Rationality and Traffic Attraction: Incentives for Honest Path Announcements in BGP

Sharon Goldberg[*]
Princeton University

Shai Halevi
IBM Research

Aaron D. Jaggard[†]
Rutgers University

Vijay Ramachandran[‡]
Colgate University

Rebecca N. Wright[§]
Rutgers University

ABSTRACT

We study situations in which autonomous systems (ASes) may have *incentives* to send BGP announcements differing from the AS-level paths that packets traverse in the data plane. Prior work on this issue assumed that ASes seek only to obtain the best possible *outgoing path* for their traffic. In reality, other factors can influence a *rational* AS's behavior. Here we consider a more natural model, in which an AS is also interested in *attracting incoming traffic* (*e.g.,* because other ASes pay it to carry their traffic). We ask what combinations of BGP enhancements and restrictions on routing policies can ensure that ASes have no incentive to lie about their data-plane paths. We find that protocols like S-BGP alone are insufficient, but that S-BGP does suffice if coupled with additional (quite unrealistic) restrictions on routing policies. Our game-theoretic analysis illustrates the high cost of ensuring that the ASes honestly announce data-plane paths in their BGP path announcements.

Categories and Subject Descriptors

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

General Terms

Theory, Economics

1. INTRODUCTION

Interdomain routing on the Internet consists of a *control plane*, where Autonomous Systems (ASes) discover and establish paths, and a *data plane*, where they actually forward packets along these paths. The control-plane protocol used in the Internet today is the Border Gateway Protocol (BGP) [35]. BGP is a path-vector protocol in which ASes discover paths through the Internet via announcements from neighboring ASes. In BGP, each AS has routing policies that may depend arbitrarily on commercial, performance, or other considerations. These policies guide the AS's behavior as it learns paths from its neighbors, chooses which (if any) neighbor it will forward traffic to in the data plane, and announces path information to its neighbors. The design of BGP seems to encourage ASes to rely on path announcement as an accurate indication for the paths that data-plane traffic follows. However, BGP does not include any mechanism to enforce that these announcements match actual forwarding paths in the data plane.

Traditional work on securing interdomain routing (*e.g.,* Secure BGP (S-BGP) [26] and the like [5, 20, 40]) has focused on the control plane, with the loosely-stated goal of ensuring "correct operation of BGP" [26]. However, addressing the control plane in isolation ignores the important issue of how packets are actually forwarded in the data plane. Here, we explicitly focus on the security goal of ensuring that the paths announced in the control plane match the AS-level forwarding paths that are used in the data plane; this has been implicit in many previous works (on securing BGP [20, 26, 40] and incentives and BGP [8–12, 28, 33]). This way, an AS can rely on BGP messages, *e.g.,* to choose a high-performance AS path for its traffic or to avoid ASes that it perceives to be unreliable or adversarial [3, 23, 34].

This goal has recently received some attention by works [1, 29, 36, 41] that suggest auxiliary enforcement protocols that operate in the data plane. However, because such solutions typically incur a high overhead (see Section 1.1), here we consider solutions that operate in the control plane alone. Furthermore, most works on BGP security assume ASes can be arbitrarily malicious. Here, we instead follow the literature on BGP and incentives by assuming that ASes are *rational*, *i.e.,* act in a self-interested manner. In our work, we define this to mean that ASes both (1) try to obtain the best possible outgoing path for their traffic, while (2) also attracting incoming traffic (see Section 1.3). We look for conditions under which rational ASes have *no incentive to lie* about about their forwarding paths in their BGP path announcements. We find that protocols like S-BGP [26] are generally *not* sufficient to prove that ASes have no incentive to lie about forwarding paths; we also require unrealistically strong assumptions on the routing policies of

[*] Partially supported by NSF award CNS–0627526. Work done in part while at IBM Research.

[†] Partially supported by NSF awards CNS–0753492 and DMS–0239996 and by ONR award N00014–05–10818. Work done in part while at Tulane University.

[‡] Partially supported by NSF awards CNS–0753061 and CNS–0524139. Work done in part while at the Stevens Institute of Technology.

[§] Partially supported by NSF awards CNS–0751674 and CNS–0753061.

SIGCOMM'08, August 17–22, 2008, Seattle, Washington, USA.
Copyright 2008 ACM 978-1-60558-175-0/08/08 ...$5.00.

every AS in the network. Our results emphasize the high cost of ensuring that control- and data-plane paths match, even if we assume that ASes are rational (self-interested), rather than arbitrarily malicious.[1]

In the rest of this section, we motivate our approach, discuss related work, outline our results and discuss their implications. The model we use is defined in Sections 2–3, and our results are detailed in Sections 4–6. Related work is discussed further in Section 7. Proofs are in the full version [18].

1.1 Matching the control and data planes.

One way to enforce honest path announcements in BGP is to deploy AS-path measurement and enforcement protocols that run in the data plane. However, determining AS-level paths in the data plane is a nontrivial task even in the absence of adversarial behavior (*e.g.,* [30] discusses the difficulty of determining AS-level paths from traceroute data). When dealing with ASes that may have incentives to announce misleading paths in the control plane, we need AS-path enforcement protocols that cannot be "gamed" (*e.g.,* by ASes that send measurement packets over the path advertised in the control plane, while sending regular traffic over a different path). Thus, data-plane enforcement protocols [1, 29, 32, 41] must ensure that measurement packets are indistinguishable from regular traffic, resulting in high overheads that are usually proportional to the amount of traffic sent in the data plane. Also, while secure *end-to-end* data-plane protocols can robustly monitor performance and reachability, *e.g.,* [2, 19], these protocols do *not* trace the identities of the ASes on a data-plane path; securely tracing AS paths requires participation of every AS on the path [1, 29, 32, 41].

Alternatively, one could hope to ensure that control- and data-plane paths match by ubiquitously deploying S-BGP [26] and the like [5]. This provides a property called path verification [28], which ensures that no AS can announce a path to its neighbors unless that path was announced to it by one of its neighbors. While path verification defends against announcement of paths that do not exist in the Internet topology [26], it does not, by itself, ensure that control- and data-plane paths match. For example, an AS a with two different paths announced by two different neighbors can easily lie in its path announcements—announcing one path in the control plane, while sending traffic over the other path in the data plane.

While it is tempting to argue that ASes are unlikely to lie about their forwarding paths because they either fear getting caught or creating routing loops, this argument fails in many situations. The hierarchy in the Internet topology itself often prevents routing loops from forming, *e.g.,* if the lie is told to a stub AS, or see also [4]. (We analyze the effect of lies on forwarding loops in the full version [18].) Furthermore, empirical results indicate that catching lies can be difficult, because even tracing AS-level paths that packets traverse in the data plane is prone to error [30]. Finally, to minimize the likelihood of getting caught, an AS could lie only when it has a good idea about where its announcements will propagate.

1.2 The game-theoretic approach.

In this work we explore the extent to which we can use *only control-plane mechanisms*, in conjunction with assumptions on AS policies, to motivate ASes to honestly announce data-plane paths in their BGP messages. Our exploration is carried out within the context of distributed algorithmic mechanism design [9, 31], which is rooted in game theory. This paradigm asserts that ASes are *rational players* that participate in interdomain routing because they derive utility from establishing paths and forwarding packets; ASes will do whatever they can to maximize their own utility. The task of mechanism design is to ensure that the incentives of rational players are aligned with accomplishing the task at hand.

The paradigm of algorithmic mechanism design in the context of routing was first suggested by Nisan and Ronen [31]. Feigenbaum *et al.* [9] brought *distributed* algorithmic mechanism design to the study of incentives in routing and shifted the focus to *interdomain* routing and BGP in particular. Rather than a centralized mechanism that sets up paths, the model in [9] postulates that paths are set up in a distributed fashion by the economically interested ASes themselves. The model was further developed in a sequence of works [6, 8–12, 28, 33]. Our model builds upon the work of Levin, Schapira, and Zohar [28], who brought a fully formal game-theoretic and distributed-computational model to this line of research (Section 2). In prior work, the prescribed behavior includes that ASes honestly announce to their neighbors the forwarding paths that they choose. If every AS follows this behavior, then the control plane and the data plane will match. In this sense, all work within this paradigm implicitly addressed matching the control and data planes. In this work, we highlight this matching (which is strictly weaker than the goal in prior work) as a standalone security property that needs to be addressed on its own. See more discussion in the full version [18].

1.3 Modeling utility with traffic attraction.

Recent work of Levin *et al.* [28] shows that if ASes are rational, then path verification (*e.g.,* S-BGP) is sufficient for honest path announcements, even when ASes have arbitrary routing policies. This encouraging result improved on earlier work [8–12] that explored restricted classes of routing policies. For example, Feigenbaum *et al.* [10, 12] found that it is sufficient to require policy consistency, a generalization of shortest-path routing and next-hop policy that requires that the preferences of neighboring ASes regarding different paths always agree. However, prior results [8–12, 28, 33] were obtained under the assumption that the utility an AS derives from interdomain routing *is entirely determined by the outgoing path that traffic takes to the destination*. (See also Section 7.) In reality, however, the utility of an AS is likely to be influenced by many other factors. For example, the utility of a commercial ISP may increase when it carries more traffic from its customers [24], or a nefarious AS might want to attract traffic so it can eavesdrop, degrade performance, or tamper with packets [3, 23, 34].

Here, we use a more realistic utility model (see Section 2.3), focusing in particular on the effect of traffic attraction, where the utility of one AS increases when it transits *incoming* traffic from another AS. We consider three models of traffic attraction. In our first model, traffic-volume attractions, utility depends only the origin of the incoming traffic, but not on the path that it takes. This captures the notion that an AS may be interested in increasing the volume of its incoming traffic or that a nefarious AS might want to attract traffic from a victim AS, in order to, say, perform traffic analysis. Our second model, generic attractions, encompasses all forms of traffic attraction; the utility of an AS may depend on the path incoming traffic takes. Our third model, customer attractions, is more restrictive. This model assumes that utility increases only if an AS attracts traffic from a neighboring customer AS that *routes on the direct link* between them; this models the fact that

[1] We do not consider situations when the control and data plane do not match due to malfunction or misconfiguration; we consider this *irrational* behavior. We also do not consider control- and data-plane mismatches caused by path aggregation [30], since typically only last hop of the (data-plane) AS-path is omitted from the BGP path announcement.

Control-plane verification	Model of AS utility			
	No traffic attraction	Increase volume of incoming traffic (Section 4)	Attract customer traffic via direct link (Section 6)	Generic traffic attraction (Section 5)
None	Policy consistency Consistent export [10, 12]	Next-hop policy All-or-nothing export	No known restrictions suffice	
Loop			Policy consistency Gao-Rexford conditions Next-hop at attractees Consistent export	Next-hop policy All-or-nothing export
Path	Arbitrary [28]	Policy consistency Consistent export		

Table 1: For each utility model and type of control-plane verification, the additional restrictions that ensure that ASes in a network with no dispute wheel have no incentive to dishonestly announce paths.

service contracts in the Internet are typically made between pairs of neighboring ASes [24] (Section 3.3).

1.4 Overview of our results.

In this work, we want to argue that under some set of conditions, any utility that an AS can obtain by lying in BGP announcements could also be obtained with honest announcements. Unfortunately, we find that conditions from previous work do not suffice when we consider traffic attraction: neither path verification [28] nor policy consistency [10, 12] alone is sufficient. (See Figures 2, 3, and 4 for examples.) These disappointing results motivate our search for new combinations of conditions (on control-plane verification, routing policy and export rules) that ensure that ASes have an incentive to honestly announce paths.

In addition to path verification (*e.g.,* S-BGP), we introduce a weaker form of control-plane verification called loop verification (Section 5.3), which roughly captures the setting in which an AS is caught and punished if it falsely announces a routing loop. Loop verification can be thought of as a formalization of "the fear of getting caught," and it may be easier to deploy than path verification.

In addition to policy consistency, we also consider the more restrictive next-hop policy, which roughly requires ASes to select paths to a destination based only on the immediate neighbor that advertises the path (Section 3.2). We also consider the Gao-Rexford conditions [14] (Section 3.3). These conditions, which are believed to reflect the economic landscape of the Internet [24], assume routing policies are restricted by business relationships between neighboring ASes, *i.e.,* by customer-provider relationships (the customer pays the provider for service) and peer-to-peer relationships (peer ASes transit each other's traffic for free).

Finally, we consider several classes of export rules (Section 3.4) that dictate whether or not an AS announces paths to its neighbors. An all-or-nothing export rule requires that, for each neighbor, an AS either announces every path or no paths. We also consider a more realistic consistent export rule [10] that roughly requires that ASes' export rules agree with their routing policies.

For many combinations of the conditions discussed above, we can still find examples in which ASes have an incentive to lie about their data-plane paths. However, for some combinations we obtain positive results, as sketched in Table 1. (These results all assume a network condition called "no dispute wheel" [21]; see Section 3.1.) Furthermore, our results are "tight", in that for every combination of the considered conditions, either one of our positive results applies or one of our negative examples does (as summarized in Tables 2–4).

Our positive results show that, for *every network* satisfying some combination of conditions, any utility an AS gains by lying can equivalently be obtained if that AS had instead *honestly announced paths to only an subset of its neighbors* and announced no paths to all other neighbors. That is, we show the existence of an *export rule* for which each AS obtains its optimal utility. As in previous work [10, 12, 28], our positive results for traffic-volume attractions (Section 4) and customer attractions (Section 6.2) also explicitly define an optimal export rule. Our positive result for generic attractions (Section 5.4) shows that an optimal export *exists*, but does not explicitly state what it is (Section 5.5). We discuss the notions used for our positive results further in the full version [18].

1.5 Implications of our results.

Our results suggest that even with control-plane enforcement mechanisms, ASes may have incentive to lie in their BGP announcements, unless very strong restrictions are imposed on their policies. As sketched in Table 1, from the set of conditions we considered, we always need *every AS in the network* to obey (1) unrealistic restrictions on its preferences (such as next-hop policy) and (2) explicit restrictions on export rules. Most of our results also require (3) full deployment of either path or loop verification. Thus, our results point to a negative answer to the question that we set out to investigate—practically speaking, it is unlikely that we could use only control-plane mechanisms to remove the incentives for ASes to announce false paths in BGP.

This suggests a choice. We can either employ expensive data-plane path enforcement techniques [1, 29, 32, 41] when it is absolutely necessary to ensure that packets are forwarded on AS-level paths that match an AS's routing policies, or dismiss this idea altogether and instead content ourselves with some weaker set of goals for interdomain routing. It is certainly possible to formulate weaker but meaningful security goals and show that certain control-plane mechanisms or data-plane protocols meet these goals. However, doing this invites the question: if we are not interested in ensuring that AS paths announced in BGP are really used in the data plane, then why use a path-vector protocol at all?

2. MODELING INCENTIVES AND BGP

We now sketch the formal model in support of our results in Sections 4–6. Details are in the full version. The model builds on the literature [9, 21, 28] and extends prior work by explicitly considering traffic attraction.

2.1 The AS graph.

An interdomain-routing system is modeled as a labeled, undirected graph called an AS graph (see Figure 1). For simplicity, each AS is modeled as a single node, and edges represent direct (physical) communication links between ASes. Adjacent nodes are called neighbors. We denote nodes by lowercase letters, typically a, b, c, d, m, and n. We follow [21] and assume the AS-graph topology does not change during execution of the protocol.

Because, in practice, BGP computes paths to each destination separately, we follow the literature [21] and assume that there is a unique *destination node* d to which all other nodes attempt to establish a path. (Thus, like most previous work, we ignore the issue of route aggregation [30].) We denote paths by uppercase letters, typically P, Q, and R.

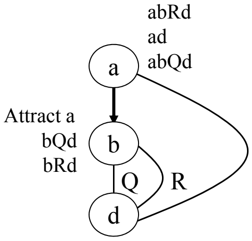

Figure 1: AS graph with traffic attraction.

2.2 The interdomain-routing game.

We extend the model of Levin *et al.* [28] that describes interdomain routing as an infinite-round game in which the nodes of the AS graph are the strategic players. In each round, one node in the graph processes the most recent path announcements (if any) from its neighbors and then performs two *actions*: (1) it decides on an outgoing link (if any) to use in the data plane; and (2) decides on paths (if any) to announce to its neighbors. Note that, just as in [28], nodes have the opportunity to announce their true data-plane path choice, but they are not forced to do so.

We assume that path announcements sent between neighbors on direct links cannot be tampered with (by a node not on the direct link). This can be enforced via the BGP TTL Security Hack [16] or via a pairwise security association between nodes using the TCP MD5 security options [22]. We further assume that each node has the opportunity to act infinitely often—*i.e.,* the game is *fair*.

Game outcome and stability. The *state* of a node n at some round in the game consists of a data-plane component (the outgoing link most recently chosen by n) and a control-plane component (the announcements most recently sent by n). This state is *transient* if it occurs only finitely many times (and it is *persistent* otherwise). The global state at some round is the collection of all node states at that round. The global outcome of a game is a global state that does not contain any transient node states.[2] If the state of a node is constant after some round then this state is locally stable. A global outcome is globally stable if all node states in it are locally stable. (This definition of stability is compatible with the original definition in [21].) We typically denote global outcomes by T or M. We may use "outcome" informally to mean the control-plane or data-plane component of the outcome when the component is clear from the context.

2.3 Utility, valuation, and attraction.

A *strategy* is a procedure used by a node to determine its actions in the game. In principle, a node can make decisions in any way that it wants, but here we assume that nodes are *rational*. In particular, each node b has a utility function $u_b(\cdot)$ mapping outcomes to integers (or $-\infty$); b tries to act to obtain an outcome T that maximizes $u_b(T)$.

We assume that every node b in the graph has a utility function of the form

$$u_b(T) = v_b(T) + \alpha_b(T) \qquad (1)$$

where $v_b(T)$ is the valuation function that depends only on the simple data-plane path from b to d in T, and $\alpha_b(T)$ is the attraction function that depends only on the simple data-plane paths from other nodes to b in T. (We write the utility function as a *sum* of the valuation and attraction functions; in fact, our results require only that utility increases monotonically with both valuation and attraction.) The components of utility in this work depend on the data-plane component of outcome alone, because the control-plane component may not correspond to actual traffic flow in the network.

The valuation function $v_b(\cdot)$ is the same as was considered in previous work on incentives and BGP [6, 8–12, 28, 33]. It is meant to capture the intrinsic value of each outgoing path (*e.g.,* as related to the cost of sending traffic on this path, its reliability, the presence of undesirable ASes on it, *etc.*).

Modeling nefarious ASes. We assume that $v_b(T) = -\infty$ implies $u_b(T) = -\infty$, so that nodes cannot derive any utility from outcomes in which they cannot reach the destination. Our negative examples do not depend on this assumption, but our positive results do. This means that our positive results do *not* hold if a manipulating node wants to attract traffic for nefarious purposes, like tampering or eavesdropping, *when it does not have a path to the destination.*

The attraction function $\alpha_b(T)$ is the new component of utility that we add in this work. Because we are interested in situations where nodes may want to attract traffic (and not deflect it), our most general form of the attraction function only requires that $\alpha_b(\cdot)$ does not increase when edges leading to b are removed from the data-plane outcome. Formally, for an outcome T and node b, let $T(b)$ be the set of edges along simple paths from other nodes to b in the data-plane component of T (*e.g.,* if T's data-plane links form a routing tree, then $T(b)$ is the subtree rooted at b). We assume that for every two outcomes T and T' and every node b, if $T'(b) \subseteq T(b)$, then $\alpha_b(T') \leq \alpha_b(T)$. This general condition covers many forms of traffic attraction; *e.g.,* attraction can depend on which links are traversed by incoming traffic at a node, and not just the nodes from which that traffic originates.

We also consider two specific forms of traffic attraction. First, traffic-volume attraction requires that $\alpha_b(T)$ depends only the origin of the incoming traffic, but not on the path that it takes. More formally, if $T(b)$ and $T'(b)$ include the same nodes then $\alpha_b(T) = \alpha_b(T')$. This also captures the idea of nefarious ASes who want to attract traffic for eavesdropping on or tampering with traffic (but see also our note above).

Another specific form of attraction is customer attraction, in which the AS graph is assumed to have underlying business relationships, and $\alpha_b(T)$ depends only on customer nodes a that route through b on the direct a-b link between them. We further discuss this form of attraction and customer-provider relationships in Section 3.3.

We say that there is an *attraction relationship* between a and b if the attractor b increases its utility when the attractee a routes traffic through it (*e.g.,* as in Figure 1). In Figure 1, we depict the utility function of each node next to that node: say that the attraction function of b is such that it earns 100 points of utility when it attracts traffic from a, and that the valuation function of b is such that it earns 10 points of utility when using the path bQd and only 1 point of utility when using the path bRd. Then, following Equation 1, the use of data-plane path $abRd$ earns b 101 points of utility.

2.4 BGP-compliant strategies.

Recall that we are interested in ensuring that the interdomain-routing control and data planes match. When all nodes follow the rules prescribed by the BGP RFC [35] in their execution of the protocol, this is achieved. We call a strategy that obeys these rules a *BGP-compliant strategy*, as formalized below.

Definition 2.1. *A BGP-compliant strategy for node n depends on two functions: A ranking function $r_n(\cdot)$ mapping each path to*

[2] Note that there could be more than one such global state; our results in this work hold regardless of which of these is taken to be the global outcome.

an integer or $-\infty$; and, an export rule $e_n(\cdot)$ *that maps each path P to the set of neighbors to which n is willing to announce the path P. A path P is* admitted at n *if $r_n(P) > -\infty$. Paths that include routing loops or that do not reach the destination are not admitted at any node. We require that, for any two paths P and Q admitted at n that begin with different next hops, it holds that $r_n(P) \neq r_n(Q)$. Note that $r_n(\cdot)$ and $e_n(\cdot)$ act on path announcements, rather than game outcomes.*

The strategy of node n is BGP-compliant*, with $r_n(\cdot)$ and $e_n(\cdot)$ as defined above, if n does the following in each round in which it participates. Node n first chooses the path P such that (a) P has highest rank of all the most recently announced paths received from neighbors, and (b) the first node a of P is the neighbor that announced P to n. Then, n performs the following two actions: (1) n chooses the outgoing link to a in the data plane; and (2) n announces the path nP to all neighbors in $e_n(P)$.*

This definition explicitly assumes that the all traffic to the destination is routed over a single next-hop. (We do not address here the question of modeling multipath routing.) Also, we assume that, if n does not receive any announcements with an admitted path, then n does not route on any outgoing link or announce any paths to its neighbors. (Notice that we model *ingress filtering* using the concept of admitted paths and *egress filtering* using the concept of an export rule.)

Control-plane announcements from a node executing a BGP-compliant strategy match its next-hop choices in the data-plane. Thus, if *all* nodes in the network use BGP-compliant strategies, then the control and data planes will match. (We may informally call a node executing a BGP-compliant strategy a *BGP-compliant node*, or sometimes an *honest node*.) In the positive results from previous work [10,12,28] included in Table 1, the prescribed strategies are examples of BGP-compliant strategies in the sense of Definition 2.1. Thus, those results achieved agreement between the control and data planes, but contrary to the current work, they do not consider traffic attraction.

We stress that Definition 2.1 gives BGP-compliant nodes the leeway to choose their ranking and export functions to try and achieve a utility-maximizing outcome in the game. In the next subsection, we discuss the relationship between utility and the ranking and export functions in a way that encompasses earlier work (without traffic attraction) and the results in this work (with traffic attraction).

2.5 From utility to ranking and export.

In the real-world implementation of BGP [35], the two actions of the game are executed by setting parameters that we have modeled as the ranking and export functions in Definition 2.1 above. In previous work [12, 28], the utility of an AS was defined to be its valuation function,[3] and that valuation directly determined the ranking function (we denote this $r_n(\cdot) \equiv v_n(\cdot)$)[4]: the larger the valuation of a path, the higher its rank. This direct translation from valuation to ranking does not always hold in our setting of traffic attraction: announcing an outgoing path with low valuation could be preferred because it brings incoming traffic from attractees. For example, in Figure 1, node b's valuation function ranks path bQd over path bRd; but, b has higher utility when it claims that it routes on bRd because it then attracts traffic from node a.

Although this direct translation does not always hold, we do assume that BGP-compliant ASes are able to "compile" their utility functions (which depend on both valuation and attraction as in Equation 1) into ranking and export functions that then consistently determine their actions in the game, *i.e.,* their behavior during the BGP protocol. This compilation might be viewed as transforming utilities into functions that act on path announcements by, *e.g.,* setting BGP local preference. We think of the compilation process as being done "once and for all," and we analyze the network with respect to fixed ranking and export functions. We note that this is not entirely realistic: the "compilation" can, in principle, model an ongoing process in which an AS reacts to changes in network conditions, contractual agreements, new information that ASes learn about each other, *etc.*, to better attempt to maximize its utility. However, the time scale for compilation is usually much longer than the time scale for BGP itself (say, hours versus seconds); so, a once-and-for-all modeling may still be reasonable. (See also Section 7.)

In general, we mostly sidestep the question of how to compile the utility into ranking and export policy. However, our counterexamples work for any ranking function "reasonably compiled" from the utility function, and our positive results all hold for the setting $r_b(\cdot) \equiv v_b(\cdot)$.

2.6 Incentives to lie.

Because nodes are rational—acting to maximize their utility in the global outcome—they may have an incentive to follow a strategy that is not BGP-compliant. As discussed in Section 1.1, although an AS knows the outgoing link on which it forwards traffic (and the next AS at the end of that link), it may not know the AS-path that the traffic takes further downstream. For example, in Figure 1, node b could deviate from BGP-compliance by announcing the path bRd in order to attract traffic from node a, while actually sending traffic over the path bQd; as a result the control and data planes would not match, unbeknownst to a.

Hence, in this work, as in [10, 12, 28, 33], we address the following high-level question: Are there sufficient conditions on the network that ensure that all nodes are honest (*i.e.,* use BGP-compliant strategies)? The earlier work studied this question using the game-theoretic notion of "incentive compatibility." In contrast to some uses of this notion in earlier work (*e.g.,* Thm. 3.2 in [28]), our positive results give nodes some additional flexibility in choosing their strategies, as long as these strategies are BGP-compliant. (We discuss this difference in some detail in the full version [18].)

Ideally, we would like conditions that ensure that nodes have no incentive to be dishonest, no matter what the other nodes do. Unfortunately, it is extremely difficult to find such conditions; see [10, 12, 28, 33]. Instead, we look for conditions that ensure that a node has no incentive to be dishonest *if it knows that everyone else is honest*. That is, we try to ensure that no node has an incentive to *unilaterally* deviate from using BGP-compliant strategies.

We discuss our technical formalizations after each of our positive results (Theorems 4.1, 5.1, and 6.1).

3. DEFINITIONS: POLICY AND EXPORT

3.1 No dispute wheel.

Griffin, Shepherd, and Wilfong [21] described a global condition on the routing policies in the AS graph, called *"no dispute wheel,"* that ensures that BGP always converges to a unique stable outcome. Roughly, a dispute wheel is a set of nodes, each of which prefers to route through the others rather than directly to the destination. More formally, there is a dispute wheel in the valuations if there exist nodes $n_1, \ldots, n_t$ such that, for each node n_i, there exists a simple path Q_i from n_i to the destination d and a simple path R_i

[3] Some previous work [8–11, 33] allowed utilities that depend on monetary transfers, which we do not consider here.

[4] This is a slight abuse of notation, because r is formally defined on paths and v on outcomes. We ignore this formality from now on.

from n_i to n_{i+1} for which $v_{n_i}(R_iQ_{i+1}) > v_{n_i}(Q_i)$.[5] (The index i is taken modulo t.) A dispute-wheel in the ranking functions (for BGP-compliant nodes) is defined similarly with r_{n_i} replacing v_{n_i}. Following the literature [12,28], we *always* consider networks with no dispute wheels in the valuations. The result of [21] in our terminology states that, if all nodes use BGP-compliant strategies with $r_n(\cdot) \equiv v_n(\cdot)$ and there is no dispute wheel in the valuations, then the game's outcome is unique and globally stable.

3.2 Policy consistency and next-hop policy.

Node a is *policy consistent* [10, 12] in valuations with one of its neighbors b if, whenever b prefers some path bPd over bRd (and neither path goes through a), then a prefers $abPd$ over $abRd$. Formally, for any two simple paths $abPd$ and $abRd$, if $v_b(bPd) \geq v_b(bRd)$, then $v_a(abPd) \geq v_a(abRd)$. We say that *policy consistency* holds for the problem instance if every node is policy consistent with each of its neighbors. (Policy consistency is a generalization of next-hop routing and shortest-path routing; see [10, 12].)

Next-hop policy requires that a node only care about the neighbor through which its traffic is routed and nothing else. This class of routing policies is more restrictive than policy consistency (*e.g.,* node c in Figure 3 is policy consistent but does *not* use next-hop policy with node m). Formally, a uses next-hop policy with b if for every two simple paths $abPd$ and $abRd$ it holds that $v_a(abPd) = v_a(abRd)$. Notice that if a uses next-hop policy with b then it must either admit all simple paths through b or (ingress) filter all of them (*cf.,* discussion in [7, 37]).

Similar definitions apply also to the ranking functions.

3.3 Gao-Rexford & customer attractions.

Gao and Rexford [14] described a set of conditions that are induced by business relationships between ASes [24]. In Gao-Rexford networks there are two kinds of edges: customer-provider edges (where typically the customer pays the provider for connectivity) and peer-to-peer edges (where two nodes agree to transit each other's traffic for free). A Gao-Rexford network obeys the following three conditions (GR1–GR3):

GR1. Topology. There are no customer-provider cycles in the AS graph, *i.e.,* no node is its own indirect customer.

GR2. Export. A node b only exports to node a paths through node c if at least one of nodes a and c are customers of node b.

GR3. Preferences. Nodes prefer outgoing paths where the next hop is a customer over outgoing paths where the next hop is a peer or a provider, and prefer peer links over provider links.

GR3 always applies to the valuation functions of each node in a Gao-Rexford network, and can also apply to the ranking functions.

We also model customer attractions within the Gao-Rexford setting. Namely, we consider a fourth condition (AT4) that models the fact that service contracts in the Internet are made between pairs of neighboring nodes, where a customer pays its provider when it sends traffic over their shared link [24]. AT4 restricts the set of traffic attraction relationships that we allow in the AS graph, and thus does *not* model settings where, *e.g.,* an AS wants to attract traffic from ASes that are a few hops away.

AT4. Attractions. A node b may only have attraction relationships with its own customers. Furthermore, b only increases its utility if its attractee-customer a sends traffic over the direct a-b link.

We remark that within our framework, the conditions GR1, GR3, and AT4 are viewed as restrictions over the problem instance (because they apply to the valuation and attraction functions), whereas the condition GR2 is a restriction on the actions nodes may take.

When we draw Gao-Rexford networks, we represent a customer-provider relationship by a directed edge from customer to provider, and a peer-to-peer relationship by an undirected edge. We represent an AT4 attraction relationship with a **bold** arrow from attractee to attractor (*e.g.,* see Figure 2).

3.4 Export rules.

Our results about BGP-compliant strategies that achieve matching control and data planes in the setting of traffic attraction involve several types of export rules. The export-all rule (used, *e.g.,* in Thm. 3.2 of [28]) requires that a node exports all its admitted paths to all its neighbors. An all-or-nothing rule for a node n means that, for each neighbor a of n, either n exports all admitted paths to a or none at all. The consistent export rule [10] means that, if n exports to a neighbor a some path R, then it must also export every other path that is ranked at least as high as R; *i.e.,* if $r_n(Q) \geq r_n(R)$ and n exports R to a, then n must also export Q to a. Finally, in Gao-Rexford networks, the export rules used by BGP-compliant nodes satisfy GR2.

The export-all rule implies the all-or-nothing export rule, which in turn implies the consistent export rule. We emphasize that both the export-all and the all-or-nothing rules are often incompatible with the Gao-Rexford export condition GR2. As one example, the export-all rule may require an AS to export a path through one of its peers or providers to another one of its peers or providers, a violation of GR2.

4. RESULTS: VOLUME ATTRACTIONS

We start with some results for traffic-volume attractions, as defined in Section 2.3. We stress that this is a rather restricted form of traffic attraction, as it excludes the possibility of the utility depending on the path along which incoming traffic arrives. We begin with a series of counterexamples, demonstrating that even for this very restricted form of traffic attraction, ensuring that nodes have no incentive to lie is far from easy. (Most of our counterexamples are Gao-Rexford networks that obey GR1–GR3 and sometimes also AT4 from Section 3.3.) We then present a positive result (Section 4.3), showing two sets of conditions, each of which suffices to ensure that a node honestly announces paths. Our results are summarized in Table 2.

4.1 Path verification is not enough.

Path Verification is the focus of most traditional work on securing BGP [5]; roughly, it ensures that nodes cannot announce paths that are not in the network. More formally, path verification is a control-plane mechanism that ensures that every node a only announces a path abP to its neighbors if its neighbor b announced the path bP to a. Path verification can be guaranteed when S-BGP [26] or IRV [20] is *fully* deployed in the network.

For the setting of no traffic attraction, a recent result of Levin *et al.* [28] shows that, in a network with path verification and no dispute wheel, no node has an incentive to unilaterally deviate from a BGP-compliant strategy with $r_n(\cdot) \equiv v_n(\cdot)$ and an export-all rule. They also show (in [27]) that the same is true in Gao-Rexford networks, but with an export rule that exports all paths except those that would violate GR2. However, we show that when there are traffic-volume attractions, a node can have an incentive to make a dishonest announcement, even when the network has path verification:

[5] For readability, we somewhat abuse notation and use $v_n(P)$ to mean n's valuation of any outcome T in which its traffic uses the data-plane path P.

Verification?	Policy	Export	Incentive to Lie?	Result
⋆	No restriction	⋆	Yes	INCONSISTENT POLICY
None / Loop	Consistent	⋆	Yes	NONEXISTENT PATH
Path / Loop	Next-hop	Inconsistent	Yes	See full version
Path	Consistent	Consistent	No	Theorem 4.1
⋆	Next-hop	All-or-nothing	No	Theorem 4.1

Table 2: Summary of our results for traffic-volume attractions. We also require no dispute wheel.

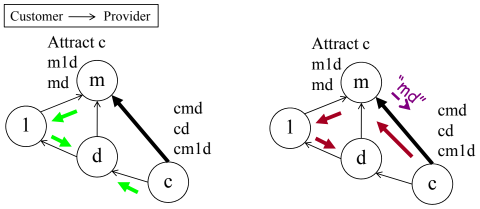

Figure 2: INCONSISTENT POLICY

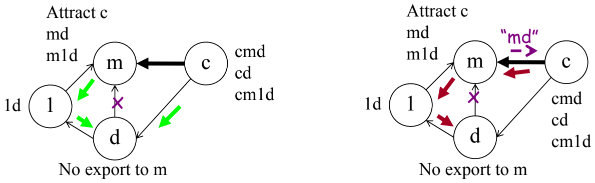

Figure 3: NONEXISTENT PATH

Figure 2: INCONSISTENT POLICY demonstrates that a policy inconsistency between a manipulator m and its customer c can give m an incentive to dishonestly announce its forwarding path in order to attract traffic from c. On the left we show the outcome T that results when each node n uses a BGP-compliant strategy with $r_n(\cdot) \equiv v_n(\cdot)$, exporting all paths except those that would violate GR2. On the right, we show the manipulated outcome M, in which only a single manipulator node m does not use a BGP-compliant strategy. Here, m has an incentive to announce the path md to node c, while actually using path $m1d$, in order to attract c's traffic. Notice that this announcement can be made even with path verification, because node 1 announced $1d$ to m. In the outcome M, node m gains not only a traffic-volume attraction (because c routes through m in M but not in T), but also an AT4 attraction (because c is a customer that routes on the direct c-m link in M). (Note that INCONSISTENT POLICY is a Gao-Rexford network with no dispute wheel that obeys AT4.)

We remark that the situation in INCONSISTENT POLICY could arise quite naturally in practice. As an example, while c is a customer of both m and d, the service contracts of c with m and d are such that usage-based billing on the m-c link is lower than billing on the d-c link. Then, c could prefer a path through m over the direct path to d as long as this path only increases AS-path length by a single hop. On the other hand, m could prefer to send traffic via 1 because 1 is, say, geographically closer to m than d.

4.2 Policy consistency alone is not enough.

Notice that, in INCONSISTENT POLICY, node c is not policy consistent with node m (Section 3.2). It is natural to ask if requiring policy consistency is sufficient to ensure that there is no incentive to lie. Indeed, for the setting of no traffic attraction, Feigenbaum *et al.* [10, 12] proved that in a network with policy consistency and no dispute wheel, then no node has an incentive to *unilaterally* deviate from a BGP-compliant strategy with $r_n(\cdot) \equiv v_n(\cdot)$ and consistent export. Perhaps surprisingly, it turns out that policy consistency is not sufficient to ensure that nodes have no incentive to lie when we consider traffic-volume attractions:

Figure 3: NONEXISTENT PATH demonstrates that, even in a policy consistent network, a manipulator m can have an incentive to announce a nonexistent path in order to attract traffic from its customer c. The outcome T, shown on the left, results when each node uses a BGP-compliant strategy with $r_n(\cdot) \equiv v_n(\cdot)$, where node d's export rule obeys consistent export but exports nothing to node m, and all other nodes export all paths allowed by GR2 (which implies consistent export). On the right, we show the manipulated outcome M, where only the manipulator m deviates from the BGP-compliant strategies described above. Here, the manipulator m has an incentive to announce to node c a false path "md" that *is not available to* m (because d does not export this path to m) in order to attract c's traffic. Again, node m gains both a traffic-volume attraction and an AT4 attraction in M that it could not have obtained by using a BGP-compliant strategy. (Note that NONEXISTENT PATH is a policy-consistent Gao-Rexford network with no dispute wheel that obeys AT4.)

Notice that c has the same preferences in both NONEXISTENT PATH and INCONSISTENT POLICY. However, in NONEXISTENT PATH, c is policy consistent with m; both prefer the nonexistent shorter path through md over the longer path through $m1d$.

4.3 But adding path verification or next-hop policy is enough!

In NONEXISTENT PATH, the manipulator m announces a path "md" was that was not announced to it by d (which would not be possible if the network had path verification), and that announcement matters because node c does not use a next-hop policy with m. It turns out that requiring *either* path verification (on top of policy consistency) *or* next-hop policies is sufficient to ensure honesty in any network with only traffic-volume attraction functions. In these settings, if each node sets its ranking equal to its valuation and honestly exports *all* paths to all neighbors, then no node has an incentive to *unilaterally* deviate from this behavior.

THEOREM 4.1. *Consider an AS graph with no dispute wheel in the valuations. Suppose that all nodes, except a single manipulator node m, use BGP-compliant strategies and set their ranking equal to their valuations ($r_n(\cdot) \equiv v_n(\cdot)$ for every node n). Suppose further that m has a traffic-volume attraction function, and that at least one of the following two conditions hold:*

***a.** The valuations function of all nodes are next-hop and the export functions of all the nodes but m obey all-or-nothing export; or*
***b.** The valuations function of all nodes are policy consistent, the export functions of all the nodes but m obey consistent export, and the network has path verification.*

Then there is a BGP-compliant strategy for m that sets $r_m(\cdot) \equiv v_m(\cdot)$ and obeys all-or-nothing export (and therefore also consistent export), such that this strategy is optimal (utility-maximizing) for m. In particular, using the export-all rule is one such optimal strategy.

Notice that Theorem 4.1 not only establishes the *existence* of an optimal consistent export rule for m, but also asserts that export-all

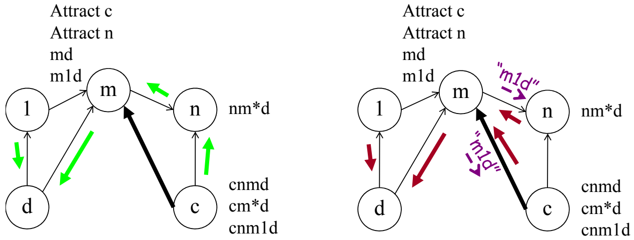

Figure 4: Bowtie

is one such optimal rule. Hence it actually establishes a single strategy from which no node has an incentive to deviate. (This notion of a single strategy is the same notion used in prior works including [10,12,28,33]. In the mechanism-design literature, this is called *incentive-compatibility in ex-post Nash equilibrium*; see [33].) The proof of Theorem 4.1 is presented in the full version [18]. In the full version, we also present a counterexample that shows we cannot drop the requirement for consistent export from Theorem 4.1.

5. RESULTS: GENERIC ATTRACTIONS

We now consider our most general notion of traffic attraction, in which the utility that nodes derive from attracting traffic can depend arbitrarily on the path that incoming traffic takes (see Section 2.3). For this general case, we show in Section 5.4 that nodes have no incentive to lie when all nodes use next-hop policy and all-or-nothing export and the network has path verification. (In fact, we show that a weaker enforcement mechanism called *loop verification* is also sufficient; see Section 5.3.) These conditions are extremely strong, but we show via a sequence of counterexamples that we cannot drop any one of these conditions without allowing an incentive to lie. The theorems and counterexamples in this section are summarized in Table 3.

5.1 Policy consistency & path verification is not enough.

In networks with only traffic-volume attraction, we were able to show that adding path verification to a policy-consistent AS graph is sufficient to ensure that nodes have no incentive to lie (Section 4.3). Unfortunately, this is not the case when we consider more general attraction relationships:

Figure 4: Bowtie demonstrates that, even in a a network that is policy consistent and has path verification, a manipulator m can have an incentive to lie about its forwarding path in order attract traffic from a customer c on the direct m-c link. Suppose node m has an attraction function such that (1) m has an AT4 attraction relationship with its customer c, and (2) m has a traffic-volume attraction with its provider n. The outcome T that results when every node uses a BGP-compliant strategy with $r_n(\cdot) \equiv v_n(\cdot)$ and exports all paths allowed by GR2, is shown on the left. The manipulated outcome M is shown on the right, where only node m deviates from the BGP-compliant strategy we described above.

Here, m has an incentive to dishonestly announce the path "$m1d$" to all of its neighbors in order to attract traffic from the attractee c on the direct c-m link. Node m can make this announcement, even with path verification, because node 1 announced the path $1d$ to m. Moreover, there is no BGP-compliant strategy for m that allows it to attract traffic from both c and n while maintaining its preferred data-plane forwarding path md. (Note that Bowtie is a policy-consistent, Gao-Rexford network with path verification *that does not obey AT4* and has no dispute wheel in the valuations.)

We remark that even though c's traffic is routed via m in both T and M (*i.e.,* m does *not* gain a traffic-volume attraction), the manipulation in Bowtie is quite reasonable in practice. For example, m might prefer the outcome in M over the outcome in T for load-balancing purposes, because incoming traffic from c and n is spread over two links in M. As another example, m might prefer the outcome M because it has a usage-based billing contract with c on the m-c link, whereas node m is not able to bill its provider n for carrying c's traffic (which occurs in outcome T).

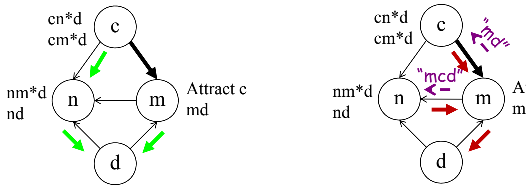

Figure 5: False Loop

5.2 Next-hop policy alone is not enough.

From Bowtie, we learn that policy consistency is not sufficient to ensure honest announcements (even when using path verification). So we throw up our hands and ask if it suffices to require that every node uses next-hop policy. With next-hop policy, it is tempting to conclude that lying about an outgoing path will not help an attractor convince an attractee to 'change its mind' and route through it in a manipulated outcome. (Notice that the manipulations in Inconsistent Policy, Nonexistent Path and Bowtie were of this form.) Furthermore, next-hop policy is sufficient when considering only traffic-volume attractions (Section 4.3).

Quite surprisingly, this intuition fails. We now present our most important counterexample, which shows that if the network does not have path verification, then even requiring next-hop policy is not sufficient:

Figure 5: False Loop demonstrates that, even in a network where all nodes use next-hop policies, a manipulator m can gain traffic from its customer c by falsely announcing a path through c to m's other neighbors. Suppose that m announces no paths to neighbor n and all paths to everyone else, and that all other nodes export all paths allowed by GR2. On the left is the outcome T, where each node compiles $r_n(\cdot) \equiv v_n(\cdot)$ and uses the BGP-compliant strategy with the export rules described above. The manipulated outcome M is on the right, where only m deviates from the BGP-compliant strategy above. In M, the manipulator m has an incentive to announce a false outgoing path "mcd" to n in order to attract traffic from its attractee c (on the direct c-m link). Notice that the outcome M results whenever there is no control-plane verification mechanism such as path verification, since the 'false loop' "$nmcd$" will either cause node n not to announce any path to node c, or instead cause node c to ignore the announcement. Also, m has no BGP-compliant strategy that allows it to gain an AT4 attraction from c, since c would have sent his traffic on the c-n link if m had either (a) honestly announced some path to n, or (b) announced no path to n (as in outcome T). (Note that False Loop is a Gao-Rexford network with no dispute wheel that obeys AT4, in which all nodes use next-hop policies.)

5.3 Introducing loop verification.

To deal with the manipulation in False Loop, we introduce loop verification, a new control-plane mechanism that deals with detecting and preventing "false loops."

BGP allows two different approaches for detecting and preventing routing loops. One is sender-side loop detection, where a

Verification?	Policy	Export	Incentive to Lie?	Result
None	⋆	⋆	Yes	FALSE LOOP
⋆	Consistent	⋆	Yes	BOWTIE
⋆	Next-Hop	Consistent	Yes	GRANDMA
Path / Loop	Next-Hop	All-or-Nothing	No	Theorem 5.1

Table 3: Summary of our results for generic attractions and no dispute wheel.

node a will *not* announce path aRd to node b if b happens to be on the path R. The other is receiver-side loop detection where a *will* announce the path aRd to b, so that b will detect the loop and discard that announcement. Receiver-side loop detection has the advantage of allowing a node b to hear announcements that *falsely* include a path that b did not announce. Notice that for b to detect a "false loop," b need only perform a *local* check to see if the path it receives matches the one that b actually announced. (This local check is less onerous than the one that is required for path verification, which requires participation from all ASes on the path.)

Loop verification encourages ASes to avoid lying in BGP announcements because they should fear getting caught. We define loop verification as the use of receiver-side loop detection by *all* nodes in a network, with the additional requirement that when node b receives an announcement of a path $P = QbRd$, such that b did not announce the path bRd to its neighbors, then b "raises an alarm." Then, the first node who announced a path that includes bRd will be punished with utility reduced to $-\infty$. This punishment process models the idea that b can catch and shame the node that announced the false loop, *e.g.,* via the NANOG list.

The properties of loop verification are strictly weaker than those of path verification. Namely, if a network has path verification, then no node will raise an alarm in loop verification. This follows from the fact no node can announce a path that includes bRd unless b announces the path bRd.

5.4 Next-hop policies & loop verification is enough!

Now that we defined loop verification, we are ready to present the main result of this section. If we add loop verification to a next-hop network with no dispute wheel, we can eliminate the manipulation performed by m in FALSE LOOP. We also require all nodes to use an all-or-nothing export rule. The following holds even if the network does *not* obey the Gao-Rexford conditions:

THEOREM 5.1. *Consider an AS graph where the valuation functions are next-hop and contain no dispute wheel. Suppose that all nodes, except a single manipulator node m, use BGP-compliant strategies where they set their ranking equal to their valuations ($r_n(\cdot) \equiv v_n(\cdot)$ for every node n), and obey all-or-nothing export. Suppose further that the network uses either loop verification or path verification. Then there exists a BGP compliant strategy for m that uses $r_m(\cdot) \equiv v_m(\cdot)$ and obeys all-or-nothing export, which obtains the best possible* stable outcome *in terms of the utility function of m.*

On an intuitive level, Theorem 5.1 proves that any gains a manipulator gets from lying can be obtained by using a clever export rule.[6] That is, Theorem 5.1 shows the *existence* of an optimal all-or-nothing export rule for the manipulator; however, this optimal export rule for m depends on the export rules chosen by the other nodes in the network. Furthermore, unlike prior work or the result from Section 4, this result does *not* explicitly describe this optimal export rule. The proof of Theorem 5.1 is quite technically involved, and we present it in the full version [18].

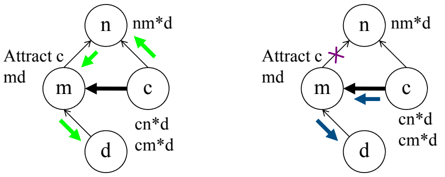

Figure 6: ACCESS DENIED.

5.5 Export-all is not always optimal.

Theorem 5.1 unfortunately does *not* explicitly describe the optimal export rule for the manipulator. We now show that the export-all rule (which was shown to be optimal in Theorem 4.1) is not necessarily optimal in this setting:

Figure 6: ACCESS DENIED demonstrates that m can attract traffic from its customer c over the direct m-c link by denying export to some of m's other neighbors. Here, the network has path and loop verification, next-hop policies at every node, and m is interested in attracting traffic only from c (but not from n) in an AT4 attraction. Suppose that all nodes, including m, honestly announce paths. On the left we present the outcome when every node, including m, uses export-all. On the right, we illustrate the outcome when m uses a different all-or-nothing export rule: in particular, m announces all paths (honestly) to c, and no paths to n. As a result, m attracts traffic from c on the direct c-m link. If m had announced paths to n, then c would not have sent its traffic on the c-m link, as in the outcome on the left. Thus, we see that the export-all rule is not optimal for m. (Note that ACCESS DENIED is a network that obeys GR1, GR3, and AT4, and has no dispute wheel.)

We pause here to observe that in the outcome on the right, n has no path to the destination if node c only exports the paths allowed by GR2. We discuss this issue in Section 6.3.

5.6 Theorem 5.1 needs all-or-nothing export.

The requirement that all nodes use an all-or-nothing export policy in Theorem 5.1 is extremely strong, especially because most networks that obey the Gao-Rexford conditions (in particular GR2) violate this export rule. We now present our most devastating (and complicated) counterexample that shows Theorem 5.1 does not hold with a more realistic export rule like consistent export:

Figure 7: GRANDMA demonstrates that a manipulator m can have an incentive to lie in order to attract traffic from a customer c if some other node a does not use an all-or-nothing export policy. Furthermore, GRANDMA shows that this is possible even when all nodes use path verification and next-hop policies.

In GRANDMA, m has an AT4 attraction relationship with its customer c, a traffic-volume attraction relationship with its provider b, and no other attractions. Suppose now that all nodes export all paths allowed by GR2; thus, a does *not* export paths through its peer 1 to its peer c. While a uses a consistent export rule (since a filters only its lowest ranked path through 1), a does not use

[6] We remark that this result only rules out the possibility of obtaining a better *stable* outcome by lying, it does not rule out the possibility of m gaining utility by inducing a non-stable outcome. See Section 2.2.

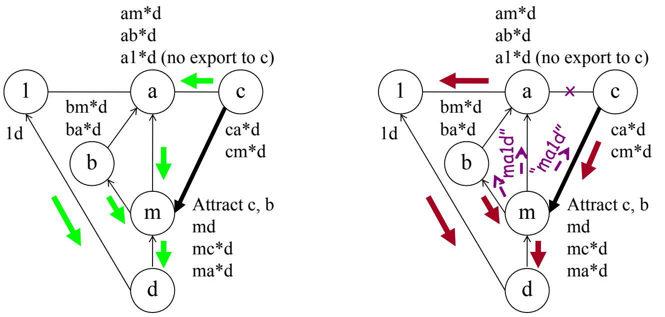

Figure 7: GRANDMA.

all-or-nothing export rule. On the left is the outcome T that results when all nodes act honestly, *i.e.,* use BGP-compliant strategies with $r_n(\cdot) \equiv v_n(\cdot)$ and the export rules above. The manipulated outcome M is shown on the right, where only the manipulator m deviates from the BGP-compliant strategies above.

In M, the manipulator m dishonestly announces the path "$ma1d$" while actually routing on md. To arrive at the outcome M on the right, node m sits quietly until node a exports "$a1d$" to it. Then m announces "$ma1d$" to all nodes, while routing on md in the data plane. Node a cannot route through m (because it thinks that m routes through it); so, a continues to route on $a1d$. Next, because a does not export paths through 1 to its peer node c, node c has no choice but to route through node m. Meanwhile, m's machinations have no effect on b, who routes through m regardless. Notice that loop or path verification would not help, since node a is indeed routing along "$a1d$". Furthermore, m manages to retain in M its traffic-volume attraction with b and gain an AT4 attraction with customer c. Also, m has no BGP-compliant strategy that obtains as large a utility as it obtains from M. (Note that GRANDMA a Gao-Rexford network with no dispute wheel *that does not obey AT4*, where all nodes use next-hop policy with all their neighbors.)

6. RESULTS: CUSTOMER ATTRACTIONS IN GAO-REXFORD NETWORKS

We now focus on Gao-Rexford networks (see Section 3.3). In Section 5, we used GRANDMA (Figure 7) to show that Theorem 5.1 does not hold with consistent export in place of the unrealistic all-or-nothing export rule (which is usually not compatible with GR2). Fortunately, GRANDMA did not obey the AT4 attraction condition. Thus, we now weaken the assumption of all-or-nothing export by focusing on the AT4 setting, in which an attractor can increase its utility only if a *customer* routes on the direct link between them. It turns out that AT4 also allows us to weaken the next-hop-policy restrictions required in Theorem 5.1. Our results are summarized in Table 4, which also shows how dropping any one of the conditions in our positive result (Section 6.2) may create an incentive to lie.

6.1 It's not sufficient to restrict policy at attractees only.

The requirement in Theorem 5.1 that every node in the network uses a next-hop policy with all of its neighbors is very strong indeed. Ideally, we would have preferred to require only *attractees* to use next-hop policy with their attractors. Unfortunately, even requiring every attractee to use next-hop policy with *all its neighbors* need not remove the incentive to lie:

Figure 8: In ORION only the attractee (node c) uses next-hop policy with all its neighbors (nodes m, n). (Every other node uses next-hop policy with its peers and providers, but not necessarily with its customers.) Notice that node a is not policy consistent

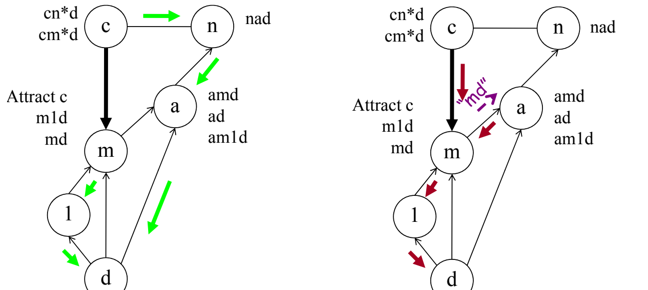

Figure 8: ORION.

with its customer m: node m prefers path $m1d$ to path md (say, because it is cheaper to route directly to 1), while node a prefers the path amd to the path $am1d$ (say, because it prefers shorter paths).

On the left is the outcome T that results when each node uses a BGP-compliant strategy with $r_n(\cdot) \equiv v_n(\cdot)$, exporting all paths allowed by GR2. The manipulated outcome M is shown on the right, where the manipulator m deviates from this BGP-compliant strategy. In the manipulated outcome M, m dishonestly announces the outgoing path "md" to all of its neighbors so that node a decides to route through m on the amd path. However, node n does not admit the path amd and thus is left with no path to the destination d. The attractee c has no choice but to route through m, increasing m's utility. Observe that m has no BGP-compliant strategy that obtains as large a utility as it obtains from M. (ORION is a Gao-Rexford network with no dispute wheel that obeys AT4.)

Notice that n uses a "forbidden-set policy" [8], in which it prefers using no path at all over using a path through m. Such preferences could arise in practice if node n does not trust node m to carry its traffic (say, because it perceives node m to be adversarial).

6.2 Policy consistency everywhere with next-hop policy at attractees is enough!

Earlier, we saw that, even in the Gao-Rexford setting with AT4, dropping either path or loop verification may create an incentive to lie (as in FALSE LOOP in Figure 5). Furthermore, from ORION above, we learn that policy restrictions only on attractees can leave an incentive to lie. The manipulation in ORION is possible because node a is not policy consistent with node m; we now show that requiring policy consistency, along with other conditions satisfied by ORION, is enough to ensure no incentive to lie.

THEOREM 6.1. *Consider a policy-consistent, Gao-Rexford network that obeys AT4, in which there is no dispute wheel in the valuations and all attractees use next-hop policies with their providers and peers. Suppose that all nodes, except a single manipulator node m, uses a BGP-compliant strategy with $r_n(\cdot) \equiv v_n(\cdot)$ and a consistent export rule that satisfies GR2. Suppose further that the network has path or loop verification.*

Then there exists a BGP-compliant strategy for m with $r_m(\cdot) \equiv v_m(\cdot)$ and a consistent export rule obeying GR2 that obtains the best possible stable *outcome in terms of the utility function of m. In particular, exporting all paths to customers and no paths to providers and peers is one such optimal strategy.*

Notice that in addition to obeying the Gao-Rexford conditions, we also separately require that there is no dispute wheel in the valuations; because of how we model export and define valuations, the former condition does not imply the latter, which is a subtle difference from [13]. Further discussion of these subtleties, and the proof of this theorem, is in the full version [18].

AT4	Verification	Policy Consistency	Next-hop policy	Export	Incentive to Lie?	Result
No	⋆	⋆	⋆	Consistent	Yes	GRANDMA
Yes	None	⋆	⋆	⋆	Yes	FALSE LOOP
Yes	⋆	None	All nodes with peers & providers	⋆	Yes	ORION
Yes	None / Loop	All nodes	None	⋆	Yes	NONEXISTENT PATH
Yes	Loop / Path	All nodes	Attractees with peers & providers	Consistent	No	Theorem 6.1

Table 4: Summary of our results for Gao-Rexford networks (obeying GR1-GR3) with no dispute wheel.

6.3 It's best to export only to your customers.

Observe that Theorem 6.1 not only shows the *existence* of an optimal export rule for the manipulator, but also explicitly describes one such export rule. It therefore provides a specific strategy from which no node has an incentive to unilaterally deviate.[7] However, this strategy requires that m *never announces any paths to its peers and providers*. While this export rule obeys consistent export and GR2, a network in which every node uses this "export-nothing-to-non-customers" rule would be a very sorry network indeed: Peer paths would not exist, and nodes would never transit traffic from their providers, even if that traffic is destined for their customers!

Unfortunately, there are cases in which the optimal export rule for the manipulator is to "export nothing to non-customers." For example, consider ACCESS DENIED in Figure 6 and observe that m's optimal strategy is to announce no paths to n (which means that when c's export rule obeys GR2, n no longer has a path to the destination). Furthermore, this network obeys the strongest conditions considered in this work (next-hop policy at all nodes and path verification). Hence, within the conditions considered here, we cannot hope to get a result where m's optimal export policy necessarily allows it to announce paths to peers and providers.

This suggests that AT4 may not be a reasonable model for attraction relationships, *e.g.,* a node could improve its utility by attracting traffic from a provider or peer if it *delivers* this traffic to a customer. Finding a more appropriate model for attraction relationships in Gao-Rexford networks remains open for future research.

7. RELATED WORK

We discussed some related work in Sections 1–2. Further discussion is below. Griffin, Shepherd, and Wilfong [21] developed a formal model of BGP which assumes ASes choose paths based on an arbitrary preference function that ranks *outgoing paths*. They used this model to initiate a study of sufficient conditions to ensure that BGP converges to a unique outcome (Section 3.1). This study was continued by many subsequent works; most relevant here are the results of Gao and Rexford [14] who considered constraints that arise due to business relationships between ASes (Section 3.3), and those of Feamster, Johari, and Balakrishnan [7] who studied the effect of filtering (Section 3.4).

In contrast to the works on convergence, the game theoretic studies of BGP [6,8–12,28,33], discussed in Section 1.2 and throughout this paper, looked for mechanisms that induce incentives to comply with the protocol (which, in particular, means that ASes would have no incentive to lie). These works interpret the preference function in Griffin *et al.* [21] as a measure of utility for each AS, and model ASes as rational agents who act selfishly to maximize utility. This is equivalent to assuming that utility is uniquely determined by *outgoing paths*. To our knowledge, our work is the first to model the effect of *incoming traffic* on the *incentive* to lie in BGP announcements. Earlier versions of our work appeared as [17] and [25].

Recently, the literature on BGP convergence has begun to model the effect of *incoming traffic* on BGP dynamics. These works [15, 38, 39] focus on the context of traffic engineering, and assume that ASes honestly announce paths; they do not consider ASes that lie. Gao, Dovrolis and Zegura [15] and Wang *et al.* [38] study algorithms for traffic attraction and deflection using AS-path prepending. (Our work does not model prepending.) Wang *et al.* [39] study oscillations that can occur if the BGP decision process depends on incoming traffic as well as outgoing paths. In contrast, our work allows *utility* to depend on incoming traffic (Section 2.3) but assumes that the BGP dynamics are based on *ranking* functions (Section 2.2) that depend only on outgoing paths. The ranking functions are derived from a "compilation" of the utility function (Section 2.5). Thus, in some sense, Wang *et al.* study the oscillations that can result as ASes continuously adjust their compilation. Indeed, Figure 2 of [39] shows conditions under which INCONSISTENT POLICY in (our) Figure 2 could experience such oscillations.

8. CONCLUSIONS

In this work, we considered control-plane mechanisms that provide incentives for rational ASes to announce their true data-plane paths in BGP messages. We find that conditions previously shown to be sufficient for honesty no longer suffice if we assume that ASes can benefit by attracting incoming traffic from other ASes. We demonstrated that, within the *control-plane* mechanisms we considered here, ensuring honesty in the face of traffic attraction requires very strong restrictions on routing policy (at the very least, policy consistency everywhere, and sometimes also next-hop policy at certain ASes), as well as control-plane verification (loop-verification or path-verification protocols like S-BGP [26]). Thus, our results suggest that in practice, it will be difficult to achieve honesty without resorting to expensive *data-plane* protocols that verify and enforce AS-level paths. By highlighting the difficulty of matching the control and data planes, even under the assumption that ASes are rational (and not arbitrarily malicious), our results can also help inform decisions about whether security protocols should be deployed in the control plane, in the data plane, or in both.

Acknowledgments

We thank Jennifer Rexford, Michael Schapira and Joan Feigenbaum for discussions and valuable feedback that has greatly improved this work. We also thank Boaz Barak, Matthew Caesar, Andreas Haeberlen, Martin Suchara, Gordon Wilfong, and the anonymous reviewers for useful comments.

9. REFERENCES

[1] K. Argyraki, P. Maniatis, O. Irzak, A. Subramanian, and S. Shenker. Loss and Delay Accountability for the Internet. In *Proc. IEEE ICNP*, pp. 194–205, Oct. 2007.

[2] I. Avramopoulos and J. Rexford. Stealth Probing: Data-Plane Security for IP Routing. In *Proc. USENIX*, Jun. 2006.

[3] H. Ballani, P. Francis, and X. Zhang. A Study of Prefix Hijacking and Interception in the Internet. In *Proc. ACM SIGCOMM*, pp. 265–276, Aug. 2007.

[7] However, as in Theorem 5.1, we add the disclaimer that this result only applies to *stable* manipulated outcomes.

[4] S. Balon and G. Leduc. Can Forwarding Loops Appear When Activating iBGP Multipath Load Sharing? In *Proc. AINTEC*, LNCS 4866, pp. 213–225, Nov. 2007.

[5] K. Butler, T. Farley, P. McDaniel, and J. Rexford. A Survey of BGP Security Issues and Solutions. Technical report, AT&T Labs–Research, 2005.

[6] R. R. Dakdouk, S. Salihoglu, H. Wang, H. Xie, and Y. R. Yang. Interdomain Routing as Social Choice. In *Proc. Incentive-Based Comp. (IBC)*, Jul. 2006.

[7] N. Feamster, R. Johari, and H. Balakrishnan. Implications of Autonomy for the Expressiveness of Policy Routing. *IEEE/ACM Trans. Network.* 15(6):1266–1279, Dec. 2007.

[8] J. Feigenbaum, D. R. Karger, V. Mirrokni, and R. Sami. Subjective-Cost Policy Routing. *Theoretical Comp. Sci.* 378(2):175–189, Jun. 2007.

[9] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker. A BGP-Based Mechanism for Lowest-Cost Routing. *Distributed Computing* 18(1):61–72, Jul. 2005.

[10] J. Feigenbaum, V. Ramachandran, and M. Schapira. Incentive-Compatible Interdomain Routing. In *Proc. ACM Conf. Elec. Commerce (EC)*, pp. 130–139, Jun. 2006.

[11] J. Feigenbaum, R. Sami, and S. Shenker. Mechanism Design for Policy Routing. *Distributed Computing* 18(4):293–305, Mar. 2006.

[12] J. Feigenbaum, M. Schapira, and S. Shenker. Distributed Algorithmic Mechanism Design. Chap. 14 (pp. 363–384) in *Algorithmic Game Theory*, N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, eds. Cambridge UP, Sep. 2007.

[13] L. Gao, T. G. Griffin, and J. Rexford. Inherently Safe Backup Routing with BGP. In *Proc. IEEE INFOCOM*, vol. 1, pp. 22-26, Apr. 2001.

[14] L. Gao and J. Rexford. Stable Internet Routing without Global Coordination. *IEEE/ACM Trans. Network.* 9(6):681–692, Dec. 2001.

[15] R. Gao, C. Dovrolis, and E. Zegura. Interdomain Ingress Traffic Engineering through Optimized AS-Path Prepending. In *Proc. IFIP Networking*, May 2005.

[16] V. Gill, J. Heasley, and D. Meyer. The Generalized TTL Security Mechanism (GTSM). RFC 3682, Feb. 2004.

[17] S. Goldberg and S. Halevi. Rational ASes and Traffic Attraction: Incentives for Honestly Announcing Paths in BGP. Technical Report TR–813–08, Princeton Univ. Dept. of Comp. Sci., Feb. 2008.

[18] S. Goldberg, S. Halevi, A. D. Jaggard, V. Ramachandran, and R. N. Wright. Rationality and Traffic Attraction: Incentives for Honest Path Announcements in BGP. Technical Report TR–823–08, Princeton Univ. Dept. of Comp. Sci., Jun. 2008.

[19] S. Goldberg, D. Xiao, E. Tromer, B. Barak, and J. Rexford. Path Quality Monitoring in the Presence of Adversaries. In *Proc. ACM SIGMETRICS*, Jun. 2008.

[20] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and A. Rubin. Working Around BGP: An Incremental Approach to Improving Security and Accuracy of Interdomain Routing. In *Proc. NDSS*, Feb. 2003.

[21] T. G. Griffin, F. B. Shepherd, and G. Wilfong. The Stable Paths Problem and Interdomain Routing. *IEEE/ACM Trans. Network.* 10(2):232–243, Apr. 2002.

[22] A. Heffernan. Protection of BGP Sessions via the TCP MD5 Signature Option. RFC 2385, Aug. 1998.

[23] K. J. Houle and G. M. Weaver. Trends in Denial of Service Attack Technology. Technical Report, CERT Coordination Center, Oct. 2001.

[24] G. Huston. Interconnection, Peering, and Settlements. In *Proc. Internet Glob. Summit (INET)*, Jun. 1999.

[25] A. D. Jaggard, V. Ramachandran, and R. N. Wright. Towards a Realistic Model of Incentives in Interdomain Routing: Decoupling Forwarding from Signaling. Technical Report 2008–02, DIMACS, Rutgers Univ., Apr. 2008.

[26] S. Kent, C. Lynn, and K. Seo. Secure Border Gateway Protocol (S-BGP). *J. Selected Areas in Communications* 18(4):582–592, Apr. 2000.

[27] H. Levin, M. Schapira, and A. Zohar. The Strategic Justification for BGP. Technical Report, Hebrew Univ. of Jerusalem, 2006.

[28] H. Levin, M. Schapira, and A. Zohar. Interdomain Routing and Games. In *Proc. ACM STOC*, May 2008.

[29] X. Liu, X. Yang, D. Wetherall, and T. Anderson. Efficient and Secure Source Authentication with Packet Passports. In *Proc. USENIX Wkshp. Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, Jul. 2006.

[30] Z. Mao, J. Rexford, J.Wang, and R. H. Katz. Towards an Accurate AS-Level Traceroure Tool. In *Proc. ACM SIGCOMM*, pp. 365–378, Aug. 2003.

[31] N. Nisan and A. Ronen. Algorithmic Mechanism Design. *Games and Economic Behavior* 35(1–2):166–196, Apr. 2001.

[32] V. Padmanabhan and D. Simon. Secure Traceroute to Detect Faulty or Malicious Routing. *Proc. HotNets-I*, pp. 77–82, Oct. 2002.

[33] D. C. Parkes and J. Shneidman. Specification Faithfulness in Networks with Rational Nodes. In *Proc. ACM PODC*, pp. 88–97, Jul. 2004.

[34] A. Ramachandran and N. Feamster. Understanding the Network-Level Behavior of Spammers. In *Proc. ACM SIGCOMM*, pp. 291–302, Sep. 2006.

[35] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271, Jan. 2006.

[36] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. H. Katz. Listen and Whisper: Security mechanisms for BGP. In *Proc. USENIX NSDI*, Mar. 2004.

[37] F. Wang and L. Gao. On Inferring and Characterizing Internet Routing Policies. In *Proc. ACM IMC*, pp. 15–26, Oct. 2003.

[38] H. Wang, R. K. Chang, D.-M. Chiu, and J. C. Lui. Characterizing the Performance and Stability Issues of the AS Path Prepending Method. In *Proc. ACM SIGCOMM Asia Workshop*, Apr. 2005.

[39] H. Wang, H. Xie, Y. R. Yang, L. E. Li, Y. Liu, and A. Silberschatz. On the Stability of Rational, Inbound-Dependent Interdomain Route Selection. In *Proc. IEEE ICNP*, pp. 40–52, Nov. 2005.

[40] R. White. Deployment Considerations for Secure Origin BGP (soBGP). Internet Draft (expired), `draft-white-sobgp-bgp-deployment-01.txt`, Jun. 2003.

[41] E. L. Wong, P. Balasubramanian, L. Alvisi, M. G. Gouda, and V. Shmatikov. Truth in Advertising: Lightweight Verification of Route Integrity. In *Proc. ACM PODC*, pp. 147–156, Aug. 2007.