Stable Internet Routing Without Global Coordination

Lixin Gao, & Jennifer Rexford,

Computer Networks
Dr. Jorge A. Cobb

UTD

Problem

- Too much overhead to maintain path histories (dynamic execution solution)
- Difficult to gather all policies and check for cycles in dispute graphs
- Can we select some practical guidelines?
  - If the guidelines are followed, the system is stable (always reaches a steady state)
- Their approach: hierarchies
  - You avoid circular conflicts by introducing hierarchies of nodes
  - Policies are based on the hierarchy

Background

- We need some background before presenting the technique
- In particular, the difference between export, import, and route selection policies.

BGP Import, Export, Path Selection

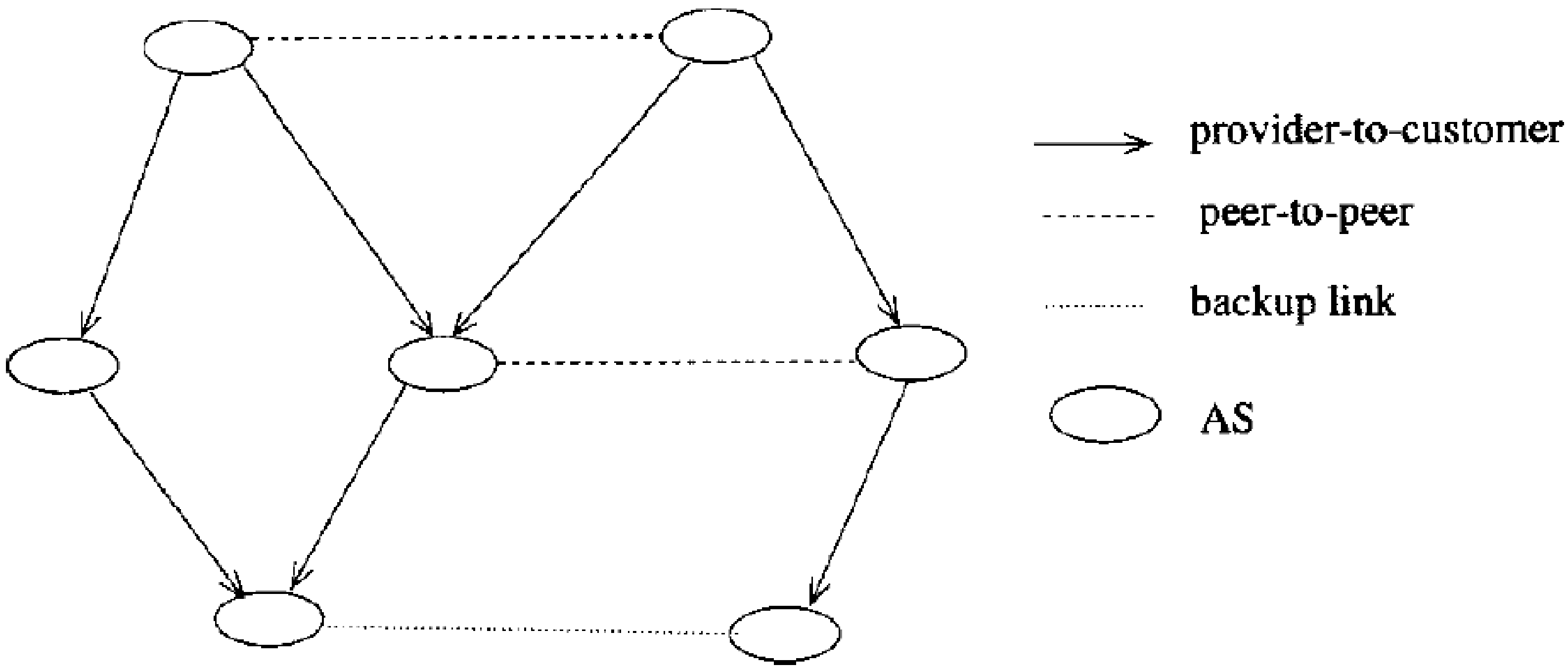
- There are three types of policies:
  - Import Policy: of the paths offered by my neighbors, which ones will I allow? (i.e. "import" the path)
  - Path Selection: given the paths offered by my neighbors which satisfy my import policy, which one do I like the best?
  - Export Policy: given my current path, will I tell my neighbor of this path or tell my neighbor that I have no path? (i.e., "export" the path)
- Note: in the Griffin paper, import/export policy was not explicit, only "list of allowed paths" was given.
  - A path is allowed in an "SPP instance" only if it exists in the export policy of my neighbour and in my import policy

UTD

Neighbor Relationships

- For each neighbour B of AS A, we have one of the following three relationships:
  - B is the service provider of A: in this case, B is the “access point” (or perhaps one of several) over which A accesses the rest of the Internet
  - B is a customer of A: in this case, A is the service provider or B
  - B is a peer of A: neither is a provider or customer of the other. They are simply “peers” who help each other.

AS relationship graph



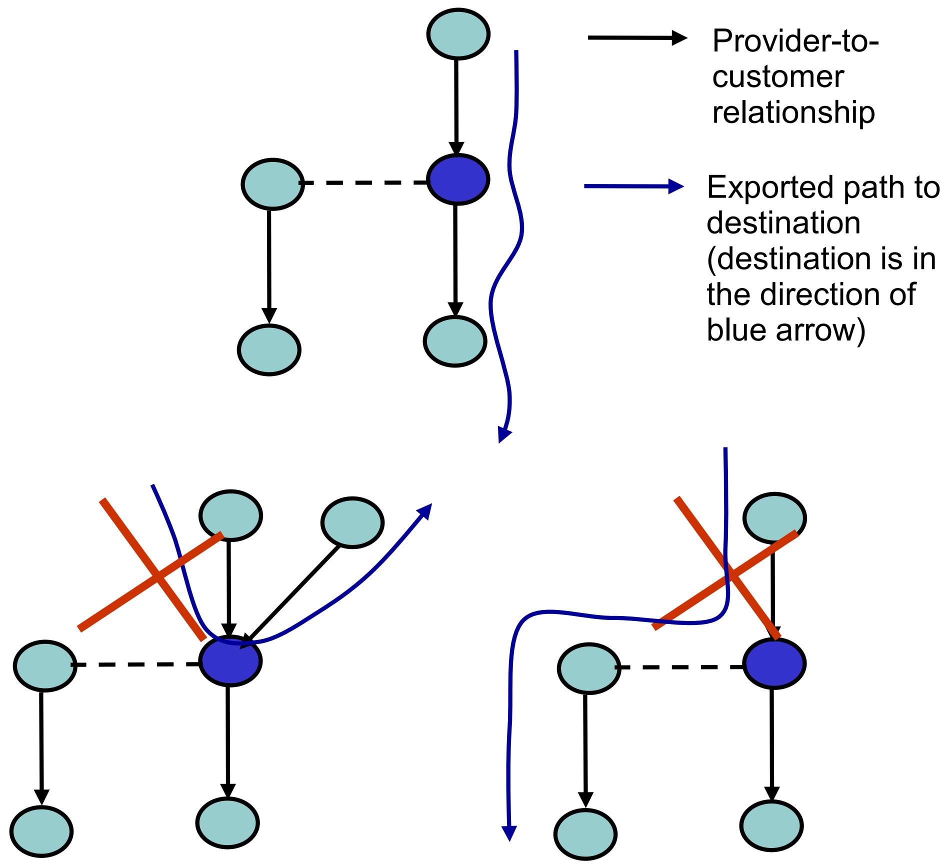
Provider-to-customer edges must form a directed acyclic graph (DAG)

Export policy restrictions

- Convergence is assured (almost) by restricting the export policies
  - i.e., you can't tell everything to everyone

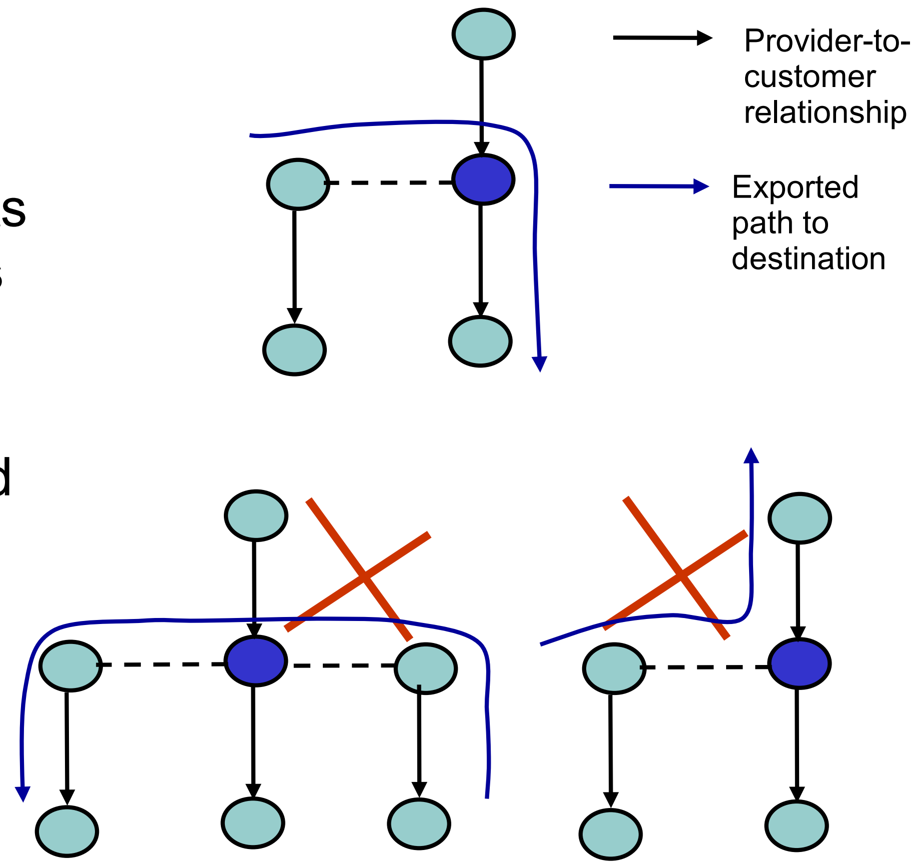
Export policy restrictions - providers

- **Exporting to a provider:** In exchanging routing information with a provider:
  - An AS can **only** export its networks and **the routes of its customers.**
  - However, it can not export routes learned from other providers or peers.
  - That is, an AS does *not* provide transit services for its provider.



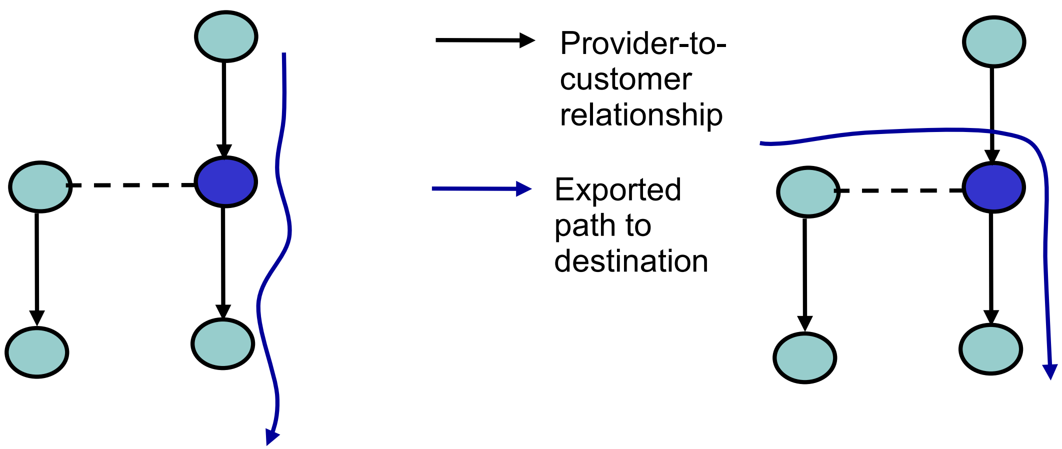
Export policy restrictions - peers

- **Exporting to a peer:** In exchanging routing information with a peer:
  - An AS can **only** export its networks and the **routes of its customers**
  - However, it can not export the routes learned from other providers or peers.
  - That is, an AS does *not* provide transit services for its peers.



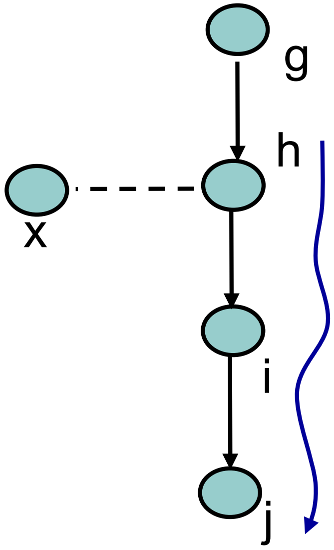
In Summary (previous rules)

- A node can export to its peer and to its provider only paths that it has learned from its customers



Exporting to peers or providers

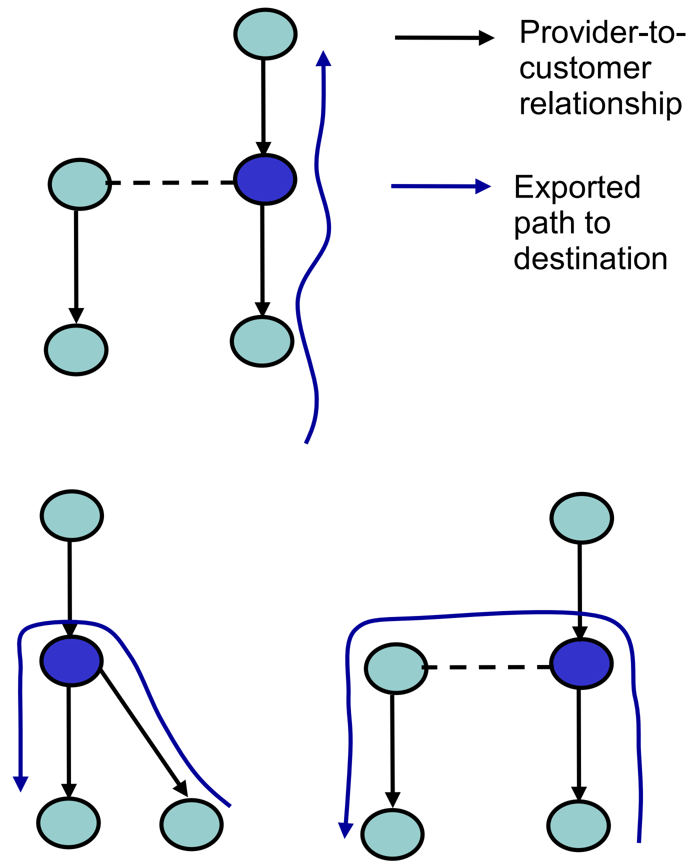
- I say that a route r is a "cust_only" route if for every pair of adjacent nodes (A,B) in the route, A is a service provider of B
- Lemma 0 : a node can only export a "cust_only" path to its peers and service providers



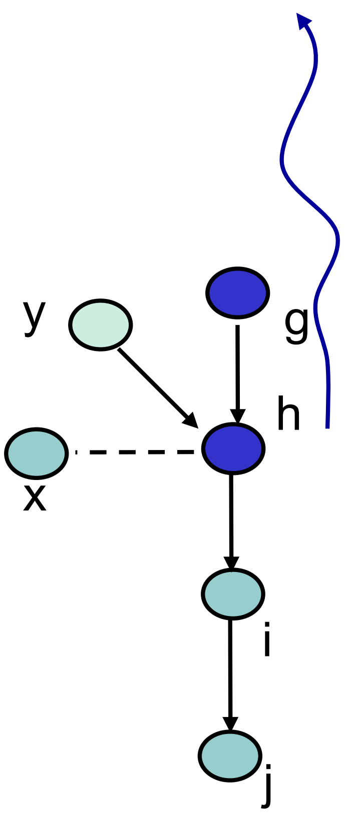
- What can h export to its provider g or its peer x?
  - only what it receives from a customer (i.e. i)
- What can i export to its provider h?
  - only what it receives from its customer (i.e. j)
- What can j export to its provider i?
  - only what it receives from its customer …

Export policy restrictions - customers

- **Exporting to a customer:** In exchanging routing information with a customer:
  - An AS **can export everything**: its customer routes, as well as routes learned from its providers and peers.
  - That is, an AS *does* provide transit services for its customers.



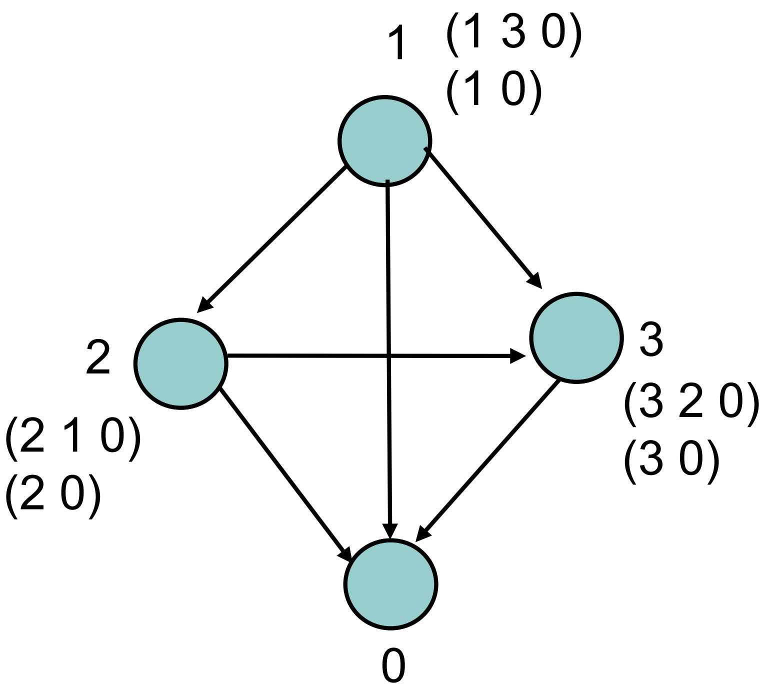
Exporting to customers



- **Lemma 1:** A path exported by a provider g to a customer h can only be subsequently exported by providers to their customers (exported via provider → customer edges).
  - If g exports the path to h, h cannot export it to y nor to x
    - h can export to y or x only a path via a customer
  - You can repeat the argument between h and i, etc.

Still not good enough

- The following system, even though it satisfies the export policies, is not safe
- Every node starts directly to zero
- Then choose the neighbour clockwise
- Have we seen this one before?



Additional Policy Requirements

Consider an AS *a*, and this AS has two routes it can choose from, r_1 and r_2.

Must prefer customer paths over those of peers and providers

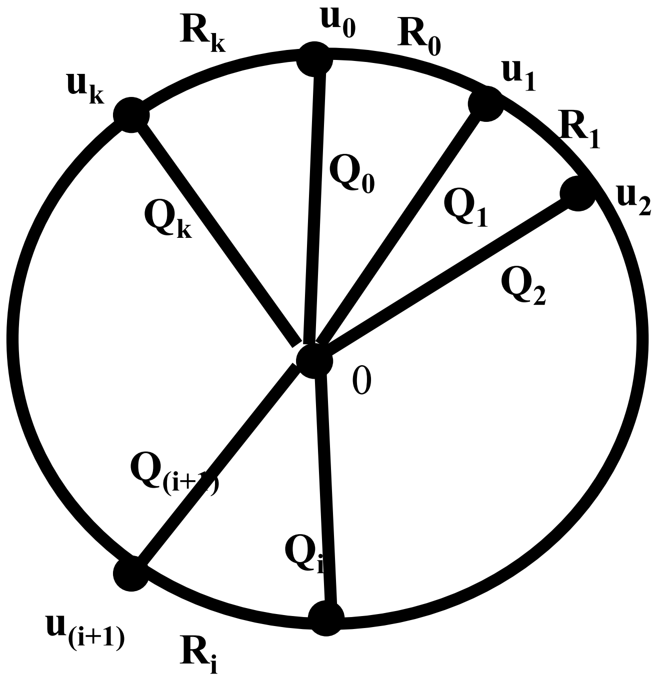
Guideline A
if $((first(r_1.as\_path) \in customer(a))$ and $(first(r_2.as\_path) \in peer(a) \cup provider(a)))$ then $r_1.loc\_pref > r_2.loc\_pref$

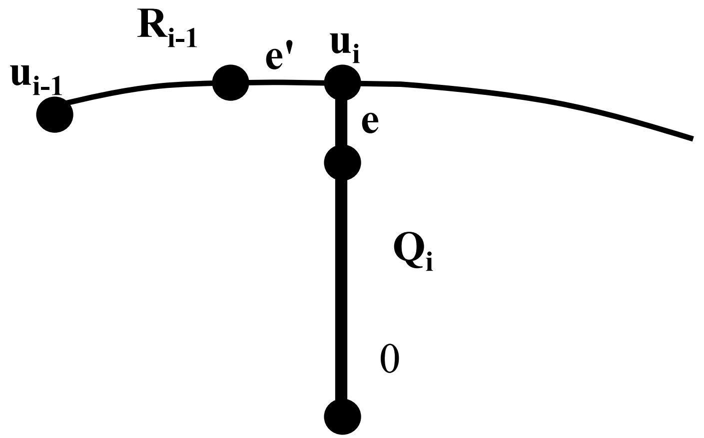
The end result

- **Theorem 1**: For a BGP system that has only customer–provider and peer-to-peer relationships (no backup links), if all ASms follow guideline A, and the provider-customer graph is acyclic, and the export policies are respected, then the BGP system is inherently safe (always converges)

- We will show that the system cannot have a dispute wheel.

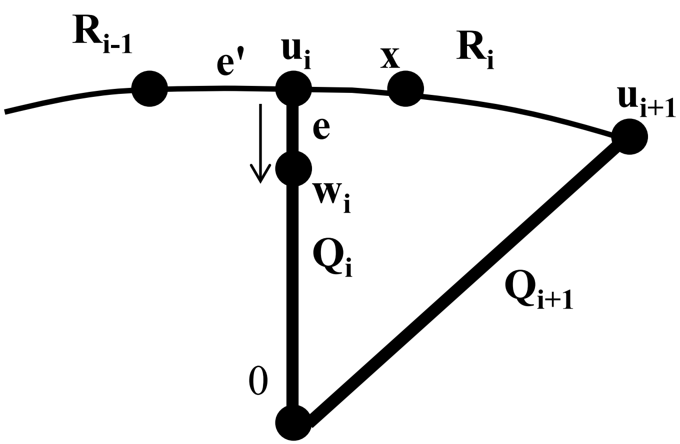
Proof





- Let e be the first edge in Q_i , and e', be the last edge in R_{i-1}.

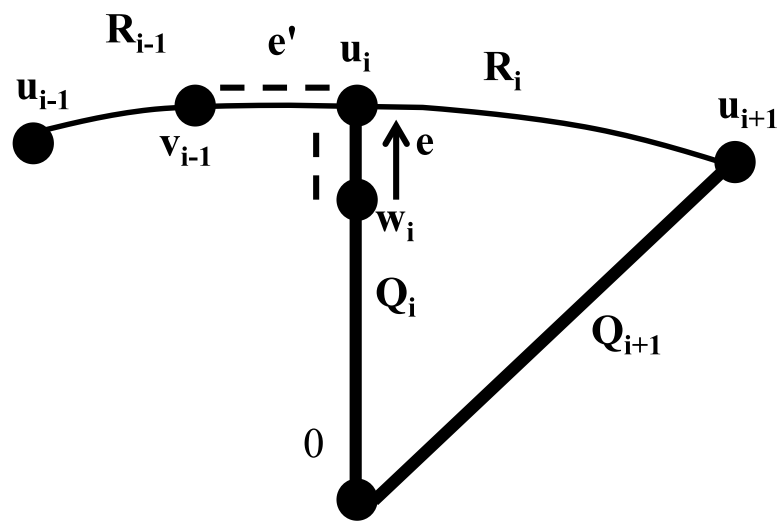
Proof (continued)



- Assume e is provider-customer (u_i is provider of w_i).
- Since u_i prefers customer paths, and from the wheel $\lambda(Q_i) < \lambda(R_i Q_{i+1})$
  - This implies the first hop in R_i is via a customer x.
  - From lemma 0, the entire path $R_i Q_{i+1}$ is a cust_only path.
  - In particular, first edge in Q_{i+1} is a provider-customer edge
- Repeat the argument at u_{i+1}, and you have that all edges on the rim of the wheel are provider-customer edges (actually, all edges in the entire wheel ☺)
- This is not possible (provider-customer graph is acyclic)

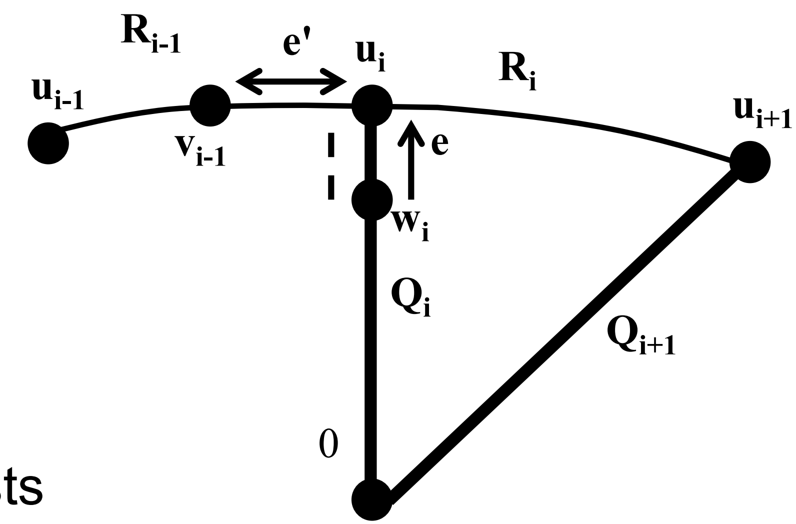
Proof (continued, remaining cases)

- Recall that e CANNOT be provider-customer edge
- Consider now e'
- If e' is peer-to-peer,
  - Because w_i is not a customer of of u_i (e is not provider-customer), it is illegal for u_i to export Q_i to its peer v_{i-1}.



Proof (continued, remaining cases)

- If e' is provider-customer edge (v_{i-1} provider, u_i customer)
  - Same thing.
- If e' is a customer-provider edge (v_{i-1} customer, u_i provider)
  - v_{i-1} is customer of provider u_i.
  - From Lemma 1, all of R_{i-1} consists of customer-provider edges
  - We can repeat this argument at all nodes, so all R_i paths are customer-provider, which violates the acyclic provider graph.



Remarks

- If links and or nodes are added/deleted, or if the policies change, as long as they satisfy the given requirements the system will remain stable.

Multiple BGP speakers

- How does a BGP speaker select a route when there are multiple BGP speakers in its AS?

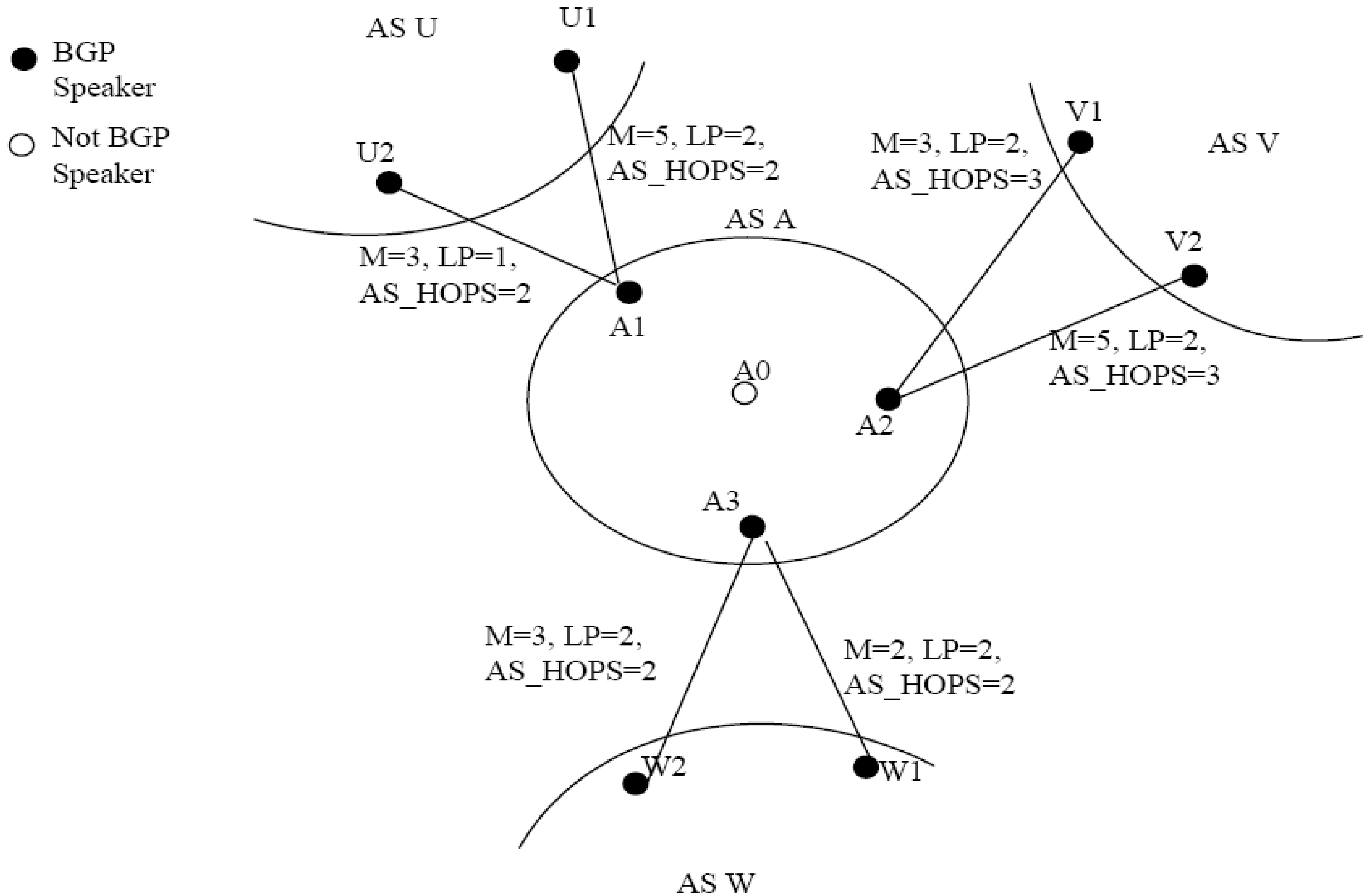
- Can the paths be different for BGP speakers in the same AS?

Route fields

- Assume an AS A receives a route r from its neighbor B
- Route r has several fields
  - r.as_path: the list of ASms from B to the destination
  - r.MED: if multiple links between A & B, MED determines which one B prefers that A uses (multi-exit discriminator)
  - r.community: you can organize ASms into "communities", and mentioned the community to which this path belongs (irrelevant for us)
  - r.local_pref: how desirable the path is (local preference), and is assigned by A.

Path Selection Policy

- If we have multiple BGP speakers per AS then, path selection is as follows:
  a) From the paths that satisfy the import policies
     - Choose those with the greatest local_pref (I assume local_pref is consistent among all routers in the same AS)
  b) Then, out of the remaining paths from above choose the ones with least # of AS hops to the destination
  c) Then, for each neighboring AS B,
     - Of the paths chosen above, if multiple paths from B (one per BGP border router), choose the one with lowest MED value
  d) Choose the path whose internal cost to reach the border router is least.
  e) Choose the path whose peer has the lowest IP address



Consider the collection of ASms above, with ASms: A, U, V, and W. To reach the destination, A must go through one of the other three ASms. Each BGP router in U, V, and W advertises to its neighbor in A a path to the destination. The attributes of each path are: MED value - the smaller the better local preference value - the larger the better, and number of hops to the destination.

(a) For each of the three BGP speakers in A (A1, A2, A3), show me what path each will follow, i.e., to which border link it will forward traffic addressed to the destination. Please show your intermediate work.

(b) For the non-BGP speaker A0, how will it know to whom it should forward messages to the destination?

Answer to previous slide

- Path via (A1, U2) is discarded due to its low local preference value of 1 (all other paths have a local pref value of 2)
- Next, paths via AS V are also discarded because # AS HOPS is larger than the other ASms
- We are left with 3 paths: (A1,U1), (A3,W2) and (A3,W1)
- (A3, W2) is discarded since its MED is greater than (A3,W1)
- We are left with two paths (A1,U1) and (A3,W1).
- Then, A1, A2, and A3 choose their path according to the internal cost to reach the border router
  - A1 chooses to go via U1, since reaching the border costs 0 (cost from A1 to A1 is zero, but from A1 to A3 is > 0)
  - Similarly, A3 chooses to go via W1
  - A2 chooses to go via A1 or A3 depending on which one is closer (in terms of local cost) to A2.

Note: no centralized behavior

- Each router executes independently on its own
  - I.e., it looks at what its neighbors are offering (both internal and external neighbors) and chooses the best path
  - It then advertises its new path to all its neighbors
- What I showed on the previous slide although it is the ultimate result, there are several intermediate steps since, again, each router acts independently.