UC SANTA CRUZ

Attaining Stable and Loop-Free Inter-Domain Routing without Path Vectors

J.J. Garcia-Luna-Aceves

University of California at Santa Cruz

jj@soe.ucsc.edu

Outline

- **Border Gateway Protocol (BGP)**, its limitations, and prior work that has tried to address them
- **Research Questions:**
- Can we define a framework (i.e., a routing algebra) focused on “etiquettes” of polite router behavior rather than protocols based on network-wide optimality criteria?

  OPERA (in prior paper)
- Is it possible to define a routing protocol that is loop-free and is guaranteed to converge in a finite time based on local private preferences?

  Loop-free convergence w/o optimality
- If so, is it possible to adapt BGP to attain stable loop-free routing across ASes, without using path vectors or requiring careful tuning of routing policies in multiple ASes?

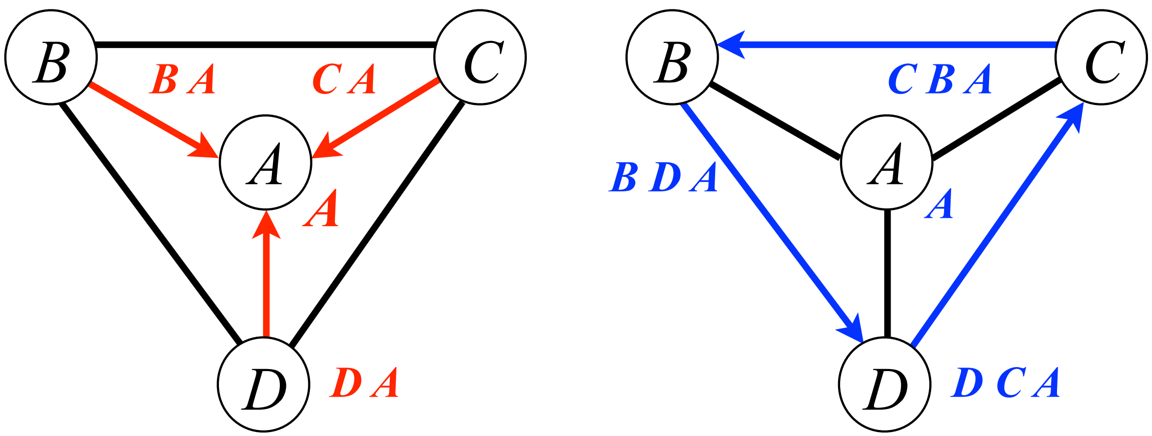
BGP Enhanced for Loop Freedom (BGP-ELF)

YES!

BGP

- **Goal: Routing across autonomous systems (AS)**
- **BGP path selection:**
  - Does not use system-wide optimality criteria to select paths
  - Path selection is based on local preferences that may be private
- **Main procedures:** neighbor acquisition, neighbor reachability, and network reachability
- Conceptually, BGP uses three policy mechanisms for routing:
  - **Import transformation** to accept paths for consideration
  - **Preference function** used to compare valid paths and select one
  - **Export transformation** used to announce preferred paths
- **Single-path policy-based routing:**
  - **Loop detection:** A valid path cannot include the AS of the router

BGP Route Oscillation Problems: (e.g., BAD-GADGET [9])



- Routers start by announcing direct AS paths to destination AS *A*
- Routers in each AS prefer two-hop paths through clockwise neighboring AS over direct paths, change and announce the new paths
- Updates make routers reverse to direct paths
- Direct-path updates make routers reverse to two-hop paths, etc.

- **No solution exists based on BGP signaling: Route oscillations persist!**
- Note the ack of coordination among ASes,and no ordering of paths selected by routers in different ASes

[9] T.G. Grifin and G. Wilfong, "An Analysis of BGP Convergence Properties," *Proc. ACM SIGOMM '99*, Aug. 1999.

Prior Work

- **Static BGP approaches:** Programs verify that routing policies do not create conflicts that prevent BGP from converging:
  - Not practical, see results in [9].
- **Dynamic BGP approaches:** Introduce modifications in BGP signaling to reduce or eliminate oscillations and looping.
  - To date, none can guarantee convergence, avoid the occurrence of temporary loops, or ensure faster convergence.
- **BGP alternatives:** New protocols based on complete AS topology information or path information
  - To date, no new inter-domain routing protocol has been shown to be loop-free and guarantee convergence

Prior Work (Conc.)

- **Routing algebras:** Many extensions to shortest-path routing algebras have been proposed, especially to address the use of path information
  - To date, no good framework exists to model routing based on local private preferences (at least w/o path vectors)
- **Safe, loop-Free routing protocols:** Several protocols have been proposed that are loop-free and converge within a finite time.
  - To date, all these protocols have assumed network-wide optimality criteria, rather than local private preferences.
  - Path-vector protocols proposed to date ***detect*** loops but ***do not eliminate them***.

Loop-Free Routing with Private Policies

- **Labeled Path Length:** The labeled path length of the nth path $P_d^k(n)$ from node k to destination d is the tuple $\ell_d^k(n) = (k,\ h_d^k(n))$, where k is an identifier and $h_d^k(n)$ is the number of hops

- **Ordering on Labeled Path Lengths:** Node a is ordered along path $P_d^k(n)$ with respect to its next hop along the path if

$$\text{L}: \ \ell_d^b(m) \prec_\ell \ell_d^a(n) \ \equiv [h_d^b(m) < h_d^a(n)] \vee [(h_d^b(m) = h_d^a(n)) \wedge (b < a)]$$

i.e., the route from b is strictly shorter, or has the same hop count and b is lexicographically smaller than a.

Loop-Free Routing with Private Policies

- Relation $\prec_\ell$ is such that, for any three different ℓ_d^a, ℓ_d^b, and ℓ_d^k , we have **Total Ordering of path labels, meaning:**

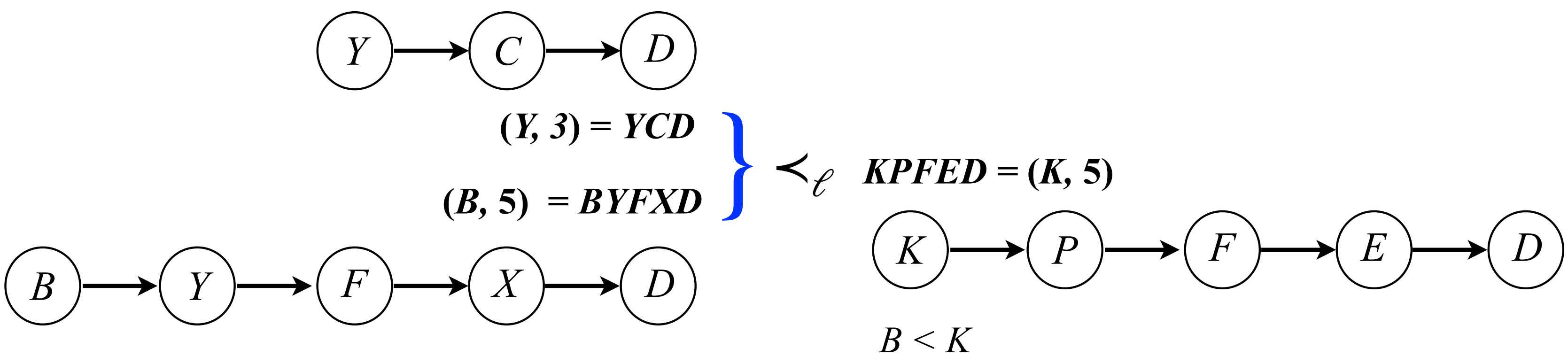
  - **Irreflexivity:** $\ell_d^a \not\prec_\ell \ell_d^a$
  - **Transitivity:**
    $[(\ell_d^a(i) \prec_\ell \ell_d^b(j)) \wedge (\ell_d^b(j) \prec_\ell \ell_d^c(k))] \rightarrow (\ell_d^a(i) \prec_\ell \ell_d^c(k))$
  - **Totality:** $(\ell_d^a(i) \prec_\ell \ell_d^b(j)) \vee (\ell_d^b(j) \prec_\ell \ell_d^a(i))$

- These properties imply **TOTAL ORDERING** as with integers

Convergence without Optimality

- **Definition of label ordering:**

$$\text{L}: \ \ell_d^b(m) \prec_\ell \ell_d^a(n) \ \equiv [h_d^b(m) < h_d^a(n)] \vee [(h_d^b(m) = h_d^a(n)) \wedge (b < a)]$$



- **Label ordering along loop-free paths:** Node k is ordered along path $P_d^k(n) = kP_d^q(m)$ with respect to its next hop q based on path labels if

$$\text{L}: \ell_d^q(m) \prec_\ell \ell_d^k(n)$$

Convergence without Optimality

Theorem 1: A routing protocol is guaranteed to be loop-free if it ensures that L is satisfied at every instant by every node (router or AS) for any destination.

Proof: By contradiction. Assume w.l.o.g. that a node has a single route to *d*.

Assume that loop $P = \{n(1) \rightarrow n(2) \rightarrow \ldots \rightarrow n(h-1) \rightarrow n(1)\}$ is created at some point.

Because **L** is true, we have $\ell_d^{n(1)} \prec_\ell \ell_d^{n(h-1)}$ and $\ell_d^{n(i)} \prec_\ell \ell_d^{n(i-1)}$ for $1 \leq i \leq h-1$

This is a contradiction, because it implies that $\ell_d^{n(i)} \prec_\ell \ell_d^{n(i)}$, which is false because of the irreflexivity property of $\prec_\ell$.

QED

BGP-ELF

- Path vectors are replaced with labeled path lengths
- Updates, queries and replies are used to ensure that ***L*** is always true
- Nodes send updates with their ***reported labels*** (current labeled path lengths) as long as ***L*** is true.
- If ***L*** is false, a node sends a query stating a ***requested label*** equal to the value of its own labeled path length prior to the input event that prompted the query
- Nodes that cannot satisfy the requested label in a query must forward it
- A node whose next hop has a reported label that is smaller than the ***requested label*** sends a reply
- Reply states a ***reference label*** sent back to the origin of the query
- Nodes communicate the largest labeled path lengths that satisfy ***L***
- Node that sent a query accepts the first valid response with

  reference label < requested label

BGP-ELF

- **Ordered Import Transformation:**
  - Let $\ell_d^k[r]$ be the path label of the path reported by node k for destination d
  - Accept route reported by q if $\mathrm{BE}_i : \ell_d^q[r] \prec_\ell \ell_d^k[r]$.
  - i.e., **accept routes only if they are totally ordered**
  - If route is accepted then $\ell_{dq}^k \leftarrow \ell_d^q[r]$ else $\ell_{dq}^k \leftarrow \ell_\infty$
- **Multi-path Local Preference Function:**
  - Use the similar local preference functions of BGP implementations
  - Compute the largest label $\ell_{d\,max}^k$ that can be accepted for each destination d
- **Ordered Export Transformation:**
  - Compute the reported route for destination d according to
  - $\mathrm{BE}_e : \ell_d^k[r] = \left(k, \quad 1 + h_{d\,max}^k \right)$
  - Update all or only some of neighbor routers in other ASes with value of $\ell_d^k[r]$

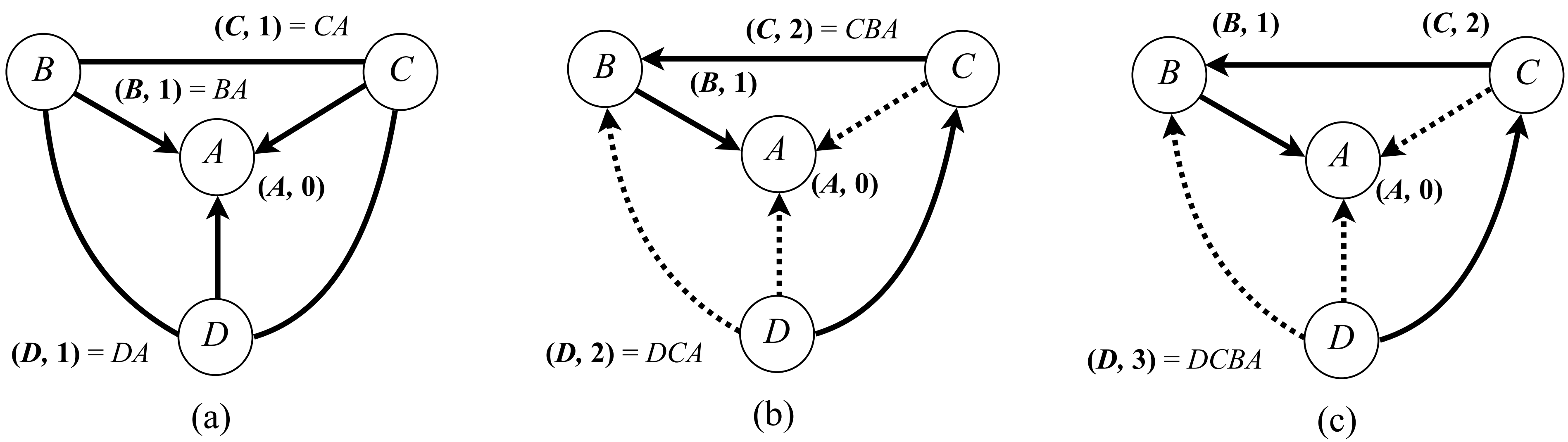
BGP-ELF: Filtering Using AS Classes

- Set of $|C|$ AS classes are defined
- Each AS is a member of one or multiple AS classes
- Class vector of size $|C|$ states the classes to which an AS belongs (1 means "belongs")
- Node k stores a vector u_d^k of unwanted AS classes (1 states "unwanted")
- The reported label from k to d has a label and a class vector v_d^k with a bit set in a position if any AS in the route belongs to class (1 means "belongs")
- Extended import transformation:

$$\mathrm{BE}_i : \left(\ell_d^q[r] \prec_\ell \ell_d^k[r] \right) \wedge \left(u_d^k \cap \nu_{dq}^k = \bar{0} \right)$$

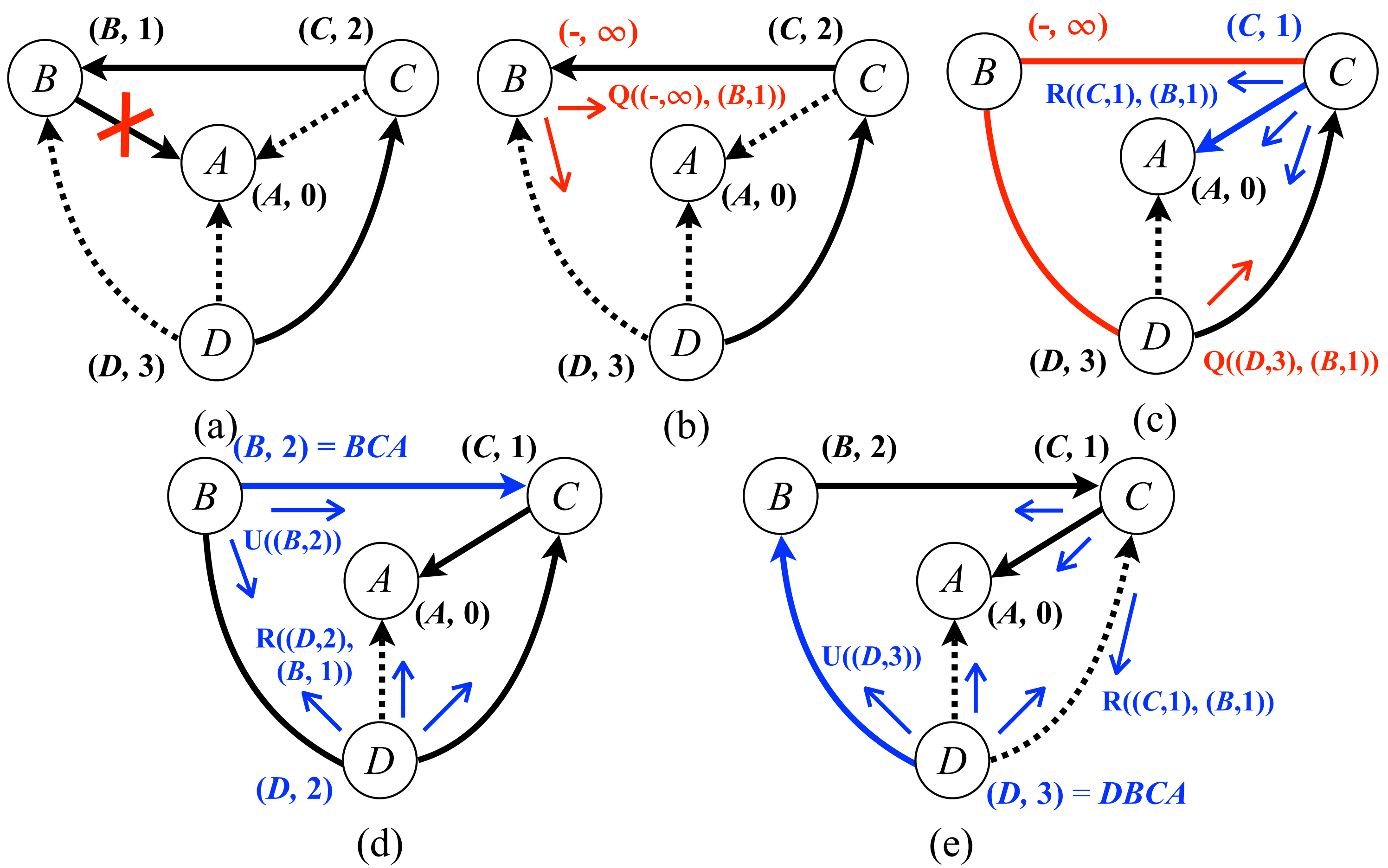
i.e., route is totally ordered and does not traverse any unwanted AS

Example: BGP-ELF in BAD-GADGET



- Nodes announce routes with largest labeled path lengths
- Multiple routes can be used locally
- No loops are created
- Preferences are constrained like shortest-path selections are constrained as in some intradomain routing protocols

Example: Failure in BAD-GADGET



Queries prevent loops from being created

Nodes accept first valid response

Multiple routes are used locally

Conclusions & Next Steps

- Appendix shows that BGP-ELF is loop-free and converges to stable routes within a finite time
- First approach based on BGP that eliminates path vectors
- BGP-ELF signaling is more efficient than DUAL (only loop-free Internet routing protocol used in practice)
- **Next:**
  - **Extend BGP-ELF to account for ASes that are not fully meshed**
  - **Explore the integration of policies and performance metrics in BGP-ELG**

Thank you!