CS269I: Incentives in Computer Science
Lecture #8: Incentives in BGP Routing*

Tim Roughgarden†

October 19, 2016

1 Routing in the Internet

Last lecture we talked about delay-based (or "selfish") routing, which is common in local area networks. Today's lecture is about a different type of routing, which is used between different local networks.

The Internet is really a "network of networks" (Figure 1). An *autonomous system (AS)* is a centrally controlled collection of routers—a bunch of routers with a common administrator. The Internet has around 42,000 ASes, and they span the gamut—Internet Service Providers (ISPs), universities, businesses, governments, etc. For example, Stanford University is an AS, and so is its Internet Service Provider (Cogent), and these two ASes are directly connected by physical links. There are of course other ASes that Stanford is not directly connected to, and sending traffic to them involves routing over a path with multiple hops.

1.1 Routing Within an AS

Routing within an AS is usually done using shortest-path routing. These routing protocols resemble the shortest-path algorithms that you studied in undergraduate algorithms (the greedy Dijkstra algorithm, and the dynamic programming-based Bellman-Ford algorithm), with many additional details. Shortest-path routing requires a definition of a "shortest path," which in turn requires a definition of "edge length" (in this context, called "link weight"). The simplest example is hop-count (where all link weights are 1), in which case breadth-first search can be used to compute shortest paths. Alternatively, different edges can have different link weights, for example depending on the recent average delay along the link. Such delay-based routing brings us back to the selfish routing model we studied last lecture.

†Department of Computer Science, Stanford University, 474 Gates Building, 353 Serra Mall, Stanford, CA 94305. Email: tim@cs.stanford.edu.

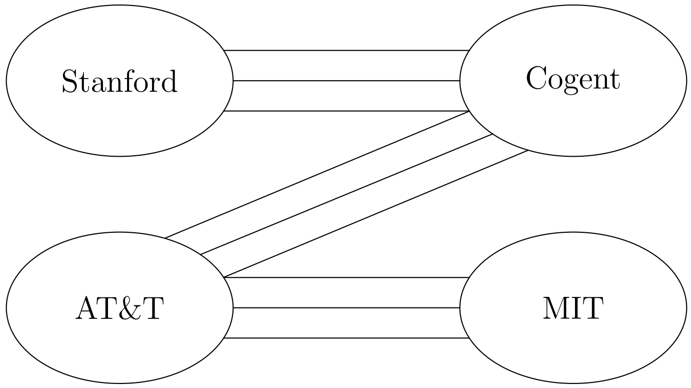

Figure 1: The Internet: a "network of networks."

Note that in shortest path routing, all routers effectively agree on which paths are the best (the shortest ones). These common preferences are dictated by the central administrator, through the choice of link weights.

1.2 Routing Between ASes

Routing between ASes is a different story. There is no central administrator to whom all ASes report (remember these ASes span the globe). Different ASes have different preferences. For example, an AS may prefer paths with the smallest monetary cost to the AS, which is a function of the AS's business agreements with other ASes, and different ASes have different sets of business agreements. For example, ISP#1 may want to route through ISP#2 only as a last resort (if the former has to the pay the latter a high cost). Link weights are insufficient to model such general preferences.

For example, consider the AS network in Figure 2. Assume that the node d is the destination for all traffic. The ASes 1 and 2 are labeled with their favorite path (at the top) and their second-favorite paths. Note that each AS prefers to route to the other one over routing directly to d. (Maybe d charges anyone who sends traffic to it.) These preferences are inconsistent with shortest-path routing (for any choice of link weights).

2 The Border Gateway Protocol (BGP)

In the Internet, routing between ASes is done using the *Border Gateway Protocol (BGP)*. We next explain the most important aspects of how it works (there are of course many more details in the actual protocol).

Fix a destination AS d. BGP runs in parallel for all choices of the destination d, and the computations for different destinations are completely unrelated. (So when one AS sends a message to another AS, it includes a full-blown routing table, with the AS's routing plan for all $\approx 42K$ possible destinations.)

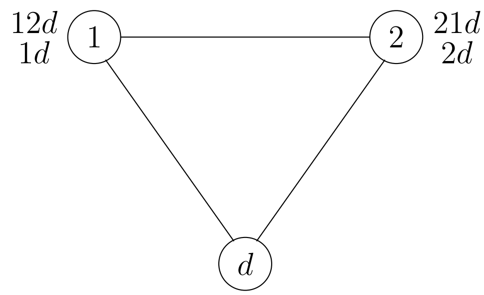

Figure 2: An example of an AS network where each AS prefers to route to the other one over routing directly to d.

Next we describe the intended behavior for ASes in BGP. Later we'll discuss whether ASes have an incentive to deviate from this intended behavior (recall the BitTorrent discussion in Lecture #5).

The protocol resembles the Bellman-Ford shortest-path algorithm, except with the distance-based update step replaced with a more abstract update step. Every AS v will maintain a path P_v to d (or the empty path, if a path from v to d hasn't been found yet). Initially, d sets P_d to the empty path (d can reach itself via the empty path), and broadcasts this fact to all of its neighbors (ASes to which it has a direct physical connection). For example, in Figure 2, the protocol begins with d alerting the other two ASes about its existence.

All of the ASes are then supposed to execute the following procedure in parallel, continuously and asynchronously.[1]

BGP Update Step

At an AS u with neighbors N:

1. For each $v \in N$:
   (a) Let P_v be the last path (from v to d) that v announced to u (if any).
2. Reset P_u to u's favorite cycle-free path of the form $(u, v) \oplus P_v$ (if any, otherwise P_u is the empty path).[2]
3. If P_u changes, announce the new value of P_u to all neighbors $v \in N$.

Note that an AS u has to take care to avoid cycles—if P_v includes u in it, then u cannot route using P_v (traffic would then loop forever).

For example, let's return to the AS graph in Figure 2. What is the ultimate result of BGP? One possible fixed point of the protocol is shown in Figure 3(a), with 1 routing directly

[1]At the Internet scale, you can't really make any assumptions about the timing of when different events occur.

[2]The "$\oplus$" operator denotes path concatenation.

to d and 2 routing through 1. AS 2 obviously doesn't want to update its path, since it's using its favorite path. AS 1 isn't thrilled about routing directly to d, but it has no other choice: it can't switch to routing via 2, since this would create a cycle. This is not the only possible fixed point of BGP in the AS graph in Figure 2; Figure 3(b) shows another (symmetric) one. Which fixed point do we expect BGP to reach? It depends on the timing of the messages—whichever of 1, 2 finds out first that the other is using a direct path to d can switch paths and "win".

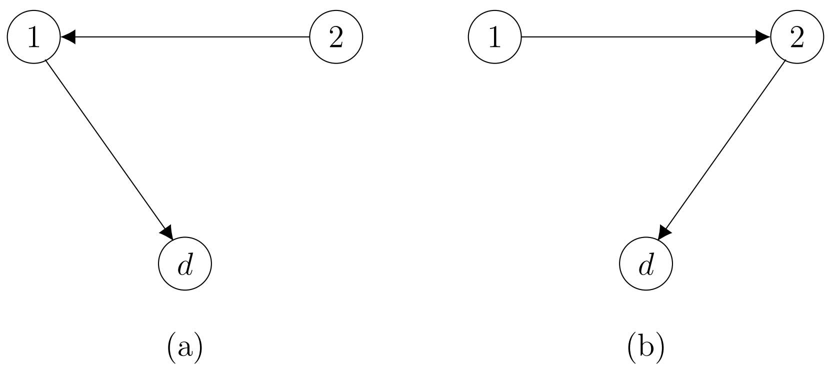

Figure 3: Two stable routings in the AS graph of Figure 2.

3 Stable Routings

We call a fixed point of BGP a *stable routing*. That is, in a stable routing, no AS wants to change its path to d, given the path choices of other ASes and the corresponding options available to u.[3] Thus Figure 3 shows two stable routings in the AS graph of Figure 2.

A stable routing must be a tree, directed into the destination d. BGP maintains the invariant that the out-degree of every AS is at most 1. It also explicitly prevents cycles from forming. The only possible sink (other than isolated ASes) is d. If the AS graph is not connected, or if some ASes prefer the empty path to some of its paths to d, then the tree need not span all of the ASes.

We've already seen that an AS graph can have more than one stable routing. The next example shows that there might be no stable routing at all. Consider a hypothetical stable routing in the network shown in Figure 4. The resulting tree must include at least one edge between d and another AS, say AS 1. (Otherwise all ASes are isolated, but every AS prefers the direct path to d over the empty path). In any stable routing that includes the link from 1 to d, AS 3 must be using the path $31d$ (since this path is its favorite). Given this, AS 2 must use the path $2d$, since its preferred path $23d$ is not available. But if AS 2 uses the path $2d$,

[3]This should remind you of the Nash equilibrium concept mentioned in Lectures #5 and #6. We can think of the ASes as the players, the neighboring ASes of an AS u as u's available strategies, and with payoffs induced by an AS's abstract preferences over paths.

then AS 1 wants to switch to the path $12d$. We conclude that there cannot be any stable routing.

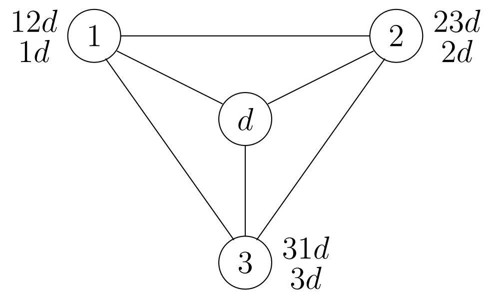

Figure 4: An example of an AS network with no stable routing.

4 Dispute Wheels

Ultimately, we want to discuss incentives in BGP—do ASes want to follow the intended behavior of the BGP protocol, or are there opportunities to game the system? But the examples in Figures 2 and 4 show that "the output of BGP" is not even well defined in some cases. In this section we identify conditions on an AS graph under which "the outcome of BGP" makes sense, and then in Section 6 we address incentive issues.

4.1 No Dispute Wheel Implies BGP Convergence

The following theorem identifies a condition ("no dispute wheel," which we'll define shortly) under which BGP has several nice properties.

Theorem 4.1 ([3]) *Suppose an AS graph has no dispute wheel. Then:*

(a) There exists a stable routing.

(b) This stable routing is unique.

(c) BGP is guaranteed to converge to the stable routing.

Note that part (c) of the theorem is strictly stronger than (a). BGP can only converge to a stable routing, so (c) implies (a). But (a) does not imply (c) in general: there are AS graphs with a unique stable routing in which BGP can cycle forever. (In light of Theorem 4.1, we know that such examples must have a dispute wheel.)

The missing definition is as follows. A *dispute wheel* consists of (for some $k \geq 2$):

1. distinct ASes $u_1, \ldots, u_k$;

2. (cycle-free) paths $P_1, \ldots, P_k$, where P_i is a path from u_i to d; and

3. (cycle-free) paths $Q_1, \ldots, Q_k$, where Q_i is a path from u_i to u_{i+1},

such that, for each $i = 1, 2, \ldots, k$, u_i prefers the "indirect path" $Q_i \oplus P_{i+1}$ to the "direct path" P_i. (Where by u_{k+1} and P_{k+1} we mean u_1 and P_1.) See Figure 5. The figure is somewhat misleading, as the P_i's and Q_i's need not be internally disjoint. For example, a vertex u_i is allowed to appear in the middle of all of the paths P_j for $j \neq i$.

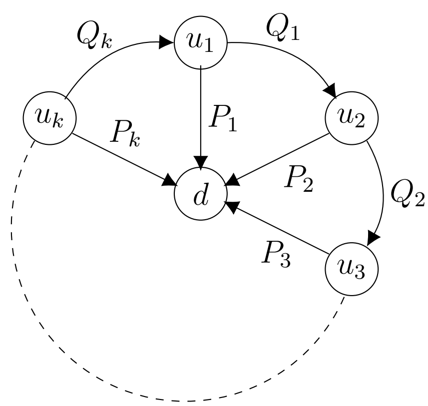

Figure 5: A dispute wheel. The P_i's and Q_i's need not be internally disjoint.

This definition generalizes the two examples we've seen thus far. The AS graph in Figure 2 is a dispute wheel, as seen by taking P_1 as the path $1d$, P_2 the path $2d$, Q_1 the path 12, and Q_2 the path 21. (Recall each AS prefers to route to the other over routing directly to d.) The AS graph in Figure 4 is also a dispute wheel: take $P_i = id$ for $i = 1, 2, 3$, $Q_1 = 12$, $Q_2 = 23$, and $Q_3 = 31$.

4.2 Are There Dispute Wheels in Practice?

When someone subjects you to a new definition, especially a weird one like the one above, you should demand two things: interesting consequences of satisfying the definition, and interesting examples that satisfy the definition. Theorem 4.1 takes care of the first part—AS graphs without a dispute wheel have many desirable properties. But how strong is the "no dispute wheel" condition?

For a "sanity check" example that is not very interesting in its own right, suppose all ASes prefer shorter paths to longer ones, with respect to some set of nonnegative link weights (with all ASes using the same link weights). Then, it's not hard to prove that the corresponding AS graph has no dispute wheel (see Exercise Set #4). But this is not very satisfying—the whole point of BGP is to move beyond shortest-path routing.

Gao and Rexford [1] gave a much more satisfying justification of the "no dispute wheel" condition, and argued that the condition should generally hold for realistic AS preferences.

Formally, they proposed what are now known as the "Gao-Rexford conditions" on an AS graph, and proved that these conditions rule out any dispute wheels.

Studying the details of the Gao-Rexford conditions would take us too far afield, but here's the idea. To first order, pairs of ASes are in one of two possible relationships. They might be *peers*, who agree to carry traffic to and from each other. (E.g., all pairs of Tier 1 ISPs are peers.) Otherwise, one is the *provider* and the other the *customer*, with the customer paying the provider for connectivity to other parts of the Internet. For example, most Tier 2 ISPs have to pay one or more Tier 1 ISPs to be able to reach the entire Internet. One of the Gao-Rexford conditions is that every AS always prefers routes through a customer over those through a peer, and those through a peer over those through a provider. This hierarchical structure of ASes prevents dispute wheels.[4] There is anecdotal evidence that the Gao-Rexford conditions approximately hold for most ASes [2].

5 Proof of Theorem 4.1

We now prove part(b) of Theorem 4.1. This will be the only proof of the lecture, and it is chosen to be representative of the types of arguments used in the other proofs.[5] In particular, we'll see why dispute wheels naturally come up in such arguments.

We'll prove the contrapositive. Suppose an AS graph has two different stable solutions, S and T. (Both are trees, directed into d.) The plan is to exhibit a dispute wheel.

As a thought experiment, imagine doing breadth-first search backward from d along edges in both S and T, and let H be the vertices reached. That is, define H as the subset of vertices that use the same path to d in both of the stable solutions. See Figure 6. If nothing else, H includes the destination AS d.

We claim that there is an edge (u_1, v_1) such that: (i) $u_1 \notin H$; (ii) $v_1 \in H$; and (iii) (u_1, v_1) is in either S or T. Note that such an edge cannot be in both S and T (since v_1 is reachable backward from d by edges of $S \cap T$, if (u_1, v_1) were also in $S \cap T$, then u_1 would also be in H, which it isn't).

We prove the claim by contradiction. If not, then no edges of S or T cross the boundary of H (i.e., have one endpoint in H and the other outside H). This means that in both S and T, the vertices that have a path to d are precisely H. (If some vertex outside H has a path to d in S or T, then the path would have to cross the boundary of H at some point.) But S and T are identical inside H (by definition), so this would mean that $S = T$ (a contradiction).

So pick an edge (u_1, v_1) as in the claim. Say the edge is in S but not T (the case where it's in T but not S is symmetric). Since $(u_1, v_1) \in S$, the path u_1 uses to reach d in S has (u_1, v_1) as the first hop, followed by whatever path P_{v_1} uses to reach d in S. Denote the path $(u_1, v_1) \oplus P_{v_1}$ by P_1.

[4] Another important condition is that ASes refuse to carry transit traffic (i.e., traffic that both originates from and is destined for other ASes), except from customers. This refusal is implemented by announcing non-empty routes only to your customers. The final condition is uncontroversial: there should be no cycle of provider-customer relationships.

[5] You will prove part (a) in Exercise Set #4.

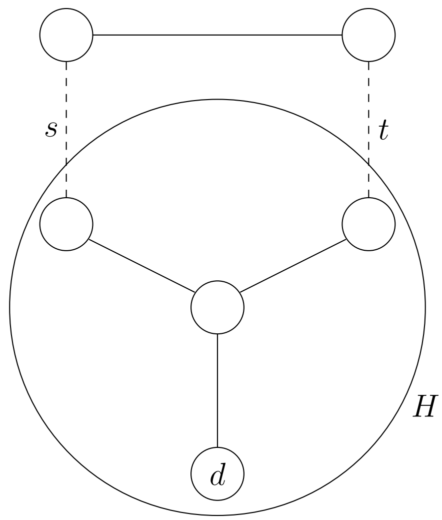

Figure 6: The set H in the proof of Theorem 4.1. The solid edges belong to both S and T. The dashed edges s and t belong to only S and T, respectively.

Now switch to the other stable solution, T. Since S and T are identical in H, and $v_1 \in H$, P_{v_1} is also in T. Thus, the u_1-d path P_1 is an available option to u_1 in T. Since $(u_1, v_1) \notin T$, u_1 does not use the path P_1 in T, and instead uses some other path P'. P' cannot be the empty path: if u_1 preferred the empty path over P_1, it would have also chosen the empty path in the stable solution S (instead of P_1). Similarly, the first hop of P' cannot have the form (u_1, w) for some $w \in H$; if u_1 preferred $(u_1, w) \oplus P_w$ over P_1 in T, then it would do so in S as well (since $w \in H$, P_w is available in S). So, P' must have the form shown in Figure 7: a nonempty path from u_1 to some other vertex $u_2 \notin H$, followed by an edge (u_2, v_2) crossing the boundary of H, followed by the path P_{v_2} that v_2 uses to get to d (in both S and T). Define Q_1 as the first part of the path (traveling from u_1 to u_2), and P_2 as the second part (from u_2 to d). By definition, u_1 prefers the path $P' = Q_1 \oplus P_2$ to P_1.

We now repeat the argument starting from u_2. The edge (u_2, v_2) lies in T (because u_2 chooses the path P_2 in T), and since it crosses the boundary of H, it cannot also lie in S (see the argument above). Since $v_2 \in H$ and S, T agree inside H, the path P_2 was one of u_2's available options in S. Since $(u_2, v_2) \notin S$, in S, u_2 took some other path to d, necessarily of the form shown in Figure 7. This path has the form $Q_2 \oplus P_3$, for a path Q_2 from u_2 to some other vertex u_3 (different from u_2, possibly the same as u_1), and for a path P_3 from u_3 to d, with the first hop of P_3 crossing the boundary of H.

We can iterate this argument as many times as we want, to define u_i's, P_i's, and Q_i's. At some point, a u_i will repeat (e.g., say $u_1 = u_{k+1}$). We claim that the vertices $u_1, \ldots, u_k$ with the paths $P_1, \ldots, P_k$ and $Q_1, \ldots, Q_k$ form a dispute wheel. The proof just checks all the conditions: (i) the u_i's are distinct, and $k \geq 2$; (ii) each path P_i is from u_i to d; (iii) each path Q_i is from u_i to u_{i+1} (with path P_k from u_k to u_1); and (iv) each u_i prefers $Q_i \oplus P_{i+1}$ to P_i (by construction). This completes the proof.

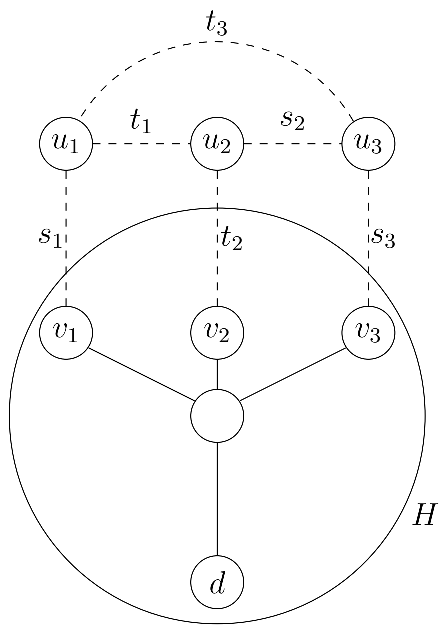

Figure 7: Illustration for the proof of Theorem 4.1. The solid edges belong to both S and T. The dashed edges s_1, s_2, s_3 and t_1, t_2, t_3 belong to only S and T, respectively. Here, $P_1 = s_1 \oplus P_{v_1}$, $Q_1 = t_1$, $P_2 = t_2 \oplus P_{v_2}$, $P_2 = t_2 \oplus P_{v_2}$, $Q_2 = s_2$, $P_3 = s_3 \oplus P_{v_3}$, and $Q_3 = t_3$.

6 Incentive Issues

We now focus on incentive issues in AS graphs that have no dispute wheel (where the "result of BGP" is well defined). Can ASes game BGP? Is there a beneficial unilateral deviation from the intended behavior described in Section 2?

6.1 Types of Deviations

There are several ways an AS could deviate from BGP. Here are three types of deviations:

1. Choose your path P_u to be something other than your favorite among the available options.

2. Withhold information about your path from (some of) your neighbors.

3. Announce a path to (some of) your neighbors that is different from the one you're actually using, possibly a path that doesn't even exist in the network.

It's clear that an AS is in a position to execute the first two types of deviations. What about the third type? Can an AS really get away with announcing fake paths to its neighbors?

It may surprise you that fake path announcements are happening in BGP *all the time*. One of the more famous examples was in February of 2008. Pakistan Telecom wanted to block access from Pakistan to YouTube, because of some offensive videos. They implemented this by rerouting YouTube traffic to a local server (just a Web page saying "access blocked."). For whatever reason (accidentally?), Pakistan Telecom announced this new route to YouTube to some of its neighbors. An ISP in Hong Kong switched its route to YouTube in seconds, and much of the Internet quickly followed suit.[6] The results were not good for anybody: wide swaths of the Internet could not reach YouTube, and Pakistan Telecom was buried under all the requests that were being directed to it.[7]

The vulnerability of BGP to such mistakes and attacks is obviously a problem, and for many years there have been ongoing efforts to roll out a new and more secure version of BGP, such as the proposed BGPsec protocol [4]. As you can imagine, with 42,000 ASes and no central authority, it's not easy to deploy major changes to the protocol.

6.2 Examples

First we note that the example in Figure 2, where there are two stable routings, already shows that withholding information about your path can be beneficial. Recall that each of the 2 ASes wants to route through the other rather than directly to the destination. If AS 1 never announces any paths and AS 2 follows the BGP protocol, then the protocol converges to the stable routing that AS 1 prefers (with 2 routing directly to d). But this example is a dispute wheel, and we already know that BGP doesn't function well in the presence of dispute wheels. So what if there are no dispute wheels?

To see that announcing fake paths can be beneficial, consider the AS graph in Figure 8. It is similar to the AS graph in Figure 4, except there is no direct physical connection between AS 3 and the destination. If the edge $3d$ were in the network and AS 3 preferred this path less than $31d$, there would be a dispute wheel. With the edge missing, however, there is no dispute wheel (as you're invited to check).

It may look weird that AS 2 is stating a preference for the non-existent path $23d$. But remember that there are 42,000 ASes in the Internet, and they are not always very open about who they're connected to, and the network is always changing. How is AS 2 supposed to know whether or not there is a direct physical link between AS 3 and d? An AS has to be ready to express preferences over any paths that it might encounter.

Suppose all of the ASes are honest. Since there is no dispute wheel, BGP converges to a unique stable routing, shown in Figure 9(a). AS 3 gets its second-favorite path. If AS 3 announces the fake path $3d$ to its neighbors, however, then BGP converges to the routing shown in Figure 9(b). The figure shows the paths that are actually used; AS 2 has been duped into thinking that its traffic is following its preferred path $23d$, when in reality it's

[6]Why did ISPs prefer the new fake route over the tried-and-true real one? Because one commonly used rule is longest prefix matching, which prefers more targeted routes to less specific ones. The true path was listed for a block of 1024 IP addresses (i.e., with the first 22 of the 32 bits of the IP address specified) while the path advertised by Pakistan Telecom was for a subset of 256 of these IP addresses (i.e., with 2 additional bits specified). This trick is sometimes called "prefix hijacking."

[7]An earlier example occurred in 2004, when a Turkish ISP basically pretended to be the entire Internet, resulting in lots of traffic getting redirected to it.

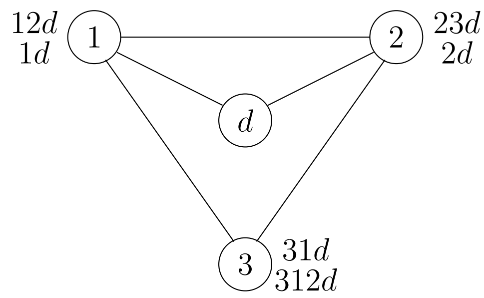

Figure 8: An example of an AS network in which announcing a fake path can be beneficial.

being routed also through AS 1. (If AS 2 knew the actual path being used, the routing would not be stable.) In this new routing, AS 3 gets its favorite path, and we conclude that announcing fake paths can be beneficial even when there is no dispute wheel.[8]

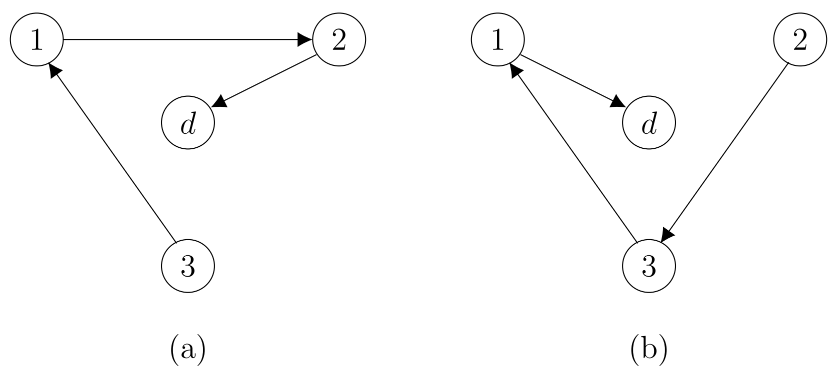

Figure 9: Stable routings in the AS graph of Figure 8 if (a) all of the ASes are honest; (b) AS 3 announces the fake path $3d$.

6.3 Incentive-Compatibility with Path Verification

We already noted that fake path announcements really are possible (indeed, frequent) in the current BGP protocol. But it is plausible that in the future, faking paths will not be so easy. For example, the proposed BGPsec protocol uses cryptographic signatures to verify the existence of announced paths. Do incentive issues persist if fake path announcements are disallowed?

Theorem 6.1 ([5]) *Assume that:*

1. *No AS can announce a path that doesn't exist.*

[8] In effect, the fake route announcement creates a "fictitious dispute wheel," which is enough to screw things up.

2. The AS graph does not have a dispute wheel.

Then, no AS has an incentive to unilaterally deviate from the intended behavior in BGP (assuming all other ASes follow BGP).

That is, no AS can benefit from withholding paths from neighbors, from choosing a path other than its favorite available option, or from falsely announcing a different route that actually exists in the network.

Theorem 6.1 is satisfying and surprising, as many network protocols used in practice are at least somewhat vulnerable to gaming by participants, even under restrictive assumptions (e.g., recall the BitTorrent discussion in Lecture #5). The theorem can also be extended to protect against coalitions of deviators—if a group of ASes deviates in a coordinated way from BGP and at least one of the coalition members becomes strictly better off, then a different member will be worse off [5].

We won't prove Theorem 6.1, but the intuition is similar to that for the proof of uniqueness given in Section 5. Roughly, if an AS had a profitable deviation, then the stable routings reached before and after the deviation play the same role as the different stable routings in the proof of Theorem 4.1 (leading to a dispute wheel). This argument isn't quite right, because after the deviation the routing is only stable in the minds of the ASes, who may have been misled about which path their traffic is being routed on (recall Figure 9(b)). It turns out that, as long as ASes can only be misled with existing paths, an analogous argument can still be made.

References

[1] L. Gao and J. Rexford. Stable Internet routing without global coordination. *IEEE/ACM Transactions on Networking*, 9(6):681–692, 2001.

[2] P. Gill, M. Schapira, and S. Goldberg. A survey of interdomain routing policies. *Computer Communication Review*, 44(1):28–34, 2014.

[3] T. Griffin, F. B. Shepher, and G. T. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Transactions on Networking*, 10(2):232–243, 2002.

[4] Internet Engineering Task Force. BGPsec protocol specification. Version 18. August 18, 2016.

[5] H. Levin, M. Schapira, and A. Zohar. Interdomain routing and games. *SIAM Journal on Computing*, 40(6):1892–1912, 2011.