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

Aaron D. Jaggard

DIMACS
Rutgers University

Joint work with Vijay Ramachandran (Colgate) and Rebecca N. Wright (Rutgers)
Partially supported by NSF and ONR

26 March 2008
DIMACS/DyDAn Workshop on Secure Internet Routing

Outline

Network model

Graph with a single destination d and other nodes trying to route data to d.

Each node v has:

Forwarding preference function $\phi_v : \mathcal{P}_v \to \mathbb{Z}$. If $\phi_v(P) > \phi_v(Q)$, then v prefers to use P instead of Q for forwarding data (if both are available).

Signaling preference functions For each neighbor w of v, a function $\sigma_{v,w} : \mathcal{P}_v \to \mathbb{Z}$. If $\sigma_{v,w}(P) > \sigma_{v,w}(Q)$, then v prefers to announce P instead of Q to w (if both are available).

Note that these preferences are static.

For now, we care about the ordering but not the cardinal values.

Assignments and Solutions

A *stable (signaling) solution* σ is essentially the same as for SPP:

- Each vertex v learns routes from its neighbors $(\{v\sigma(u, v)\}_u)$
- The route $\sigma(v, w)$ that v announces to its neighbor w is the route known to v that maximizes the signaling preference function $\sigma_{v,w}$

The *forwarding digraph induced by* σ captures how nodes forward when the paths in σ are signaled; v chooses the path it knows that maximizes its forwarding preference function ϕ_v

Solution Characteristics

Number of solutions Given an FS-SPP instance, it may have zero, exactly one, or multiple signaling solutions, just as in SPP.

(A)cyclic forwarding Given a solution to a FS-SPP instance, the induced forwarding assignment may correspond to a digraph that is either cyclic or acyclic (*i.e.*, both are realizable)

- Forwarding loops in a stable solution require that at least one node lies about its forwarding
- Even if an FS-SPP solution induces an acyclic forwarding digraph, forwarding may or may not agree with signaling.

Solution Characteristics

Number of solutions Given an FS-SPP instance, it may have zero, exactly one, or multiple signaling solutions, just as in SPP.

(A)cyclic forwarding Given a solution to a FS-SPP instance, the induced forwarding assignment may correspond to a digraph that is either cyclic or acyclic (*i.e.*, both are realizable)

- Forwarding loops in a stable solution require that at least one node lies about its forwarding
- Even if an FS-SPP solution induces an acyclic forwarding digraph, forwarding may or may not agree with signaling.

Solution Characteristics

Number of solutions Given an FS-SPP instance, it may have zero, exactly one, or multiple signaling solutions, just as in SPP.

(A)cyclic forwarding Given a solution to a FS-SPP instance, the induced forwarding assignment may correspond to a digraph that is either cyclic or acyclic (*i.e.*, both are realizable)

- Forwarding loops in a stable solution require that at least one node lies about its forwarding
- Even if an FS-SPP solution induces an acyclic forwarding digraph, forwarding may or may not agree with signaling.

Agreement between Forwarding and Signaling

Definition

For a signaling solution σ, we say that *forwarding and signaling disagree in* σ if there is some node that chooses one path for forwarding but whose data is forwarded along a different path.

Combinations of Solution Characteristics

Signaling solutions?			Forwarding loops?		
None	Unique	Multiple	Yes	No; F-S agree?	
				No	Yes
X					
	X		X		
	X			X	
	X				X
		X	X		
		X		X	
		X			X

Table: Solution characteristics of various FS-SPP examples.

S-Dispute Wheels

The classic dispute wheel translates naturally to the FS-SPP framework. Because this involves only signaling, we refer to these as *S-dispute wheels*.

Classic SPP results carry over immediately to the signaling aspects of FS-SPP. In particular:

Theorem (Essentially Griffin-Shepherd-Wilfong)

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

Note that this does not guarantee anything about forwarding.

S-Dispute Wheels

The classic dispute wheel translates naturally to the FS-SPP framework. Because this involves only signaling, we refer to these as *S-dispute wheels*.

Classic SPP results carry over immediately to the signaling aspects of FS-SPP. In particular:

Theorem (Essentially Griffin-Shepherd-Wilfong)

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

Note that this does not guarantee anything about forwarding.

Unique Stable Signaling with a Forwarding Loop

In particular, an FS-SPP instance may be S-dispute-wheel-free and thus have a unique signaling solution, but the induced forwarding digraph need not be acyclic.

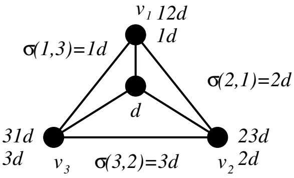


Figure: S-DW-free FS-SPP instance whose unique signaling solution induces a forwarding loop.

Nodes prefer to signal their direct paths and forward along their indirect paths.

FS-Dispute Wheels

Define a new type of wheel structure, the *Forwarding/Signaling Dispute Wheel* (FS-Dispute Wheel).
Similar to regular dispute wheels, but:

- Pivots prefer to forward along rim instead of spoke
- Pivots prefer to signal spoke path (to neighbor along next rim segment) instead of rim path

Theorem

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

Note that FS-DW-freeness does not guarantee a unique stable solution or agreement between forwarding and signaling.

FS-Dispute Wheels

Define a new type of wheel structure, the *Forwarding/Signaling Dispute Wheel* (FS-Dispute Wheel).
Similar to regular dispute wheels, but:

- Pivots prefer to forward along rim instead of spoke
- Pivots prefer to signal spoke path (to neighbor along next rim segment) instead of rim path

Theorem

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

Note that FS-DW-freeness does not guarantee a unique stable solution or agreement between forwarding and signaling.

Motivation for Gao-Rexford Constraints

- An AS *does* provide transit services for its customers
  - In SPP, may export any route to customers
  - In FS-SPP, may signal any route to customers
- An AS *does not* provide transit services for its non-customers
  - In SPP, may export only customer routes to non-customers
  - In FS-SPP, may signal any route to non-customers, but only when forwarding through a customer; when forwarding through a non-customer, must not signal *any* route at all
- Prefer routes learned from customers (because no payments to customers to carry traffic)
  - In SPP, prefer routes learned from customers
  - In FS-SPP, prefer to *forward* through customers; no preference about which routes to signal

Motivation for Gao-Rexford Constraints

- An AS *does* provide transit services for its customers
  - In SPP, may export any route to customers
  - In FS-SPP, may signal any route to customers
- An AS *does not* provide transit services for its non-customers
  - In SPP, may export only customer routes to non-customers
  - In FS-SPP, may signal any route to non-customers, but only when forwarding through a customer; when forwarding through a non-customer, must not signal *any* route at all
- Prefer routes learned from customers (because no payments to customers to carry traffic)
  - In SPP, prefer routes learned from customers
  - In FS-SPP, prefer to *forward* through customers; no preference about which routes to signal

Motivation for Gao-Rexford Constraints

- An AS *does* provide transit services for its customers
  - In SPP, may export any route to customers
  - In FS-SPP, may signal any route to customers
- An AS *does not* provide transit services for its non-customers
  - In SPP, may export only customer routes to non-customers
  - In FS-SPP, may signal any route to non-customers, but only when forwarding through a customer; when forwarding through a non-customer, must not signal *any* route at all
- Prefer routes learned from customers (because no payments to customers to carry traffic)
  - In SPP, prefer routes learned from customers
  - In FS-SPP, prefer to *forward* through customers; no preference about which routes to signal

Motivation for Gao-Rexford Constraints

- An AS *does* provide transit services for its customers
  - In SPP, may export any route to customers
  - In FS-SPP, may signal any route to customers
- An AS *does not* provide transit services for its non-customers
  - In SPP, may export only customer routes to non-customers
  - In FS-SPP, may signal any route to non-customers, but only when forwarding through a customer; when forwarding through a non-customer, must not signal *any* route at all
- Prefer routes learned from customers (because no payments to customers to carry traffic)
  - In SPP, prefer routes learned from customers
  - In FS-SPP, prefer to *forward* through customers; no preference about which routes to signal

Motivation for Gao-Rexford Constraints

- An AS *does* provide transit services for its customers
  - In SPP, may export any route to customers
  - In FS-SPP, may signal any route to customers
- An AS *does not* provide transit services for its non-customers
  - In SPP, may export only customer routes to non-customers
  - In FS-SPP, may signal any route to non-customers, but only when forwarding through a customer; when forwarding through a non-customer, must not signal *any* route at all
- Prefer routes learned from customers (because no payments to customers to carry traffic)
  - In SPP, prefer routes learned from customers
  - In FS-SPP, prefer to *forward* through customers; no preference about which routes to signal

Motivation for Gao-Rexford Constraints

- An AS *does* provide transit services for its customers
  - In SPP, may export any route to customers
  - In FS-SPP, may signal any route to customers
- An AS *does not* provide transit services for its non-customers
  - In SPP, may export only customer routes to non-customers
  - In FS-SPP, may signal any route to non-customers, but only when forwarding through a customer; when forwarding through a non-customer, must not signal *any* route at all
- Prefer routes learned from customers (because no payments to customers to carry traffic)
  - In SPP, prefer routes learned from customers
  - In FS-SPP, prefer to *forward* through customers; no preference about which routes to signal

FS-GR Constraints

- Consistent classification of neighbors
- Unconstrained signaling when forwarding through a customer
- Only signal to customers when forwarding through a non-customer
- Prefer to *forward* through customers

  Preference for what to *signal* is unconstrained
- No customer-provider cycles in network

What FS-GR Guarantees for FS-SPP

Theorem (Essentially Gao-Griffin-Rexford)

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

Theorem

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

FS-GR constraints alone don't guarantee the network will converge, but if it does there won't be forwarding loops.

What FS-GR Guarantees for FS-SPP

Theorem (Essentially Gao-Griffin-Rexford)

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

Theorem

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

FS-GR constraints alone don't guarantee the network will converge, but if it does there won't be forwarding loops.

What FS-GR Guarantees for FS-SPP

Theorem (Essentially Gao-Griffin-Rexford)

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

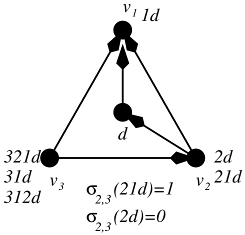
Theorem

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

FS-GR constraints alone don't guarantee the network will converge, but if it does there won't be forwarding loops.

An FS-GR Example

The FS-GR constraints do not guarantee that nodes will be truthful.

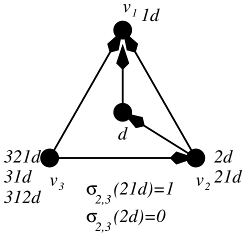


(More generally, this shows that even an FS-DW-free network need not have agreement between forwarding and signaling.)

How do we ensure agreement between forwarding and signaling? Look at incentive compatibility of best-reply dynamics (including truthful announcements).

An FS-GR Example

The FS-GR constraints do not guarantee that nodes will be truthful.

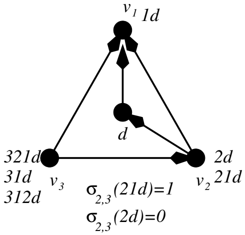


(More generally, this shows that even an FS-DW-free network need not have agreement between forwarding and signaling.)

How do we ensure agreement between forwarding and signaling? Look at incentive compatibility of best-reply dynamics (including truthful announcements).

An FS-GR Example

The FS-GR constraints do not guarantee that nodes will be truthful.



(More generally, this shows that even an FS-DW-free network need not have agreement between forwarding and signaling.)

How do we ensure agreement between forwarding and signaling? Look at incentive compatibility of best-reply dynamics (including truthful announcements).

The Network Routing Game

As in FS-SPP, we implicitly assume route verification: nodes only announce routes that they have learned, but they may announce a (known) route other than the one used for forwarding.

Game otherwise the same as before, but utility functions differ.

Solution concept

Definition (Best-reply dynamics)

v follows *best-reply dynamics* if it:

- Receive current route updates from neighbors
- Select the 'best' forwarding route from the known routes
- Signal the selected forwarding route to neighbors (filtering as required/allowed)

Use *ex-post* Nash equilibrium solution concept throughout.
If every node other than v follows best-reply dynamics, then v has no incentive deviate from best-reply dynamics. In particular, nodes signal the route they use for forwarding.

Bi-quasi-linear Utilities

We assume that v's utility in a stable signaling solution σ has the form:

$$U_v(\sigma) = F_v(\sigma) + S_v(D_{\sigma \to v})$$

- F_v is v's *forwarding utility*; this depends on the route that v chooses (which is *not* necessarily the route along which v's data are forwarded, but which seems more likely to motivate v's decisions)
- S_v is v's *signaling utility*; this depends on the part of forwarding digraph induced by σ from which v is reachable

Bi-quasi-linear Utilities

We assume that v's utility in a stable signaling solution σ has the form:

$$U_v(\sigma) = F_v(\sigma) + S_v(D_{\sigma \to v})$$

- F_v is v's *forwarding utility*; this depends on the route that v chooses (which is *not* necessarily the route along which v's data are forwarded, but which seems more likely to motivate v's decisions)
- S_v is v's *signaling utility*; this depends on the part of forwarding digraph induced by σ from which v is reachable

Lots of examples to show that dropping various conditions allows networks that are not incentive-compatible.
Four conditions

- Policy consistency
- Consistent filtering
- Route verification
- No dispute wheel

have been studied (in various combinations) to guarantee incentive-compatibility of BGP with usual utilities. Dropping any one of these allows a network in which BGP is not incentive-compatible with these utility functions.

Nodes Eventually Routing through v

Assume the signaling utility increases if a node is added to the set of nodes whose traffic is (eventually) forwarded through v.

Other nodes may or may not be removed from this set

Theorem

If every node has next-hop preferences and filtering is not allowed (except as a strategic action), if v unilaterally acts strategically such that its forwarding path is unchanged but its signaling utility increases, then the forwarding preferences induce a dispute wheel with two pivots.

Nodes Eventually Routing through v

Considering the proof of the preceding theorem, we can even do a little bit better.

Corollary

In the preceding scenario (next-hop, no non-strategic filtering, adding a node to the set that eventually routes through v), if the network is in one stable solution, then v cannot act unilaterally to force the network into the other stable solution.

Nodes Directly Routing through v

Theorem

If every node has next-hop preferences, filtering is not allowed (except as a strategic action), and there is no dispute wheel, if v unilaterally acts strategically such that its forwarding path is unchanged but one or more 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.

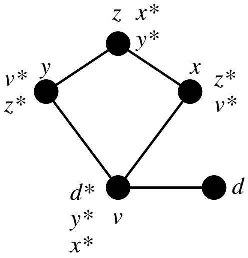
Nodes Directly Routing through v

Corollary

If every node has next-hop preferences, filtering is not allowed (except as a strategic action), and there is no dispute wheel, if v *unilaterally acts strategically such that its forwarding path is unchanged but one or more nodes are added to the set of its neighbors that choose routes whose next hop is* v*, then the size of this set cannot increase as a result of the strategic action.*

Example for Nodes Directly Routing through *v*

Unlike the previous case, *v* may be able to act strategically to choose *which* (but not how many) neighbors choose *v* as their next hop (even if the network had converged to a different solution).



Conclusions

- Defined FS-SPP framework to decouple forwarding from signaling
    - No FS-DW guarantees stable solutions have acyclic forwarding
    - FS-GR constraints preclude FS-DWs and, with additional signaling restriction, also preclude S-DWs
- Studied bi-quasi-utility functions in network routing game
    - Examples start to show boundary of incentive-compatibility
    - Incentive-compatibility conditions for different assumptions on signaling utilities