Inferring Autonomous System Relationships in the Internet

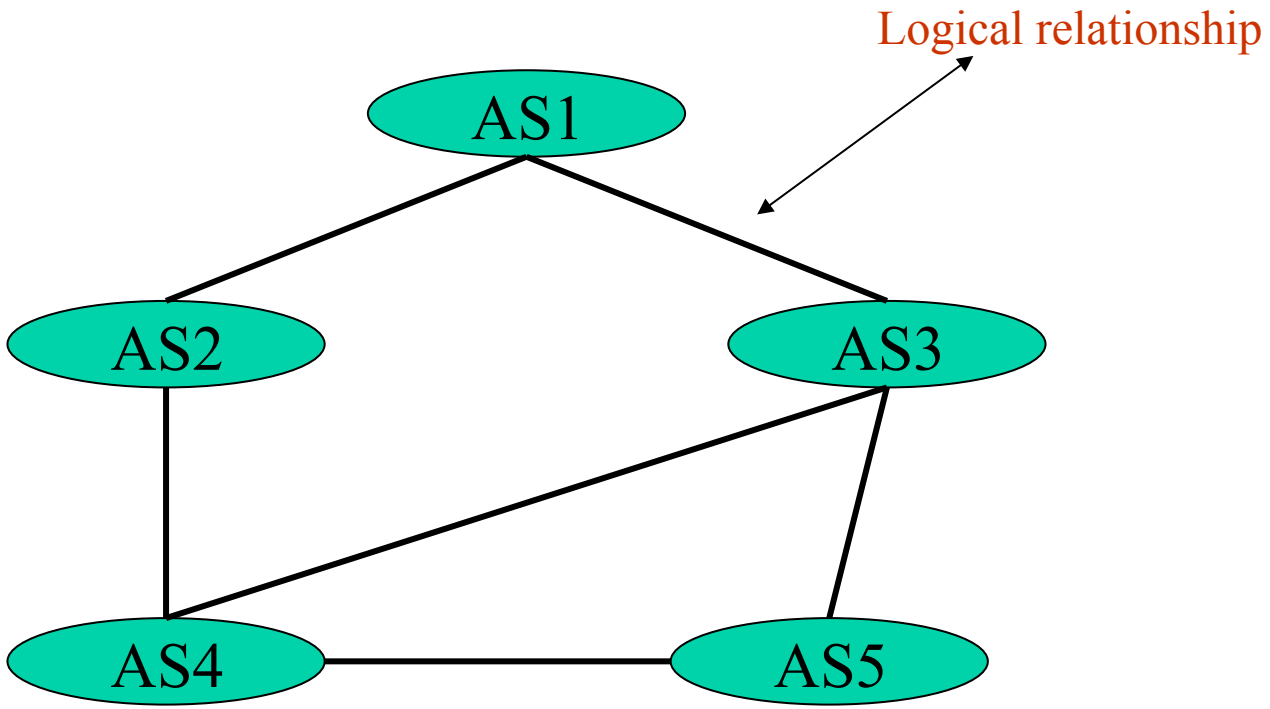
Lixin Gao

Motivation

- Routing policies are constrained by the contractual commercial agreements between administrative domains
- *For example*: AS sets policy so that it does not provide transit services between its providers
- Therefore connectivity does not imply reachability
- Policies not just connectivity influence the structural properties of the Internet

Background

Connectivity between ASes can be modeled using an AS graph **G = (V,E)**

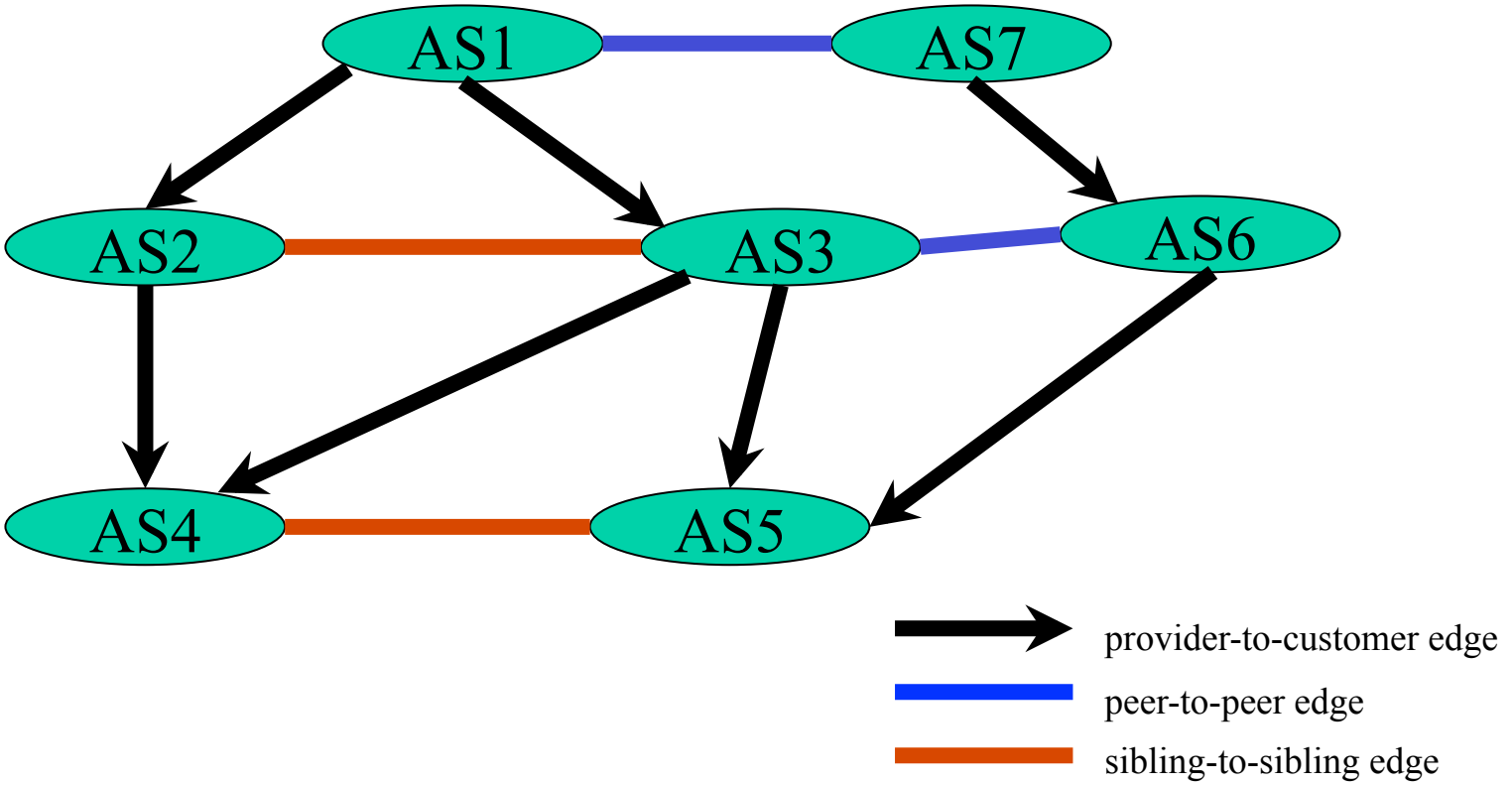


AS Relationships

- The commercial agreements between pairs of administrative domains can be classified into:
  - customer-provider relationship
  - peering relationship
  - mutual-transit relationship
- Classification for relationship of pairs of Autonomous Systems:
  - customer-to-provider relationship
  - provider-to-customer relationship
  - peer-to-peer relationship
  - sibling-to-sibling relationship

Annoted AS graph

Partially directed graph labeled with relationship

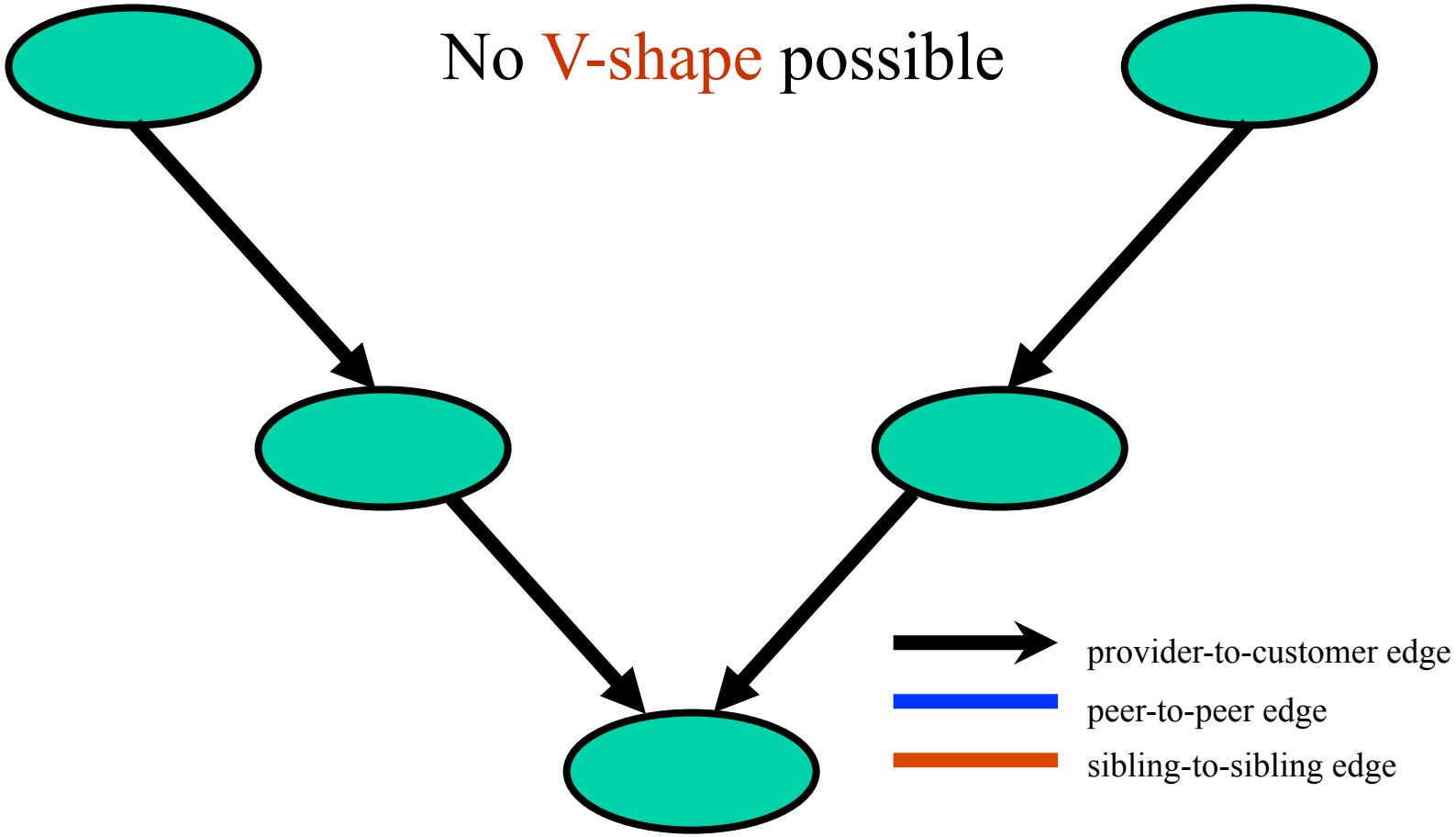


Rules governing BGP export policy

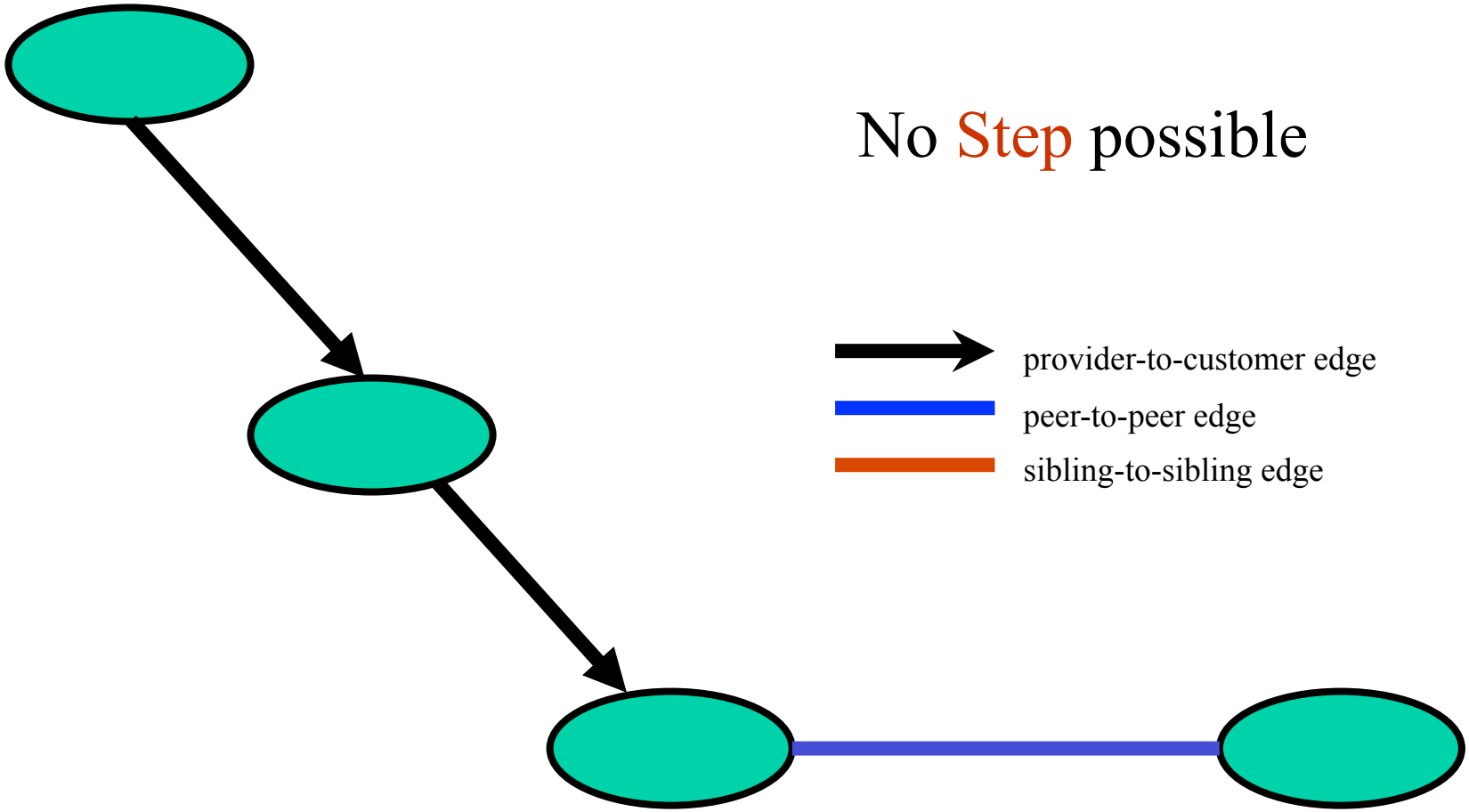
	Own Routes	Customer's Routes	Sibling's Route	Provider's Route	Peer's Route
Exporting to a Provider	×	×	×		
Exporting to a Customer	×	×	×	×	×
Exporting to a Peer	×	×	×		
Exporting to a Sibling	×	×	×	×	×

Selective export rules indicate that a BGP routing table entry should have a certain pattern

Valley-free property

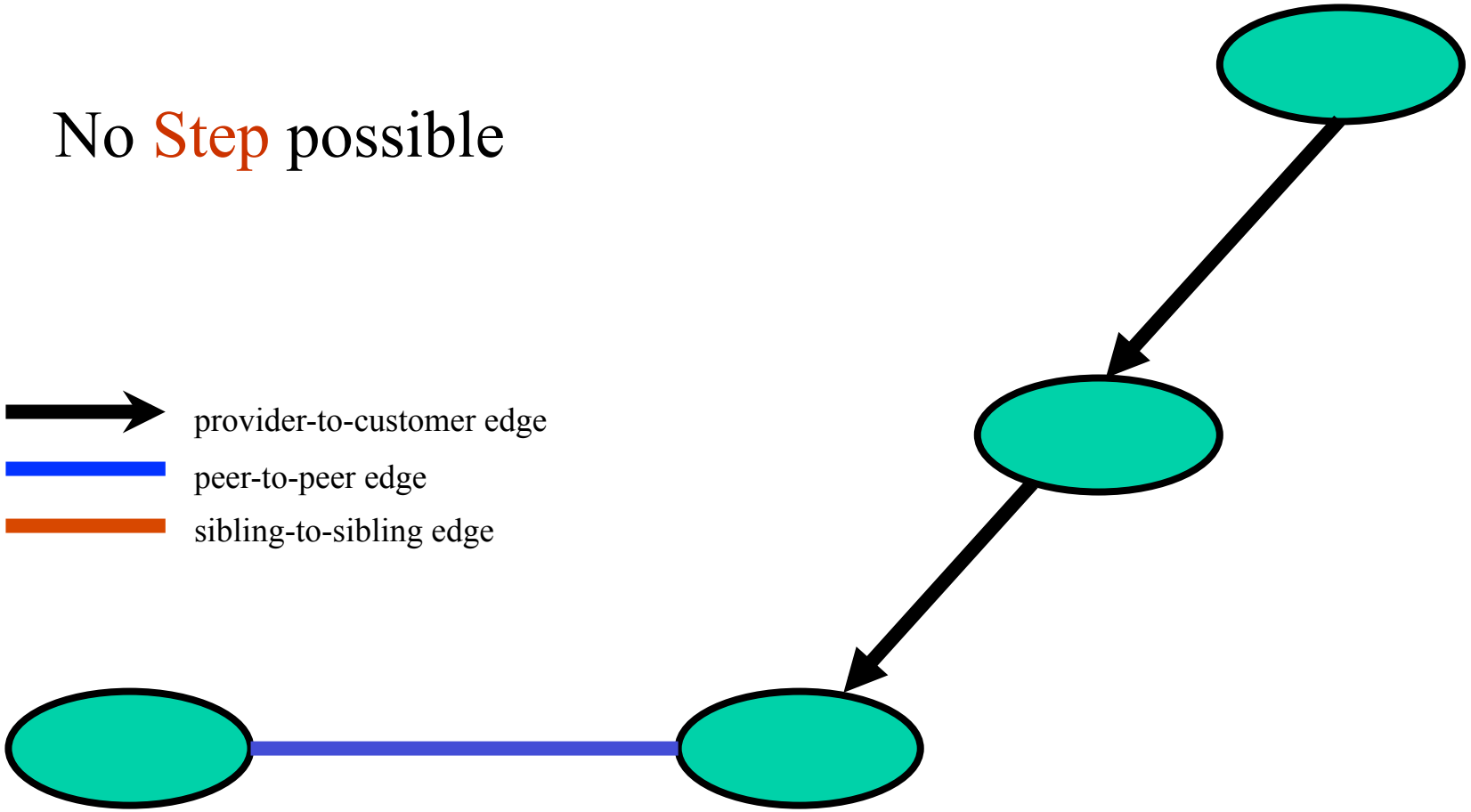


Valley-free property

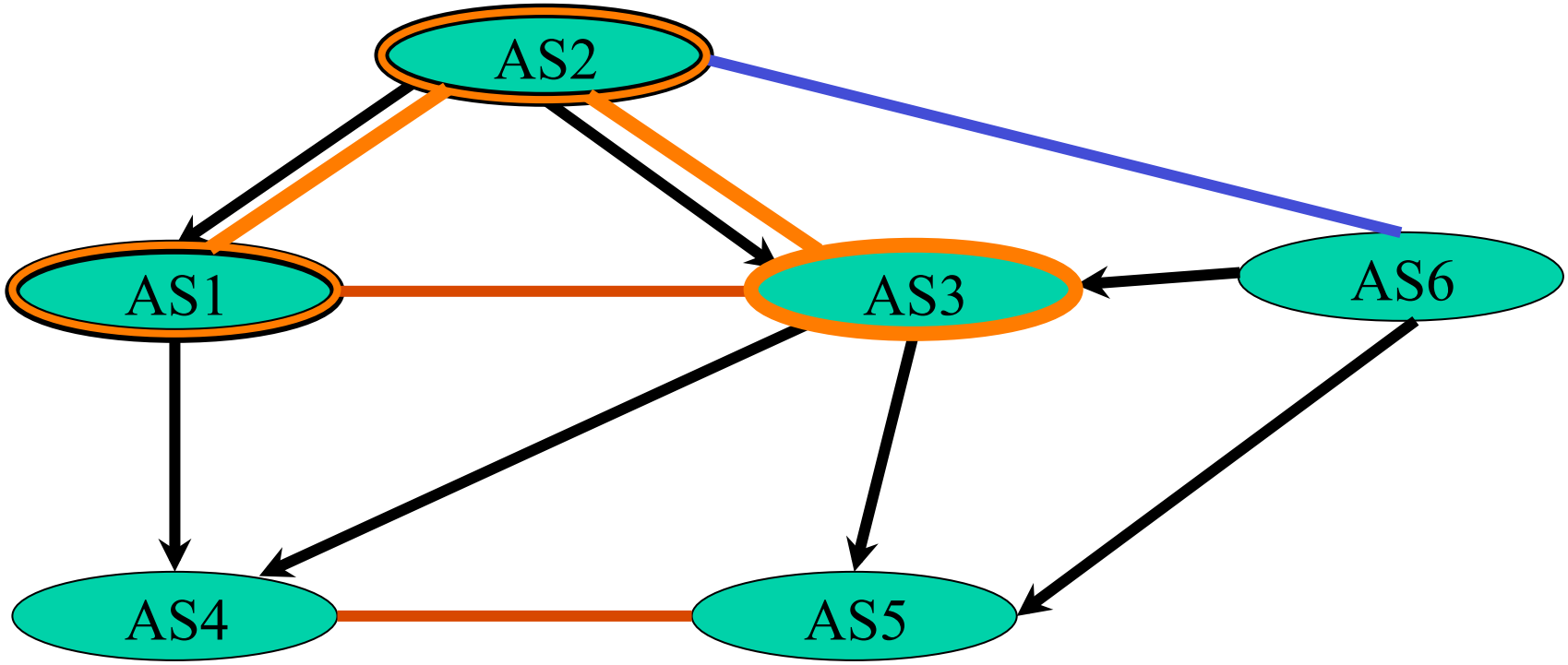


Valley-free property

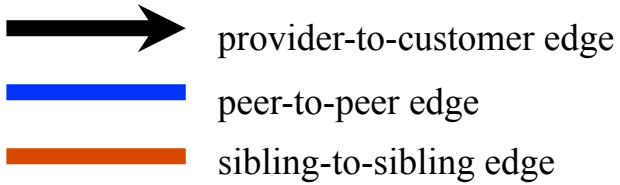
No Step possible



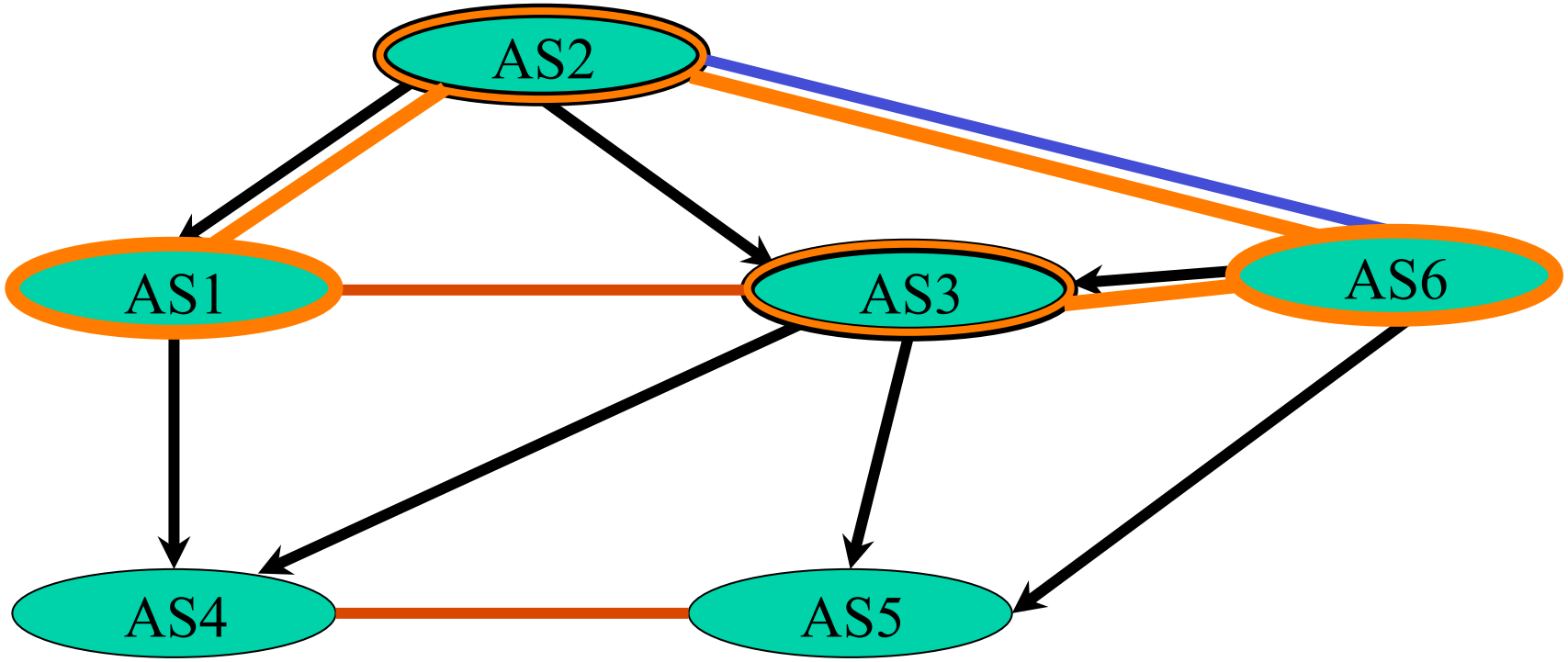
Valley-free property



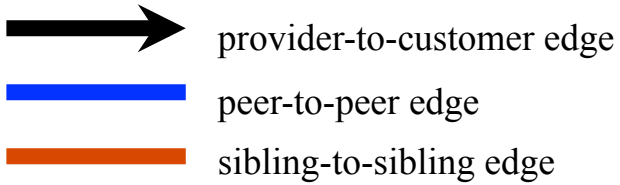
AS path (1,2,3) is valley-free



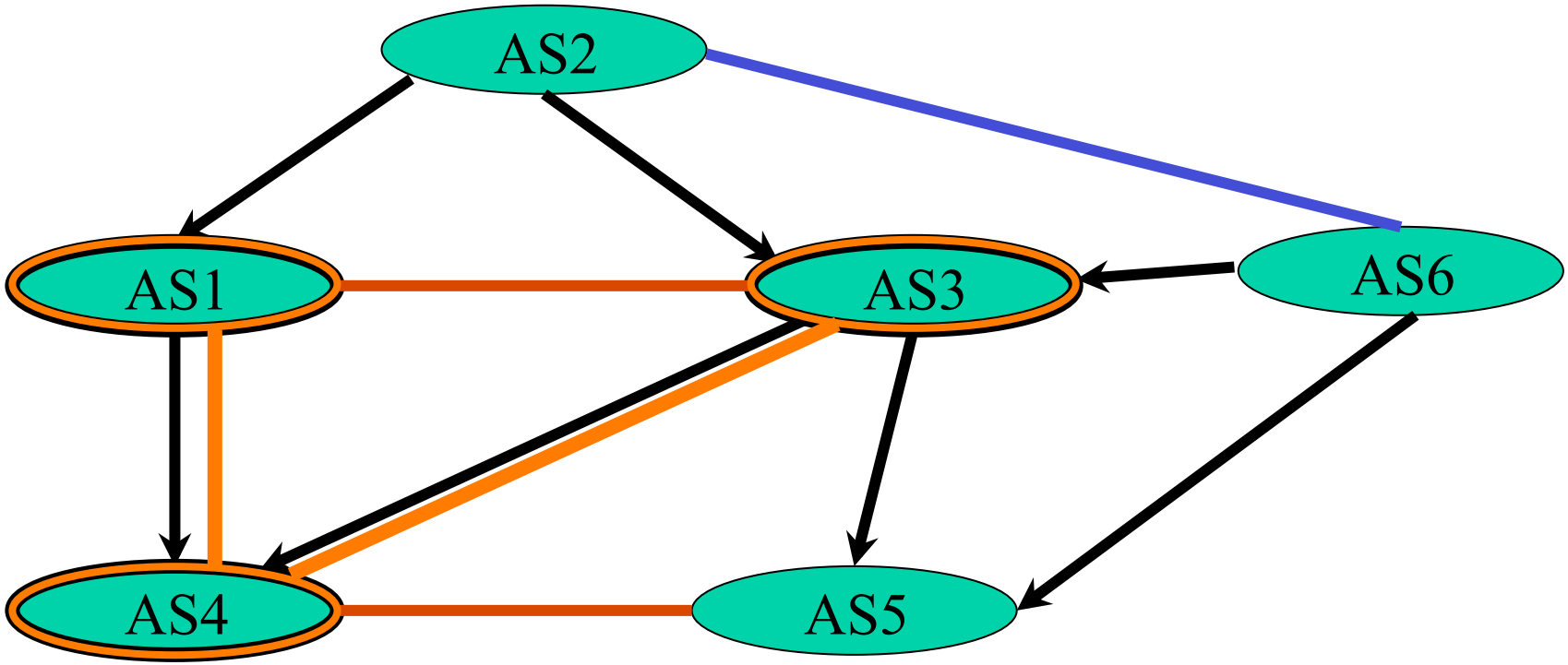
Valley-free property



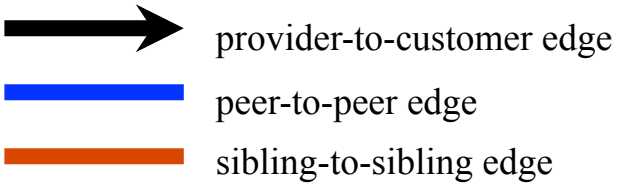
AS path (1,2,6,3) is valley-free



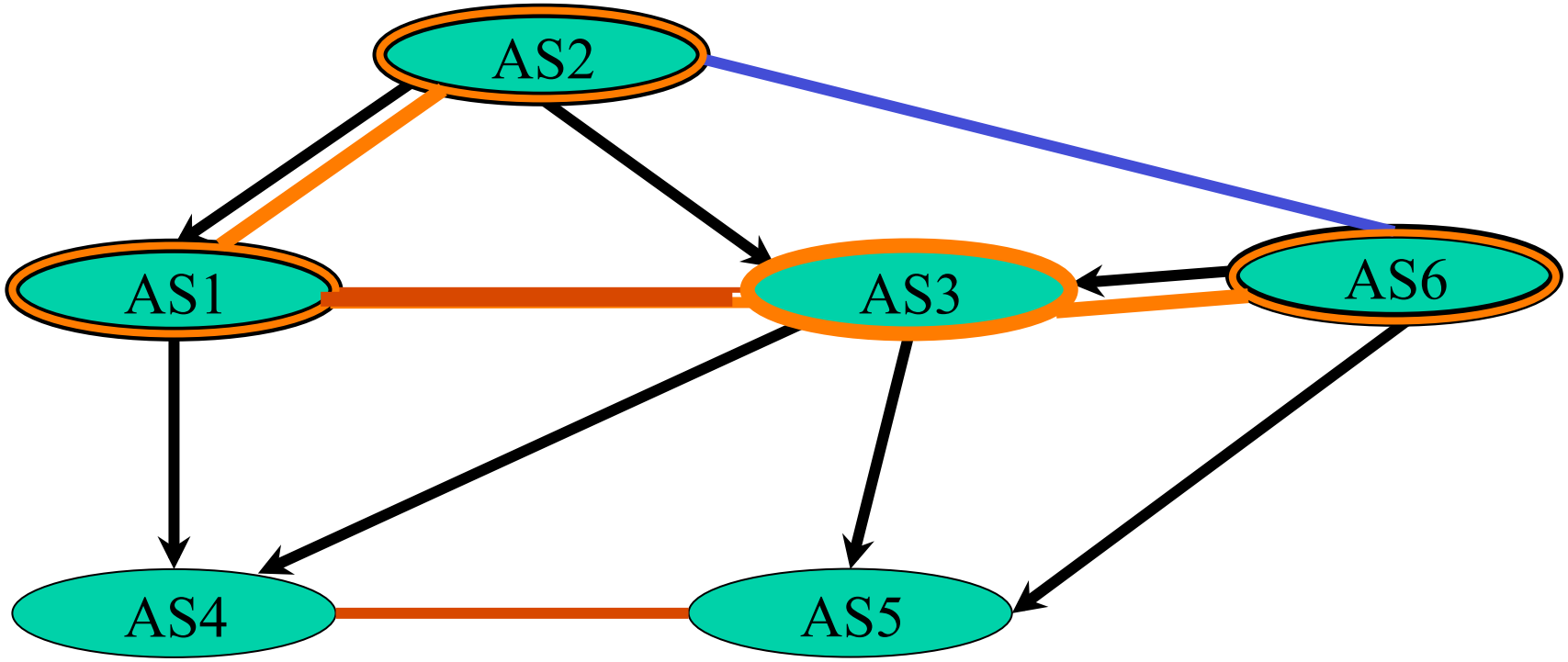
Valley-free property



AS path (1,4,3) is not valley-free



Valley-free property



AS path (2,1,3,6) is not valley-free

provider-to-customer edge

peer-to-peer edge

sibling-to-sibling edge

Valley-free property

- After traversing a provider-to-customer or peer-to-peer edge, the AS path can not traverse a customer-to-provider or peer-to-peer edge.

Routing Table Entry Patterns

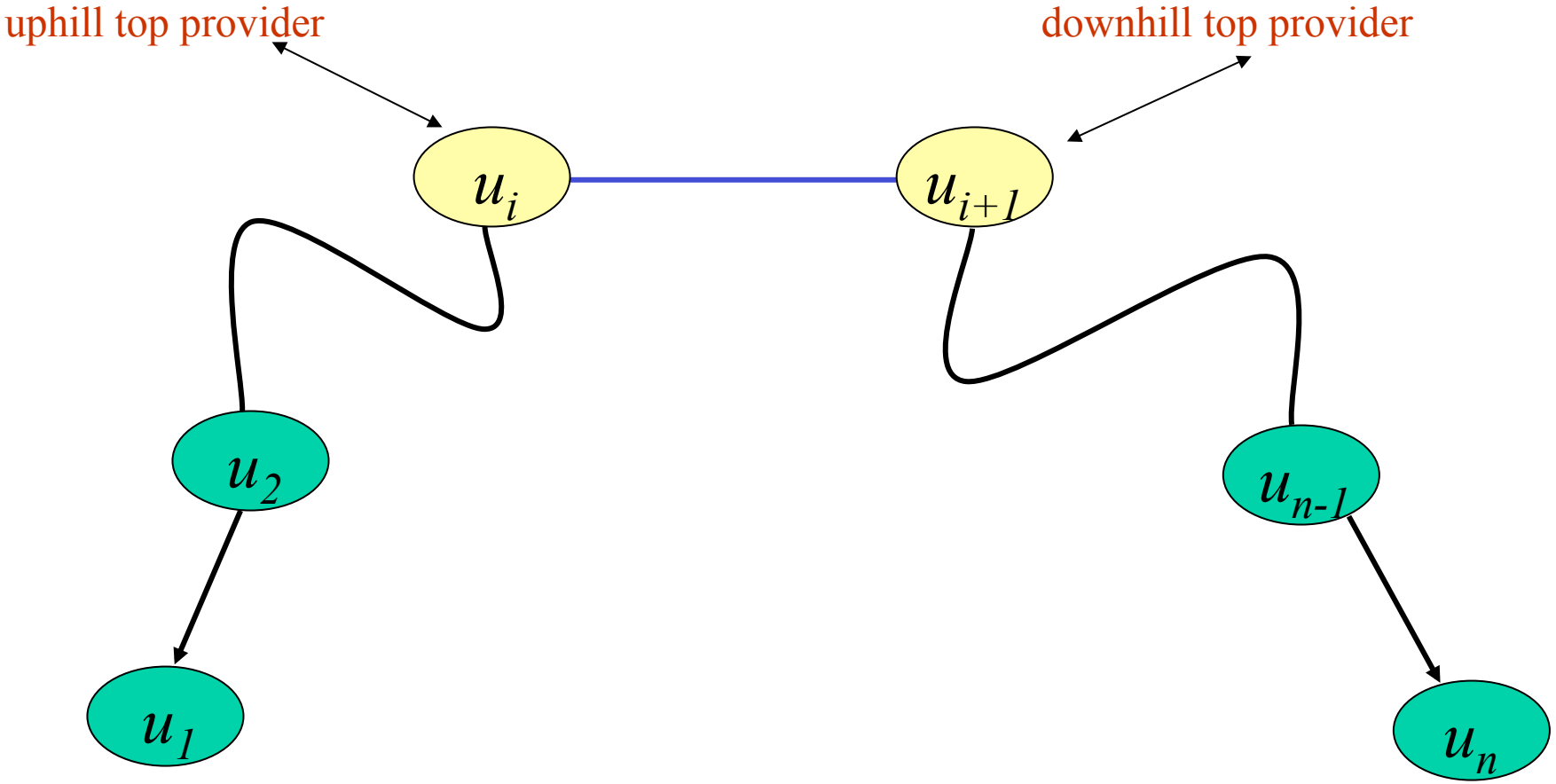
- **Downhill Path:** a sequence of edges that are either provider-to-customer or sibling-to-sibling

- **Uphill Path:** a sequence of edges that are either customer-to-provider or sibling-to-sibling

Routing Table Entry Patterns

- An AS path of a BGP routing table entry has one of the following patterns:
  - an uphill path followed by a peer-to-peer edge followed by a downhill path
  - an uphill path
  - a downhill path
  - an uphill path followed by a downhill path
  - an uphill path followed by a peer-to-peer edge
  - a peer-to-peer edge followed by a downhill path

Routing Table Entry Patterns



Heuristic Algorithms

- The Algorithm for inferring AS relationships is based on the fact that ASes set up their export policies according to the relationships and on the resulting patterns on BGP routing table entries
- It is also based on the intuition that a provider typically has a larger size than its customer and the size of an AS is typically proportional to its degree in the AS graph

Heuristic Algorithms

- top provider of an AS path is the AS that has the highest degree among all ASes in the path
- we can infer that consecutive AS pairs on the left of the top provider are customer-to-provider or sibling-to-sibling edges and on the right are provider-to-customer or sibling-to-sibling edges

Basic Algorithm

- **Input:** BGP routing table RT
- **Output:** Annotated AS graph G

- *Phase 1*: Compute the degree for each AS

- *Phase 2*: Parse AS path to initialize consecutive AS pair relationship

- *Phase 3*: Assign relationship to AS pairs

Refined Algorithm

- Top provider may not have the highest degree - possibility of incorrect inference of relationships
- let each routing table entry vote on the relationship of an AS pair
- if a sibling-to-sibling relationship is concluded by only one entry, we ignore it

Refined Algorithm

- If all routing table entries agree that an AS pair has a provider-to-customer (or customer-to-provider) relationship, then the AS has that relationship
- If only one routing table entry infers that an AS pair has a provider-to-customer (or customer-to-provider) relationship and more than one entry infer that an AS pair has a customer-to-provider (provider-to-customer) relationship, then the AS pair has a customer-to-provider (provider-to-customer) relationship
- For all other cases, the AS pair has a sibling-to-sibling relationship

Refined Algorithm

- **Input:** BGP routing table RT
- **Output:** Annotated AS graph G

- *Phase 1*: Compute the degree for each AS

- *Phase 2*: Count the number of routes that infers an AS pair as having a provider-to-customer or customer-to-provider relationship

- *Phase 3*: Assign relationship to AS pairs

Algorithm for Inferring Peer-to-Peer Relationships

- Peer-to-peer edge between top provider and one of its neighbors only
- If the top provider has sibling-to-sibling relationship with one of its neighbors, then it has a peer-to-peer relationship with the other neighbor
- We use the heuristic that peer-to-peer edge is between the top provider and its neighboring AS that has a higher degree because such edges are between ASes of comparable sizes
- We also use the heuristic that the degrees of two peers do not differ significantly - ASes having peer-to-peer relationship do not differ by more than **R** times

Final Algorithm

- **Input:** BGP routing table RT
- **Output:** Annotated AS graph

- *Phase 1*: Use either *Basic* or *Refined* algorithm to coarsely classify AS pairs into having provider-to-customer or sibling-to-sibling relationships
- *Phase 2*: Identify AS pairs that can not have a peer-to-peer relationship
- *Phase 3*: Assign peer-to-peer relationships from rest of the connected AS pairs as long as the pair degrees do not differ by more than **R** times

Inference Results

	TOTAL ROUTING ENTRIES	TOTAL EDGES	SIBLING-TO-SIBLING EDGES INFERRED BY BASIC (PERCENTAGE)	SIBLING-TO-SIBLING EDGES INFERRED BY REFINED (IGNORED ENTRIES)	PEER-TO-PEER EDGES INFERRED BY FINAL [R=] (PERCENTAGE)	PEER-TO-PEER EDGES INFERRED BY FINAL [R=60] (PERCENTAGE)
1999/9/27	968674	11288	149 (1.3%)	124 (25)	884 (7.8%)	733 (6.5%)
2000/1/2	936058	12571	186 (1.47%)	135 (51)	838 (6.7%)	668 (5.3%)
2000/3/9	1227596	13800	203 (1.47%)	157 (46)	857 (6.2%)	713 (5.7%)

Verification of Inferred Relationships by AT&T

OUR INFERENCE	AT&T INFORMATION	PERCENTAGE OF AS
Customer	Customer	99.8%
	Peer	0.2%
Peer	Peer	76.5%
	Customer	23.5%
Sibling	Sibling	20%
	Peer	60%
	Customer	20%
Nonexistent	Customer	95.6%
	Peer	4.4%

Comparing inference results from Basic and Final($R=\infty$) with AT&T internal information

Verification of Inferred Relationships by AT&T

OUR INFERENCE	AT&T INFORMATION	PERCENTAGE OF AS
Customer	Customer	99.5%
	Peer	0.5%
Peer	Peer	76.5%
	Customer	23.5%
Sibling	Sibling	25%
	Peer	50%
	Customer	25%
Nonexistent	Customer	95.6%
	Peer	4.4%

Comparing inference results from Refined and Final($R=\infty$) with AT&T internal information

Verification of Inferred Relationships by AT&T

OUR INFERENCE	AT&T INFORMATION	PERCENTAGE OF AS
Customer	Customer	99.8%
	Peer	0.2%
Peer	Peer	100%
Sibling	Sibling	20%
	Peer	60%
	Customer	20%
Nonexistent	Customer	95.6%
	Peer	4.4%

Comparing inference results from Basic and Final(R=60) with AT&T internal information