao-Rexford policy. In some cases, as many as 18,000 AS pairs get disconnected because one AS has its access links failed. However, under either the *1-step* or the *any-step* policies, no AS pairs are disconnected in this failure scenario.

2) Tier-1 de-peering: This corresponds to a scenario where two tier-1 ASes decide to terminate their connection. As the study in [21] shows, tier-1 de-peering can have a huge impact on the reachability of ASes single-homed to the de-peered tier-1 ASes. We select some well-known tier-1 AS pairs [19] and let them de-peer in our experiments. Not unexpectedly, the *1-step* policy does not offer any improvement over the Gao-Rexford policy. However, as shown in TABLE I, the *any-step* policy is able to entirely eliminate any loss of connectivity.

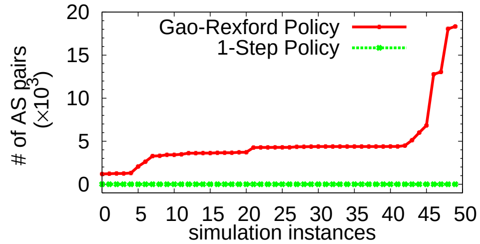


Fig. 10. Number of disconnected AS pairs in access link failures when the Gao-Rexford policy and the *1-step* policy are adopted, respectively. The result for *any-step* policy is the same as the *1-step* policy result.

This is because the *any-step* policy allows AS paths with multiple consecutive peering links (now they have the mutual transit semantics) to be used. As a result, the de-peered tier-1 ASes can use other tier-1 ASes to bypass the failed peering link.

TABLE I
NUMBER OF DISCONNECTED AS PAIRS UNDER TIER-1 DE-PEERING.

peering link	**# of disconnected AS pairs**	
	Gao-Rexford	any-step
1239 - 3356	546	0
1239 - 7018	294	0
701 - 1239	273	0
701 - 3356	338	0

3) AS partition: This last scenario considers failures that partition a tier-1 AS into two disconnected components. Using the NetGeo service [24], we classify the US customers of a tier-1 AS into three categories: east coast customers, west coast customers, and other customers. We assume that after a partition the east coast customers and west coast customers of the tier-1 AS cannot reach each other through that tier-1 AS. We test two well-known tier-1 ASes, Quest and AT&T, and present the results of disconnected AS pairs in TABLE II. As with the tier-1 de-peering scenario, the *any-step* policy offers full protection against AS partition failures. This is again because the *any-step* policy allows a second tier-1 AS to transit traffic between the east coast and west coast customers of the partitioned tier-1 AS.

TABLE II
NUMBER OF DISCONNECTED AS PAIRS UNDER TIER-1 AS PARTITION.

tier-1 AS	**# of disconnected AS pairs**	
	Gao-Rexford	any-step
209 Quest	86	0
7018 AT&T	113	0

VIII. CONCLUSION

This paper studies the fundamental problem of safely accommodating diverse mutual transit agreements in inter-domain routing. These mutual transit agreements can take several possible forms and some of them already exist in the Internet, e.g., when two ASes merge or two ASes establish a sibling relation. We propose a series of policy guidelines that support mutual transit agreements with progressively richer semantics and study the safety and robustness of those policy guidelines. Based on those theoretical insights, we further discuss how diverse mutual transit agreements can be safely established and easily implemented in BGP. We also demonstrate the benefits, in terms of routing reliability under various representative failure scenarios, of extending Internet peering agreements to mutual peering agreements.

ACKNOWLEDGMENT

The authors are sincerely grateful to the editor, Prof. Olivier Bonaventure, and the anonymous reviewers for many helpful comments and constructive suggestions. This work is supported by NSF grants CNS-0627004, CNS-0626808, and CNS-0626617.

REFERENCES

[1] V. Valancius, N. Feamster, R. Johari, and V. Vazirani, "MINT: A Market for INternet Transit," in *CoNEXT '08: Proceedings of the 2008 ACM CoNEXT Conference*. New York, NY, USA: ACM, 2008, pp. 1–6.
[2] J. Feigenbaum, V. Ramachandran, and M. Schapira, "Incentive-compatible interdomain routing," in *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, 2006, pp. 130–139.
[3] J. R. Lane and A. Nakao, "Path brokering for end-host path selection: toward a path-centric billing method for a multipath internet," in *CoNEXT '08: Proceedings of the 2008 ACM CoNEXT Conference*. New York, NY, USA: ACM, 2008, pp. 1–6.
[4] T. G. Griffin and G. Wilfong, "An analysis of BGP convergence properties," *SIGCOMM Comput. Commun. Rev.*, vol. 29, 1999.
[5] L. Gao and J. Rexford, "Stable internet routing without global coordination," in *SIGMETRICS '00: Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*. New York, NY, USA: ACM, 2000, pp. 307–317.
[6] L. Gao, T. G. Griffin, and J. Rexford, "Inherently safe backup routing with BGP," in *Proceedings of INFOCOM'01*, 2001.
[7] N. Feamster, R. Johari, and H. Balakrishnan, "Implications of autonomy for the expressiveness of policy routing," in *Proceedings of SIGCOMM '05*, Philadelphia, Pennsylvania, USA, 2005.
[8] L. Gao, "On inferring autonomous system relationships in the Internet," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 733–745, 2001.
[9] K. Varadhan, R. Govindan, and D. Estrin, "Persistent route oscillations in inter-domain routing," *Computer Networks*, vol. 32, no. 1, Jan. 2000.
[10] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "Policy disputes in path-vector protocols," in *Proceedings of ICNP '99*. Washington, DC, USA: IEEE Computer Society, 1999.
[11] T. G. Griffin and G. Wilfong, "A safe path-vector protocol," in *Proceedings of INFOCOM' 00*, 2000.
[12] C. T. Ee, V. Ramachandran, B.-G. Chun, K. Lakshminarayanan, and S. Shenker, "Resolving inter-domain policy disputes," in *SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2007, pp. 157–168.
[13] A. D. Jaggard and V. Ramachandran, "Robustness of class-based path-vector systems," in *ICNP '04: Proceedings of the 12th IEEE International Conference on Network Protocols*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 84–93.
[14] J. Sobrinho, "Network routing with path vector protocols: theory and applications," in *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 2003, pp. 49–60.
[15] J. L. Sobrinho, "An algebraic theory of dynamic network routing," *IEEE/ACM Trans. Netw.*, vol. 13, no. 5, pp. 1160–1173, 2005.
[16] T. G. Griffin and J. L. Sobrinho, "Metarouting," in *SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2005, pp. 1–12.
[17] G. Huston, "Interconnection, peering, and settlements," in *Proceedings of INET' 99*, Jun. 1999.
[18] "Tiscali acquires Pipex broadband business." [Online]. Available: http://www.computing.co.uk/computing/news/2194229/tiscali-acquires-pipex
[19] "Tier 1 network." [Online]. Available: http://en.wikipedia.org/wiki/Tier_1_carrier
[20] "The RouteViews Project," http://www.routeviews.org.

[21] J. Wu, Y. Zhang, Z. M. Mao, and K. G. Shin, "Internet routing resilience to failures: analysis and implications," in *Proceedings of CoNEXT '07*, New York, NY, Dec 2007.
[22] Y. He, G. Siganos, M. Faloutsos, and S. Krishnamurthy, "Lord of the links: a framework for discovering missing links in the internet topology," *IEEE/ACM Trans. Netw.*, vol. 17, no. 2, pp. 391–404, 2009.
[23] J. Xia and L. Gao, "On the evaluation of AS relationship inferences," in *Proceedings of Globecom'04*, 2004.
[24] "NetGeo API." [Online]. Available: http://www.caida.org/tools/utilities/netgeo/NGAPI/index.xml

APPENDIX

A. Dispute Wheel

The safety and robustness of our routing policy guidelines are established by a sufficient condition proved in [10], i.e., no dispute wheel ensures safety and robustness. A *dispute wheel* W of size m, as shown in Fig. 11, is a triple $(\mathcal{U}, \mathcal{Q}, \mathcal{R})$, where $\mathcal{U}$ is a sequence of m nodes $u_0, u_1...u_{m-1}$ called the *pivot nodes*; $\mathcal{Q}$ is a sequence of m non-empty paths $Q_0, Q_1...Q_{m-1}$, which are often referred to as the *spoke paths*; and $\mathcal{R}$ represents m non-empty paths $R_0, R_1...R_{m-1}$. This triple is such that for each $0 \leq i < m$, we have (1) R_i is a path from u_i to u_{i+1}; (2) Q_i and R_iQ_{i+1} are valid paths at u_i; and (3) u_i prefers R_iQ_{i+1} over Q_i. All subscripts are to be interpreted modulo m. R_iQ_{i+1} is often called the *rim path*. $R_0R_1...R_{m-1}$ is often referred to as the rim of W.

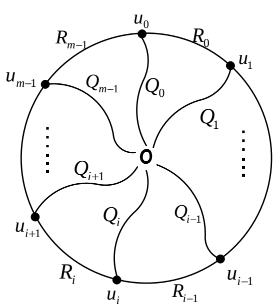


Fig. 11. A dispute wheel $W = (\mathcal{U}, \mathcal{Q}, \mathcal{R})$ of size m.

Yong Liao graduated with a B.S. degree in 2001 from University of Science and Technology of China. In 2004, he received his M.S. degree from the Graduate School of Chinese Academy of Sciences. Since fall 2004, 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, data center network, and network virtualization.

Lixin Gao (F IEEE '10) is a professor of Electrical and Computer Engineering at the University of Massachusetts at Amherst. She received her Ph.D. degree in computer science from the University of Massachusetts at Amherst in 1996. Her research interests include multimedia networking, and Internet routing and security. Between May 1999 and January 2000, she was a visiting researcher at AT&T Research Labs and DIMACS. She is an Alfred P. Sloan Fellow and received an NSF CAREER Award in 1999. She has served on a number of technical program committees including SIGCOMM2006, SIGCOMM2004, SIGMETRICS2003, and INFOCOM2004, and is on the Editorial Board of IEEE Transactions on Networking.

Roch Guérin (F IEEE '01 / F ACM '06) received an engineer degree from ENST, Paris, France, and M.S. and Ph.D. degrees in EE from Caltech.

From 1986 till 1998 he was with the IBM T. J. Watson Research Center, Yorktown Heights, NY. In 1998 he joined the Electrical and Systems Engineering department of the University of Pennsylvania, as the Alfred Fitler Moore Professor of Telecommunications Networks. His research has spanned core networking issues, e.g., routing, traffic engineering, quality-of-service, etc., as well as topics related to the use of networks by applications and the exploration of how economic factors affect the evolution of networked systems.

Dr. Guérin has been an editor for several ACM and IEEE publications, and has been involved in the organization of numerous ACM and IEEE sponsored activities. He was General Chair of the IEEE INFOCOM'98 conference, Technical Program co-chair of the ACM SIGCOMM'01 conference, General Chair of the ACM SIGCOMM'05 conference, and Program co-chair of the ACM CoNEXT'07 conference. He currently serves on the ACM CoNEXT steering committee. In 1994 he received an IBM Outstanding Innovation Award for his work on traffic management and the concept of equivalent bandwidth, and in 2010 he received the IEEE INFOCOM Achievement Award for "Pioneering Contributions to the Theory and Practice of QoS in Networks." He was on the Technical Advisory Board of France Telecom for two consecutive terms from 2001 to 2006, and on that of Samsung Electronics in 2003-2004. He joined the Scientific Advisory Board of Simula in 2010.

Zhi-Li Zhang received a B.S. degree in Computer Science from Nanjing University, China, in 1986 and his M.S. and Ph.D. degrees in computer science from the University of Massachusetts at Amherst in 1992 and 1997. In 1997 he joined the Computer Science and Engineering faculty at the University of Minnesota, where he is currently a professor. From 1987 to 1990, he conducted research in Computer Science Department at Arhus University, Denmark, under a fellowship from the Chinese National Committee for Education. He has held visiting positions at Sprint Advanced Technology Labs, IBM T.J. Watson Research Center, Fujitsu Labs of America, Microsoft Research China, and INRIA, Sophia-Antipolis, France.