Formal Models for Path-Vector Protocols

Aaron D. Jaggard (Tulane)
Vijay Ramachandran (Stevens)

Partially supported by ONR, NSF, and the Stevens Technogenesis Program

Goals

- Are there areas for collaboration?
  - What have we been doing?
  - What have others been doing?
- Feedback on previous/current work
  - In which ways this could be refined to be more useful?
- Questions for future work
  - Are there aspects of routing policies that would be good to look at next?

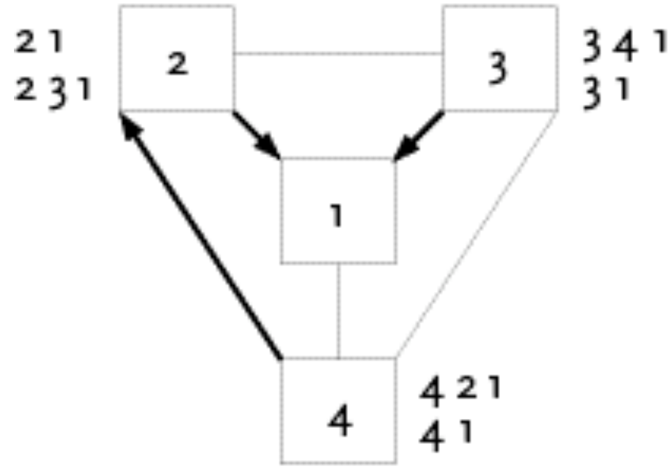
The Design Space of Path-Vector Protocols [GJR '03]

- Robustness: Predictable routing tree, even after link/node failure
  - Primary concern
- Expressiveness: What routing policies are permitted?
  - Use the *Stable Paths Problem* as semantic domain
- Autonomy: What degree of independence do operators have in local-policy configuration?
  - One example: *next-hop policies*, which can contradict shortest-paths routing
- Global Constraint: What assumptions about the network are needed?
- Protocol details:
  - Policy Opaqueness: Can local route settings be kept private?
  - Protocol Transparency: How directly does the protocol apply local policy to route data?

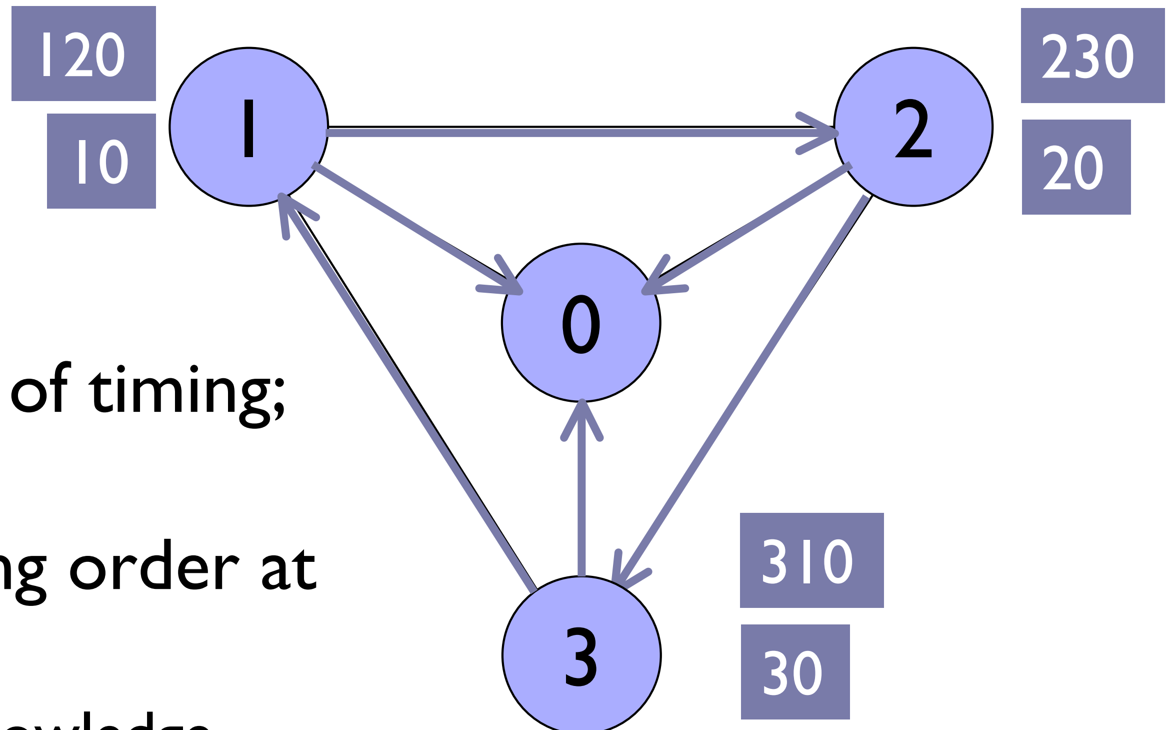
Formally Modeling Policy Semantics

- The Stable Paths Problem (SPP) models the underlying theoretical problem that eBGP is trying to solve [Griffin-Shepherd-Wilfong '02]
- SPP solvability is NP-complete; solvability $\not\Rightarrow$ convergence.

An SPP instance is a graph in which each node represents one AS and has a policy in the form of a linear preference ordering on paths.

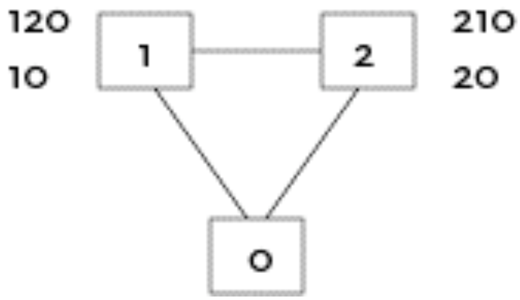


Bad Gadget [GSW'02]

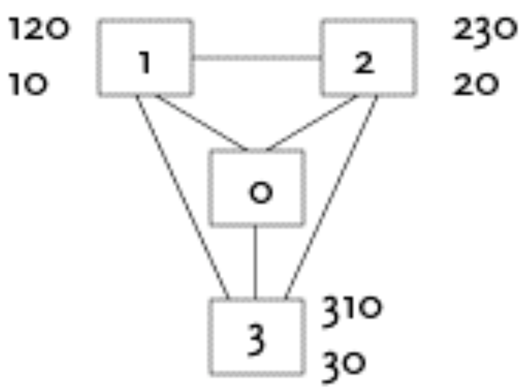


- Cycles regardless of timing; no stable solution
- Can fix by changing order at any node
  - Fix uses global knowledge
- Note: Each node prefers rim to spoke

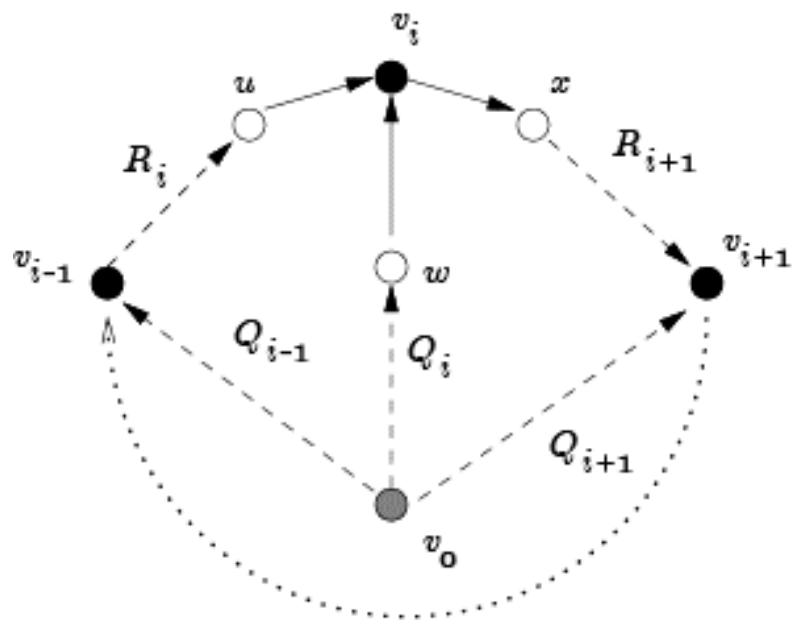
SPP Results [GSW '02]



DISAGREE (multiple solutions)



BAD GADGET (no solution)



Dispute Wheel

No dispute wheel implies robust convergence.

Path-Vector Policy Systems

[GJR '03]

Formal model of path-vector routing:

$$(PV, PL, K)$$

Path-Vector System:

The underlying message-exchange system for route information. What is exchanged and how?

Global Constraint:

What assumptions about the network must be true to achieve robustness?

Policy Language:

How can policies be described? PL acts as a local constraint on the expressiveness of policies.

Question:

What role do these components play in achieving protocol design goals?

Path-Vector Algebras [Sobrinho '03]

- A *path-vector algebra* defines:
  - Signatures (path data objects)
  - Labels (combines import and export policies)
    - Apply label to signature to obtain new signature (the path data after export/import)
  - Weight function on signatures (rank)
  - Operation to apply labels to signatures
  - Rank criteria for tie-breaking
- These abstract away some protocol-level details

Robustness Condition [GJR '03, S '03]

Theorem: A protocol in which a path's (global) rank always increases as it is extended (by export/import) is dispute-wheel-free (and thus robust).

(Assume that we prefer the path with smallest rank, as with cost.)

- Increasing systems generalize cost functions
  - Cost now assigned to (path, edge) pairs

Trade-Offs in Implementation [GJR '03]

Theorem. A transparent, robust PVPS that supports next-hop policies and is at least as expressive as shortest paths must have a non-trivial global constraint.

Corollary. A globally unconstrained, robust PVPS that is expressive enough to capture all increasing configurations either does not support next-hop policies or is not transparent, or both.

Hierarchical BGP (HBGP)

- Partition neighbors into customers, providers, and peers
- Local constraints on policies
  - Scoping: Share route data from customers with everyone, share data from everyone with customers, do not share other data
  - Relative-preference: Prefer peer routes to provider routes, customer routes to both peer and provider
- No customer/provider cycles
- HBGP is robust [Gao-Rexford '01]
  - Are constraints violated often? Why?

Extending HBGP [JR' 04]

- Use the PVPS framework to generalize the HBGP constraints of [GR' 01, GGR' 01].
- A class-based PVPS is described by:
  - A set of classes (types of neighbor assignments, e.g., customer/provider/peer) and consistency relationships between them
  - Scoping rules
  - Relative-preference rules
- These systems are transparent and support next-hop policies enough to require a nontrivial global constraint.

Class-Based Robustness [JR '04]

From the class description alone, we can construct a global constraint involving a check on pairs of class assignments.

- Prevent cycles that could form dispute wheel rims by checking two cases
- Networks obeying this constraint are robust.
- Networks violating this constraint allow nodes to write policies that induce routing anomalies.

Dispute Rings [FJB '05]

- Dispute rings specialize dispute wheels
  - No node appears more than once
- Safety under filtering generalizes robustness
  - Allow arbitrary filtering, not just all paths through a certain node or edge

Theorem: Dispute-ring-freeness is necessary for safety under filtering

- Still open: Is there a necessary condition for robustness?

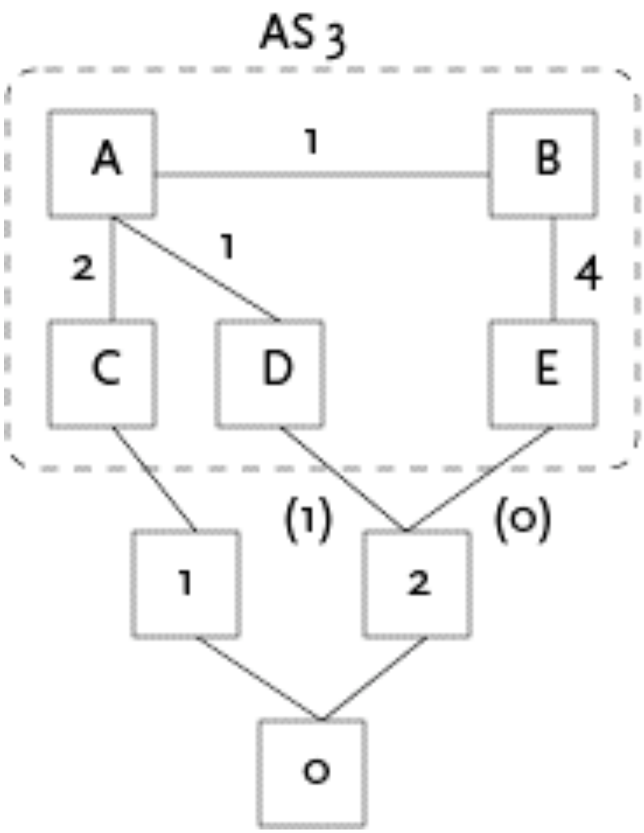
How to Model MEDs?

- Have a selection function choose one route (according to local policy) from a set of routes
  - No longer ranking paths linearly
- A singleton-valued selection function f satisfies Independent Route Ranking if, for T containing S,

  $$f(T) = P_2 \text{ implies } f(S) = P_2 \text{ or } P_2 \text{ is in } T \setminus S$$

  - Learning new routes shouldn't cause the selection of a new, previously known route
  - Potentially violated by use of MED attribute
- Second condition for set-valued selection functions
  - Source of future work; focus on singleton-valued here

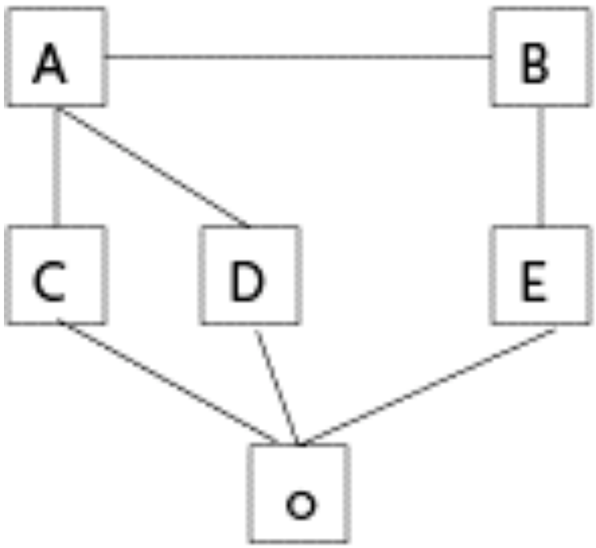
Generalized SPP [GW '02]



MED-EVIL [GW'02] (no solution)

- BGP selection:
  - lowest MED value from paths to the same AS; then
  - shortest IGP distance.
- IGP distances are shown near intra-domain links.
- MED values are shown in parentheses near inter-domain links.
- This example oscillates.

Violate IRR Using MEDs



MED-EVIL [GW'02]
(condensed)

$$\sigma_A(AC0, AD0) = AD0$$
$$\sigma_A(AD0, ABE0) = ABE0$$
$$\sigma_A(AC0, ABE0) = AC0$$
$$\sigma_A(AC0, AD0, ABE0) = AC0$$

$$\sigma_B(BE0) = BE0$$
$$\sigma_B(BAD0, BE0) = BE0$$
$$\sigma_B(BAC0, BE0) = BAC0$$

Generalized Path Relations [JR'06]

Subpath $P_1 \ominus P_2$ iff $P_1 = v \cdots v_0, P_2 = u \cdots v_0$, and $uP_1 = P_2$

Linear Selection $P_1 \oslash P_2$ iff $P_1 = v \cdots v_0, P_2 = u \cdots v_0$, and
$\exists\, S : \sigma_u \left(\{uP_1,\ P_2\} \cup S\right) = uP_1$

Nonlinear Selection 1 $P_1 \odot_1 P_2$ iff $P_1 = v \cdots v_0,\ P_2 = u \cdots v_0$,
and there exists a set of routes $S \not\ni uP_1$ such that
$\sigma_u \left(\{P_2\} \cup S\right) \neq P_2$ and $\sigma_u \left(\{uP_1,\ P_2\} \cup S\right) = P_2$

Nonlinear Selection 2 $P_1 \odot_2 P_2$ iff $P_1 = v \cdots v_0,\ P_2 = u \cdots v_0$,
and there exists a set of routes $S \not\ni uP_1$ such that
$\sigma_u(S) = P_2$ and $\sigma_u \left(\{uP_1\} \cup S\right) \notin \{uP_1,\ P_2\}$

Generalized Dispute Wheels [JR'06]

- Extend original notion of dispute wheel to include new relations between paths
  - Gives sufficient condition for robustness in generalized SPP
- Not considered in initial generalization of SPP
  - GW'02 translated limited class of GSPPs to SPP, applied SPP convergence conditions
  - Here, generalized dispute wheels apply directly to GSPP

Summary of Previous Work

- PVPS framework for study of path-vector protocols
  - Conditions needed for robustness
  - Tradeoffs involved in implementing these conditions
- Concrete and reasonable guidelines for class-based systems
- Extended framework to allow nonlinear selection
  - Start to model interactions between internal and external routing

Questions (I)

- Feedback on our work
  - How to refine it to make it more useful?
    - Can people work within class-based routing?
    - Should we focus on next-hop (plus tweaks)?
    - What are pressing (non-implementation) questions?
  - Related areas for joint work?
    - In RRG?

Questions (II)

- What sort of policies should we look at?
  - What sort of policies are typically written?
  - Are there policies you'd like to write but can't?
  - How do you want to be able to reconfigure policies?
  - What anomalies do you see? (Often?)
  - What are typical iBGP policies?

adj@math.tulane.edu
vijayr@cs.stevens.edu

www.math.tulane.edu/~adj/

www.cs.stevens.edu/~vijayr/