DIMACS Technical Report 2008-02
Version of 2008–04–27
April 2008

Towards a Realistic Model of Incentives in Interdomain Routing: Decoupling Forwarding from Signaling

by

Aaron D. Jaggard[1,2]
DIMACS
Rutgers University
adj@dimacs.rutgers.edu

Vijay Ramachandran[3]
Department of Computer Science
Colgate University
vijayr@cs.colgate.edu

Rebecca N. Wright[4,5]
DIMACS and Department of Computer Science
Rutgers University
rebecca.wright@rutgers.edu

[1]DIMACS Assistant Research Professor
[2]Partially supported by NSF award CNS–0753492 and ONR award N00014–05–10818. Work done in part while at the Dept. of Mathematics, Tulane University, partially supported by NSF award DMS–0239996 and ONR award N00014–05–10818.
[3]Partially supported by NSF award CNS–0753061. Work done in part while a DIMACS postdoc at the Stevens Institute of Technology, partially supported by NSF award CNS-0524139, ONR award N00014–05–10818, and by the Stevens Technogenesis program.
[4]DIMACS Deputy Director
[5]Partially supported by NSF awards CNS–0751674 and CNS–0753061.

ABSTRACT

We model the task of interdomain routing—the task of connecting the networks that compose the Internet—as an iterative, highly distributed, asynchronous game. Unlike previous examinations of this game that assumed quasi-linear utilities, we assume that each node has a *quasi-bilinear* utility depending not only on the route it believes it is assigned in the outcome, but also on other nodes assigned to route through it. This more realistic model captures out-of-band business relationships that may affect nodes' behavior in the game and the difficulty of monitoring traffic flows on the Internet. We show by example that conditions that guarantee incentive compatibility when utility does not depend on signaling do not provide this assurance in the model we study. We also extend the Stable Paths Problem to decouple forwarding from signaling and show that this allows stable signaling solutions to have forwarding loops, and we give a sufficient condition to prevent this. Finally, we provide positive results about incentive compatibility when using utility functions that depend on both forwarding and signaling; this relies on nodes having *next-hop policies* (so that their forwarding preferences depend only on the next hops of available routes) and certain other assumptions. In conjunction with these results, we provide examples of networks that violate these conditions and in which nodes have incentive to lie about their chosen paths.

Contents

1 Introduction

The Internet comprises many subnetworks, called Autonomous Systems (ASes), that are administered by independent entities. Most Internet traffic is destined for an AS other than the one in which it originates; because most ASes are not directly inter-connected, traffic often traverses several subnetworks on its path from source to destination. The routes used to forward traffic depend on a composition of decisions made throughout the Internet, which depend on information exchanged among ASes. The task of determining these routes is called *interdomain routing*, and the protocol currently used to exchange information for that purpose is called the *Border Gateway Protocol* (BGP).

The process as prescribed by BGP is asynchronous and iterative: Each time an AS learns new information about available routes, it recalculates its best option based on its own preferences and alerts neighboring ASes of its choice. A stable route assignment is reached if this process converges to a point at which each AS has chosen its best available option, and no route changes are announced. Thus, interdomain routing can be modeled naturally as a multi-round game [LSZ06] in which nodes make strategic choices about what route to use to *forward* data and what routes, if any, to *signal* to neighbors for them to use; a stable route assignment is a Nash equilibrium of the game. (A discussion of various equilibria and related game-theoretic concepts is given in Appendix A.) While an AS's forwarding choice affects the path taken by traffic from that AS to the destination, an AS's signaling action affects the flow of traffic from other ASes because signaling announcements are how other ASes learn about potential routes they can use.

Physical inter-AS connections related to long-term, out-of-band business relationships that are often predicated on assumptions about traffic flow between ASes [Hus99]. An AS may thus want traffic to flow in ways that match certain assumptions in order to benefit from these relationships. An AS may also want to engage in *ingress traffic engineering*, distributing incoming traffic among various connections for maintenance or cost reasons [YXW+05], or it may want to maliciously snoop on other traffic in the network. In light of these factors, utility functions in the interdomain routing game should consider incoming traffic in order to most realistically model the real-world incentives of ASes. However, in previous game-theoretic analyses of interdomain routing a node's utility was defined to depend only the route that its traffic takes (which might differ from the route that the node selects) and, possibly, payments from a mechanism. Using that general approach, it has been shown that following BGP as prescribed—choosing the best forwarding option based on one's preferences and signaling that route to others—is (1) incentive compatible in ex-post Nash and welfare-maximizing with certain policy restrictions [FRS06] and (2) incentive compatible in ex-post Nash if certain security assumptions hold [LSZ06]. We show here by example that these conditions no longer guarantee incentive compatibility (in ex-post Nash) when our utility functions, which also depend on incoming traffic, are used. Our utility functions also differ from earlier work in that they depend on the route selected by a node and not the route taken by data; we believe this more realistically captures nodes' motivation.

From a networking perspective it is also realistic to decouple forwarding from signaling.

Internet forwarding is destination-based, and once packets are sent to the next router along their path to the destination, the packets become the next router's responsibility. There is no mechanism for the source of a traffic flow to have control over the actual path followed, and it is difficult to detect the path actually taken [PS02]. Not only is it possible for the route actually by data to differ from the route apparently taken (*i.e.*, the route selected), but this is actually seen in the Internet [MRWK03].

This motivates our work here, in which we examine the interdomain-routing game with *quasi-bilinear* utilities that have separate components for forwarding and signaling. As mentioned above, conditions that guarantee incentive compatibility when utility does not depend on signaling do not provide this assurance in the model we study. In the rest of this section, we first discuss related work and then provide a detailed summary of our work, which includes positive results both for the static analysis of policies and for the interdomain-routing game.

1.1 Related work

The interdomain-routing problem and BGP have been studied from both networking and game-theoretic perspectives. A static model for the interdomain-routing problem, called the *Stable Paths Problem*, was introduced by Griffin, Shepherd, and Wilfong [GSW02]. They showed that determining whether or not a stable route assignment exists (*i.e.*, whether or not playing the interdomain-routing game by following BGP would converge to a stable forwarding tree) is NP-complete. However, they gave a condition on nodes' preferences sufficient to guarantee that a unique, stable route assignment exists—akin to a Nash equilibrium for the interdomain-routing game. This condition involves the absence of a structure called a *dispute wheel* that involves the relative preferences of various paths in the network. Checking that an SPP instance is dispute-wheel free requires complete knowledge of nodes' preferences, which is unrealistic in the Internet. Gao and Rexford [GR01] showed that if nodes' preferences were dictated by business relationships that underlie today's commercial Internet [Hus99], running BGP would guarantee reaching a stable route assignment. Follow-up work showed that those assumptions on nodes' preferences precluded the existence of a dispute wheel [GGR01].

Feigenbaum *et al.* [FPSS05] introduced an efficient, distributed, strategyproof mechanism for computing lowest-cost routes in which nodes' private information is a per-packet transit cost. This mechanism relies on using VCG payments to incentivize nodes. Without the dispute-wheel freeness conditions on nodes' preferences, generalizing the mechanism beyond lowest-cost routing led mainly to negative results [FKMS05, FSS04], showing that obtaining welfare maximization is hard. (In general, any metric-based preference implies that there is no dispute wheel [FRS06].) Welfare maximization was also studied from a networking perspective in [Sob03], in which routing-policy constraints were identified that permitted a "globally optimal" route assignment.

The first positive result towards more general welfare-maximizing mechanisms for interdomain routing appeared in work of Feigenbaum *et al.* [FRS06], which combined the networking and mechanism-design approaches. They showed that if three properties (including no dispute wheel) hold on nodes' preferences, following BGP leads to a welfare-maximizing route allocation; in addition, following BGP is incentive compatible in ex-post Nash equilibrium. Unlike the lowest-cost-route mechanism of [FPSS05], no payments are required when nodes' utilities are positive (*i.e.*, when no node incurs a cost to participate). Levin *et al.* [LSZ06] strengthened this result, showing that following BGP is incentive compatible in ex-post Nash equilibrium when only the dispute-wheel-freeness condition holds on nodes' preferences, as long as nodes are not able to announce routes that do not exist (a property called *route verification*). These results assume that a node's utility is based only on that node's forwarding route; in particular, signaling actions are assumed not to affect a node's utility unless it changes the forwarding route.

Earlier work in network security points out the difficulty of route verification and traffic detection. Various work—*e.g.*, [KLS00]—studies the threat of injecting false routes into the interdomain-routing process and includes proposed protocol changes to prevent it; however, many of these proposals require deployment of a private-key infrastructure, something that has been met with resistance. Padmanabhan and Simon [PS02] outline the threat of thwarting traces to detect what route traffic actually follows; again, their tools require deployment of a secure routing infrastructure. These difficulties motivate the examination of an interdomain-routing model that includes the ability to take advantage of these threats in a strategic way.

1.2 Our contributions

We start by extending the Stable Paths Problem (SPP) to a new framework, the Forwarding/Signaling-Stable Paths Problem (FS-SPP), in a way that decouples the forwarding and signaling actions that are distinct in real-world routing. This allows for a static analysis of policies in a way that better models the possible actions of ASes. We translate the dispute-wheel-freeness condition to a natural analogue that guarantees *signaling stability* in FS-SPP. Unlike the original SPP, our new model allows stable signaling solutions that still have forwarding loops in which data is forwarded endlessly without reaching the destination (this requires nodes to lie about their routes and thus cannot be modeled in SPP). We define a condition, similar to dispute-wheel-freeness but without a direct analogue in SPP, that precludes stable signaling solutions with forwarding loops.

We translate the Gao-Rexford constraints on SPP to our new framework by considering the motivation that makes these constraints "natural" in today's Internet. We show that if the resulting constraints are satisfied by an FS-SPP instance, then any stable signaling solution for that instance will not have forwarding loops; however, these natural constraints are no longer enough to guarantee robust signaling. This provides additional motivation to study the incentive compatibility of truthful signaling.

Motivated by this static analysis and the intrinsic interest of having a more realistic economic model for network routing, we introduce utility functions in which forwarding and signaling actions can be decoupled for benefit (which we call *quasi-bilinear utilities*). In this setting, we consider the incentive compatibility of BGP in the ex-post Nash solution concept. We show that following BGP is not incentive-compatible with quasi-bilinear utilities in networks missing any of the properties (*e.g.*, dispute-wheel freeness) needed for the positive results (with quasi-linear utilities) of [LSZ06, FRS06]. We give several examples of negative results to demonstrate this.

We examine utility functions in which the contribution to an AS's utility from signaling actions is based on *all* nodes whose traffic is carried by the AS (not just neighbors). We show that following BGP is always incentive compatible when nodes have next-hop policies, if nodes do not arbitrarily exclude (filter) routes. However, we also show that following BGP is not always incentive compatible, even with next-hop policies, if nodes are able to arbitrarily exclude routes. We also examine utility functions in which the signaling contribution to an AS's utility depends only on the nodes that forward along routes on which the AS in question is the *next hop* (not just appearing at some point on the path). We show that a node can unilaterally act strategically to add a particular node to this set, but that if the network does not have a dispute wheel then unilateral strategic action cannot increase the size of this set (*i.e.*, a strategic node may be able to affect which of its neighbors choose it as their next hop, but the node cannot increase the total number of such neighbors).

Finally, this work illuminates additional nuances of the relationship between interdomain routing and incentive compatibility. Decoupling forwarding from signaling leads to a more precise treatment of filtering, which we study as non-strategic behavior (in keeping with real-world networks). By looking at positive and negative results, we study the effects of different policy conditions on incentive compatibility. Not only does our work consider forwarding utility as only part of the overall utility, but we also shift the focus from the route followed by data (which is unknown to an AS) to the route chosen by an AS (which we believe is more closely tied to the ASes' behavior).

In Sec. 2 we extend SPP to decouple forwarding from signaling and consider the properties of numerous examples; we also use the motivation for the Gao-Rexford conditions to describe analogous natural conditions for this new model and describe consequences of these conditions. In Sec. 3 we briefly review the interdomain-routing game. Sections 4 and 5 present our results for this model with our quasi-bilinear utility functions, the former section focusing on negative results by example and the latter section focusing on positive results.

2 Extending SPP

2.1 Network model

The Internet topology is given as an undirected graph $G = (V, E)$ in which nodes V correspond to ASes and links E correspond to connections between ASes. Let $N(v)$ be the set of v's *neighbors*, *i.e.*, $N(v) = \{w \in V \mid (w, v) \in E\}$. Because routes to different destinations are computed independently, we assume that V contains a predetermined destination node d.

Unless noted otherwise, we only consider *simple routes*, *i.e.*, routes that are loop-free. Let $\mathcal{S}_v$ represent the simple routes in G from source node v to the destination d, including ϵ, the null route; let $\mathcal{S} = \cup_{v \in V} \mathcal{S}_v$ be the set of all simple routes. We write routes as a sequence of nodes from source to destination, *e.g.*, $R_1 = v_1 v_2 \cdots v_k d$. The *next hop of* v_i along R is the node appearing after v_i in the sequence R, *e.g.*, v_k's next hop on R_1 is d. Let $\mathsf{next}(R)$ to be the source node's next hop along R, *e.g.*, $\mathsf{next}(R_1) = v_2$. We write xR to denote the route R extended to x (x to the source node of R, then R to the destination); in general, QR represents the concatenation of routes Q and R.

The goal is to find a *forwarding-route assignment* $\pi : V \rightarrow \mathcal{S}$ such that $\pi(v)$ is a simple route from v to d. These routes represent the routes used to send traffic from each AS to d. Ideally, we want the assignment to be consistent: π should form a confluent tree to the destination d. Unlike previous work in this area, we make a distinction between the routes that nodes *believe* their traffic follows and the routes that nodes *actually use* to forward traffic. As we discuss below, inconsistent information exchanged during the routing process could lead to a *perceived forwarding assignment* that is a confluent tree, while the actual forwarding routes form a digraph containing cycles. Because we are concerned about the potential for data loss resulting from forwarding loops, our goal will be to assign actual forwarding routes that form a tree.

We argue that without some sort of hardware authentication, a router cannot *prove* to one of its neighbors that it is forwarding data along the particular route—*i.e.*, along the physical link to the next hop of the route—that it is advertising; this argument is based on the disconnect between the physical *data or forwarding plane* and the way routes are advertised in the *control or signaling plane*. This is compounded by the possibility of thwarting tools to detect the actual paths of network flows, as outlined by Padmanabhan and Simon [PS02]. Furthermore, while after-the-fact detection of misrouting may be possible, it is not useful if there is there is significant harm done or loss incurred by a short-term misroute/disruption. These difficulties provide motivation for our study of when ASes have no incentive to lie about how they are forwarding data.

2.2 A model for signaling/forwarding

Decoupling of signaling from forwarding cannot be captured by the original definition of the *Stable Paths Problem* (SPP) [GSW02] that provides a formal model for analyzing policy interactions in interdomain routing. We define a more general model, the *Forwarding/Signaling Stable Paths Problem* (or FS-SPP) as follows.

Definition 2.1. An instance of the *Forwarding/Signaling Stable Paths Problem* (an *FS-SPP instance*) is

- An undirected graph $G = (V, E)$ with a distinguished vertex d.
- For each node v a set $\mathcal{P}_v$ of *permitted paths* to d, each of which starts at v, ends at d, and does not contain any loops and such that if $vuP \in \mathcal{P}_v$ and $u \neq d$, then $uP \in \mathcal{P}_u$.
- For each node v and each adjacent node $w \in N(v)$, a *signaling-preference function* $\sigma_{v,w} : \mathcal{P}_v \rightarrow \mathbb{Z}$.

- For each node v, a *forwarding-preference function* $\phi_v : \mathcal{P}_v \to \mathbb{Z}$.

We will write $\mathcal{P}$ for the union of all sets of permitted paths in an instance, Σ for the collection of signaling-preference functions, and Φ for the collection of forwarding-preference functions; we will then write $(G, \mathcal{P}, \Sigma, \Phi)$ for the instance. Larger values of the various preference functions indicate routes that are more preferred.

Note that $\sigma_{v,w}(P)$ is defined even if P is a route that goes through w. It appears that this is necessary if we want FS-SPP to generalize SPP; in particular, we use it to realize the SPP Disagree in Example 2.7.

Throughout this section, we use σ and ϕ for functions related to signaling and forwarding, respectively; the existence and type of the subscripts determine which aspect of signaling/forwarding is being considered in a particular context. When subscripted with a vertex (*e.g.*, ϕ_v) or pair of vertices (*e.g.*, $\sigma_{v,u}$), the function is a preference function; these are defined on paths and capture the local policies of v (with respect to its neighbor u in the case of $\sigma_{v,u}$).

A function ϕ or σ without subscripts is typically an assignment function (Def. 2.2); this determines the next-hop for forwarding (ϕ) or the path signaled by one vertex to another (σ) and does not depend on the preference functions (although the stability of an assignment does depend on preferences).

Finally, ϕ_σ (Def. 2.2) is a particular forwarding assignment that is induced by the signaling assignment σ.

Definition 2.2 (Path assignments and solutions)**.** Given an FS-SPP instance $(G, \mathcal{P}, \Sigma, \Phi)$, a *signaling assignment* for the instance is a partial function $\sigma : V \times V \rightharpoonup \mathcal{P}$ such that $\sigma(v, w)$ is defined iff $w \in N(v)$, and $\sigma(v, w) \in \mathcal{P}_v$ whenever it is defined. For a signaling assignment σ, we define the set of *known paths at* v by $\mathcal{K}_v(\sigma) = \{v\sigma(u, v) | u \in N(v),\ v\sigma(u, v) \in \mathcal{P}_v\}$. These are the paths that v learns from its neighbors (*i.e.*, they are announce by v neighbors and permitted at v). A *forwarding assignment* is a function $\phi : V \setminus \{d\} \to V$ such that $\phi(v) \in N(v)$ for every node v.

We say that a signaling assignment is a *(stable) signaling solution* if every node v announces to each of its neighbors w the most-preferred (according to its signaling-preference function for announcements to w) path that it knows. More formally, σ is a stable signaling solution if $\sigma(v, w) \in \mathcal{K}_v(\sigma)$ whenever $\sigma(v, w)$ is defined and

$$\forall Q \in \mathcal{K}_v(\sigma),\ Q \neq \sigma(v, w) \Rightarrow \sigma_{v,w}(Q) < \sigma_{v,w}(\sigma(v, w)).$$

For a signaling assignment σ, the *forwarding assignment induced by* σ is obtained by mapping each node $v \neq d$ to the next hop on v's most-preferred (according to its forwarding-preference function) known route; we denote this function by ϕ_σ. More formally, for each $v \in V \setminus \{d\}$ we choose the route $P_v \in \mathcal{P}_v$ that satisfies $P_v \in \mathcal{K}_v(\sigma)$ and

$$\forall Q \in \mathcal{K}_v(\sigma),\ Q \neq P_v \Rightarrow \phi_v(Q) < \phi_v(P_v)$$

and let $\phi_\sigma(v) = \mathsf{next}(P_v)$. We say that a forwarding assignment ϕ is *loop-free* if the digraph on V whose edges are $\{(v, \phi(v))\}_{v \in V}$ is acyclic. (If $\phi = \phi_\sigma$, we call this digraph the *forwarding digraph induced by* σ, and we denote it by D_σ.) Similarly, we will say that a signaling assignment σ is (forwarding-)loop-free if ϕ_σ is loop-free, and we will say that σ is stable and loop-free if σ is a signaling solution and ϕ_σ is loop-free.

Definition 2.3 (Forwarding and signaling agreement)**.** Given a signaling solution σ, we say that *forwarding and signaling disagree in* σ if there is some node that chooses a path for forwarding but whose data is sent along a different path. More formally, we there is disagreement in σ if there are two adjacent nodes v and w such that $P = \sigma(v, w)$, $\phi_\sigma(w) = v$, and $\phi_\sigma(v) \neq \mathsf{next}(P)$. If there is not disagreement in σ, then we say that *forwarding and signaling agree in* σ.

Considering the definitions above, we now consider examples of network instances that have different combinations of these properties—the existence and multiplicity of signaling solutions, the presence or absence of forwarding loops, and whether or not forwarding and signaling agree. Table 1 summarizes the properties of the different examples described in the remainder of this section.

Example	Signaling solutions?			Forwarding loops?		
	None	Unique	Multiple	Yes	No; F-S agree?	
					No	Yes
2.4		X		X		
2.5	X					
2.6		X				X
2.7			X			X
2.8		X			X	
2.9			X	X		
2.10			X		X	

Table 1: Solution characteristics of various FS-SPP examples. Possible characteristics are whether an instance has no, exactly one, or multiple signaling solutions and, if it does have one or more signaling solutions, whether the induced forwarding digraphs may have loops or, if not, whether forwarding and signaling must agree in each signaling solution.

An FS-SPP instance may have a stable signaling solution that is not loop-free. For example, we may modify the Bad Gadget example of [GSW02] to obtain the following example, shown in Fig. 1

Example 2.4. d is the destination; the sets of permitted paths are $\mathcal{P}_d = \{d\}$, $\mathcal{P}_1 = \{1d, 12d\}$, $\mathcal{P}_2 = \{2d, 23d\}$, and $\mathcal{P}_3 = \{3d, 31d\}$; the signaling-preference functions are: $\sigma_{d,v}(d) = 1$ for every $v \neq d$, $\sigma_{1,3}(1d) = 1$, $\sigma_{2,1}(2d) = 1$, $\sigma_{3,2}(3d) = 1$, and all other $\sigma_{u,v}(P)$ values are 0; and the forwarding-preference functions are $\phi_1(12d) = 1$, $\phi_1(1d) = 0$, $\phi_2(23d) = 1$, $\phi_2(2d) = 0$,

$\phi_3(31d) = 1$, and $\phi_3(3d) = 0$. Let σ be defined by $\sigma(d, v) = 0$ for all $v \neq d$, $\sigma(1, 2) = 1d$, $\sigma(2, 3) = 2d$, $\sigma(3, 1) = 3d$, and $\sigma(i, j) = \epsilon$ otherwise. The resulting sets of known paths are $\mathcal{K}_d(\sigma) = \{d\}$, $\mathcal{K}_1(\sigma) = \{1d, 12d\}$, $\mathcal{K}_2(\sigma) = \{2d, 23d\}$, and $\mathcal{K}_3(\sigma) = \{3d, 31d\}$; it is easy to verify that σ is a stable signaling solution. The forwarding assignment induced by σ is $\phi_\sigma(1) = 2$, $\phi_\sigma(2) = 3$, and $\phi_\sigma(3) = 1$; this produces the cycle $1 \to 2 \to 3 \to 1$ in the forwarding digraph induced by σ.

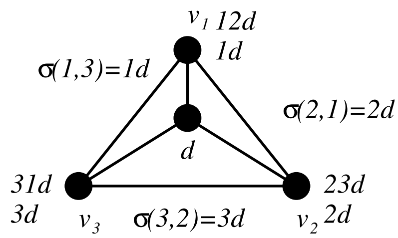

Figure 1: Network from Example 2.4

Viewed from a routing perspective, in Example 2.4 each of the non-destination nodes announces its direct path to the destination (perhaps to gain traffic if there is a per-packet-profit to be made) while in fact forwarding data through its 'legal' non-destination neighbor. As a result, all traffic is forwarded around the cycle on the outside of the graph until it is dropped, never reaching the destination.

In general, if for every $v \in V$, $w \in N(v)$, and $P \in \mathcal{P}_v$ we have $\sigma_{v,w}(P) = \phi_v(P)$, then the FS-SPP instance reduces to an SPP instance. As a result, we have that the problem of deciding whether an FS-SPP instance has a solution is NP-complete: it is easy to verify the stability of a candidate FS-SPP solution, while SPP is known to be NP-complete [GSW02].

Example 2.5. In Example 2.4, if we let $\sigma_{v,w}(P) = \phi_v(P)$ for all $v \neq 0$ and all permitted paths P, then we obtain the original BAD GADGET (an instance of SPP) from [GSW02]; this shows that an FS-SPP instance need not have a stable solution.

Example 2.6. We may also modify Example 2.4 by changing the forwarding preferences to agree with the signaling preferences (so that, *e.g.*, $\phi_1(10) = 1$ and $\phi_1(120) = 0$). In the resulting FS-SPP instance the signaling assignment σ defined above is again stable; with the modified forwarding preferences, the forwarding assignment induced by σ becomes $\phi_\sigma(1) = 10$, $\phi_\sigma(2) = 20$, and $\phi_\sigma(3) = 30$.

In general, whether or not a signaling assignment is stable for an FS-SPP instance does not depend on the forwarding-preference function of the instance. However, the forwarding assignment induced by the signaling assignment does depend on the forwarding-preference function.

Example 2.7. Figure 2 presents the SPP DISAGREE [GSW02] as an FS-SPP instance; let $\mathcal{P}_1 = \{1d, 12d\}$, $\mathcal{P}_2 = \{2d, 21d\}$, $\sigma_{1,2}(12d) = \sigma_{2,1}(21d) = 1$, $\sigma_{1,2}(1d) = \sigma_{2,1}(2d) = 0$,

$\phi_1(12d) = \phi_2(21d) = 1$, and $\phi_1(2d) = \phi_2(2d) = 0$. This instance has two signaling solutions σ^1 and σ^2 defined by: $\sigma^1(d,1) = \sigma^1(d,2) = d$, $\sigma^1(1,2) = 1d$, and $\sigma^1(2,1) = 21d$; and $\sigma^1(d,1) = \sigma^1(d,2) = d$, $\sigma^1(1,2) = 12d$, and $\sigma^1(2,1) = 2d$. For σ^1, we have $\mathcal{K}_{\sigma^1}(1) = \{1d\}$ and $\mathcal{K}_{\sigma^1}(2) = \{2d, 21d\}$; for σ^2, we have $\mathcal{K}_{\sigma^2}(1) = \{1d, 12d\}$ and $\mathcal{K}_{\sigma^2}(2) = \{2d\}$. The edges in the D_{σ^1} are $(1,d)$ and $(2,1)$, while the edges in D_{σ^2} are $(2,d)$ and $(1,2)$. For both σ^1 and σ^2, the induced forwarding digraph is loop-free and each forwarding path (*i.e.*, the paths in the forwarding digraph) is known by each node along that path.

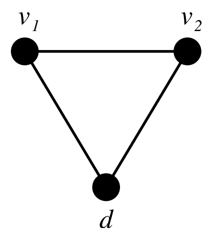

Figure 2: Network from Example 2.7

The following example shows that we may have an FS-SPP with a unique stable solution and an acyclic induced forwarding digraph that disagrees with the signaled paths.

Example 2.8. Figure 3 shows a four-node network in which d is the destination. Because $1d$ is the only permitted path at 1, the only nontrivial signaling-preference function is $\sigma_{2,3}$; we take $\sigma_{2,3}(2d) = 0$ and $\sigma_{2,3}(21d) = 1$. There is then a unique signaling solution σ with: $\sigma(d,1) = \sigma(d,2) = d$, $\sigma(1,2) = \sigma(1,3) = 1d$, $\sigma(2,3) = 21d$, and the values of $\sigma(2,1)$, $\sigma(3,1)$, and $\sigma(3,2)$ irrelevant (but uniquely determined by the signaling-preference functions $\sigma_{2,1}$, $\sigma_{3,1}$, and $\sigma_{3,2}$). The forwarding digraph induced by σ has edges $(1,d)$, $(2,d)$, and $(3,2)$; this disagrees with the signaled paths because $\phi_\sigma(3) = 2$, $\sigma(2,3) = 21d$, but $\phi_\sigma(2) = d \neq \mathsf{next}(\sigma(2,3))$. Switching the values of 2's signaling-preference function produces an instance with a different unique solution *and* agreement.

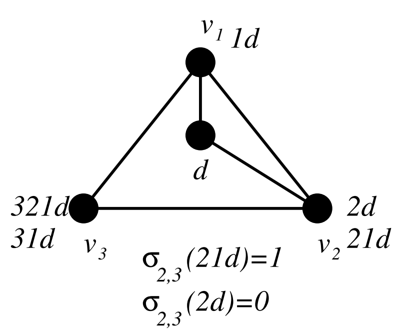

Figure 3: Network for Example 2.8

If we exchange the values of $\sigma_{2,3}$ so that $\sigma_{2,3}(21d) = 0$ and $\sigma_{2,3}(2d) = 1$, we obtain a different FS-SPP instance with a unique signaling solution $\hat{\sigma}$ that differs from σ in that $\hat{\sigma}(2,3) = 2d$; we then have $\phi_{\hat{\sigma}}(3) = 1$, so forwarding and signaling agree in $\hat{\sigma}$.

We could further modify Example 2.8 by also exchanging the values of ϕ_2 so that $\phi_2(2d) = 0$ and $\phi_2(21d) = 1$; $\hat{\sigma}$ remains the unique signaling solution, but in the forwarding digraph induced by $\hat{\sigma}$, 2 now forwards to 1 instead of to d. This illustrates that a node can lie about its forwarding in a solution (here $\hat{\sigma}(2,3) = 2d$ although $\phi_{\hat{\sigma}}(2) = 1$) but we can still have forwarding and signaling agree in that solution (because the only node that lies about forwarding is a leaf node in the forwarding tree).

Example 2.9. Consider the network shown in Fig. 4 with nodes $d, 0, 1, 2, 3$, where d is the destination node, and edges $\{d, i\}$ and $\{i, i+1\}$ for $0 \leq i \leq 3$ (with $i+1$ interpreted modulo 4). Let $\mathcal{P}_d = \{d\}$ and $\mathcal{P}_i = \{id, i(i-1)d, i(i-1)(i-2)d\}$ for $0 \leq i \leq 3$; let $\sigma_{i,i+1}(i(i-1)d) = 2$, $\sigma_{i,i+1}(id) = 1$, and $\sigma_{i,i+1}(i(i-1)(i-2)d) = 0$ (the other functions $\sigma_{i,v}$ are irrelevant given $\mathcal{P}_v$ as above); let $\phi_i(i(i-1)d) = 1$, $\phi_i(i(i-1)(i-2)d) = 1$, and $\phi_i(id) = 0$. (Note that the tie doesn't cause problems because the routes involved all have the same next hop.)

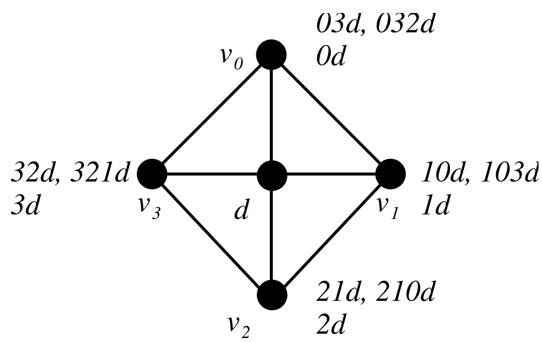

Figure 4: Network for Example 2.9.

This FS-SPP instance has two solutions σ^e and σ^o (the superscripts indicate whether the even or odd nodes are used to forward traffic directly to d). These are defined by $\sigma^e(d,i) = d$, $\sigma^e(0,1) = 0d$, $\sigma^e(1,2) = 10d$, $\sigma^e(2,3) = 2d$, and $\sigma^e(3,0) = 32d$, with corresponding knowledge sets $\mathcal{K}_{\sigma^e}(0) = \{0d, 032d\}$, $\mathcal{K}_{\sigma^e}(1) = \{1d, 10d\}$, $\mathcal{K}_{\sigma^e}(2) = \{2d, 210d\}$, and $\mathcal{K}_{\sigma^e}(3) = \{3d, 32d\}$; and $\sigma^o(d,i) = d$, $\sigma^o(0,1) = 03d$, $\sigma^o(1,2) = 1d$, $\sigma^o(2,3) = 21d$, $\sigma^o(3,0) = 3d$ with corresponding knowledge sets $\mathcal{K}_{\sigma^o}(0) = \{0d, 03d\}$, $\mathcal{K}_{\sigma^o}(1) = \{1d, 103d\}$, $\mathcal{K}_{\sigma^o}(2) = \{2d, 21d\}$, and $\mathcal{K}_{\sigma^o}(3) = \{3d, 321d\}$.

For both solutions σ^e and σ^o, the induced forwarding digraph has exactly the edges $\{(i, i-1)\}_{0 \leq i \leq 3}$; this gives a forwarding loop involving all of the non-destination edges.

We may generalize Example 2.9 to have $2k$ non-destination nodes (for $k \geq 2$) by simply letting i range over the integers in $[0, 2k-1]$. For $k \geq 3$, we may modify the forwarding-preference functions to obtain an FS-SPP instance with multiple loop-free solutions in which forwarding and signaling disagree as in the following example.

Example 2.10. Consider the network on $d, 0, \ldots, 5$, shown in Fig. 5, where d is the destination node, and edges $\{d, i\}$ and $\{i, i+1\}$ (interpreted modulo 6) for $0 \le i \le 5$. The sets of permitted paths and the signaling-preference functions are as in Example 2.9 (now allowing the parameter i to range over $0 \le i \le 5$). Also as before, we let $\phi_i(i(i-1)d) = 1$ and $\phi_i(i(i-1)(i-2)d) = 1$ for $0 \le i \le 5$ and $\phi_i(id) = 0$ for $i = 1, 2, 4, 5$; now, however, we let $\phi_0(0d) = \phi_3(3d) = 2$.

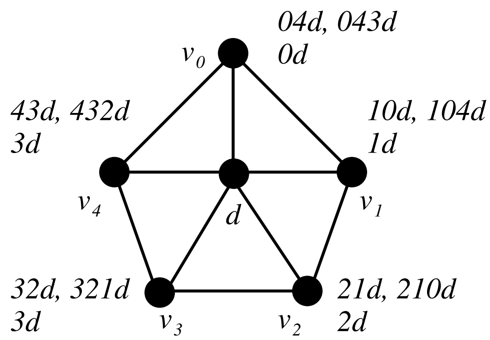

Figure 5: Network for Example 2.10.

There are again two signaling solutions σ^e and σ^o. The corresponding knowledge sets are $\mathcal{K}_{\sigma^e}(i) = \{id, i(i-1)(i-2)d\}$ for even i and $\mathcal{K}_{\sigma^e}(i) = \{id, i(i-1)d\}$ for odd d, and $\mathcal{K}_{\sigma^o}(i) = \{id, i(i-1)d\}$ for even i and $\mathcal{K}_{\sigma^o}(i) = \{id, i(i-1)(i-2)d\}$ for odd d, with $0 \le i \le 5$ in each case. The edges in D_{σ^e} and D_{σ^o} are the same: $\{(0, d), (1, 0), (2, 0), (3, d), (4, 3), (5, 4)\}$. This gives disagreement between forwarding and signaling: for σ^e, we have $\phi_{\sigma^e}(4) = 3$, $\sigma^e(3, 4) = 32d$, and $\phi_{\sigma^e}(3) = d$; for σ^o, we have $\phi_{\sigma^o}(1) = 0$, $\sigma^o(0, 1) = 05d$, and $\phi_{\sigma^o}(0) = d$.

2.3 Basic results about the model

We may extend the basic notion of dispute wheels in SPP instances to account for the possibility of announcing different routes to different neighbors in an FS-SPP instance.

Definition 2.11 (S-dispute wheel)**.** A *signaling-dispute wheel (S-dispute wheel)* in an FS-SPP instance consists of k nodes $y_0, \ldots, y_{k-1}$ and k paths $R_0Q_1, \ldots, R_{k-1}Q_0$ such that:

- for each i, $R_iQ_{i+1} \in \mathcal{P}_{y_i}$;
- for each i, R_i is a path from y_i to y_{i+1} (so $Q_{i+1} \in \mathcal{P}_{y_{i+1}}$); and
- for each i, if v_i is the neighbor of y_i on the path R_{i-1}, then $\sigma_{y_i,v_i}(R_iQ_{i+1}) > \sigma_{y_i,v_i}(Q_i)$ (*i.e.*, y_i prefers to announce the 'indirect' route R_iQ_{i+1} to its neighbor on R_{i-1} instead of the 'direct' route Q_i).

In all of these conditions, we interpret the subscripts modulo k. Note that there is no requirement that the paths be disjoint, although each permitted path is (by definition) simple.

S-dispute wheels in FS-SPP are essentially the same as dispute wheels [GSW02] in SPP—allowing a node to signal different paths to different neighbors does not affect the preference criteria that are part of the dispute wheel definition.

Theorem 2.12. *If an FS-SPP instance does not contain any S-dispute wheels, then it has a unique signaling solution.*

The proof of Thm. 2.12 is essentially the same as for the analogous result in SPP.

Proof. This is virtually identical to the proof for dispute wheels in SPP [GSW02]—allowing nodes to advertise different paths to different neighbors does not change the argument. (Essentially, we start with a node whose announcements to a neighbor oscillates or differs between two solutions—depending on whether we consider the no or multiple solution case—and follow the route it prefers to announce until we reach the last node on that path that oscillates or differs, depending on the case. This node is a pivot, and we iterate until we close the wheel after some finite number of steps.) □

Remark 2.13. We should be able to define signaling dispute digraphs, essentially the same as dispute digraphs in SPP, whose acyclicity is equivalent to the absence of an S-dispute wheel in the relevant FS-SPP instance. We do this explicitly below for our new wheel-like structure in Def. 2.16 and Prop. 2.17.

Wheel-type structures play an additional role when signaling and forwarding are decoupled; we will also make use of the following definition, which incorporates both signaling-preference and forwarding preference functions (in contrast to S-dispute wheels, which use just the former).

Definition 2.14 (FS-dispute wheel)**.** A *forwarding/signaling-dispute wheel (FS-dispute wheel)* in an FS-SPP instance consists of:

- k nodes $y_0, \ldots, y_{k-1}$ and
- k paths $R_0Q_1, \ldots, R_{k-1}Q_0$

such that

- for each i, $R_iQ_{i+1} \in \mathcal{P}_{y_i}$;
- for each i, R_i is a path from y_i to y_{i+1} (so $Q_{i+1} \in \mathcal{P}_{y_{i+1}}$);
- for each i, $\phi_{y_i}(R_iQ_{i+1}) > \phi_{y_i}(Q_i)$ (*i.e.*, y_i prefers to forward traffic along the 'indirect' route R_iQ_{i+1} instead of along the 'direct' route Q_i); and
- for each i, if v_i is the neighbor of y_i on the path R_{i-1}, then $\sigma_{y_i,v_i}(Q_i) > \sigma_{y_i,v_i}(R_iQ_{i+1})$ (*i.e.*, y_i prefers to announce the 'direct' route Q_i to its neighbor on R_{i-1} instead of the 'direct' route R_iQ_{i+1}).

In all of these conditions, we interpret the subscripts modulo k. Note that there is no requirement that the paths be disjoint, although each permitted path is (by definition) simple.

Theorem 2.15. *If an FS-SPP instance is FS-dispute wheel-free, then every signaling solution for the instance induces an acyclic forwarding digraph.*

Proof. If the instance has no stable signaling solutions then the theorem holds trivially. Otherwise, let σ be a (not necessarily unique) signaling solution for the instance and assume that it has a forwarding loop in which x_i forwards traffic to x_{i+1} for $0 \leq i \leq m-1$ (interpreting the subscript in x_j modulo m); all of these nodes forward traffic for d to another node, so every $x_i \neq d$. Because x_i forwards traffic to x_{i+1}, $\sigma(x_{i+1}, x_i)$ must be a nonempty path and $x_i\sigma(x_{i+1}, x_i)$ must be simple. There must be some i such that $\sigma(x_i, x_{i-1}) \neq x_i\sigma(x_{i+1}, x_i)$ (*i.e.*, x_i announces a route other than the one it thinks it is using for forwarding), otherwise each node would have a nonsimple path; let i be one such value, and set $y_0 = x_i$, $P_0 = x_i\sigma(x_{i+1}, x_i)$ (the path that y_0 thinks it is using for routing), and $Q_0 = \sigma(x_i, x_{i-1})$ (the path that y_0 announces to the node that sends traffic to y_0 in the forwarding loop).

Having defined y_0 and P_0, let y_1 be the first node (when moving from y_0 to d) x_j on P_0 such that $\sigma(x_j, x_{j-1}) \neq x_j\sigma(x_{j+1}, x_j)$. (This node must be part of the forwarding loop because y_0 forwards to a node on the forwarding loop, as does that node, *etc.*, so the first node on the announced route that lies about how it forwards must also be on the forwarding loop.) Such a node must exist otherwise P_0 would go through y_0 again. Let $P_1 = x_j\sigma(x_{j+1}, x_j)$, $R_0 = P_{0[y_0,y_1]}$, and $Q_1 = P_{0[y_1,d]}$.

Having defined y_i and P_i, define y_{i+1}, P_{i+1}, R_i, and Q_{i+1} in the same way that y_1, P_1, R_0, and Q_0 were defined from y_0 and P_0. By our definition of y_0 we must eventually define $y_k = y_0$ for some k (after cycling through all of the nodes x_i in the original forwarding loop, although only those that do not truthfully signal their forwarding path are chosen as y_i's).

By construction, we have $\phi_\sigma(y_i) = R_iQ_{i+1}$ and $\sigma(y_i, z_i) = Q_i$, where z_i is the next node on R_{i-1} moving from y_i to y_{i-1}. Because both R_iQ_{i+1} and Q_i are in $\mathcal{K}_v(\sigma)$, we have $\phi_{y_i}(R_iQ_{i+1}) > \phi_{y_i}(Q_i)$ and $\sigma_{y_i,z_i}(Q_i) > \sigma_{y_i,z_i}(R_iQ_{i+1})$; thus, the FS-SPP instance contains an FS-dispute wheel. □

Definition 2.16 (FS-dispute digraph)**.** Given an FS-SPP instance, its *FS-dispute digraph (FS-DDG)* is the digraph on the set $\mathcal{P}$ of all permitted paths from the instance whose edges are defined by the relations

$$P_1 \oslash P_2 \text{ if } P_2 = vP_1 \text{ for some } v$$

i.e., if P_2 extends P_1 by an edge, and $P_1 \ominus wP_2$ if:

1. P_1 and P_2 are in $\mathcal{P}_v$ for some v;

2. $\sigma_{v,w}(P_2) > \sigma_{v,w}(P_1)$; and

3. $\phi_v(P_1) > \phi_v(P_2)$,

i.e., if wP_2 is received by w instead of wP_1 (because of the signaling-preference function at v) even though v prefers to forward data through P_1.

Proposition 2.17. *An FS-SPP instance is FS-dispute wheel-free iff its FS-dispute digraph is acyclic.*

Proof. If the instance contains an FS-DW with y_i, v_i, R_i, and Q_i as in Definition 2.14, then for every i, $R_iQ_{i+1} \ominus v_iQ_i$ (recall that v_i is the next node on R_{i-1} when moving from y_i toward y_{i-1}). We also have $v_iQ_i \oslash^* R_{i-1}Q_i$, *i.e.*, $R_{i-1}Q_i$ is a (not necessarily proper) superpath of v_iQ_i. Iterating this argument, we obtain a cycle in the FS-DDG that contains one $\ominus$-edge for each pivot in the FS-DW.

If the FS-DDG contains a cycle, then the cycle must contain at least one $\ominus$-edge (otherwise a permitted path in the instance would be a proper subpath of itself). Assume this is $P_1 \ominus w_1Q_1$. We cannot have $wQ_1 \oslash^* P_1$ because both P_1 and Q_1 are permitted at some v, so this would imply that v appears twice in P_1 (as its endpoint as on the next hop of the subpath wQ_1), a permitted (and thus simple) path. Thus we have at least one more $\ominus$-edge; call this $P_2 \ominus w_2Q_2$, with $w_1Q_1 \oslash^* P_2$ (so that this is the next $\ominus$ edge in the cycle). In general, having defined w_iQ_i, find the next $\ominus$ edge in the cycle in question and let this be $P_{i+1} \ominus w_{i+1}Q_{i+1}$, so that $w_iQ_i \oslash^* P_{i+1}$; for each i, let v_i be the node such that $P_i, Q_i \in \mathcal{P}_{v_i}$. Eventually, we exhaust the cycle and return to P_1.

Letting R_{i+1} be the subpath of P_{i+1} from v_{i+1} to v_i (so that $P_{i+1} = R_{i+1}Q_i$) and inverting the ordering of the subscripts, we see that the resulting structure satisfies Def. 2.14 and the instance contains an FS-DW. □

This result is useful because it allows us to easily verify FS-dispute-wheel-freeness of FS-SPP instances.

FS-dispute-wheel-freeness and the existence of a unique solution does not imply agreement between forwarding and signaling. (Example 2.26 illustrates this, motivated by a slightly different context.) Also, the absence of an FS-dispute wheel in an FS-SPP instance does not guarantee that there will be a unique stable solution.

2.4 Gao-Rexford constraints and FS-SPP

Gao and Rexford [GR01] identified conditions that guarantee inherent safety in BGP instances in which nodes classify their neighbors as customers, providers, and peers; in fact, these conditions imply that the SPP instance has no dispute wheel [GGR01]. Here we adapt these conditions to reflect the motivation for the original conditions in the context of decoupled forwarding and signaling; this allows route advertisements that were not possible in the original Gao-Rexford framework (*e.g.*, announcing a non-customer route to a non-customer, allowed here if a node is forwarding to a customer). Satisfaction of these adapted conditions implies FS-DW-freeness, but not S-DW-freeness, for FS-SPP. These also do not imply agreement between forwarding and signaling, which motivates our studies, later in this paper, of the incentive compatibility of truthful signaling.

The following definition implements the original motivation for the Gao-Rexford constraints in the FS-SPP framework.

Definition 2.18 (FS-GR constraints)**.** A FS-SPP instance satisfies the *Gao-Rexford constraints for FS-SPP* (the *FS-GR* constraints) if the following conditions hold on the instance.

Classification of neighbors: Each node classifies every one of its neighbors as a *customer*, *provider*, or *peer*. A node v classifies one of its neighbors w as a customer iff w classifies v as a provider, and v classifies w as a peer iff w classifies v as a peer.

Signaling while forwarding to customers If a node is forwarding along a customer route, then for each of its neighbors, the node may announce *any* route that it knows to that neighbor. (This may be beneficial for the node, because it does not need to pay its customer to carry the traffic.) Note that this does not actually constrain nodes' behavior.

Signaling while forwarding to providers/peers If a node is forwarding along a non-customer route, the it may announce routes only to its customers. The node thus does not provide transit service to its providers or peers.

Preference for customer routes If a node has learned a route from one of its customers, then it chooses one its customer routes for forwarding.

No customer-provider cycles The digraph on the set of ASes in which edges point from customers to their providers is acyclic.

Remark 2.19. These conditions are trivially satisfied if nodes never advertise routes to their providers and peers. However, this would mean that routes to the destination are advertised by the originating AS to its providers and then by the AS and these providers only to some subset (depending on filtering) of their direct and indirect customers; such a situation is clearly unacceptable from a reachability perspective. Thus, if ASes are to satisfy the FS-GR constraints, their signaling policies must depend on the currently selected forwarding route (because of the decoupling between routing and signaling; this was not an issue in [GR01] when these were coupled). In particular, ASes cannot expect to have rankings (or utilities) for forwarding that depend only on the paths known and separate rankings (or utilities) for signaling that depend only on the paths known (and perhaps the neighbor to whom announcements are being made) such that these may be changed independently and arbitrarily without violating the FS-GR constraints.

The preceding remark shows that nontrivial policies that satisfy the FS-GR constraints are not separable in the sense of the following definition.

Definition 2.20 (Policy separability)**.** We say that a node's routing and signaling policies are *separable* if they may be configured independently and the signaling policy is independent of the node to whom announcements are being made, and *per-node-separable* if they may be configured independently and the signaling policy may depend on the node to whom announcements are being made.

Unfortunately, the FS-GR constraints do not prevent S-dispute wheels or guarantee signaling convergence. This is seen in the following example.

Example 2.21. We may orient the edges of BAD GADGET so that d is a customer of all other nodes, each node prefers to forward along the direct route to d, and the rim edges do not form a cycle. Because each node is forwarding to a customer, the signaling preferences may be assigned arbitrarily; in particular, we may use the BAD GADGET preferences. Figure 6 shows one way to do this; each node's signaling preferences are shown in the left list at the node and the node's forwarding preferences are shown in the right list at the node.

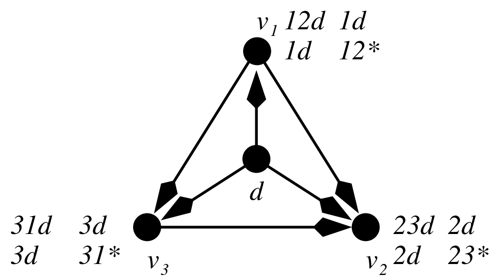

Figure 6: Network from Example 2.21

Remark 2.22. This is certainly resolved if we add the restriction that only customer paths are announced to non-customers (as in the following theorem); whether or not a weaker condition suffices remains unclear.

Theorem 2.23. *If an instance of FS-SPP satisfies the FS-GR constraints and, additionally, the only paths announced to non-customers are customer paths, then the instance is S-dispute-wheel-free.*

Proof. Assume the instance contains an S-dispute wheel with pivots y_i, rim paths R_i, and spoke paths Q_i ($0 \leq i \leq k-1$) as in Def. 2.11. Some announcement on the rim must be from a node to one of its non-customers, otherwise the rim would form a customer-provider cycle. The node that does this must be announcing one of its customer paths (because it is being announced to a non-customer); the customer from which the path was learned must, in turn, be announcing one of *its* customer paths (for the same reason). Following this around the rim, we obtain a customer-provider cycle. □

Theorem 2.24. *If an instance of FS-SPP satisfies the FS-GR constraints, then the forwarding digraph induced by any stable solution is acyclic.*

Proof. Once data is forwarded along a customer edge (*i.e.*, from a provider to a customer), every subsequent forwarding hop will be across a forwarding edge (because if x forwards to its customer y, then y must advertise *some* route to its provider x; this means that y must be forwarding along a customer edge, although the path it announces to x need not satisfy this). Because there are no customer-provider cycles in the network, no forwarding loop may have any customer edges in it.

If data is forwarded from x to y along either a provider or peer edge (*i.e.*, y is a provider or peer, respectively, of x), then y must then forward that data along a customer edge; if y does not forward data along a customer edge, it will not announce any route to its non-customer x and so x will not be forwarding data to y. □

Remark 2.25. The preceding example and theorems are of particular interest because they refine our understanding of what is guaranteed by the very natural Gao-Rexford conditions. In particular, Example 2.21 shows that, when modified for FS-SPP, these conditions *do not* guarantee signaling stability as they do in SPP. However, Thm. 2.24 shows that if the network does converge to a stable signaling solution (whether or not the FS-GRconstraints and the additional requirement of Thm. 2.23 are satisfied), then the induced forwarding digraph is acyclic.

Example 2.26. The FS-GR constraints do not guarantee that nodes will be truthful. In particular, we may modify Example 2.8 to obtain an instance that follows the FS-GR constraints and which has a unique signaling solution, but in which the signaling solution disagrees with the induced forwarding digraph. Figure 7 shows the resulting network; $12d$ is now permitted at 1, with $\phi_1(1d) > \phi_1(12d)$ (these are both customer routes for 1), and the extension of this path is permitted at 3. $\sigma_{1,3}(1d) = 1 > \sigma_{1,3}(12d) = 0$, and $\phi_3(321d) > \phi_3(31d) > \phi_3(312d)$ (these are all provider paths for 3). Note that 3 only learns routes from providers, so it can never forward to a customer and thus never announces any of its paths to its neighbors (both providers for 3). Thus 2 does not learn any customer paths and is free to choose between the paths it learns from 1 and d; $\phi_2(2d) > \phi_2(21d)$.

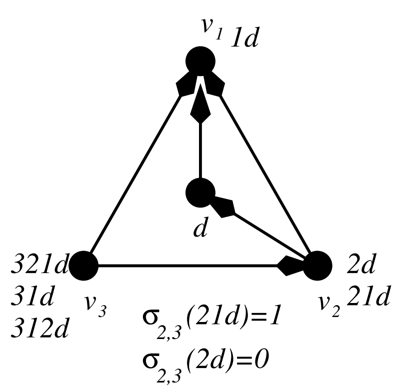

Figure 7: Network for Example 2.26. Arrows go from customers to providers.

More generally, this example also shows that an FS-SPP instance with a unique solution and no FS-DW need not have agreement between forwarding and signaling.

3 The Interdomain-Routing Game

3.1 The motivation for game theory

The static analysis above shows that systems with unique solutions do not necessarily have acyclic forwarding digraphs; even if the forwarding digraph is acyclic, the paths that are signaled may disagree with the forwarding tree. While forwarding loops will only occur in a stable signaling solution if some node is lying, dishonest signaling will not necessarily cause a forwarding loop; as a result (*e.g.*, if the natural FS-GR constraints are satisfied), if nodes have an incentive to lie about their forwarding they may be able to do so without causing obvious disruptions to the network. While detecting signaling dishonesty may be possible after the fact, this may not prevent such behavior; it seems very difficult (if not impossible) for a node to *prove* to one of its neighbors that it is forwarding data to a particular next hop. In order to mitigate this risk, we thus turn to the problem of when truthful signaling is incentive compatible when our more realistic utility functions are used.

Another motivation for studying the game-theoretic aspects of decoupling forwarding from signaling is that game theory is a logical tool to use in studying the routing decisions made by ASes; this is seen in the previous work in the area. It is thus natural to consider the more realistic view of decoupling forwarding from signaling in these models in addition to the static analysis studied above. In particular, we are interested in having components of the utility that depend separately on forwarding and signaling.

3.2 Game dynamics and BGP

Interdomain-route calculation is distributed, asynchronous, and iterative, based on the autonomous decisions of ASes throughout the Internet. We model this as a multi-round game with an infinite number of rounds as in [LSZ06]. In each round of the game, some subset of nodes is chosen to participate by a *scheduler*, but each node must participate in an infinite number of rounds. When a node v participates, it may perform the following actions:

1. Receive *update messages* from its neighbors; each message from some $w \in N(v)$ contains a single route $\pi'(w) \in \mathcal{S}_w$, possibly the empty route ϵ.

2. Choose a single outgoing edge (v, x) to some neighbor $x \in N(v)$, representing the choice of $\pi(v) = v\pi(x)$ as v's forwarding route, or no edge at all, representing the null route $\pi(v) = \epsilon$.

3. Send update messages to any of its neighbors, containing a route in $\mathcal{S}_v$ (possibly ϵ).

The scheduler can determine when, if at all, update messages get delivered to neighbors, although it cannot indefinitely drop messages between neighbors. The scheduler thus represents "fair but arbitrary network delays" (see [GSW02] for a formal model).

Remark 3.1. It is important to note that in this work we assume that if a node v sends an update message containing the path $vR \neq \epsilon$, then v received (earlier in the round in which it sends the update) an update message containing R.

A node's *strategy* dictates its actions when chosen to participate. The standard protocol for interdomain routing on the Internet, the Border Gateway Protocol (BGP), prescribes the following strategy, referred to as *best-reply dynamics* in [LSZ06].

1. Receive the most current route updates from all neighbors.

2. Determine the forwarding-route choice $\pi(v)$ based on configured parameters.

3. Signal the route choice $\pi(v)$ to all neighbors (unless specifically configured to signal ϵ).

The role of "configured parameters" in step 2 can be modeled as preferences over forwarding routes, just as in the standard abstract model of BGP, the Stable Paths Problem (SPP) [GSW02]. Formally, let each node v have a *forwarding-preference function* $\phi_v : \mathcal{S}_v \to \mathbb{Z}$, such that $\phi_v(R_1) > \phi_v(R_2)$ for routes $R_1, R_2 \in \mathcal{S}_v$ implies that v prefers the choice of R_1 over the choice of R_2 as its forwarding route. *Following BGP* implies choosing, in step 2, the most preferred route out of the options signaled by neighbors, and signaling that choice, in step 3, to neighbors.

We are careful to include the possibility of *filtering*, which is excluding a route from consideration. We assume that $\phi_v(\epsilon) = 0$, but we permit $\phi_v(R) < 0$, *i.e.*, there may be routes that are less preferred than having no route at all. These routes are *filtered on import*; for all $v \in V$, let the set of *permitted routes* $\mathcal{P}_v = \{R \in \mathcal{S}_v \mid \phi_v(R) \geq 0\}$ be the routes that are not filtered on import. (Note that $\epsilon \in \mathcal{P}_v$.) Analogously, if node v chooses R as its forwarding route in step 2 but signals ϵ to neighbor w, we say that R is *filtered on export* to w. We note an important distinction between our model and the model in [LSZ06]: filtering is not considered a deviation from the BGP strategy in our model, as the BGP specification (and the SPP model of BGP) permits filtering as a standard part of routing policy. Therefore, we will insist on explicitly stating any assumptions about filtering when discussing nodes' behavior in the routing game.

The utility U_v for a node v is defined as follows. Let $\pi_{S,t}(v)$ be the path chosen by v in round t when schedule S is used. If there is a value t_0 such that $\pi_{S,t}(v) = R$ for all $t \geq t_0$, then the forwarding component of v's utility under the schedule S is $\phi_v(R)$. More generally, we let

$$\pi_S(v) = \begin{cases} \lim_{t\to\infty} \pi_{S,t}(v) & \text{if the limit exists} \\ \epsilon & \text{otherwise} \end{cases}$$

and then use $\phi_v(\pi_S(v))$ in defining v's utility below.

Given a schedule S, let $D_{S,t}(v)$ be the part of the forwarding digraph induced by the path assignment $\pi_{S,t}$ from which v is reachable. Let $D_S(v)$ be the limit inferior of the sequence of diagraphs $\{D_{S,t}(v)\}_t$, *i.e.*, if $V(G)$ and $E(G)$ are the vertex and edge sets of a graph G, then we let

$$V(D_S(v)) = \bigcup_{t_0=0}^{\infty} \bigcap_{t=t_0}^{\infty} V(D_{S,t}(v)) \quad \text{and} \quad E(D_S(v)) = \bigcup_{t_0=0}^{\infty} \bigcap_{t=t_0}^{\infty} E(D_{S,t}(v)).$$

In particular, a node is in $D_S(v)$ if and only if that node's traffic is always forwarded through v in every round after some round t_0 (when schedule S is used); a directed edge (u, w) is in $D_S(v)$ if an only if traffic from u (1) is forwarded directly to w and (2) eventually passes through v in every round after some round t_0 (when schedule S is used). As illustrated by Example 3.2, $D_S(v)$ need not be connected.

Given a schedule S, the utility $U_v(S)$ of node v is

$$U_v(S) = \phi_v(\pi_S(v)) + \Sigma_v(D_S(v)),$$

where $\phi_v(\epsilon) = 0$ for every v and Σ_v assigns numerical values to digraphs. In particular, this depends on the route that v eventually always chooses (which may or may not match the route that v's data takes) under the schedule S and the set of nodes/edges that eventually always send data to v under the schedule S. As examples, the function $\Sigma_v(G)$ might be defined as the number of vertices in G (*i.e.*, the number of nodes that always route through v after some round) or the number of edges of the form (x, v) in G (*i.e.*, the number of v's neighbors that always choose v as their next hop after some round of the game).

Example 3.2. Consider the network shown in Figure 8, where the forwarding preferences for each node are listed next to the node. For the schedule

$$\{0\}, \{1\}, \{3\}, \{2\}, \{4\}, \{0\}, \{1\}, \{3\}, \{2\}, \{4\}, \ldots,$$

the induced forwarding digraph is always acyclic with the traffic from all non-destination nodes being forwarded through 0, but the nodes 1, 2, and 3 that form the instance of BAD GADGET never converge. As a result, $D_S(0)$ consists of the nodes $1, 2, 3, 4$ and the directed edge $(4, 3)$, but it does not contain any other edges; in particular, the graph is disconnected.

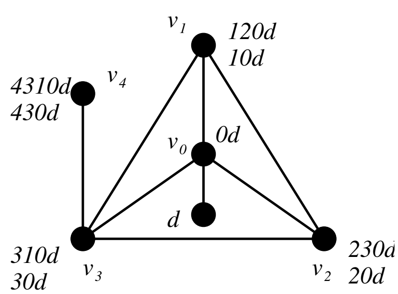

Figure 8: Network for Example 3.2.

In this schedule, if each node follows BGP and announces the route it uses for forwarding, then node 4 does not gain any utility from forwarding because (like nodes 1, 2, and 3) it never converges to a single route for forwarding. However, 3 could always announce 310d (which is always available to 3) to 4, regardless of which route 3 is using for forwarding; in this case, 4 would then gain some benefit from forwarding (even though the route taken by 4's data would vary over time).

Remark 3.3. The limiting aspect of the definition of the forwarding component of the utility is in the same spirit as the utility in [LSZ07] (other than the difference between the route taken by data and the route selected by v). The signaling aspect seems to be the correct analogue for signaling, but other definitions may also be reasonable. For example, if v benefits by eventually seeing some of the traffic from other nodes, but v doesn't benefit additionally from seeing *all* traffic instead of just *some* of it, then it might be more appropriate to consider the limit superior of the sequence of digraphs, *i.e.*, defining a digraph $D'_S(v)$ by

$$V(D'_S(v)) = \bigcap_{t_0=0}^{\infty} \bigcup_{t=t_0}^{\infty} V(D_{S,t}(v)) \quad \text{and} \quad E(D'_S(v)) = \bigcap_{t_0=0}^{\infty} \bigcup_{t=t_0}^{\infty} E(D_{S,t}(v)).$$

and then taking the signaling component of utility to be $\Sigma_v(D'_S(v))$.

Also, we may replace ϕ_v with a forwarding-utility function Φ_v, but this does not seem to provide any advantage.

Informally, node utilities have a *forwarding* and *signaling* component. The forwarding component is based on the valuation a node has for the route it chooses for forwarding traffic. *Because this choice depends on the routes signaled by the node's neighbors, the forwarding component of utility is based on the perceived, not actual, forwarding route.* We again note that, if nodes signal routes that are not actually used, detecting the inconsistency is nearly impossible. The signaling component for v is based on the consequence of signaling the route choice to others, namely, that others' traffic may be sent through the v. Therefore, it is defined as some function of the actual traffic flows through v induced by the route choices elsewhere in the network. For example, if v wishes to snoop on others' traffic, Σ_v may be a positive function based on the number of nodes sending traffic through v.

Here we note another distinction between our model and previous models of interdomain routing: The utility functions in [LSZ06,FRS06,FSS04,FKMS05] contained only a forwarding component. Signaling behavior thus affected a node's utility only when it changed the node's forwarding route. However, as we have discussed, a node may benefit from changing its signaling behavior without changing its forwarding route. We informally say that a node *lies in signaling* when a node signals a non-null route that is different than the route chosen as its forwarding route.

3.3 Equilibrium and solution concept

BGP permits distributed, asynchronous computation of interdomain routes with limited (local knowledge): Nodes choose routes autonomously, based on announcements from neighbors and on individual routing policy. Knowledge of routes in the entire network is not required to participate in the process, nor would it be possible to obtain in an efficient manner. However, we require that these uncoordinated decisions somehow form a consistent forwarding tree to each destination. Luckily, if all nodes follow BGP, consistency is naturally enforced through truthful signaling.

BGP is intended to converge to a *stable route assignment*; *i.e.*, it is assumed that, after a finite number of rounds of the interdomain-routing game, π becomes constant for all nodes.

This amounts to a Nash equilibrium in the multi-round game: Each node has chosen its best available route given the signals from its neighbors, and no further update messages need to be sent. However, it is known that certain combinations of node forwarding-preference functions and filtering decisions can cause BGP to oscillate, which amounts to the lack of a pure Nash equilibrium. Networking researchers have studied this problem from the perspective of constraints on routing policy that may eliminate divergence. The broadest-known constraint on policies is *dispute-wheel freeness* [GSW02]; we recall the definition of the dispute wheel structure in Sec. 4.

Previous results have shown that, if the dispute-wheel-freeness condition and several other assumptions hold [FRS06, FRS06, LSZ06], the strategy of following BGP is incentive compatible in ex-post Nash equilibrium. The ex-post Nash equilibrium was argued to be the most reasonable solution concept for Internet algorithms [SP04]. In ex-post Nash, we can consider several forms of rational manipulation, given that the strategic agents themselves are performing computations and reporting results, and address the question of specification faithfulness. For the interdomain-routing game, this amounts to asking whether a node can benefit by deviating from following BGP. In the remainder of this paper, we present the results of analyzing this question with quasi-bilinear utilities.

4 Negative Results

Various restrictions have proved important in the study of quasi-linear utilities [FRS06, LSZ06]; in this section, we show that a violation of any one of these four restrictions can provide an incentive for a node to lie about its forwarding choice when quasi-bilinear utility functions are used. These four conditions are:

Policy consistency $\phi_v(P) > \phi_v(Q)$ and $wP, wQ \in \mathcal{P}_w$ imply $\phi_w(wP) > \phi_w(wQ)$.

Consistent filtering $\phi_v(P) > \phi_v(Q)$ and $wQ \in \mathcal{P}_w$ together imply $wP \in \mathcal{P}_w$ if wP is a simple path.

Route verification (in this context) $\forall v, w \in N(v), \sigma, \quad \sigma(v, w) \in \mathcal{K}_v(\sigma)$

Dispute-wheel freeness The network does not contain a *dispute wheel* as defined in Def. 4.1.

Outside of the example noted in this section, we will assume that nodes do not advertise routes that they have not learned (but they are not required to *use* the routes they advertise).

The notion of a dispute wheel is a key concept in the analysis of interdomain routing; it was originally introduced in [GSW02].

Definition 4.1. A *dispute wheel* in a network consists of k nodes $y_0, \ldots, y_{k-1}$ (the *pivots*) and k paths $R_0Q_1, \ldots, R_{k-1}Q_0$ such that:

- for each i, $R_iQ_{i+1} \in \mathcal{P}_{y_i}$;
- for each i, R_i is a path (a *rim path*) from y_i to y_{i+1} (so $Q_{i+1} \in \mathcal{P}_{y_{i+1}}$); and
- for each i, if v_i is the neighbor of y_i on the path R_{i-1}, then $\phi_{y_i}(R_iQ_{i+1}) > \phi_{y_i}(Q_i)$ (*i.e.*, y_i prefers to use the route R_iQ_{i+1} instead of the *spoke path* Q_i).

In all of these conditions, we interpret the subscripts modulo k. Note that there is no requirement that the paths be disjoint, although each permitted path is (by definition) simple.

After seeing that violations of these conditions allow for instances in which nodes have incentive to lie, we will see an example in the next section of a network in which all of these conditions are satisfied but, because of the particular form of the quasi-bilinear utility function, there is still an incentive to lie.

4.1 Policy consistency

Example 4.2. Figure 9 shows a network in which policy consistency is violated; as a result, one node (node 3 here) may have an incentive to announce a route that is less preferred than the one it is actually using for forwarding if that node benefits from carrying the traffic from another node (node 4). The policies of 3 and 4 are inconsistent because $\phi_3(31d) > \phi_3(32d)$ but $\phi_4(432d) > \phi_4(431d)$. Because there is a route—$4d$—on which ϕ_4 takes a value between those it takes on $432d$ and $431d$, node 3 has an incentive to announce its less preferred route to 4 if it will benefit from carrying 4's traffic (instead of having 4 send it along $4d$).

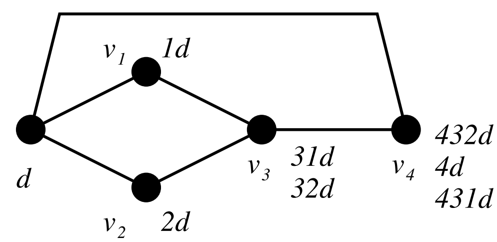

Figure 9: Network for Example 4.2.

4.2 Consistent filtering

Example 4.3. Figure 10 shows a network in which consistent filtering is violated (by filtering the route $431d$ that appeared in the FS-SPP instance in Fig. 9) and a node has an incentive to lie; here, node 3 may thus choose its more preferred route $31d$ while advertising its less preferred route $32d$ to node 4. Node 4 thus forwards data through node 3 (choosing its more preferred path $432d$) instead of directly to d (4's second choice route). Node 4's traffic will, however, be directed along the path $431d$, a path that is not permitted at 4 (because it has been filtered inconsistently).

More generally, if a there is an edge along which paths are filtered inconsistently and both paths are simultaneously available to the announcing node v, then v might have an incentive to lie by choosing its more-preferred path (which is filtered when announced to the other node w) while announcing is less-preferred path (which is not filtered) to w. Whether

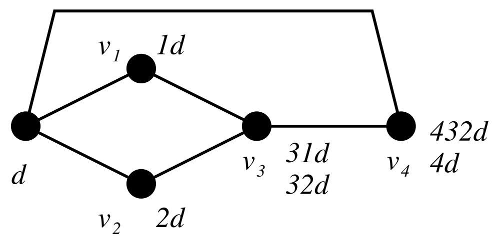

Figure 10: Network for Example 4.3

or not w chooses this path will depend on the other routes available to it, but if w's only physical link to the destination is through v, v will definitely have an incentive to lie.

4.3 Route verification

An example of a network without route verification in which a node has an incentive to lie is given in [LSZ06]

4.4 Dispute-wheel freeness

Example 4.4. Figure 11 shows the SPP instance BAD GADGET [GSW02], which is a dispute wheel with three nodes on its rim and in which nodes have incentive to lie. If any of the rim nodes i benefits by transiting extra traffic, it has an incentive to announce its less preferred route (id) to its neighbor $i-1$ while forwarding traffic to its neighbor $i+1$. (If all three nodes do this then a forwarding loop is formed.) Here, the list of paths next to each node shows the relative values of the node's forwarding-preference function with the most-preferred path at the top of the list.

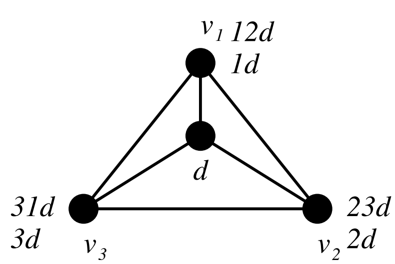

Figure 11: Network for Example 4.4.

If a network contains a dispute wheel with only two pivots then the analogous action by either pivot will produce a forwarding loop. For dispute wheels with three or more pivots, the pivots will again have an incentive to lie, but no single lie will create a forwarding loop (which might make the lie easier to detect).

5 Utilities, Policies, and Filtering

In this section we explore some of the subtle tradeoffs between different assumptions about routing policies, filtering, and incentive compatibility. We start with an example that shows that, for a certain type of quasi-bilinear utility function, satisfaction of all four of the constraints described in the previous section does not guarantee incentive compatibility.

Example 5.1. Consider the signaling utility defined by

$$\Sigma_v^{\text{nbr}}(D) = |\{w \in N(v)|(w,v) \in D\}|\,.$$

This might arise from out-of-band business relationships that v has with its neighbors; these could reflect, *e.g.*, whether or not each link carried any traffic during a specified period. The example in Fig. 12 shows that v can increase its utility (even if the network is dispute-wheel free, consistent filtering, policy consistency, and route verification) by announcing a route that it is not using to forward or by filtering the route that it is using to forward and not announcing any route. Note that nodes 5 and 6 *do not* have next-hop preferences.

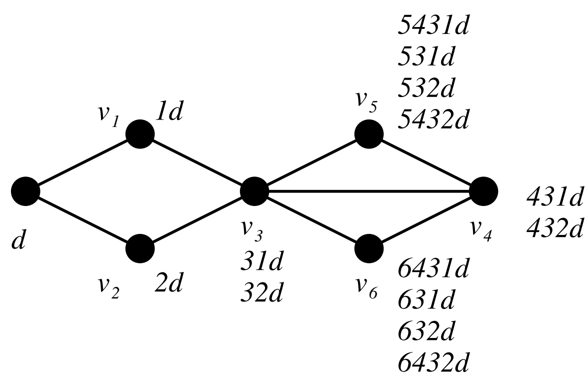

Figure 12: Network for Example 5.1

In this example, if the routes $31d$ and $32d$ are always available to node 3, this node has an incentive to announce its less-preferred route $32d$ (which it is not using for forwarding) so that node 4 cannot announce the route $431d$ to nodes 5 and 6. When 3 lies in this way, nodes 5 and 6 do not learn their most-preferred routes (through node 4) and thus choose their second-choice routes, $531d$ and $631d$, respectively; this increases the number of node 3's neighbors that forward data to it. If $432d$ were not permitted at node 4 (or if 4 had another path P with $\phi_4(431d) > \phi_4(P) > \phi_4(432d)$) then 4 would not forward to 3, but this would be offset by the gain of both 5 and 6 forwarding to 3.

Definition 5.2 (Next-hop preferences)**.** We say that a node v in a network has *next-hop preferences* if $\phi_v(P) > \phi_v(Q) > \phi_v(R)$ and $\mathsf{next}(P) = \mathsf{next}(R)$ together imply $\mathsf{next}(Q) = \mathsf{next}(P) = \mathsf{next}(R)$. We may then give an ordering on neighbors to determine which path is chosen by the node. We say that the network uses next-hop routing if every node in the network has next-hop preferences.

Unlike the previous example, the nodes in Example 5.3 all have next-hop preferences. As in Example 5.1, we use let the signaling utility be Σ_v^{nbr} (which depends on the *neighbors* of v that use v as their next hop), and nodes do not filter unless acting strategically. We see that this instance is not incentive compatible. In contrast to this is Thm. 5.4 below, in which the utility depends on the set of nodes whose data *eventually* flows through a node (not necessarily as the next hop).

Example 5.3. Figure 13 shows a network in which nodes have next-hop preferences; here, *e.g.*, the 32$*$ above the 31$*$ indicates that 3 prefers routes through 2 to routes through 1. If no node acts strategically, then 2 forwards data to 1 (who forwards data to d) while 3 and 4 both forward data to 2. If 1 filters all of its route announcements to 2, then 2 will forward through one of 3 and 4, who forwards through 1, while the other one of these two nodes will forward through 2. 1 could determine which of these solutions is chosen by additional action; it might do this if its utility depended on which node(s) choose 1 as their next-hop and not just which nodes send their data (eventually) through 1.

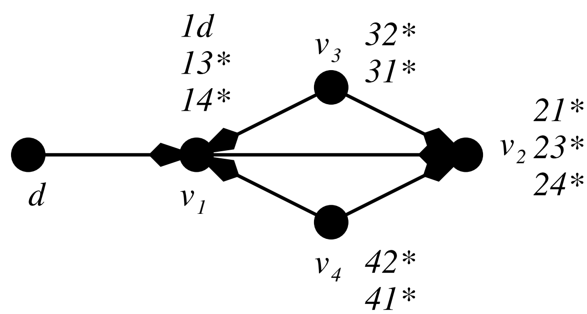

Figure 13: Network for Example 5.3

Theorem 5.4. *In a network instance in which every node has next-hop preferences and filtering is not allowed (except as part of strategic actions), if a node v unilaterally acts strategically in such a way that its forwarding path is unchanged but its signaling-utility strictly increases because a node is added to $D_S(v)$, then the forwarding preferences induce a dispute wheel with two pivots.*

Proof. Let P be the path that v chooses for forwarding when no node acts strategically, and let v act strategically in a way that P is still v's most-preferred (available) path for forwarding in the resulting scenario (in which no other node acts strategically). Assume that the traffic from some node x is now eventually forwarded to v, but that traffic from x was not forwarded through v before, and let $xRvW0$ be the path that x chooses for forwarding when v acts strategically. (Traffic from x will be forwarded along $xRvP0$ because all nodes other than v are not acting strategically, although P and W need not be distinct.) Let y be the node closest to v on R that does not forward data through v when no node acts strategically, and let T be the path that it chooses in this case; x is one such node, so y and T exist and are well-defined. Because $w = \mathsf{next}(y_0Rv)$ does not forward data through y (by choice of y and because y is not forwarding through v) and because nodes do not filter unless acting strategically, w has another path Q (because it would prefer forwarding through y over not having any path) that it announces to y. Because y still chooses T, it must prefer paths through $z = \mathsf{next}(T)$ to paths through w. Because y prefers routes whose next hop is z to those whose next hop is w one of the latter routes when v acts strategically, y must not learn any route from z in this case. Observe that 0 will always announce itself to its neighbor on T; moving along T away from 0, each node will announce its most-preferred forwarding path to the next neighbor on T (on the side away from 0) because none of these nodes acts strategically. If one of these announcements is filtered, *e.g.*, from b to a (neighbors on T, with b closer to 0 along T) then it must be because the announced path would not be simple at a; in that case, a has another path that it announces to its other neighbor along T (who might filter it because it's not a simple path, leading to another application of this argument). Iterating this argument, we see that y does not receive a valid path from z because z has chosen a path that eventually leads through y. If z prefers this path to its original subpath of T, then we may construct a dispute wheel with y and z as the only pivots. If z does not, then z would choose the path it gets from its (non-y) neighbor on T if it got a valid route from that neighbor; that neighbor must thus be choosing a route that goes through z (and which thus extends $yRvW0$, because that is the only route announced by y and extensions of this will be announced non-strategically—only v can act strategically, but v will filter every extension of this route as non-simple). We may iterate this argument until we reach a node on T that prefers an extension of $yRvW0$ over its original subpath of T; this process stops before we reach 0. We may then use this node and y as pivots on a dispute wheel (involving the forwarding preferences of nodes). □

We note that in the preceding proof, forwarding preferences were considered, but we produced a dispute wheel (defined in terms of signaling preferences). This is fine because no node except for v acts strategically and everybody (except possibly v) uses next-hop routing; we may replace the subpath P on a spoke of the (forwarding) dispute wheel with W (as announced by v) in order to obtain a proper signaling dispute wheel.

Considering the proof of this theorem, we see that once the network is stably in the first solution (when v does not act strategically), v will not be able to do anything (including withdrawing all of its routes in a sort of 'reboot') to push the network into the other forwarding solution. The following example, which has multiple solutions, illustrates this.

We may also consider utility functions where the signaling component of v's utility depends on the neighbors of v that choose routes for which v is the next hop. This might reflect out-of-band business relationships that consider whether or not a link is used, but not the level of use.

Example 5.5. Consider the network with next-hop preferences shown in Fig. 14, in which 1 prefers d over 2 and 2 prefers 1 over d. If d does not act strategically the resulting forwarding tree is $2 \longrightarrow 1 \longrightarrow d$, while if d filters its announcements to 1 (so that 1 does not learn the direct route to d), then the resulting forwarding tree is $1 \longrightarrow 2 \longrightarrow d$. We see that even if the first signaling/forwarding solution is chosen, d may act unilaterally to make a node (in this case 2) choose a route with d as the next hop when that node was originally choosing a route whose next hop was not d. Note that this does not contradict the observations above because 2 *was* forwarding data to d in the non-strategic case, just not as the next hop on 2's selected path.

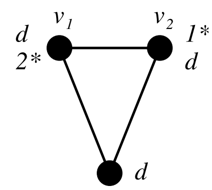

Figure 14: Network from Example 5.5

Example 5.6. In the network shown in Fig. 15, x can only forward through y; we have $\phi_y(z*) > \phi_y(v*)$ (*i.e.*, y prefers routes through z to routes through v), $\phi_z(y*) > \phi_z(d*)$, and $\phi_v(d*) > \phi_v(y*)$. There are two stable solutions: v forwards data directly to d and all other nodes forward along their respective subpaths of $xyzd$; and x and z both forward data to y, which forwards data to v, which forwards data to d. It may be that v derives greater utility from the first solution (either because of something special about transiting traffic from x and/or y, or simply because this leads to more nodes in $D_S(v)$); by the argument given in the proof of Thm. 5.4, v cannot unilaterally force a switch from the first solution to the second. (Even if v 'reboots' or, equivalently, filters all of its outgoing routes and then re-announces routes strategically.) Once the network settles into the first solution, y will always know a path through z as long as z has a path; because the path in the first solution does not go through v and because no routes are filtered, nothing v does can affect this.

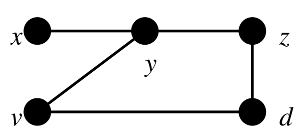

Figure 15: Network for Example 5.6

Example 5.7. If a next-hop network allows arbitrary per-route filtering, then it is possible for a node to benefit by acting strategically; one such example is shown in Fig. 16.

In this network, $\phi_v(b*) > \phi_v(c*)$ (*i.e.*, v prefers paths through b to paths through c) and no other paths are permitted at v; a and u's only permitted paths are through v; $\phi_w(u*) > \phi_w(a*)$ but $wuvbd$ is *not* permitted at w (it is filtered between u and w); $\phi_y(w*) > \phi_y(z*)$ but $ywavbd$ is *not* permitted at y; and the permitted paths at b, c, x, and z are exactly those they receive from d, d, y, and d, respectively.

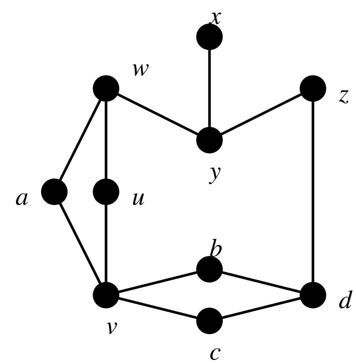

Figure 16: Network for Example 5.7

With these conditions, the nodes in this network all use next-hop routing; some of them filter specific routes (*i.e.*, $wuvbd$ and $ywvbd$). As a result, when v does not act strategically (and thus announces vbd), w chooses $wavbd$; y must then choose the path through its second-choice neighbor z. As a result, $D_S(v)$ contains exactly the vertices a, u, and w. If v's signaling utility would increase by having x route data through v, then v has an incentive to announce vcd to u, which announces $uvcd$ to w. Because this path is permitted at w and $\phi_w(u*) > \phi_w(a*)$, w now chooses the route $wuvcd$ (even though its data is forwarded along $wuvbd$) and announces this route to y; this route is permitted at y, so y now forwards data to y. As a result, x and y are added to $D_S(v)$, increasing v's utility.

Remark 5.8. Next-hop routing with no (non-strategic) filtering might seem unlikely in light of the arbitrary filtering allowed by BGP (and SPP), but it is not merely an academic special case. Next-hop routing with selective filtering of routes (the per-route filtering considered above) is arguably less-likely; it involves disregarding the actual path, other than the next hop, when considering which route to use for forwarding, but then relying on these details to determine whether or not to filter each route to each neighbor. If, instead, the per-neighbor approach to forwarding is also carried over to signaling/filtering so that each node decides whether or not to announce routes to each of its neighbors (and then follows this without consideration of the details of the routes), we may view this as removing (one direction of) the filtered edges and then not filtering at all on the remaining (directed) edges.

Example 5.9 (Strategic actions force change in solutions)**.** Figure 17 shows a network (similar to Disagree) with next-hop preferences that contains a two-pivot dispute wheel (with pivots x and z, no other rim nodes, and spokes xvd and $zyvd$); we assume that no filtering takes place, other than the removal of simple paths, unless a node is acting strategically. If no node acts strategically, then there are two solutions: one has forwarding tree $x \longrightarrow z \longrightarrow y \longrightarrow v \longrightarrow d$ and the other has forwarding tree $y \longrightarrow v \longrightarrow d \cup z \longrightarrow x \longrightarrow v \longrightarrow$. If the network is in one of these stable solutions, v may force it to go to the other one by acting strategically. In particular, v may force a move from the first solution to the second one by withdrawing its advertisement to y and then reannouncing a route to y once x

and v have started forwarding data through v; if v does not reannounce the route to y, then the network will converge to the forwarding tree $y \longrightarrow z \longrightarrow x \longrightarrow v \longrightarrow d$. Conversely, v may force a move from the second solution to the first by withdrawing its announcement to x (whether or not v reannounces a route to x after the network reconverges has no effect).

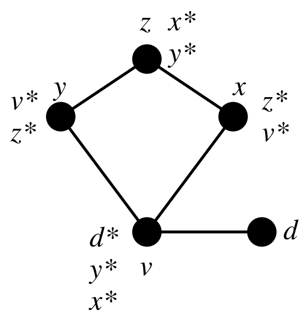

Figure 17: Network from Example 5.9

Of particular interest is the fact that, in forcing a switch from the first solution to the second solution, v may unilaterally act strategically to add a node to the set of its neighbors that choose routes whose next hop is v without any nodes being removed from this set as a result of the strategic action. (As required by Thm. 5.4, the node added to this set was previously forwarding through v, but not with v as the next hop on its chosen path.)

Theorem 5.10. *In a network instance in which every node has next-hop preferences, filtering is not allowed (except as part of strategic actions), and there is no dispute wheel, if a node v unilaterally acts strategically in such a way that its forwarding path is unchanged but nodes are added to the set of its neighbors that choose routes whose next hop is v, then some other node(s) must be removed from this set as a result of the strategic action.*

Proof. Let x be a neighbor of v that chooses to route directly through v when v acts strategically but not when v acts non-strategically. Let P be the subpath from x to v of the path that x chooses when v does not act strategically, let y be the last node before v on P, and let $z = \mathsf{next}(P)$ (not necessarily distinct from y). Note that x prefers routes from z to routes from v (because a route from v must also be available when v does not act strategically and x routes through z).

When v acts strategically, x routes through v, so x must not receive any valid route from z; thus, if z has a route, it must go through x. Note that if z prefers this route over a route that it learns from $\mathsf{next}(P_{[z,v]})$ then there is a dispute wheel with x and z as the two pivots. Thus, if z has a route, it must not learn a valid route from its neighbor $\mathsf{next}(P_{[z,v]})$; repeating the argument above, if this node chooses a route then it must go through z and x. As before, this route must be less-preferred than any route through this node's other neighbor on P, otherwise there would be a dispute wheel. Iterating this argument, we see that if y has any route then it must go through the other nodes on $P_{[x,y]}$ in reverse order (possibly with extra nodes—other than v—inserted). If this route is more preferred at y than the route directly from y to v, then we have a dispute wheel with y and x as the pivots; if not, then v must not export any route to y (else y would choose that route), so when v acts strategically y no longer routes directly through v. □

In fact, this also proves the following.

Corollary 5.11. *In a network instance in which every node has next-hop preferences, filtering is not allowed (except as part of strategic actions), and there is no dispute wheel, if a node v unilaterally acts strategically in such a way that its forwarding path is unchanged but nodes are added to the set of its neighbors that choose routes whose next hop is v, then the size of this set cannot strictly increase as a result of the strategic action.*

Proof. If v acts strategically and, as a result, nodes $\{x_i\}$ route directly through v when the did not do so when v did not act strategically, then for every i let y_i be the last node (before v) on the path that x_i chooses when v does not act strategically. The argument above shows that when v acts strategically, y_i chooses a path whose last hop (before v) is x_i. If $y_i = y_j$ for any $i \neq j$, then we must also have $x_i = x_j$. □

6 Conclusions and Future Work

We introduce a more realistic setting in which to analyze the interdomain-routing problem, that of quasi-bilinear utilities, where nodes receive benefit not only based on the route used to forward traffic, but also based on the carriage of transit traffic. In this setting, we are able to give both positive results—that certain assumptions on nodes' preferences do remove the incentive to lie in signaling—and negative results—that even when many previously studied assumptions are satisfied, nodes may have an incentive to lie. The consequences of this go beyond incentive compatibility. We extend the abstract model of interdomain routing to permit the decoupling of signaling and forwarding actions, and show that inconsistency between the two can lead to forwarding loops. Removing an incentive to lie about what routes are used is one way to prevent these loops that could lead to data loss.

There are several open questions that naturally result from this work. First, there are several examples of quasi-bilinear utilities not examined here that can be motivated by real-world scenarios. Do these also require strong assumptions on nodes' preferences to guarantee incentive compatibility of BGP? Second, many of the positive results give sufficient conditions for incentive compatibility that are not matched with necessary conditions. This mirrors much of the previous work in interdomain-routing theory; is it possible to find necessary conditions? (Our examples of incentive-incompatiblity are a step in this direction.) Finally, can even more realistic utility functions and models of interdomain routing be developed to further bridge the gap between the networking and game-theoretic perspectives?

Other problems of interest include a more exhaustive study of the effects of different filtering restrictions.

References

[FKMS05] J. Feigenbaum, D. Karger, V. Mirrokni, and R. Sami, "Subjective-Cost Policy Routing," in *Proc. WINE'05*, pp. 174–183, Dec. 2005.

[FPSS05] 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.

[FRS06] J. Feigenbaum, V. Ramachandran, and M. Schapira, "Incentive-Compatible Interdomain Routing," in *Proc. ACM EC'06*, pp. 130–139, Jun. 2006.

[FSS07] J. Feigenbaum, M. Schapira, and S. Shenker, "Distributed Algorithmic Mechanism Design," Chapter 14 in *Algorithmic Game Theory*, Cambridge, 2007.

[FSS04] J. Feigenbaum, R. Sami, and S. Shenker, "Mechanism Design for Policy Routing," in *Proc. ACM PODC'04*, pp. 11–20, Jul. 2004.

[GGR01] L. Gao, T. G. Griffin, and J. Rexford, "Inherently Safe Backup Routing with BGP," in *Proc. IEEE INFOCOM 2001*.

[GR01] L. Gao and J. Rexford, "Stable Internet Routing Without Global Coordination." *ACM/IEEE* Trans. Net. **9**(6):681–692, Dec. 2001.

[GSW02] T. G. Griffin, B. F. Shepherd, and G. Wilfong, "The Stable Paths Problem and Interdomain Routing." *ACM/IEEE* Trans. Net. **10**(2):232–243, Apr. 2002.

[Hus99] G. Huston, "Interconnection, Peering, and Settlements," in *Proc. INET'99*, The Internet Society, Jun. 1999.

[KLS00] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (S-BGP)," *IEEE J. Sel. Areas Comm. (JSAC)* **18**(4), Apr. 2000.

[LSZ06] H. Levin, M. Schapira, and A. Zohar, "The Strategic Justification for BGP." Leibniz Cent. Research in Comp. Sci., Tech. Rep. 2006–80, Nov. 2006.

[LSZ07] H. Levin, M. Schapira, and A. Zohar, "Interdomain Routing and Games."

[MRWK03] Z. M. Mao, J. Rexford, J. Wang, and R. Katz, "Towards an Accurate AS-Level Traceroute Tool," in *Proc. ACM SIGCOMM'03*, pp. 365–378, Aug. 2003.

[PS02] V. N. Padmanabhan and D. R. Simon, "Secure Traceroute to Detect Faulty or Malicious Routing," in *Proc. HotNets-I*, pp. 77–82, Oct. 2002.

[SP04] J. Shneidman and D. C. Parkes, "Specification Faithfulness in Networks with Rational Nodes," in *Proc. ACM PODC'04*, pp. 88–97, Jul. 2004.

[Sob03] J. L. Sobrinho, "Network Routing with Path Vector Protocols: Theory and Applications," in *Proc. ACM SIGCOMM'03*, pp. 49–60, Aug. 2003.

[TV07] É. Tardos and V. V. Vazirani, "Basic Solution Concepts and Computational Issues," Chapter 1 in *Algorithmic Game Theory*, Cambridge, 2007.

[YXW+05] Y. R. Yang, H. Xie, H. Wang, A. Silberschatz, Y. Liu, L. E. Li, and A. Krishnamurthy, "On Route Selection for Interdomain Traffic Engineering," *IEEE Net.* **19**(6), Nov./Dec. 2005.

A Equilibrium Concepts

Here we provide a brief overview of different game-theoretic solution concepts. These are drawn very closely from [TV07] (dominant strategy and Nash equilibrium) and [FSS07] (ex-post Nash equilibrium); we include them here for the convenience of the reader.

We let S_v be the set of strategies that node v may play, and $S = \times_{v \in V} S_v$ to be the resulting possible strategy vectors. We use $U_v(\vec{s})$ for v's utility when the strategy vector $\vec{s}$ is played (*i.e.*, when each u plays s_u); $\vec{s}_{-v}$ is the tuple obtained by dropping s_v, and $s'_v, \vec{s}_{-v}$ is the strategy vector obtained by replacing s_v with s'_v in $\vec{s}$.

Definition A.1 (Dominant strategy)**.** A strategy vector s is a *dominant strategy* if

$$\forall v \in V,\ \forall \vec{s}' \in S,\ U_v(s_v, \vec{s}'_{-v}) \geq U_v(\vec{s}'),$$

i.e., for every agent v, *regardless of the strategies* $\vec{s}'_{-v}$ *chosen by the other nodes in the network,* v can do no better than playing the strategy s_v.

A strictly weaker solution concept is that of Nash equilibrium, which is defined for pure strategies as follows.

Definition A.2 (Pure strategy Nash equilibrium)**.** A strategy vector s is a *Nash equilibrium* if

$$\forall v \in V,\ \forall s'_v \in S_v,\ U_v(s_v, \vec{s}_{-v}) \geq U_v(s'_v, \vec{s}_{-v}),$$

i.e., for every agent v, *if the agents other than* v *choose the strategies in* $\vec{s}_{-v}$*,* then v can do no better than playing s_v.

The ex-post Nash equilibrium has been argued [SP04] to be the most relevant notion of equilibrium for distributed games; this is the notion of equilibrium used in Sec. 5

Definition A.3 (*Ex-post* Nash equilibrium)**.** In this case each node v has a private type t_v; the behavior of v, and the outcome of the game, when v uses strategy s_v depends on t_v. In this case the utility of an agent v may depend on its private type in addition to the behavior of the various agents; we thus write $U_v(\vec{s}(\vec{t}); t_v)$ for v's utility when its private type is t_v and agent u has private type t_u and chooses strategy s_u (thus behaving in accordance with $s_u(t_u)$). Then a strategy vector $\vec{s}$ is an *ex-post Nash equilibrium*if

$$\forall v \in V,\ \forall s'_v \in S_v,\ \forall \vec{t},\ U_v(\vec{s}(\vec{t}); t_v) \geq U_v(s'_v(t_v), \vec{s}_{-v}(\vec{t}_{-v}); t_v)$$

i.e., for every agent v, *regardless of the private types of the other agents and of v's private type, if the agents other than v choose the strategies in $\vec{s}_{-v}$,* then v can do no better than playing s_v.

Thus the concept of ex-post Nash equilibrium falls between those of dominant strategy and Nash equilibrium: In a dominant strategy, no knowledge of the other nodes is assumed when asserting that v can do no better than playing s_v; in a Nash equilibrium, this conclusion holds if the strategies of the other agents are known (without holding information back in the form of private types); while in an ex-post Nash equilibrium, this conclusion follows when *some* information is known about the other agents (their strategies) while some is not (their private types).

Finally, we recall the way in which "incentive compatible" is used in this (and earlier) work on game theory and BGP.

Definition A.4 (Incentive compatibility in ex-post Nash)**.** Following [LSZ06], we say that a BGP is *incentive compatible in ex-post Nash* if the strategy vector for the interdomain-routing game in which every agent's strategy is to follow BGP is an ex-post Nash equilibrium.