On the Correctness of IBGP Configuration

Timothy G. Griffin
AT&T Research
griffin@research.att.com

Gordon Wilfong
Bell Labs, Lucent Technologies
gtw@research.bell-labs.com

ABSTRACT

The Border Gateway Protocol (BGP) has two distinct modes of operation. External BGP (EBGP) exchanges reachability information between autonomous systems, while Internal BGP (IBGP) exchanges external reachability information within an autonomous system. We study several routing anomalies that are unique to IBGP because, unlike EBGP, forwarding paths and signaling paths are not always symmetric. In particular, we focus on anomalies that can cause the protocol to diverge, and those that can cause a router's chosen forwarding path to an egress point to be *deflected* by another router on that path. Deflections can greatly complicate the debugging of routing problems, and in the worst case multiple deflections can combine to form persistent forwarding loops. We define a correct IBGP configuration to be one that is anomaly free for *every* possible set of routes sent by neighboring autonomous systems. We show that determination of IBGP configuration correctness is NP-hard. However, we give simple sufficient conditions on network configurations that guarantee correctness.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing Protocols*

General Terms

Algorithms, Theory

Keywords

Border Gateway Protocol, BGP, Internal BGP, BGP Configuration

1. INTRODUCTION

The Border Gateway Protocol (BGP) [19, 15] has two distinct modes of operation. External BGP (EBGP) exchanges reachability information between autonomous systems, while Internal BGP (IBGP) exchanges external reachability information within an autonomous system. IBGP was not designed to maintain reachability *within* an autonomous systems. This task is normally performed by an Interior Gateway Protocol (IGP) such as OSPF, IS-IS and RIP [16]. IBGP's primary task is to allow internal sources to reach external destinations and external sources to reach internal destinations. BGP routing of external routes and interior routing interact — IBGP's route selection process can prefer one route over another because it has a lower IGP "distance" associated with its egress point. This paper investigates two types of anomalous behavior that can arise from this interaction.

EBPG and IBGP differ in both their treatment of certain attributes and in the manner in which BGP messages are forwarded. For our study, the forwarding model of IBGP will be central. EBGP normally assumes that a BGP session is established between two routers that share a physical network. That is, EBGP messages are normally not routed. IBGP on the other hand is routed within an autonomous system using the connectivity provided by the IGP. With pure EBGP, signaling messages and forwarding traffic flow along the same paths, but in opposite directions. We use the term *path symmetry* to describe this scenario. Since IBGP is routed, there is no guarantee that the forwarding and signaling paths are symmetric.

In this paper we focus on IBGP routing anomalies that arise from path asymmetry. It is known that route oscillations can occur in IBGP due to the use of the Multi-Exit Discriminator (MED) path attribute [18]. However, it has been observed that MED-induced oscillation has a different root cause [14]. In particular, the MED attribute is treated in a distinct manner by the BGP route selection algorithm. The specification of BGP in RFC 1771 [19] states that the route selection process should conform to what we call the *rule of independent ranking*, which states that the preference of any route should not be a function of the existence or nonexistence of other routes in the BGP table. The selection algorithm's treatment of the MED attribute violates this rule, and can lead to oscillation problems [18, 4]. In this paper we ignore the MED attribute.

The first IBGP anomaly we consider is one that has been carefully studied for EBGP — the fact that the protocol can fail to converge [22, 13, 11, 8, 10, 7]. That is, in an otherwise stable network, BGP speaking routers can exchange routing messages forever without establishing a stable set of routes. We show that, even in the absence of MED, IBGP can diverge as a result of path asymmetry.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGCOMM'02, August 19-23, 2002, Pittsburgh, Pennsylvania, USA.
Copyright 2002 ACM 1-58113-570-X/02/0008 ...$5.00.

The second type of IBGP anomaly we study involves inconsistent forwarding paths within an autonomous system. When a BGP speaking router selects a best route to an external destination, that route has a BGP next-hop value that identifies a particular *point of egress* from the autonomous system. The forwarding path from the router to its selected point of egress is supplied by IGP routing. However, it may happen that along this forwarding path another BGP speaking router has chosen a *different* point of egress for the same external destination. In this case we will say that the forwarding path has been *deflected.* Again, this problem arises due to path asymmetry. Deflections can greatly complicate the debugging of routing problems, and in the worst case multiple deflections can combine to form a persistent forwarding loop.

We consider an autonomous system's *network configuration* to be comprised of the network topology, the IGP metrics, the IBGP sessions, and the BGP routing policies. Given a network configuration, what could it mean to say that it is free of the anomalies described above? Network operators have complete control over their network configurations, but they have little control over the routing messages sent by external neighbors. For this reason, it is helpful to view a network configuration as a program that takes as input external routing messages and produces as output a stable routing for the autonomous system. Configuration correctness is then taken to be correctness of this program — we require that, for every possible set of inputs, the program terminates with a stable routing that is free of deflections. We call such a configuration *correct.* Note that this is essentially a way of defining configuration sanity within an isolated autonomous system without requiring any assumptions concerning global routing sanity.

In addition to providing this framework for discussing the correctness of IBGP configurations, we prove several complexity results concerning configuration correctness. In particular, we show that determining if a configuration will converge for all possible inputs is an NP-hard problem. We also show that deciding if, for all inputs, a configuration determines a deflection-free routing is NP-hard as well. For both types of routing anomalies, we define sufficient conditions that guarantee configuration correctness.

We will concentrate on IBGP configurations that use route reflectors [2]. BGP confederations [21] represent another technique used to scale the internal configuration of BGP. Although confederations can introduce path asymmetries that lead to the routing anomalies outlined here, we will not cover them explicitly since many of the arguments are similar.

We believe that our observations have significance beyond technical details narrowly concerned with BGP configuration. Protocol designers using "vectoring" techniques should be aware of the potential anomalies that can arise when the relationships between forwarding and signaling paths are unconstrained.

1.1 Related Work

A comparison of the scaling abilities of route reflection and BGP confederations, as well as the simple looping example we use in Figure 4, are presented in [5, 6].

Determination of EBGP convergence was shown to be NP-complete in [13]. The current paper looks at IBGP correctness, and focuses on the interaction of IBGP and interior routing. Although many of our examples may not arise directly in practice, they do show that implementing algorithms for automatically checking IBGP configuration correctness will be more complex than expected. In addition, BGP misconfigurations are not uncommon [17], and they may interact with network failures in surprising and unexpected ways.

We take the BGP protocol as given, and attempt to develop correct configurations to avoid routing anomalies. Another approach is to redesign the protocol so that anomalies cannot occur. For example, [1] proposes a modification to the manner in which IBGP routes are propagated. Although this work was aimed at solving the "MED oscillation problem," it does address the signaling anomalies we focus on. In addition, their solution guarantees that a routing will have no loops, although there may still be deflections. However, their solution comes at the cost of increased memory utilization for all BGP speakers within an autonomous system, which may be incompatible with one of the main motivations for using route reflection.

1.2 Outline

Section 2 reviews the BGP route selection process, and Section 3 presents a formal definition of an IBGP configuration. Section 4 presents examples of both signaling and forwarding anomalies in IBGP. NP-hardness results for signaling correctness are presented in Section 5, and those for forwarding correctness are presented in Section 6. Section 7 concludes with some remarks concerning the implications of this work for protocol design.

2. BGP ROUTE SELECTION

We study a single autonomous system that may or may not be using route reflection [2]. An autonomous system consists of a network of routers connected by some underlying physical network where each physical link has been assigned a distance. Routers will interchange routing information via IBGP [15]. At any given time, any number of the border routers may be announcing EBGP routes into the autonomous system. The description of the physical connectivity, IBGP connectivity, and border router will be called a *configuration* of an autonomous system. All of our analysis will be in terms of one particular external destination, called the *origin.*

The external routes that are being announced into the autonomous system will be called the *egress paths* and the border routers that are announcing these paths will be called the *egress nodes.* In the presence of route reflection, a cluster refers to a set of route reflectors $\mathcal{R}$ and a set of clients $\mathcal{C}$ where each router in $\mathcal{R}$ acts as a route reflector for each client in $\mathcal{C}$.

Each route update message has a number of attributes associated with it and the route chosen by a router is based on these attributes. The standard route selection procedure of BGP (see, for example [15]) selects first those routes with highest **local_pref** values, then from these the ones with shortest **as_path** length and then of these, the ones with lowest **med** values (for those routes with the same next hop). At any instant of time, there will be some set of egress paths to the origin and we consider the subset of these paths that have not been filtered out by virtue of having low value of **local_pref**, long **as_path** length or a **med** value that is higher than some other egress path to the same neighboring autonomous system. Thus, according to the route selection preference rules for IBGP, the choice for each router in the autonomous system of which of the remaining egress paths to take is based on

(i) first selecting any egress paths learned via EBGP over those learned by IBGP

(ii) and then from these, select those whose egress nodes are nearest according to the distance values assigned to the links in the physical network

(iii) and finally, from the remaining egress paths, use some tie breaking rules based on the **oid** attribute (the id number of the egress node), the length of the cluster list **cluster_list** (a list of the clusters the message has passed through) and the **pid** (the id of the router from which the announcement of the route was received).

Note that IBGP does not require that every router within an autonomous systems choose the same egress point for a given destination.

3. IBGP CONFIGURATION

We define an *IBGP configuration* C to be a pair $C = (G_P, G_S)$, where G_P is the *physical graph* and G_S is the *signaling graph.*

3.1 Physical Graph

First we define a weighted graph $G_P = (V, B, E_P, d)$ called the physical graph to represent the underlying physical topology of the system. Each node $u \in V$ represents a router. We will often refer to a router u meaning the router represented by the node u. The set $B \subseteq V$ is the set of border routers that have physical connectivity to external networks. There is an edge $e = \{u, v\} \in E_P$ if there is a physical link between the routers represented by u and v. The distance $d(e)$ of an edge is the distance assigned to a physical link that is typically used by such protocols as OSPF for determining shortest paths between nodes.

In general, we indicate a path P as a sequence of nodes

$$P = u_1 u_2 \ldots u_n.$$

We concatenate paths $P = u_1 u_2 \ldots u_n$ and $Q = v_1 v_2 \ldots v_m$ to form a new path $PQ = u_1 u_2 \ldots u_n v_1 v_2 \ldots v_m$ if $u_n \neq v_1$ or $PQ = u_1 u_2 \ldots u_n v_2 \ldots v_m$ if $u_n = v_1$. We define the *length of a path* P to be the sum of the distances $d(e)$ over all edges e of P. Let P be a path between nodes u and v such that there is no path between nodes u and v whose length is less than that of P. Then the length of P is written as $M(u, v)$ ($=M(v, u)$). Such a path P is said to be a *shortest path* between u and v. For every pair of nodes u and v we define $sp(u, v)$ in G_P to be a shortest path between u and v and we assume that such paths are consistent in the sense that if $sp(v_1, v_t) = v_1 v_2 \ldots v_t$ then $sp(v_i, v_j) = v_i v_{i+1} \ldots v_j$, $1 \leq i < j \leq t$ and $sp(v_t, v_1) = v_t v_{t-1} \ldots v_1$.

3.2 Signaling Graph

We define the signaling digraph $G_S = (V, A_S)$ as follows. An arc in G_S represents an IBGP session between two routers. Since BGP sessions are implemented with routed TCP sessions, an arc in G_S corresponds to some routed *path* in G_P.

The set of arcs A_S is partitioned into three sets **over**, **down**, and **up**. An arc $(u, v) \in$ **over** represents the fact that there is a vanilla IBGP session allowing route update messages to flow from u to v. This is *routed* traffic (using interior routing) whose physical realization will not concern us. Typically we would expect to have an arc $(v, u) \in$ **over** whenever the arc $(u, v) \in$ **over**. An arc $(u, v) \in$ **down** represents the signaling of route update messages from a route reflector u to one of its clients v. In such a case, u would send route update messages to v for any routes except those announced by v. Finally, an arc $(u, v) \in$ **up** represents the signaling of route update messages from a client u to a router v acting as a route reflector for u. In this case, u sends route update messages only about the routes it announced into the system. Typically we would expect that $(u, v) \in$ **down** if and only if $(v, u) \in$ **up**.

A valid signaling path S satisfies the following properties. The path S can be partitioned into three subpaths $S = PQR$ where $P = p_1 p_2 \ldots p_a$ for some $a \geq 0$ where each $p_i \in$ **up**, $R = r_1 r_2 \ldots r_b$ for some $b \geq 0$ where each $r_i \in$ **down** and Q is either empty or consists of a single arc $q \in$ **over**. Note that any of P, Q and R may be empty.

3.3 Configuration Instance

Network operators have control over their network configurations. However, they do not have control over what routes are sent to them by their neighbors. Routes to a given destination could potentially arrive at any subset of the border routers, $X \subseteq B$. We can think of the configuration as a system that takes as input a set X and returns as output a routing. As we will see, such a system may be nondeterministic and is not even guaranteed to terminate with a routing. We are interested in investigating correctness conditions that guarantee that a configuration will always behave deterministically and terminate with a routing. Even when these guarantees are met, it is possible that the resulting routing has undesirable forwarding behavior such as forwarding loops. Therefore, we will define additional correctness conditions to prevent such forwarding anomalies.

An *egress instance* is a pair $I = (C, X)$ where C is a configuration and $X \subseteq B$. We define an egress instance to be *signaling correct* if it is guaranteed to deterministically arrive at a unique routing. A configuration C is signaling correct when every egress instance (C, X) is signaling correct. For signaling correct configurations C, we say that an egress instance is *forwarding correct* if the resulting routing contains no deflections (we define this more formally in Section 4.2). Then we say that a configuration C is forwarding correct if every egress instance (C, X) is forwarding correct. A configuration is *correct* if it is both signaling correct and forwarding correct.

4. CONFIGURATION CORRECTNESS

This section presents several examples of both signaling and forwarding anomalies in IBGP.

4.1 Examples of signaling anomalies

Previous papers have discussed situations in which EBGP systems fail to settle on a routing [12, 22]. We now show that such situations can occur with IBGP within a single autonomous system even in the absence of problems due to the **med** attribute.

In Figure 1, we see an egress instance that has no solution.

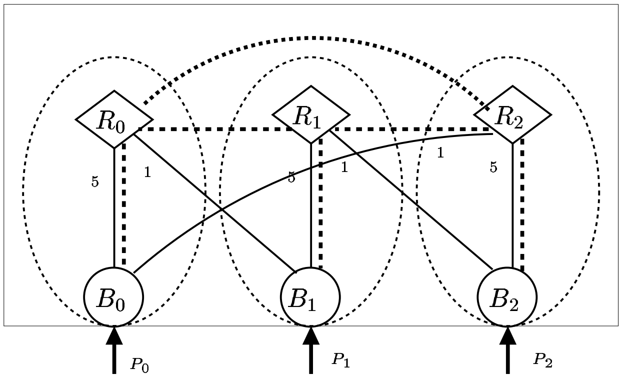


Figure 1: No solution.

Typically we use the following conventions in the figures in this paper. A client within a cluster is shown as a circle, a route reflector as a diamond and the elements of the cluster are enclosed in a dashed oval. The solid lines in the figures are the physical connections and are usually labeled with their IGP weights. Dashed lines between two routers indicate that they exchange IBGP messages. The clusters are enclosed in a box representing the autonomous system boundary. External routes (to the origin) are shown as arrows entering the autonomous system at border routers.

In Figure 1, each R_i acts as a route reflector with respect to C_i and each C_i has an immediate egress path P_i that it has learned via EBGP, $i = 0, 1, 2$. The weights on the edges make it so that the ranking of the egress paths for each R_i is such that R_i prefers P_{i+1} over P_i which is preferred over P_{i+1} where the subscripts are to be interpreted modulo 3. It can easily be verified that in this situation, the routers will never converge on a set of routes. Thus it would make no sense to ask whether such a system gives rise to fixed forwarding loops since it can never settle into a fixed routing.

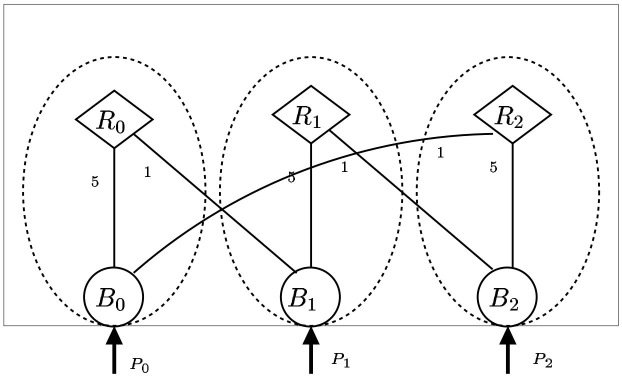


Figure 2: No solution, no clutter.

Graphical conventions: To avoid excessive clutter in figures, the IBGP sessions (dashed lines) will not be shown, but are implied by the clustering structure. That is, we assume that each client has BGP sessions between itself and each route reflector in the cluster and that all route reflectors at a given level of the hierarchy are completely meshed. Using these conventions, the configuration of Figure 1 will be presented as shown in Figure 2.

The configuration of Figures 1 and 2 has no solution and results in BGP divergence. In contrast, a BGP system may have multiple solutions resulting in nondeterministic routing — the protocol will choose one solution based on the random order of message processing. Figure 3 presents an IBGP configuration that has two solutions.

The router R_i acts as a route reflector with respect to C_i where C_i has an egress path P_i learned via an EBGP session, $i = 1, 2$. There are two possible solutions. The timing of the route update messages sent between R_1 and R_2 determine which of the two solutions would be installed. In either situation each C_i will choose its immediate egress path P_i. If R_1 informs R_2 that it has chosen P_1 first, then R_2 will also choose egress path P_1. Similarly, if R_2 tells R_1 that it has chosen P_2 first, then R_1 will then choose P_2 as its favorite egress path.

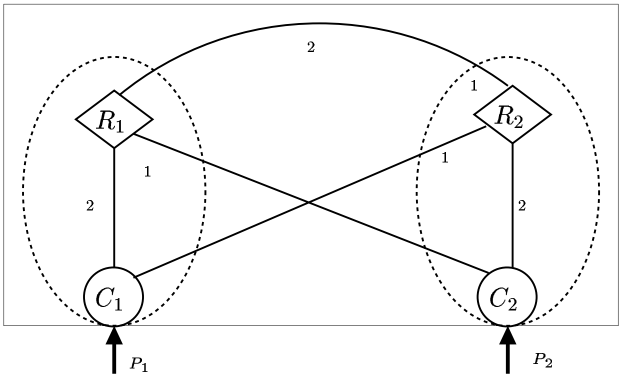


Figure 3: Two solutions.

4.2 Examples of forwarding anomalies

Recall that for each node $v \in X$ and each node $u \in V$, $sp(u, v)$ denotes some fixed shortest path from u to v in G_P. At some fixed time, let $\pi(u)$ denote the route selected by IBGP at node u to the origin. Define $f_\pi(u)$, the *forwarding path at u according to π*, to be $sp(u, v)$ where $\pi(u)$ has egress node v. We think of $f_\pi(u)$ as being directed from u to v. Thus $f_\pi(u)$ is the path along which u would send packets within the autonomous system towards the origin.

Suppose $f_\pi(u_1) = u_1 u_2 \ldots u_t$. Then we use the notation $f_\pi(u_1)[u_i, u_j]$ to mean the subpath of $f_\pi(u_1)$ starting at u_i and ending at u_j, $i \leq j$. We say that there is a *deflection* on $f_\pi(u_1)$ at u_i if $f_\pi(u_i) \neq f_\pi(u_1)[u_i, u_t]$ but for all $j < i$, $f_\pi(u_j) = f_\pi(u_1)[u_j, u_t]$. That is, u_i is the first node from u_1 along $f_\pi(u_1)$ whose forwarding path is not consistent with that of node u_1. An egress instance is *forwarding correct* if it results in routes at each node that create no deflections.

Consider the path that starts at some node u, follows $f_\pi(u)$ until a deflection, say at w, follows $f_\pi(w)$ until it meets a deflection at say x, follows $f_\pi(x)$ and so on. If such a path contains a cycle, then we say that the path assignment π contains a *forwarding loop.*

Suppose there is a deflection along $f_\pi(u)$ at some node w. If $f_\pi(w) = w$ then we say that this is a *simple deflection.* Note that in this case, w must be in X. The idea is that packets from u will not continue on to the egress node of u, but instead will be deflected out of the autonomous system at w. Thus simple deflections cannot cause a routing loop within the autonomous system. Of course, there can be more complicated deflections. It may be that $f_\pi(u_i)$ is just a different path to u_t from u_i than is implied by $f_\pi(u_1)$ or $f_\pi(u_i)$ may end at a different node of X. These types of deflections could cause a forwarding loop.

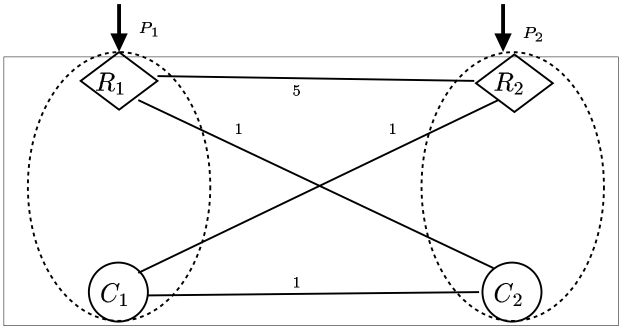


Figure 4: Example of a forwarding loop.

An example of a forward loop, taken from [5], is shown in Figure 4. The system has two clusters each with a router R_i acting as a route reflector for client router C_i. Thus route updates are sent between R_i and C_i, $i = 1, 2$. The only other IBGP session occurs between R_1 and R_2. Each R_i is also a border router announcing the egress path P_i into the network. Clearly R_i will always choose P_i and announce it to C_i. Hence C_i will only know about P_i and so it will choose P_i as well. But note that the shortest path from C_1 to P_1 is $C_1C_2R_1$ and the shortest path from C_2 to P_2 is $C_2C_1R_2$ and so there will be a forwarding loop between C_1 and C_2.

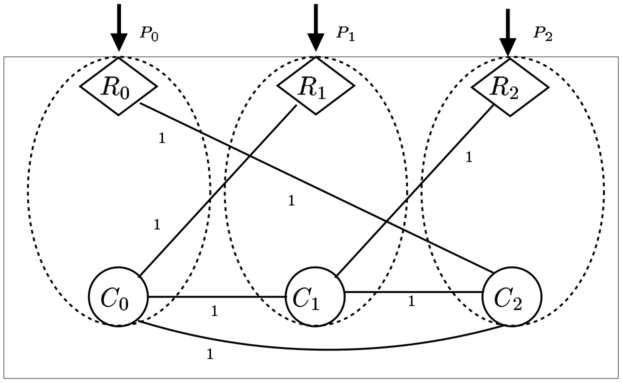


Figure 5: Forwarding loop caused by 3 forwarding paths.

There can be more complicated forwarding loops consisting of any number of edges and involving any number of forwarding paths. Figure 5 shows an example where three forwarding paths together form a forwarding loop and generalizing this example to any number of forwarding paths can be easily done. In the example, each client C_i prefers R_i as its egress node (since that is the only one its route reflector R_i will ever announce to it). All physicals links have cost 1 and so the shortest path from C_i to R_i is $C_iC_{i-1}R_i$ (where subscripts are to be evaluated modulo 3). But then we get the forwarding loop $C_0C_2C_1C_0$.

It should be noted that while the union of all the forwarding paths might contain a directed cycle, this does not necessarily imply that a forwarding loop exists. For example,

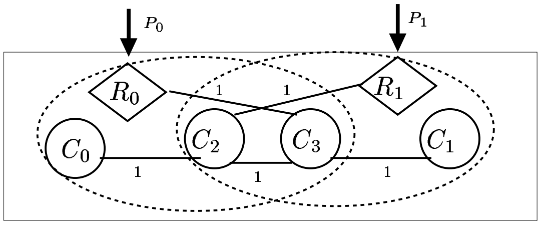


Figure 6: Union of forwarding paths contains a cycle but there is no forwarding loop.

consider Figure 6. It can be easily verified that in the solution to this system the forwarding path for C_1 is $C_1C_3C_2R_1$ and that of C_0 is $C_0C_2C_3R_0$. Thus the union of these two forwarding paths has the directed cycle $C_2C_3C_2$. However in the solution the forwarding path for C_2 is C_2R_1 and that of C_3 is C_3R_0. Thus there is no forwarding loop because the forwarding paths of C_0 and C_1 get deflected by C_2 and C_3 respectively.

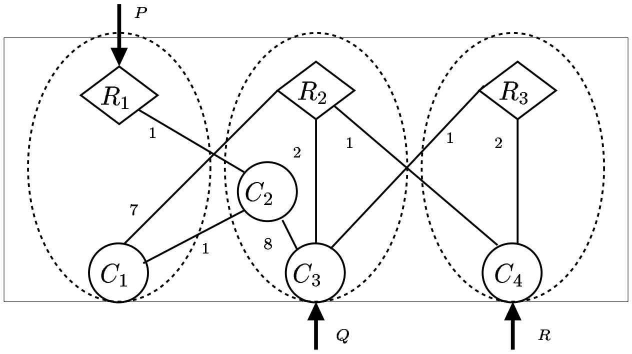


Figure 7: Timing dependent forwarding loop.

We now show that signaling and forwarding anomalies both can be present in the same configuration. Consider the egress instance shown in Figure 7. Notice that the subsystem consisting of R_2, R_3, C_3 and C_4 is the same as the egress instance shown in Figure 3 and so has two possible solutions. Also, the subsystem consisting of R_1, R_2, C_1 and C_2 is essentially the egress instance shown in Figure 4 will be shown to have a forwarding loop. Note that whichever egress path R_i chooses, router C_i will have to choose the same one since it only hears of egress paths via R_i, $i = 1, 2$. Notice that R_1 will always choose P and since C_1 will also choose P, the forwarding path for C_1 will always be $C_1C_2R_1$. Also, R_2 and R_3 will never choose P because Q and R respectively are always available to them and closer than P. The forwarding paths for R_2 and R_3 then are always the single links to either C_3 or C_4. Thus the only forwarding path that can cause forwarding loops is the forwarding path from C_2. Consider one solution where R_2 and R_3 both choose egress path Q. Then the forwarding path from C_2 is just C_2C_3 and no forwarding loops result. However, suppose both R_2 and R_3 choose egress path R. Then the forwarding path for C_2 to R is $C_2C_1R_2C_4$ with length 9 (as opposed to the path $C_2C_3R_2C_4$ with length 11) and this causes a forwarding loop between C_1 and C_2. This shows that nondeterministic routing makes it impossible to statically determine if this egress instance will have deflections.

5. RESULTS ON SIGNALING CORRECTNESS

In this section we study signaling correctness. In Section 5.1 we show how to translate a configuration into the SPP model [12]. In Section 5.2, we use the SPP model to formally define the notions of a deterministic egress instance and a convergent egress instance . In Section 5.3 we show that determining if a configuration is signaling correct is an NP-hard problem. Finally, in Section 5.4 we provide sufficient conditions that guarantee that a configuration will be signaling correct.

5.1 Translation to SPP

An instance S of SPP consists of a graph $G = (V, E)$ with a distinguished node 0, a set of permitted paths $\mathcal{P}^u$ for each node $u \neq 0$ in the graph and a ranking function λ^u defined on the set of permitted paths at each node $u \neq 0$. Then $\mathcal{P} = (\mathcal{P}^{v_1}, \mathcal{P}^{v_2}, \ldots, \mathcal{P}^{v_n})$ and $\Lambda = (\lambda^{v_1}, \lambda^{v_2}, \ldots, \lambda^{v_n})$ where $V = \{0, v_1, v_2, \ldots, v_n\}$. Thus S is defined by the triple $(G, \mathcal{P}, \Lambda)$. For any path $P = u_1 u_2 \ldots u_n$, let $P^r = u_n u_{n-1} \ldots u_2 u_1$.

For a node $u \in V$ and a set $W \subseteq \mathcal{P}^u$, define $\max(u, W) = \varepsilon$ if $W = \emptyset$ and otherwise $\max(u, W) = P0 \in W$ where $\lambda^u(P0)$ is maximal. A *path assignment* π for an instance $S = (G, \mathcal{P}, \Lambda)$ of SPP is a function such that $\pi(u) \in \mathcal{P}^u$ for each node $u \in V$. The set choices(u, π) is defined to be all $P0 \in \mathcal{P}^u$ where either $P = u \in X$ or $P0 = \{uv\}\pi(v)$ for some $\{u, v\} \in E$. A solution to S is then a path assignment π such that the following holds for each vertex $u \in V$. Then a path assignment π is said to be *stable at node* u if

$$\pi(u) = \max(u, \text{choices}(u, \pi)).$$

A path assignment is said to be *stable* if it is stable at every node in V.

A stable path assignment for S is called a *solution* for S. The SPP instance S is said to be *solvable* if there is some solution for S. If no stable assignment for S exists, then S is *unsolvable.* It should be noted that a solvable SPP instance may have more than one stable solution.

Given a egress instance $I = (C, X)$ we now show how to construct an instance, $S_I = (G_I, \mathcal{P}_I, \Lambda_I)$, of the SPP.

Let $G_P = (V, E_P, d)$ be a physical graph, $G_S = (V, A_S)$ be a signaling graph and $X \subseteq V$ be a set of egress nodes. Then for egress instance $I = (C, X)$ define the SPP instance $S_I = (G_I, \mathcal{P}_I, \Lambda_I)$ as follows. Let $G_I = (V \cup \{0\}, E)$ where there is an edge $\{0, u\} \in E$ if and only if $u \in X$ and there are edges $\{u, v\} \in E$, $u, v \neq 0$, if and only if there are arcs (u, v) and (v, u) in A_S. At node u, the set of permitted paths consists of the empty path ε and all paths $P0$ where P^r is a valid signaling path in G_S from some egress node to u. The set of permitted paths at a node u is ranked so that $\lambda^u(Pv0) < \lambda^u(Qw0)$ if $M(u, v) > M(u, w)$. The empty path is always the lowest ranked path. There are any number of ways of breaking ties. We assume the following. Each node $x_i \in X$ is assigned some distinct value $\mathbf{oid}(x_i)$. The value $\mathbf{oid}(x_i)$ is meant to represent the **oid** of x_i. Also each node u assigns some distinct number $id^u(v)$ to each node v where $(v, u) \in A_S$. This represents the **pid** of v. For path $P0$, define $|\mathbf{cluster\_list}(P0)|$ to be the number of different clusters represented by the nodes along P. If $M(u, v) = M(u, w)$ then we define $\lambda^u(uyPv0) < \lambda^u(uzQw0)$ if

(i) $\mathbf{oid}(w) < \mathbf{oid}(v)$ or

(ii) if $\mathbf{oid}(w) = \mathbf{oid}(v)$ and $|\mathbf{cluster\_list}(uzQw0)| < |\mathbf{cluster\_list}(uyPv0)|$ or

(iii) if $\mathbf{oid}(w) = \mathbf{oid}(v)$, $|\mathbf{cluster\_list}(uzQw0)| = |\mathbf{cluster\_list}(uyPv0)|$ and $id^u(z) < id^u(y)$.

We now consider how difficult it is to compute the SPP instance S_I for a given instance $I = ((G_P, G_S), X)$ where $G_P = (V, E_P, d)$, $G_S = (V, A_S)$ and $X \subseteq V$. Let $|V| = n$. The length of any valid signaling path is then bounded by n. Thus there can be at most $O(n^n)$ valid signaling paths. But in typical networks, the route reflector hierarchy is generally no more than about three. Thus the length of any valid signaling path can be considered bounded by some constant c and hence there are $O(n^c)$ possible valid signaling paths.

Computing the set of all possible valid signaling paths is straightforward. Define S_i to be the set of all length i valid signaling paths. Then S_0 is just the set of all nodes in X. Then iteratively compute S_{i+1} as follows. If $s \in S_i$ then consider the set A of all the arcs in G_S directed out of the last node in s. For $a \in A$, if sa is a valid signaling path then add sa to S_{i+1}. Thus if the maximum length of a valid signaling path is a constant, then the set of all valid signaling paths can be computed in time polynomial in n.

Determining the ranking of the valid signaling paths at the various nodes is easily accomplished after computing shortest paths in the physical graph G_P.

Thus assuming that valid signaling digraphs have length bounded by some constant, the SPP instance S_I can be computed in polynomial time.

5.2 Deterministic and signaling correct egress instances

By earlier examples, it was shown that it really only makes sense to ask if an egress instance is forwarding correct if in fact, the egress instance deterministically settles on a unique routing. Thus we now define more precisely what it means for an egress instance to deterministically result in a unique routing by defining what it means for the corresponding SPP instance to have a unique solution.

Let a configuration instance $I = (C, X)$ correspond to the SPP $S(C, X)$. By correctness of I, we want to formalize something along the lines of "for every $X \subseteq B$, the SPP $S(C, X)$ is correct". But what could we mean by "$S(C, X)$ is correct"? This should at least mean that $S(C, X)$ has a unique solution. However, even when an SPP has a unique solution, there is no guarantee that a BGP-like distributed protocol will converge to it. But how how can we discuss divergence without introducing the protocol itself? We will impose a digraph structure on the set of all path assignments, as in [11], where an arc from π to π' indicates that some set of nodes have changed their best paths as a result of the assingments to their neighbors. Such a digraph will contain regions of vertices such that there is a directed path between any two vertices in the region, and the set is minimal with respect to this property. We will call these regions *minimal steady set.* When such a set contains a single vertex, then that vertex must represent a solution. So we will interpret "$S(C, X)$ is correct" to mean that there is a unique minimal steady set (Theorem 5.2) and that this set contains a unique vertex (Theorem 5.3).

Let $S = (G = (V, E), \mathcal{P}, \Lambda)$ be some instance of SPP. We say that a set of path assignments $\Pi = \{\pi_1, \pi_2, \ldots, \pi_t\}$ is *steady* if for each i and for all subsets $U \subseteq V$ there is some j

such that for all $u \in U$, $\max(u, \text{choices}(u, \pi_i)) = \pi_j(u)$, and for $v \in V - U$, $\pi_i(u) = \pi_j(u)$. Such an assignment is said to be *minimal* if for all $\Pi' \subset \Pi$, Π' is not steady.

If $\Pi = \{\pi_1, \pi_2, \ldots, \pi_t\}$ is a minimal steady set of path assignments and $\pi_i(u) = \pi_1(u)$ for $1 \leq i \leq t$, then u is said to be *fixed* in Π. We write $f(\Pi)$ to indicate the set of all nodes where u is fixed in Π. If all $u \in V$ are fixed in Π then $|\Pi| = 1$ and Π is a stable solution.

An instance of SPP is called a *predictable instance* if it has exactly one minimal steady set of path assignments. If the SPP instance corresponding to an egress instance I is predictable, then we say that I is predictable. If an instance of SPP is predictable and in its one minimal steady set of path assignments all nodes are fixed then we say that it is a *signaling correct instance.* If the SPP instance corresponding to an egress instance I is signaling correct, then we say that I is signaling correct.

5.3 Complexity of determining signaling correctness

We are now prepared to consider the complexity of determining if a configuration is signaling correct. Unfortunately, we show that this problem is NP-hard.

For technical reasons, we now define a structure that is similar to a configuration but we allow some of the border routers to be considered to always be announcing routes into the system. We define a *generalized configuration* to be the triple $\mathcal{G} = (G_P, G_S, F)$ where $G_P = (V, B, E_P, d)$ is a physical graph, $G_S = (V, A_S)$ is a signaling digraph and $F \subseteq B$ is a set of nodes that are always egress nodes. For any subset $W \subseteq B$, let $\mathcal{G}(W)$ be the egress instance $((G_P, G_S), W \cup F)$.

The NP-hardness proofs presented will rely on reductions from 3-SAT, a known NP-complete problem [9].

We begin by considering a generalized configuration that will be a fundamental structure in the various constructions used in the reductions. We use a mixed graph (i.e., a graph with both directed and undirected edges) to more easily illustrate the signaling connectivity of a network. The graph $G_R = (V, E_R)$ is called the *relationship graph.* An undirected edge $\{u, v\} \in E_R$ means that in the corresponding signaling graph there are arcs $(u, v) \in$ **over** and $(v, u) \in$ **over**. An arc $(u, v) \in E_R$ represents the fact that u operates as a route reflector to its client v. Thus the corresponding signaling digraph would have an arc $(u, v) \in$ **down** and an arc $(v, u) \in$ **up**.

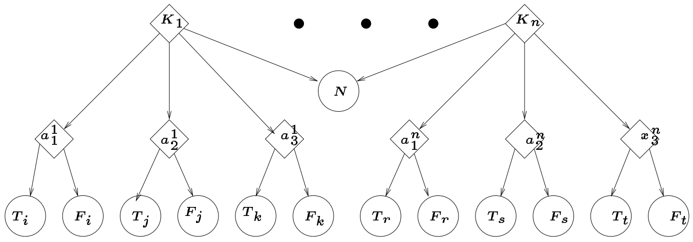


Figure 8: Relationship graph.

Let $I = K_1 \wedge K_2 \wedge \ldots \wedge K_n$ be an instance of 3-SAT where $K_i = a_1^i \vee a_2^i \vee a_3^i$ where each a_j^i is one of x_k or $\overline{x}_k$ for some k, $1 \leq k \leq m$. We define a generalized configuration $\mathcal{B}_I$ based on this instance that will be a basis for the various constructions to follow.

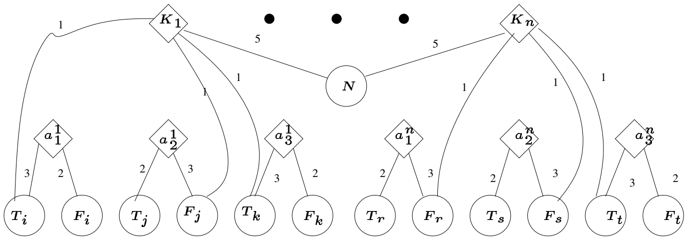


Figure 9: Physical graph.

For 3-SAT instance I, we build a relationship graph and a physical graph as shown in Figure 8 and Figure 9 where by way of example, we suppose that $K_1 = x_i \vee \overline{x}_j \vee x_k$ and $K_n = \overline{x}_r \vee \overline{x}_s \vee x_t$.

Formally, we define

$$\begin{aligned} V &= \{N\} \\ &\cup \ \{K_i : 1 \leq i \leq n\} \\ &\cup \ \{a_j^i : 1 \leq i \leq n, \ 1 \leq j \leq 3\} \\ &\cup \ \{T_i : 1 \leq i \leq m\} \cup \{F_i : 1 \leq i \leq m\}. \end{aligned}$$

Then the arcs of E_R are

$$\begin{aligned} E_R &= \{(K_i, N) : 1 \leq i \leq n\} \\ &\cup \ \{(K_i, a_j^i) : 1 \leq i \leq n, \ 1 \leq j \leq 3\} \\ &\cup \ \{(a_j^i, T_k) : \text{if } a_j^i \text{ is } x_k \text{ or } \overline{x}_k\} \\ &\cup \ \{(a_j^i, F_k) : \text{if } a_j^i \text{ is } x_k \text{ or } \overline{x}_k\}. \end{aligned}$$

The physical graph G_P is as indicated in the example shown in Figure 9. For each K_i there is an edge $\{K_i, N\}$ with distance 5. Consider some clause $K_i = (a_1^i \vee a_2^i \vee a_3^i)$. Suppose a_j^i is x_k for some k. Then there is an edge $\{a_j^i, T_k\}$ with distance 3, an edge $\{a_j^i, F_k\}$ with distance 2 and an edge $\{a_j^i, T_k\}$ of distance 1. If a_j^i is $\overline{x}_k$ for some k then there is an edge $\{a_j^i, F_k\}$ with distance 3, an edge $\{a_j^i, T_k\}$ with distance 2 and an edge $\{a_j^i, F_k\}$ of distance 1.

Let $U = \{F_i, T_i : 1 \leq i \leq m\}$ and $Y = N$. Then $\mathcal{B}_I = ((G_P, G_S), F)$. If I is satisfiable then let α be a truth assignment that satisfies I. Then define W_α to consist of all the T_i such that x_i is true in α and all the F_j such that x_j is false in α.

LEMMA 5.1. *The SPP instance determined by $\mathcal{B}(W)$ is signaling correct for all W and has the following properties:*

- *(i) If I is satisfiable, then the SPP instance defined by $\mathcal{B}(W_\alpha)$ has the property that in its stable solution no node K_i will have N as its egress node.*
- *(ii) If I is not satisfiable, then in the SPP instance defined by $\mathcal{B}(W)$ for any W, there will be some node K_i with N as its egress node in the stable solution.*

For $i = 1, 2, \ldots$, let $G_R^i(I)$ and $G_S^i(I)$ be copies of the relationship graph G_R and implied signaling graph G_S described above. Also, for $i = 1, 2, \ldots$, let $G_P^i(I)$ be a copy of the physical graph G_P described above. If I is satisfied by assignment α then we let U_α^i consist of the T_j^i's and F_k^i's such that T_j and F_k are in U_α. In the i^{th} copy we re-label nodes so that node u becomes node u^i. In the case where only one copy of each such graph is used, the superscript i

will be omitted. We show how various copies of these graphs can be combined to provide lower bounds on a number of problems involving signaling stability and routing stability.

Let $\mathcal{G} = (G_P, G_S, F)$ where $G_P = (V, E_P, d)$ be a generalized configuration consisting of a physical graph $G_P = (V, E_P, d)$, a signaling digraph $G_S = (V, A_S)$ and a set of nodes F that are always egress nodes. Suppose that for all possible subsets $W \subseteq B$, the egress instance $\mathcal{G}(W) = (G_P, G_S, X = F \cup W)$ defines an SPP instance that has exactly one minimal steady set of path assignments. Then we say that such a generalized configuration is *predictable*.

The problem UNIQUE STEADY is defined as follows. An instance of UNIQUE STEADY consists of a generalized configuration $\mathcal{G} = (G_P, G_S, F)$ and the question is whether $\mathcal{G}$ is a predictable generalized configuration.

THEOREM 5.2. UNIQUE STEADY *is NP-hard.*

Proof: From an instance I of 3-SAT we define an instance $D = (G_P, G_S, F)$ of UNIQUE STEADY. The instance D is computable in time polynomial in the size of I.

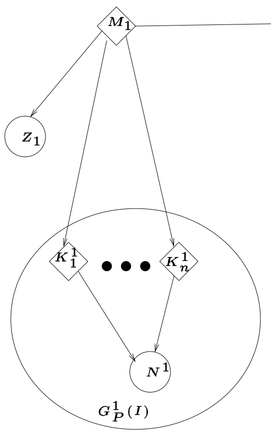


Figure 10: Relationship graph for Theorem 5.2.

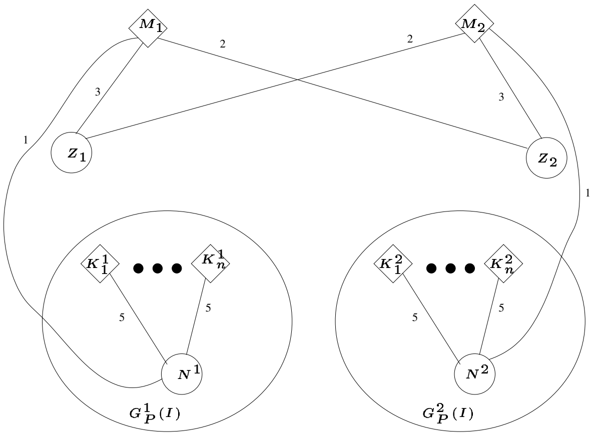


Figure 11: Physical graph for Theorem 5.2.

The physical graph G_P and signaling graph G_S are as indicated in Figure 10 and Figure 11 respectively.

Then we define F and B as $F = \{N^1, N^2\}$ and

$$\begin{aligned} B &= \{Z_1, Z_2, N^1, N^2\} \\ &\cup \{T_i^1, T_i^2 : 1 \leq i \leq m\} \\ &\cup \{F_i^1, F_i^2 : 1 \leq i \leq m\}. \end{aligned}$$

If I is satisfiable then take $W = W_\alpha^1 \cup W_\alpha^2 \cup \{Z_1, Z_2\}$. Then by Lemma 5.1 we know that no K_j^i will have N^i as an egress node in a solution where K_j^i is stable. Then M_i will not have N^i as a choice. But then its easy to see that the resulting egress instance will be equivalent to that shown in Figure 3, thus having two stable solutions and hence two minimal steady sets of path assignments.

If there is no satisfying assignment for I then by Lemma 5.1, N_i will be the egress node of some K_j^i for any W and so M^i will also have to have N^i as its egress node and this represents the only minimal steady set of path assignments. ■

Thus it will not be simple to check if a generalized configuration is predictable, that is, if any of the possible sets of egress nodes will result in an SPP instance that has a unique minimal steady set of path assignments. So we now assume that we are given a predictable generalized configuration. Let $\mathcal{G} = (G_P, G_S, F)$ where $G_P = (V, E_P, d)$ be a predictable generalized configuration consisting of a physical graph $G_P = (V, E_P, d)$, a signaling digraph $G_S = (V, A_S)$, a set of nodes F that are always egress nodes. Suppose that for all possible subsets $W \subseteq B$, the egress instance $\mathcal{G}(W) = (G_P, G_S, X = F \cup W)$ defines an SPP instance whose one minimal steady set of path assignments is signaling correct. Then we say that such a generalized configuration is *signaling correct.* We then ask the question if its possible to easily determine if a predictable generalized configuration is signaling correct.

The problem SIGNALING CORRECT is defined as follows. An instance of SIGNALING CORRECT consists of a predictable generalized configuration $\mathcal{G}$. For such an instance of SIGNALING CORRECT the question asked is whether $\mathcal{G}$ is signaling correct.

THEOREM 5.3. SIGNALING CORRECT *is NP-hard.*

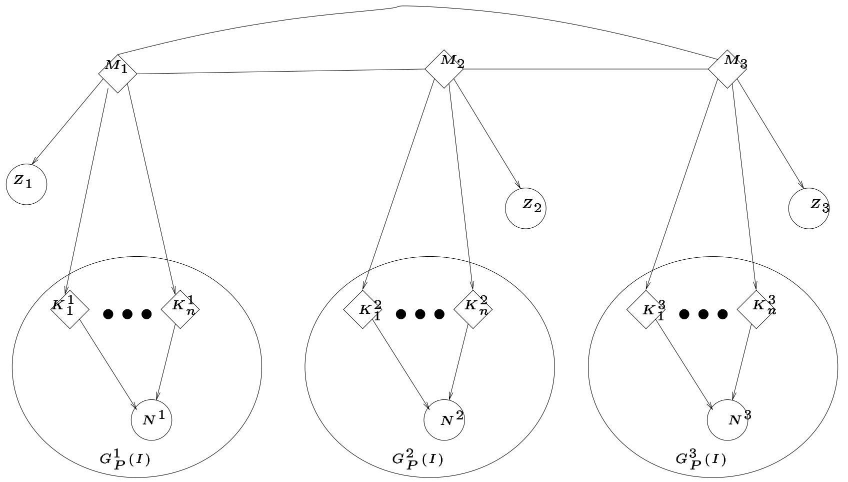
Proof: From an instance I of 3-SAT we define an instance $D = (G_P, G_S, F)$ of SIGNALING CORRECT.

The physical graph G_P and signaling graph G_S are as indicated in Figure 5.3 (a) and (b) respectively.

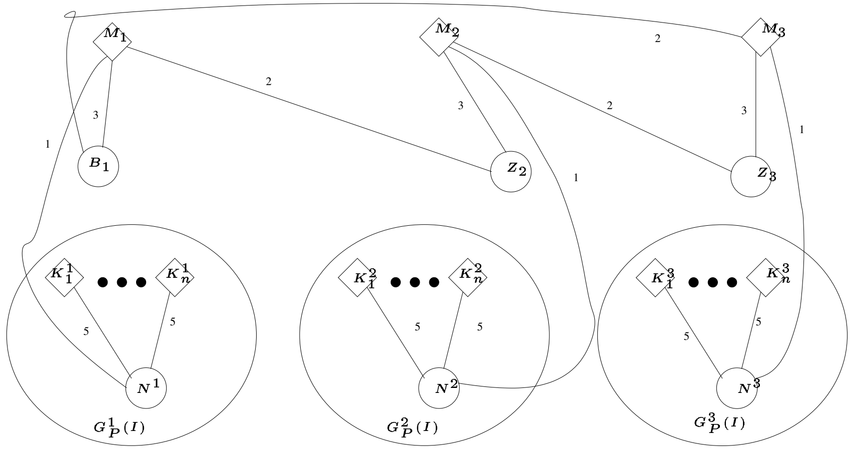
Then we define F and B as $F = \{N^1, N^2, N^3\}$ and

$$\begin{aligned} B &= \{Z_1, Z_2, Z_3, N^1, N^2, N^3\} \\ &\cup \{T_i^1, T_i^2, T_i^3 : 1 \leq i \leq m\} \\ &\cup \{F_i^1, F_i^2, F_i^3 : 1 \leq i \leq m\}. \end{aligned}$$

The arguments are analogous to those in the proof of Theorem 5.2 in showing that there is a W (containing Z_1, Z_2 and Z_3) such that each M_i will not have N^i as an egress node when M_i is stable if there is a satisfying assignment for I. But then the resulting egress instance will be equivalent to that shown in Figure 1, thus have no stable solutions. So if there is no satisfying assignment for I then if M_i is stable then it will have N^i as its egress node for any W in which case its one set of minimal steady path assignments will consist of exactly one path assignment and it will be stable. ■



(a) Relationship graph for Theorem 5.3.



(b) Physical graph for Theorem 5.3.

5.4 A sufficient condition for signaling correctness

We have shown that it will be difficult to determine if a predictable generalized configuration is signaling correct. However, we can show a sufficient condition for guaranteeing that a generalized configuration is signaling correct. The condition follows from the customer-provider, peer-to-peer model defined in [8]. This condition has been shown to guarantee signaling correctness by showing that a *dispute wheel* (a structure derived from the formal SPP model, that represents a circular sequence of router configuration incompatibilities) cannot exist [7]. The result follows from the fact that the absence of dispute wheels implies signaling correctness [12]. The customer-provider, peer-peer model says that for each AS u, and each AS v that u has a BGP session with, v is in exactly one of three sets $customer(u)$, $provider(u)$ or $peer(u)$ depending on the established relationship between u and v. These sets should be such that if $v \in customer(u)$ then $u \in provider(v)$ and if $v \in peer(u)$ then $u \in peer(v)$. Then export rules are determined by these relationships as follows:

(i) u exports only its routes and those of its customers to $v \in provider$,

(ii) u exports all routes to $v \in customer$ and

(iii) u exports its routes and those of its customers to $v \in peer$.

Define the directed graph D with a node for each AS and an arc (u, v) if and only if $v \in customer(u)$. Then it was shown that if D is a directed acyclic graph (DAG) and each AS prefers customer routes over peer and provider routes then the system would be guaranteed to converge to a stable route assignment.

But it is easily seen that saying that having an arc $(u, v) \in$ **up** is equivalent to having $v \in provider(u)$, having an arc $(u, v) \in$ **down** is equivalent to having $v \in customer(u)$ and having an arc $(u, v) \in$ **over** is equivalent to having $v \in peer(u)$. Thus if the directed graph consisting of all arcs $(u, v) \in$ **down** is a DAG and if all nodes u rank exit routes they hear from clients over exit routes they hear from non-clients then the resulting SPP instance will be signaling correct.

Thus this gives a sufficient condition for guaranteeing that an SPP instance is signaling correct. However, notice that if this sufficient condition holds for the case where all possible exit routes are present, then it certainly holds for whichever subset of possible exit routes are present. Thus it provides a sufficient condition for guaranteeing that a generalized configuration is signaling correct.

6. RESULTS ON FORWARDING CORRECTNESS

We now consider the problem of determining if a configuration that is known to be signaling correct is in fact, forwarding correct. In Section 6.1 we show that determining if a configuration is forwarding correct is NP-hard. For this reason, we would like to define sufficient conditions that guarantee that a configuration is forwarding correct. Before doing so in Section 6.3, we first discuss in Section 6.2 some of the difficulties with defining reasonable sufficient conditions.

6.1 Complexity of forwarding correctness

Suppose we are given a signaling correct generalized configuration (perhaps one guaranteed signaling correct by the conditions discussed in Section 5.4). Then we would like to know if for every possible set of egress paths, does the resulting SPP instance have the property that its one minimal steady set of path assignments results in no deflections. We say that such a signaling correct generalized configuration is *correct*. Unfortunately, this problem also turns out to be hard.

We define the problem DEFLECTION as follows. An instance of DEFLECTION consists of a signaling correct generalized configuration $\mathcal{G}$. For such an instance of DEFLECTION the question asked is whether $\mathcal{G}$ is correct.

THEOREM 6.1. DEFLECTION *is NP-hard.*

Proof: From an instance I of 3-SAT, we define an instance $D = (G_P, G_S, Y, U)$ of DEFLECTION.

Let $I = K_1 \wedge K_2 \wedge \ldots \wedge K_n$ be an instance of 3-SAT where $K_i = a^i_1 \vee a^i_2 \vee a^i_3$ where each a^i_j is one of x_k or $\overline{x}_k$ for some k, $1 \leq k \leq m$.

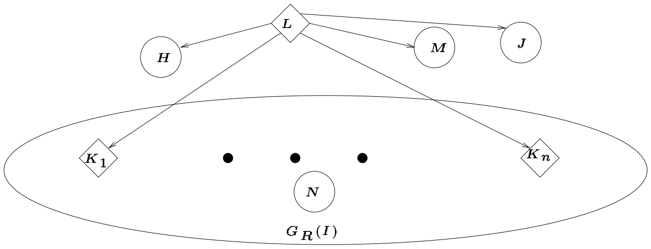


Figure 12: Relationship graph for Theorem 6.1.

We build a signaling graph $G_S = (V, A_S)$ as indicated by the relationship graph shown in Figure 12. This consists of $G_R(I)$ along with the additional vertices and arcs shown. A physical graph $G_P = (V, E_P, d)$ is also constructed as shown in Figure 13. Again this consists of $G_P(I)$ plus some additional vertices and weighted edges as shown in Figure 13.

Then we define Y and U as $Y = \{N\}$ and

$$U = \{H, J\} \cup \{T_i : 1 \leq i \leq m\} \cup \{F_i : 1 \leq i \leq m\}.$$

We first show that if there is a satisfying assignment for I then there is some $W \subseteq U$ such that taking $X = \{N\} \cup W$ as the set of egress nodes results in a SPP instance whose one stable solution has a deflection. Also, it is shown that if I is unsatisfiable then for every $W \subseteq U$, the one stable solution of the resulting correct SPP instance has no deflections.

Suppose there is a satisfying assignment α. Let $U_\alpha \subset W$. Also $J \in W$ and $H \in W$. By the proof of Lemma 5.1, when L is stable it will never have N as its egress node since no K_i does. Thus L will have H as its egress node. Of course, J will always have itself as its egress node when it is stable. The deflection occurs if L has H as its egress node since then M must then also have H as its egress node. But then M's shortest path to H goes through J and so there will be a deflection there.

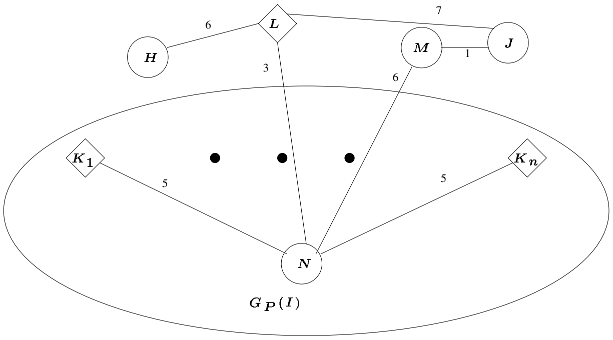


Figure 13: Physical graph for Theorem 6.1.

Now suppose there is no satisfying assignment for I. We know from the proof of Lemma 5.1 that for any choice of egress nodes from amongst the T_i's and F_i's, some K_j will have N as its egress path when K_j is stable. If any K_i has N as an egress node then L must also have N as its egress node when L is stable. But then M must also have N as its egress node. But M's shortest path to N is direct and so no deflection at J occurs. It does not matter if either or both of H and J are in X because either they will have their own egress path if it is an egress node or they will have N as their egress node but it easy to check that their shortest paths go through L or M and both of these have N as egress node as well. ■

6.2 Towards conditions for forwarding correctness

Given a signaling correct configuration, we would like to be able to determine if it is in fact, a forwarding correct configuration and hence a correct configuration. Since this is NP-hard, perhaps we can at least provide sufficient conditions that guarantee correctness. Before doing so, we first look at a few seemingly simple signaling correct egress instances that give rise to a simple deflection or a forwarding loop.

The system in Figure 14 shows an egress instance with one router R_1 acting as a route reflector for two clients C_1 and C_2. The only signaling edges are between R_1 and C_i, $i = 1, 2$. The result is that R_1 always choose P (an EBGP route), and so C_1 only learns of P and hence chooses it as well. But C_2 chooses Q and C_2 is on the forwarding path from C_1 to P. This example says that *simple deflections*, those that do not not cause forwarding loops but have a "short cut" via an unexpected egress path along a forwarding route, can occur in very innocent looking egress instances. Notice that in this example only typical IBGP sessions between clients and their route reflector are allowed and the physical graph is a complete mesh. It is not clear what is "wrong" with this egress instance, and this suggests that simple deflections, while not desirable since they can complicate debugging, may be very difficult to eliminate.

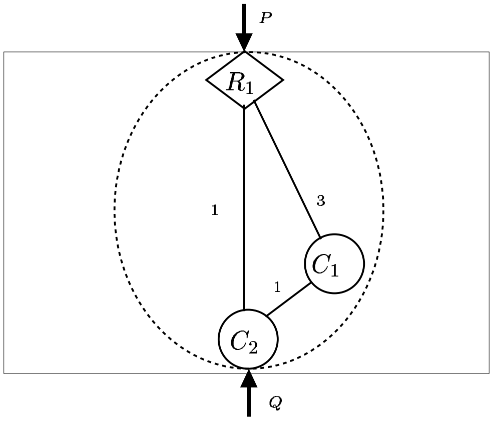


Figure 14: Simple deflection in a trivial system.

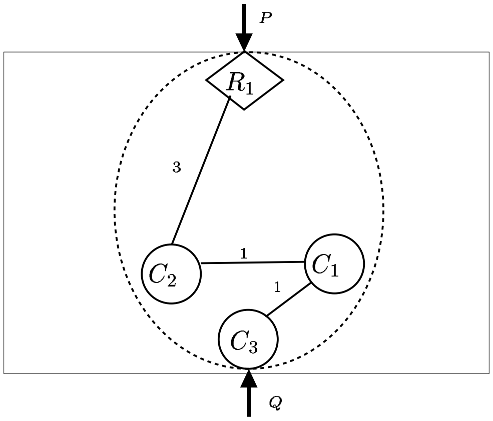


Figure 15: Trivial system that has a forwarding loop.

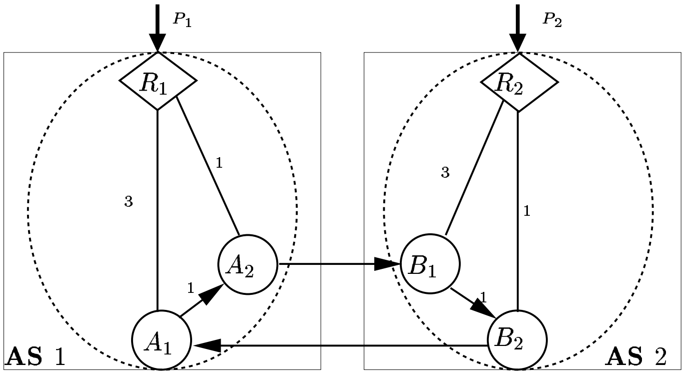


Figure 16: External forwarding loops caused by internal deflections.

Figure 15 is a fairly simple example that shows that forwarding loops can occur in egress instances that allow more IBGP sessions than would typically be defined. The IBGP sessions are the typical ones between the one route reflector R_1 and each of the clients C_i. There is one additional IBGP session between C_2 and C_3. The physical links are as shown in the figure. This results in C_1 choosing P because it never hears about Q since R_1 chooses P. But because of that extra IBGP session between C_2 and C_3, C_2 does hear about Q and chooses it thus setting up a forwarding loop between C_1 and C_2.

In fact, the presence of simple deflections can actually be the source of forwarding loops *between autonomous systems*, as opposed to the forwarding loops contained within a single autonomous system. Such an example is illustrated in Figure 16. In this network, there are two autonomous systems labeled 1 and 2 trying to set up routes to the origin. Only the physical links and the costs of the internal links are shown. Both AS 1 and 2 have turned off testing the length of the **as_path** (or they are member ASes in a confederation). Router R_1 acts as a route reflector to clients A_1 and A_2 and these IBGP sessions between R_1 and its clients are the only IBGP sessions within AS 1. Similarly, R_2 is a route reflector for clients B_1 and B_2 and there is no IBGP session between B_1 and B_2. Imagine R_2 learns of the route P_2 and announces it to its clients. Then B_1 selects that route and announces it to A_2 via EBGP. Then A_2 will select route corresponding to (router level signaling) path A_2 B_1 R_2 P_2. Meanwhile, R_1 learns of the route P_1 and announces it to its clients. This results in A_1 selecting P_1 since its the only route it hears about, while A_2 selects the route with path A_2 B_1 R_2 P_2 since it was learned via EBGP. However, A_1's shortest path to R_1 goes through A_2 and this results in a deflection. Now A_1 announces via EBGP the route corresponding to the path A_1 R_1 P_1 to B_2 and so B_2 will choose route that route. This causes the packets sent by B_1 to R_2 to get deflected at B_2 since it lies along the shortest path from B_1 to R_2. Thus there is a forwarding loop B_2, A_1 A_2 B_1 B_2 $A_1 \cdots$, indicated in the figure by directed links.

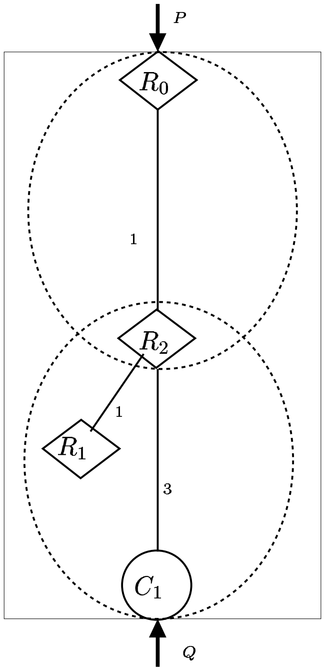


Figure 17: Favoring a route other than a client route.

When the forwarding path is not a valid signaling path, it is possible to get deflections. In Figure 17, router R_0 is a route reflector for client router R_2 where R_2 in turn acts as a route reflector, along with a second route reflector R_1, in a "lower level" cluster containing client C_1. The path $C_1R_2R_1$ is the shortest path between C_1 and R_1 and it also a valid signaling path from C_1 to R_1. But R_2 favors R_0 as egress node over C_1 by virtue of the shorter path thus not satisfying the constraint that client routes should be favored over other routes. Since $R_0R_2R_1$ is not a valid signaling path, R_1 will favor egress node C_1 by default. But this causes a deflection at R_2.

6.3 Sufficient conditions for forwarding correctness

Given that it is NP-hard to determine if deflections can occur, we now wish to give sufficient conditions that guarantee that a stable solution will not give rise to any deflections (or forwarding loops). That is, if we have a signaling correct generalized configuration satisfying these conditions, then in fact, we have a correct generalized configuration.

The conditions we require are:

(*i*) if $(v, u) \in$ **up** and $(w, u) \notin$ **up** then it must be that $\lambda^u(uwP^r0) < \lambda^u(uvQ^r0)$ for any valid signaling paths uwP and uvQ and

(*ii*) for any nodes u and v, $sp(u, v) = P$ for some valid signaling path P.

The first condition is motivated by the example in Figure 17. The second condition is motivated by the examples of Figures 14 and 15.

In what follows, we will sometimes abuse notation and write $\lambda^u(v) < \lambda^u(w)$ for nodes $u \neq w$ to mean that the length of the shortest path from u to v is greater than the length of the shortest path from u to w or if the shortest paths are equal then $\mathbf{oid}(v) > \mathbf{oid}(w)$. Thus u would rank any permitted path having egress node w over any permitted path having egress node v.

Consider the stable solution π to some given generalized configuration. Let $\mathcal{E}(u)$ denote the egress node chosen by node u corresponding to $\pi(u)$.

LEMMA 6.2. *Let P be a shortest path from u to v and let w be some node on P. If $\lambda^w(v) < \lambda^w(\mathcal{E}(w))$ then $\lambda^u(v) < \lambda^u(\mathcal{E}(w))$.*

Proof: Either the length of a shortest path from w to $\mathcal{E}(w)$ is shorter than the length of a shortest path from w to v, in which case the length of a shortest path from u to $\mathcal{E}(w)$ is also shorter than from that from u to v, or else the lengths are equal and $\mathbf{oid}(\mathcal{E}(w)) < \mathbf{oid}(v)$. Thus, in either case, $\lambda^u(v) < \lambda^u(\mathcal{E}(w))$. ■

LEMMA 6.3. *Suppose condition (i) holds. Let $v = \mathcal{E}(u) = \mathcal{E}(v)$ and suppose $sp(v, u)$ is a signaling path from v to u. If $w \in sp(v, u)$ then either $\mathcal{E}(w) = v$ or $\lambda^w(\mathcal{E}(w)) > \lambda^w(v)$.*

Proof: The proof is by induction on the number of edges in $sp(v, w)$. The lemma is trivially true if $sp(v, w)$ has 0 edges (that is, if $w = v$).

Let $sp(v, u) = v_0v_1 \ldots v_k$ where $u = v_k$ and $v = v_0$. Suppose the claim is true for any v_i where $0 \leq i < j$.

Now consider v_j. If $\mathcal{E}(v_j) = v$ then we are done. Otherwise suppose $(u_{j-1}, u_j) \in$ **up**. Then by condition (*i*), it must be that there is a valid signaling path S from $\mathcal{E}(u_j)$ to u_j consisting solely of arcs in **up**. But then the path Sv_j is also a valid signaling path and so

$$\lambda^{u_j}(\mathcal{E}(u_j)) > \lambda^{u_j}(\mathcal{E}(u_{j-1})). \tag{1}$$

By induction,

$$\lambda^{u_{j-1}}(\mathcal{E}(u_{j-1})) > \lambda^{u_{j-1}}(v) \tag{2}$$

and so by Lemma 6.2, Equation (2) implies

$$\lambda^{u_j}(\mathcal{E}(u_{j-1})) > \lambda^{u_j}(v). \tag{3}$$

Thus Inequality (1) and Inequality (3) together imply that $\lambda^{u_j}(\mathcal{E}(u_j)) > \lambda^{u_j}(v)$ as required.

Suppose $(u_{j-1}, u_j) \in$ **down** or $(u_{j-1}, u_j) \in$ **over**. Let Q be any valid signaling path from $\mathcal{E}(u_{j-1})$ to u_{j-1}. Then $(u_{j-1}, u_j) \in$ **down** and so Qu_j is a valid signaling path from $\mathcal{E}(u_{j-1})$ to u_j. Then the same arguments as in the previous case hold. ■

THEOREM 6.4. *If a generalized configuration $\mathcal{G}$ satisfies conditions (i) and (ii) above, then no solution of $\mathcal{G}$ will have any deflections.*

Proof: Let u be some node such that there is a deflection along $P = sp(v, u)$ where $v = \mathcal{E}(u)$ according to some solution π. By condition (*ii*), P must be a signaling path. Let $u_0 = v$, $u_k = u$ and $P = u_0u_1 \ldots u_k$. Let u_i be the node closest to u on P at which $\mathcal{E}(u_i) \neq v$. By Lemma 6.3, $\lambda^{u_i}(\mathcal{E}(u_i)) > \lambda^{u_i}(v)$ and so by Lemma 6.2,

$$\lambda^{u_{i+1}}(\mathcal{E}(u_i)) > \lambda^{u_{i+1}}(v) = \lambda^{u_{i+1}}(\mathcal{E}(u_{i+1})). \tag{4}$$

A similar argument to that in the proof of Lemma 6.3 shows that there is a valid signaling path Q from $\mathcal{E}(u_{i-1})$ to u_{i-1} then Qu_i is also a valid signaling path and so

$$\lambda^{u_i}(v) = \lambda^{u_i}(\mathcal{E}(u_i)) > \lambda^{u_i}(\mathcal{E}(u_{i-1}))$$

contradicting Inequality (4). ■

7. REMARKS

BGP has evolved to become a rather complicated protocol, implementing many optional features designed for scalability and increased policy expressiveness. These features can interact in complex and surprising ways. The need to optimize the implementation of the IBGP full mesh led to the development of BGP route reflection and confederations. On the one hand, these extensions have allowed IBGP configurations to encompass very large transit networks. On the other hand, this has come at the price of increased configuration complexity and the potential for realizing the kinds of routing anomalies considered in this paper.

We have focused on configuration correctness to avoid these anomalies. One could as well redesign the protocol so that they can never occur [1]. Another approach to IBGP redesign might be to make route reflectors smarter — they could calculate best paths for their clients based on the *client's* notion of distance to an egress point.

BGP represents a widely deployed instance of a *vectoring* protocol that allows *local and independent policy control.* Other protocols in the same family as BGP are now emerging, such as protocols for optical inter-networking [3] and Telephony Routing over IP (TRIP) [20]. We believe that the design space of this family of protocols is not well understood. Our work should help protocol designers better understand one aspect of this design space — potential anomalies can arise when the relationships between forwarding and signaling paths are unconstrained.

The anomalies studied here can arise after links or nodes fail in the network. Thus, IBGP configuration is not robust with resect to failures. Note however that the deflection problem vanishes if traffic is *tunneled* to egress points rather than being routed. For example, if MPLS is used to forward traffic within an autonomous system, then there will be no deflection problems.

Acknowledgments

We would like to thank F. Bruce Shepherd for many helpful discussions. Ruomei Gao, Zhuoqing Morley Mao, and Li Xiao provided many helpful comments and corrections.

8. REFERENCES

[1] Anindya Basu, C.-H. Luke Ong, April Rasala, F. Bruce Shepherd, and Gordon Wilfong. Route oscillations in I-BGP with route reflection. In *Proceedings of ACM SIGCOMM*, 2002.

[2] T. Bates, R. Chandra, and E. Chen. BGP route reflection: An alternative to full mesh IBGP. RFC 2796, 2000.

[3] Marc Blanchet, Florent Parent, and Bill St-Arnaud. Optical BGP (OBGP): InterAS lightpath provisioning. Internet Draft draft-parent-obgp-01.txt. Work in progress.

[4] Cisco. Endless BGP Convergence Problem in Cisco IOS Software Releases. Field Note, October 10 2001, `http://www.cisco.com/warp/public/770/fn12942.html`.

[5] Rohit Dube. A comparison of scaling techniques for BGP. *ACM Computer Communication Review*, 29(3), 1999.

[6] Rohit Dube and John G. Scudder. BGP scaling techniques revisited. *ACM Computer Communication Review*, 29(5), 1999.

[7] L. Gao, T. G. Griffin, and J. Rexford. Inherently safe backup routing with BGP. In *Proceedings of the 20th IEEE Conference on Computer Communications (Infocom 2001)*, Anchorage, Alaska, April 2001.

[8] Lixin Gao and Jennifer Rexford. Stable internet routing without global coordination. In *Proceedings of ACM SIGMETRICS*, June 2000.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Co., San Francisco, CA, 1979.

[10] T. G. Griffin and G. Wilfong. A safe path vector protocol. In *Proceedings of the 19th Annual IEEE Conference on Computer Communications (Infocom 2000)*, Tel Aviv, Israel, March 2000.

[11] Timothy G. Griffin, F. Bruce Shepherd, and Gordon Wilfong. Policy disputes in path-vector protocols. In *Proceedings of Seventh International Conference on Network Protocols (ICNP'99)*, pages 21–30, 1999.

[12] Timothy G. Griffin, F. Bruce Shepherd, and Gordon Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Transactions on Networking*, 2002.

[13] Timothy G. Griffin and Gordon Wilfong. An analysis of BGP convergence properties. In *Proceedings of ACM SIGCOMM*, pages 277–288, 1999.

[14] Timothy G. Griffin and Gordon Wilfong. Analysis of the MED oscillation problem in BGP. unpublished manuscript, 2002.

[15] S. Halabi and D. McPherson. *Internet Routing Architectures.* Cisco Press, Indianapolis, Indiana, second edition, 2000.

[16] C. Huitema. *Routing in the Internet.* Prentice Hall, 2000.

[17] Ratul Mahajan, David Wetherall, and Tom Anderson. Understanding BGP misconfigurations. In *Proceedings of ACM SIGCOMM*, 2002.

[18] D. McPherson, V. Gill, D. Walton, and A. Retana. BGP persistent route oscillation condition. Internet Draft `draft-ietf-idr-route-oscillation-01.txt` , Work In Progress, 2002.

[19] Y. Rekhter and T. Li. A Border Gateway Protocol. RFC 1771 (BGP version 4), March 1995.

[20] Jonathan Rosenberg, Hussein Salma, and Matt Squire. Telephony routing over IP (TRIP). RFC 3219. January 2002.

[21] P. Traina, D. McPherson, and J. Scudder. Autonomous system confederations for BGP. RFC 3056, 2001.

[22] Kanan Varadhan, Ramesh Govindan, and Deborah Estrin. Persistent route oscillations in inter-domain routing. *Computer Networks*, 32:1–16, 2000.