Towards a Logic for Wide-Area Internet Routing

Nick Feamster and Hari Balakrishnan
M.I.T. Computer Science and Artificial Intelligence Laboratory
{feamster,hari}@lcs.mit.edu

What is a Routing Logic?

Protocol designers and network operators need a way to describe and reason about protocol behavior.

Protocol	What is Correct?	Reasoning Framework
Authentication	Alice knows that she is talking to Bob, etc.	BAN Logic
Routing	**Routing Logic Properties**	**Routing Logic Rules**

- **Properties:** describe behavior
- **Rules:** reason about whether a certain property holds

What are the most important reasons not to accept this paper?

"It’s a theory paper."

This is really a paper on better system design.

Practice: How to abstract messy details?

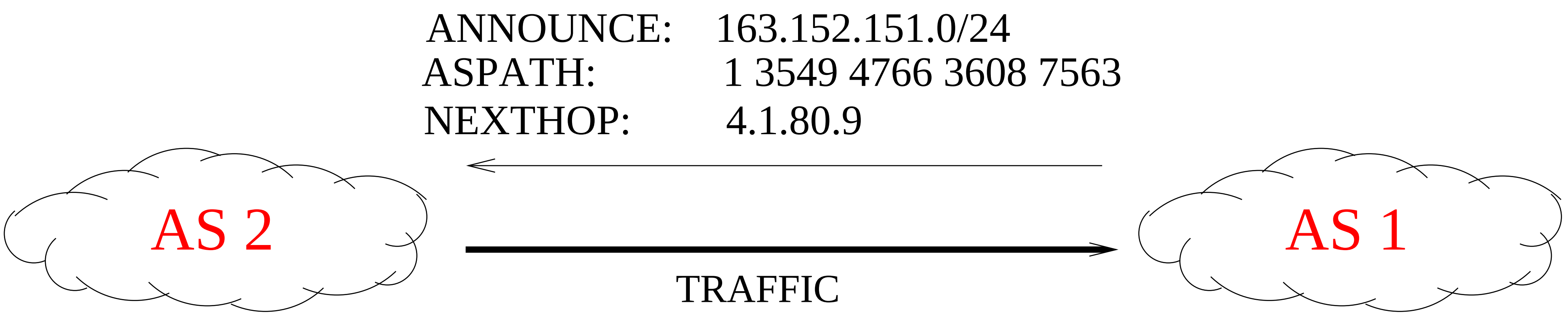
- Determine high-level properties that describe the behavior of a routing protocol.
- Define these properties in terms of rules (i.e., sufficient conditions) that are easier to reason about.
- Show examples where the logic assists reasoning.

(This work is not about automatic theorem proving, model checking, etc.)

Practical Uses for a Routing Logic

- **Reason about BGP’s behavior**
- **Verify that BGP configurations satisfy properties**
- Synthesize BGP configuration automatically
- Design protocol extensions that fix problems

BGP: Internet’s Wide-Area Routing Protocol



- Simple specification (only a 57-page RFC)
  - Compare with OSPF: ~ 250 pages
- *Complex dynamics: Beware!*

In case you've missed a few SIGCOMMs...

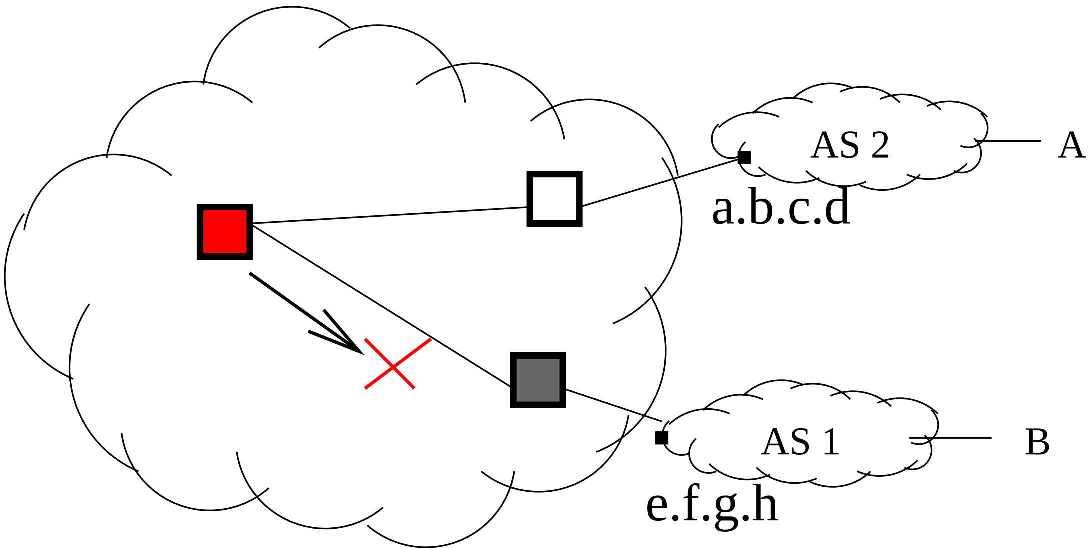
- BGP has serious problems
  - Easily misconfigured [Mahajan2002]
  - Forwarding loops [Dube1999]
  - Persistent route oscillation [Griffin1999, Varadhan2000]
  - Slow convergence/suppressed routes [Labovitz2001, Mao2002]
  - Useless routing messages [Labovitz1999, Wang2002]
  - Security weaknesses [Beard2002, Kent2000]

For every aberrant behavior,

a useful lesson and a point solution.

- Can we build on these lessons to:
  - express exactly what's wrong with BGP (or its configuration)
  - reason about proposed fixes, design modifications, etc.

BGP is Hard to Get Right



Routes from AS 1 have next-hop e.f.g.h
If e.f.g.h not injected into IGP, some routes from within AS will fail.

- Correctness is more than shortest paths!
  - Federated, asynchronous operation ("coopetition")
  - Coupling with IGP
  - BGP’s "correctness" is as much about configuration, policy, and competition as it is about pushing packets

How do we know if we’ve got it right? What is "right"?

How to define "correct" behavior?

- Does it advertise invalid routes?

Validity

- Does every valid path have a corresponding route?

Visibility

- Given a set of choices, will it converge to a unique, stable answer?

Safety

- Is that answer affected by the ordering of messages or the set of available routes?

Determinism

- Does the protocol expose information?

Information-flow control

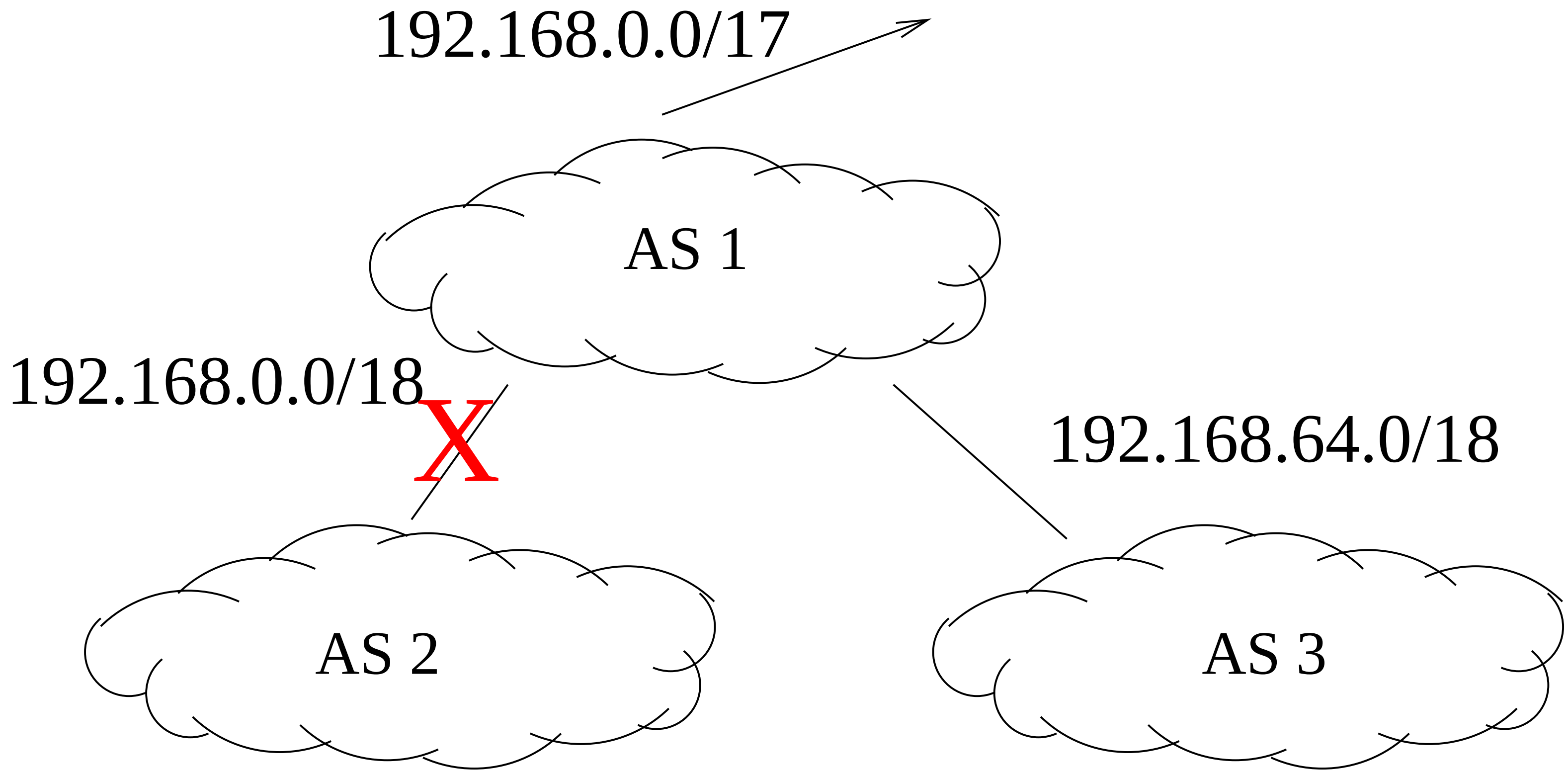
Rules: Sufficient Conditions for Each Property

- **Validity:** a route implies a corresponding valid path

A **puts route to dest for** B

- *Reachability:* A **can reach dest via route**
- *Policy conformance:* A **carries traffic to dest for** B
- *Progress:* `route.next-hop` **makes progress along route to dest**

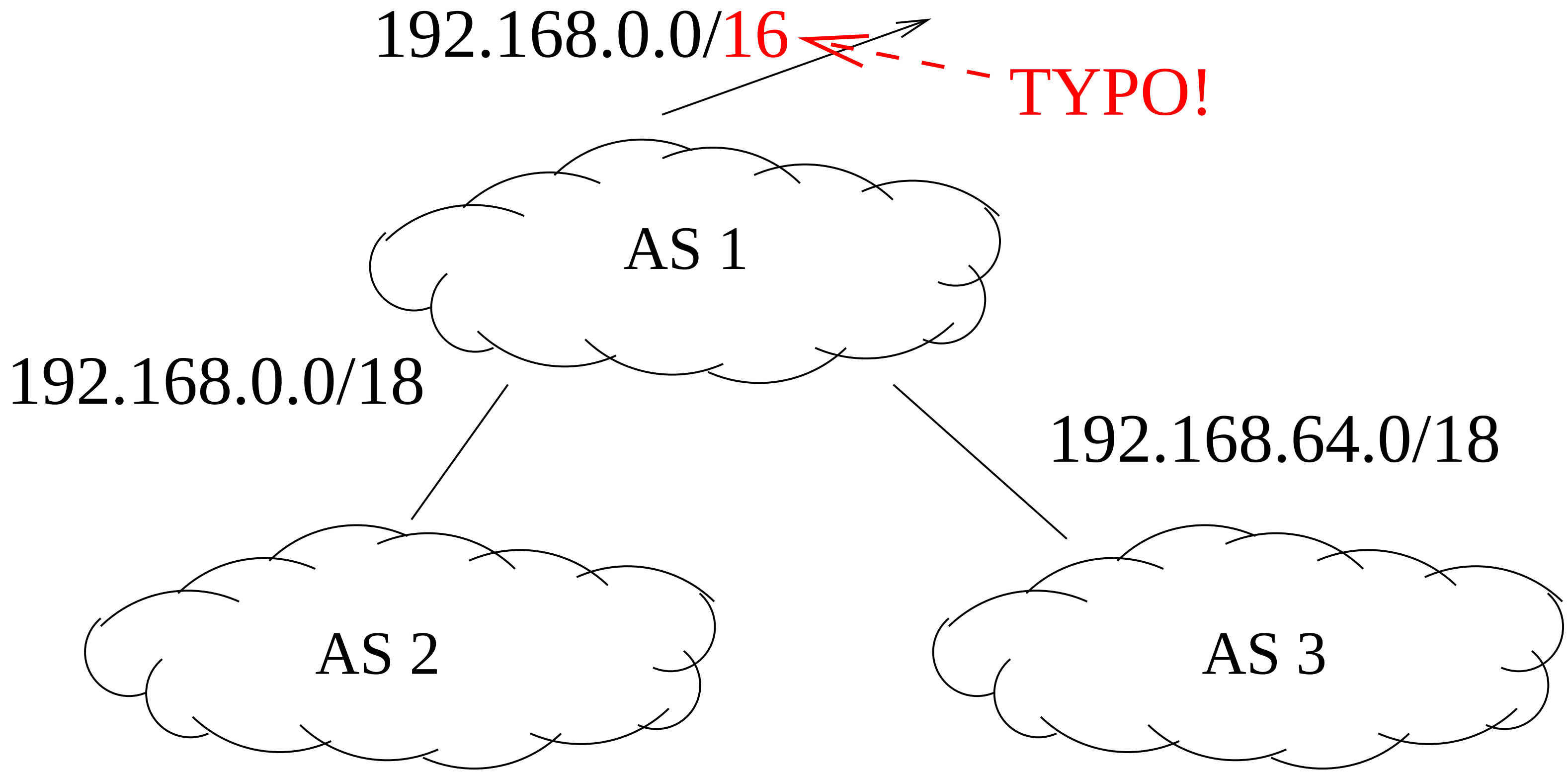
How Aggregation Affects Validity



Aggregated prefix does not accurately reflect **reachability** *of destination.*

(Operator might not care.)

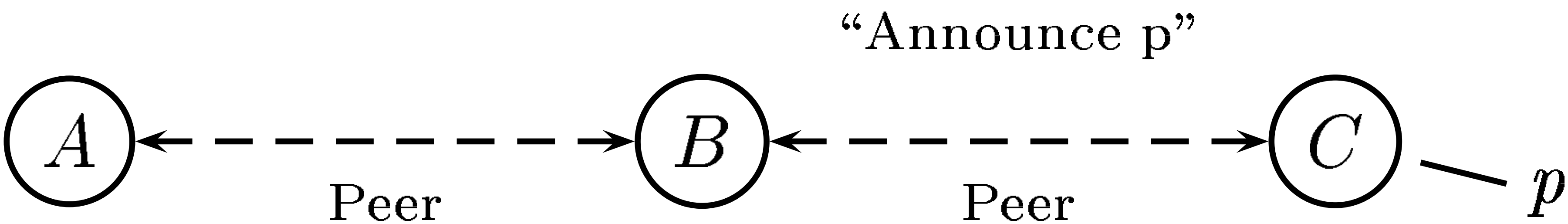
How Aggregation Affects Validity



Incorrect aggregation does not accurately reflect ***reachability*** *of destination.*
(Operator should care.)

Information-flow Control

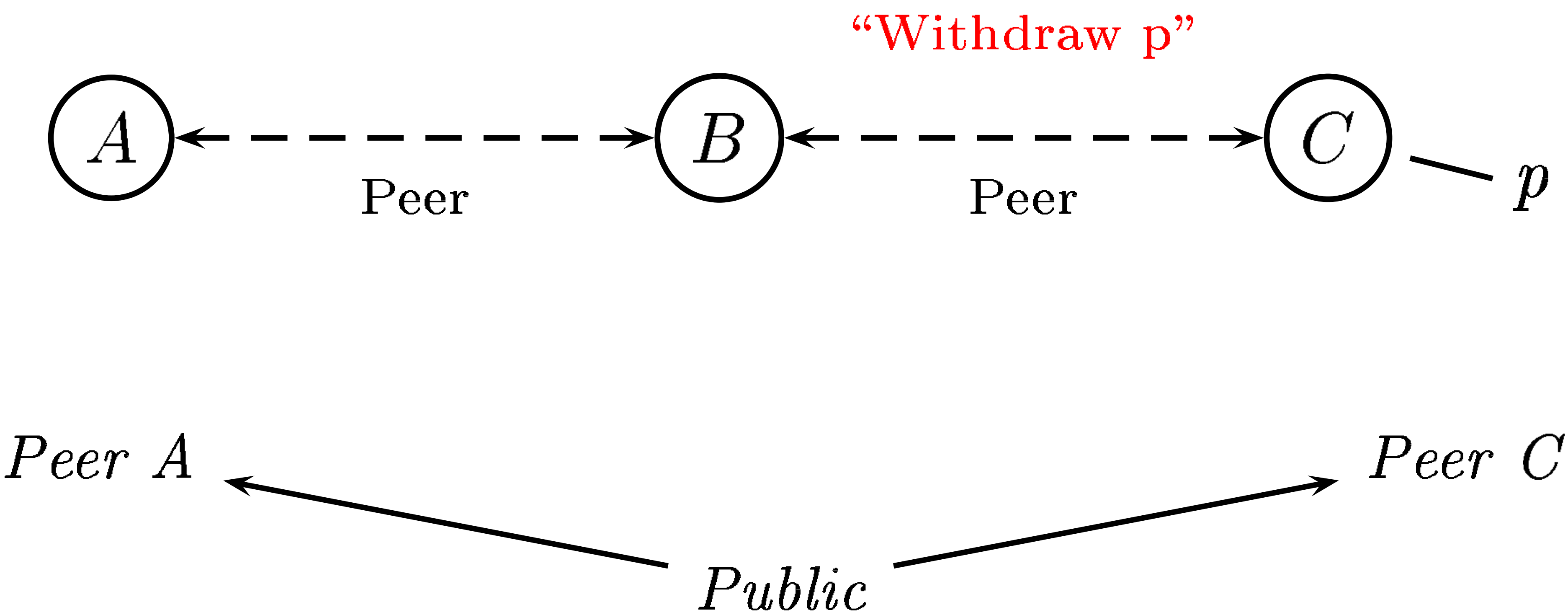
Simple rule: don't advertise routes from one peer to other peers.



Expressing constraints: Denning's lattice model

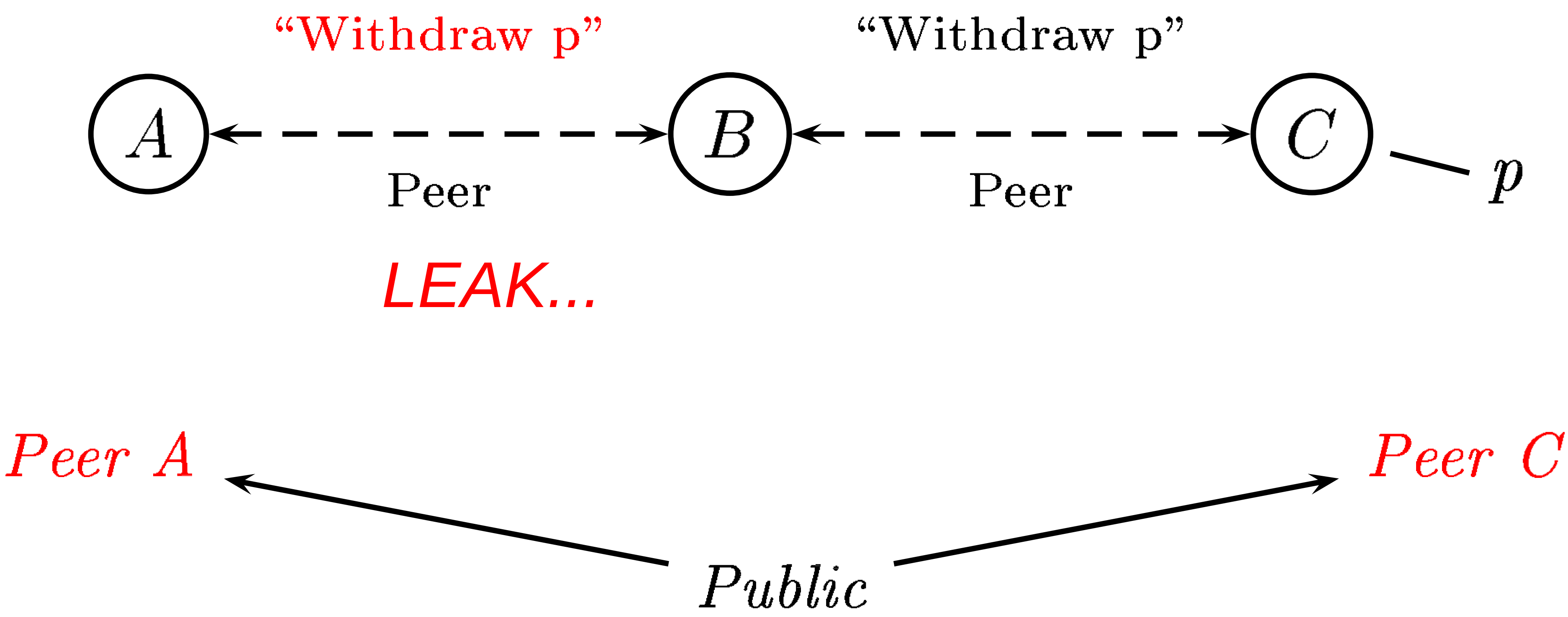
Information-flow Control

Example: "stateless" BGP implementation
(phenomenon observed by Labovitz in 1997.)



Information-flow Control

Example: "stateless" BGP implementation
(phenomenon observed by Labovitz in 1997.)



Reasoning about BGP's Behavior

The routing logic rules can be used to prove theorems about these properties.

- Verifying that an arbitrary route reflector configuration satisfies validity is NP-complete.
- Route reflectors that re-advertise all eBGP-learned routes will satisfy validity.
- Certain fixes to other problems (e.g., safety) can violate information-flow policy.

Verifying Configuration

- *Why?* Unlike most protocols, BGP's correctness depends heavily on how it is configured.

- *How?* To validate a property:
  - enumerate aspects of configuration that affect it
  - test that those aspects conform to certain rules

- *Limitations?* Some aspects involve cooperation across ASes; not really possible today.

That's OK, plenty goes wrong inside of one AS, too.

Open Questions

- Timing-related issues (e.g., convergence times, etc.)
- Some configuration validation is about verifying intent. (e.g., aggregation)
- Applicability to other routing protocols
  - (e.g., overlays, ad-hoc, etc.)
  - Perhaps some different rules or properties

Conclusion

- Network operators and protocol designers need a logic to reason about routing protocols like BGP
- The routing logic provides
  - A set of properties to describe protocol behavior
  - Rules to reason about them
- Set of properties is not complete, but it is an important and interesting set
- Promising for reasoning, verification, and design

The BGP Decision Process

Step 1:	Highest Localpref
Step 2:	AS Path Length
...	...
Step 4:	Lowest MED (routes from same AS)
Step 5:	Prefer eBGP over iBGP-Learned Routes
...	...
Step 8:	Lowest Router ID

Routing Needs a Framework for Reasoning

Protocol designers and network operators need a way to describe and reason about protocol behavior.

Protocol	What is Correct?	Reasoning Framework
Authentication	Alice knows that she is talking to Bob, etc.	BAN Logic
Routing	???	???

Routing Needs a Framework for Reasoning

Protocol designers and network operators need a way to describe and reason about protocol behavior.

Protocol	What is Correct?	Reasoning Framework
Authentication	Alice knows that she is talking to Bob, etc.	BAN Logic
Routing	**Routing Logic Properties**	**Routing Logic Rules**

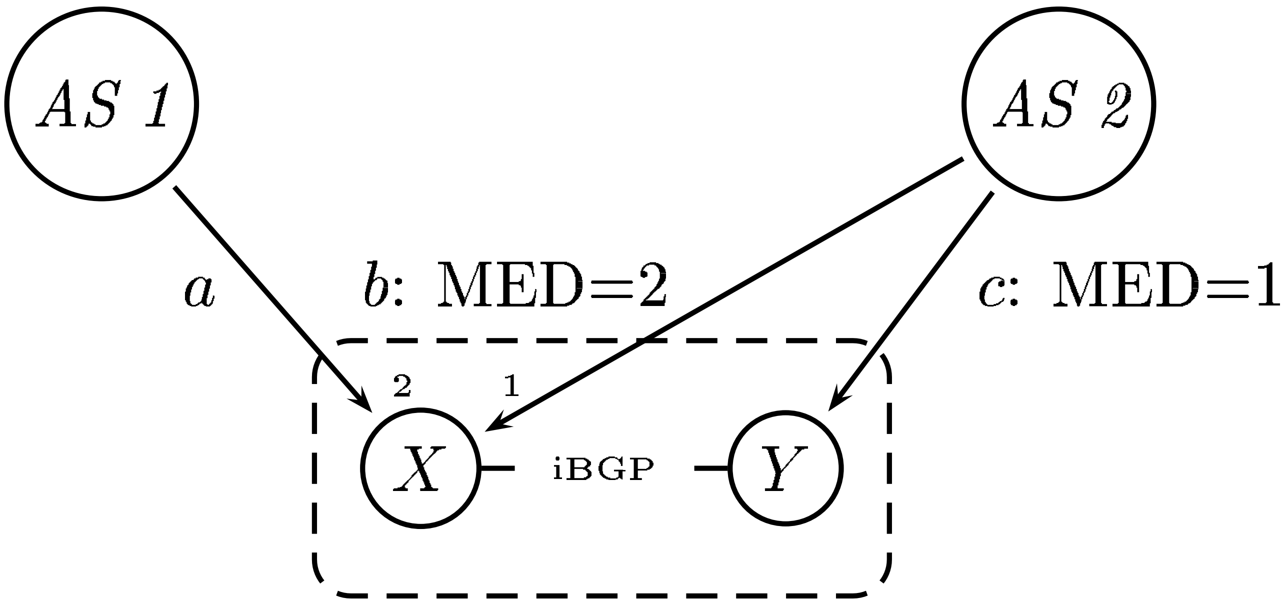
- **Properties:** describe behavior
- **Rules:** reason about whether a certain property holds

Routing Logic Properties

- Does it advertise invalid routes?
- Does every valid path have a corresponding route?
- Given a set of choices, will it converge to a unique, stable answer?
- Is that answer affected by the ordering of messages or the set of available routes?
- Does the protocol "leak" information?

Determinism: obvious, right?

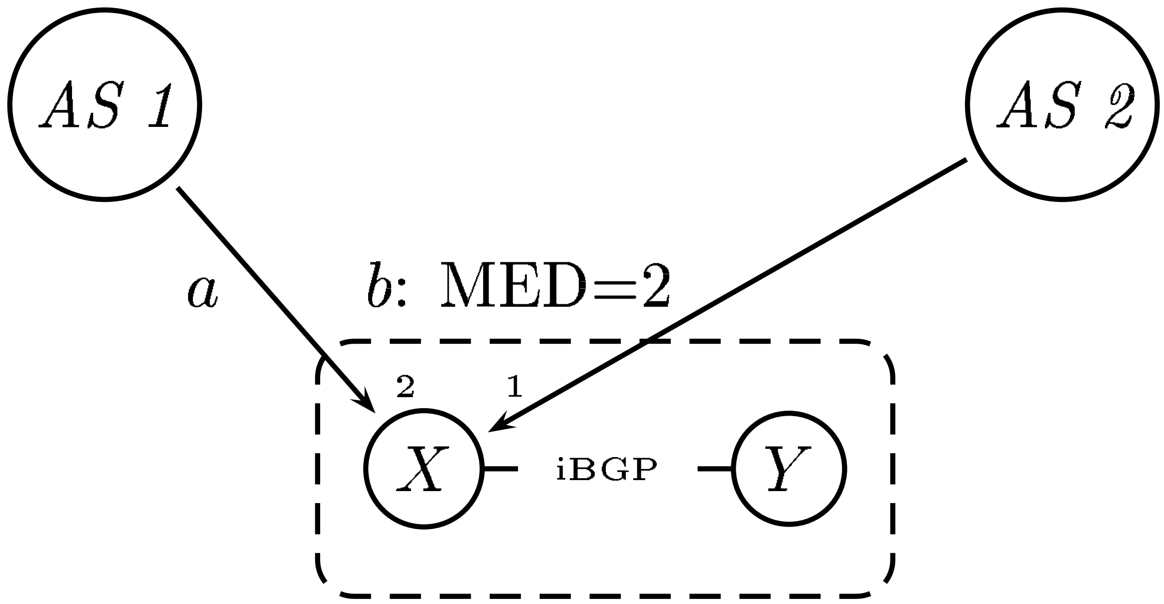
Non-transitivity: message ordering can affect outcomes.



Default behavior:
a, b, c => best route is c
b, c, a => best route is a
c, a, b => best route is b

So...let’s just get the configuration right?

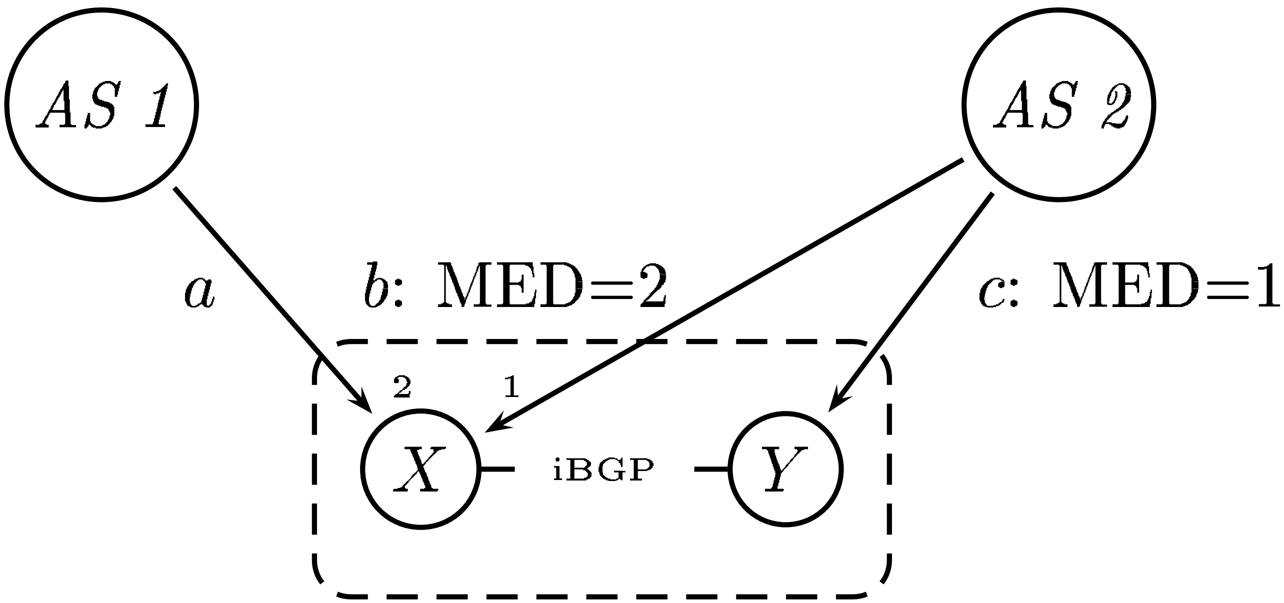
Nope...even with "deterministic-med", BGP can still violate determinism!



Best route at X: **b**.

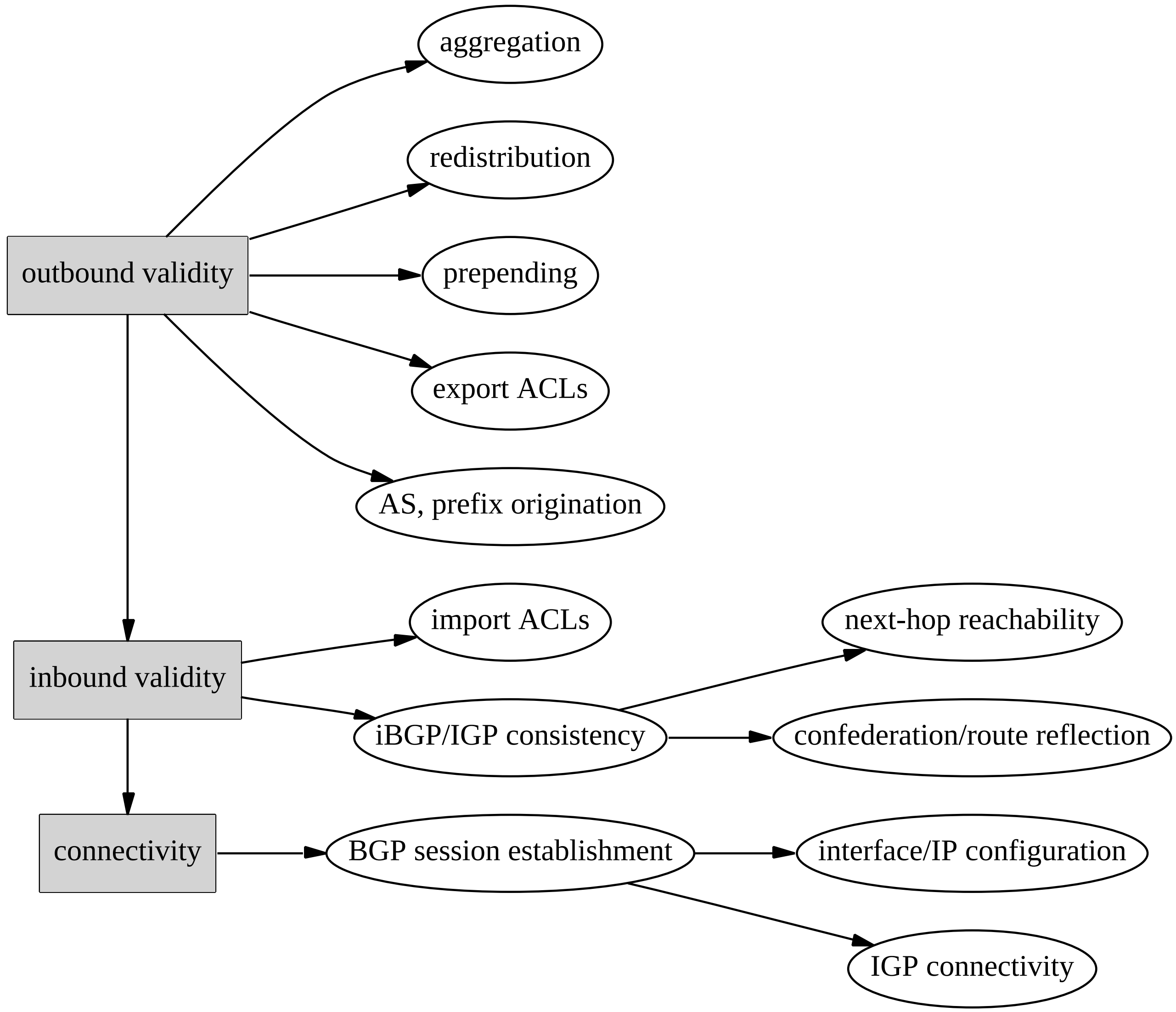
So...let’s just get the configuration right?

Nope...even with "deterministic-med", BGP can still violate determinism!



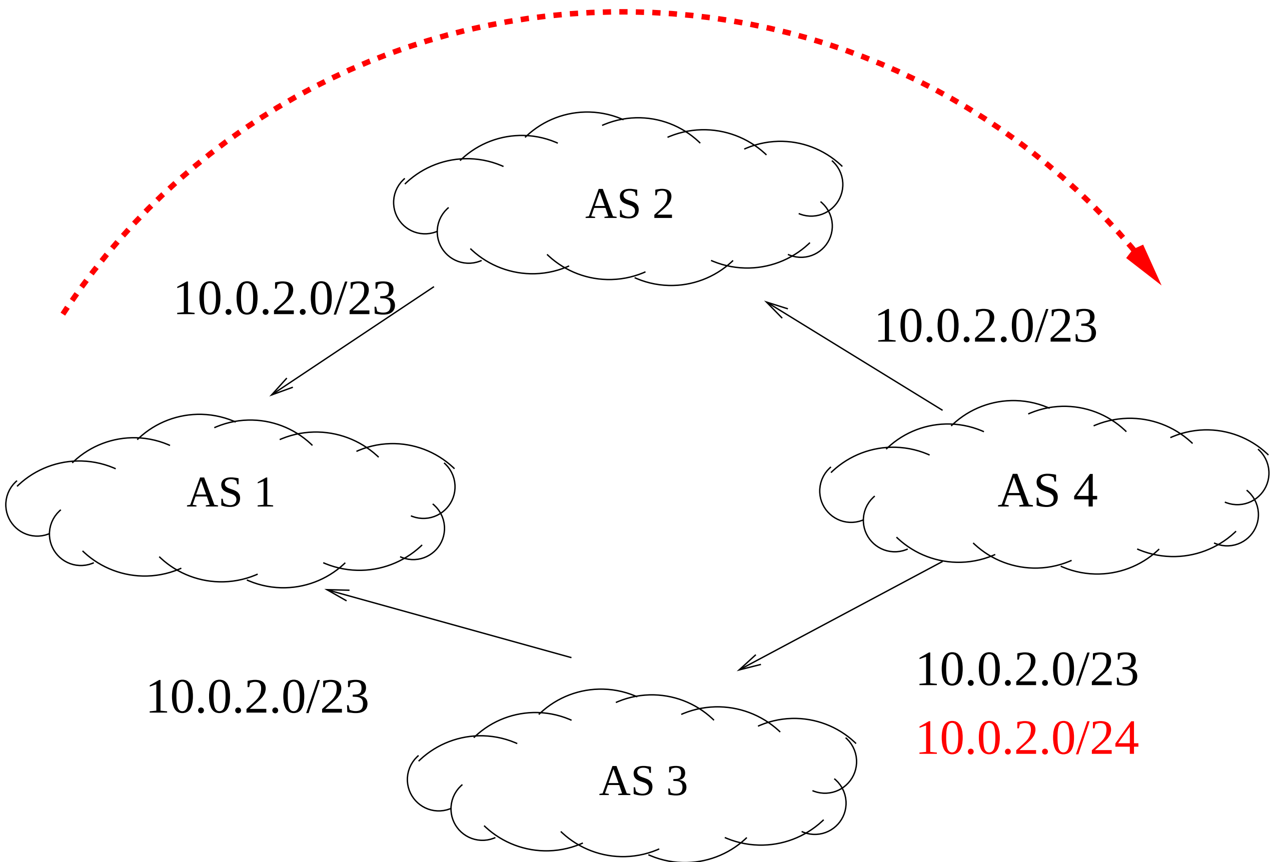
Best route at X: **a**.

Verifying Configuration



Some of these aspects are more straightforward than others.

Can we have valid paths and hide them, too?



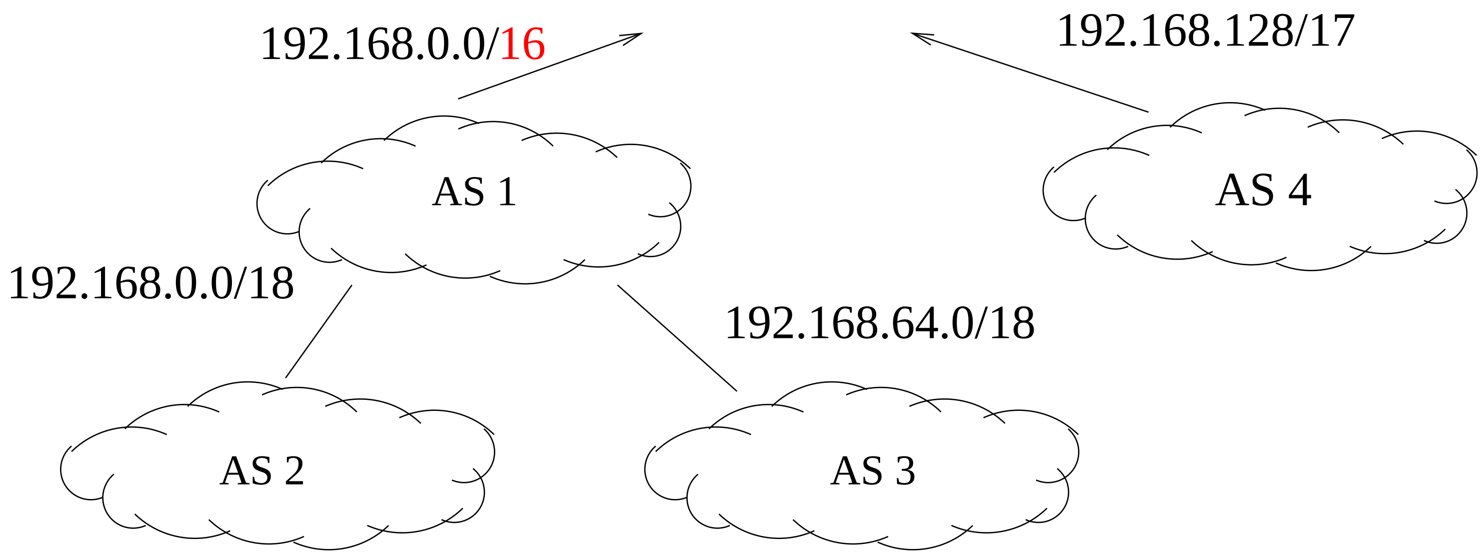
4’s policy (for "valid paths"): 3 preferred, 2 backup
3’s info-flow: Don’t accept prefixes smaller than /23

A’s path to D violates **policy conformance.**

The properties: not complete, but important

- **Validity:** Will packets that use this route get there?
  - basic correctness property
- **Visibility:** Is best route chosen from all possibilities?
  - optimal routing, robustness in failure scenarios
- **Safety:** Is there policy-induced oscillation?
  - network stability
- **Determinism:** Can a snapshot of the network state determine the result of the "computation"?
  - ease of debugging, traffic engineering
- **Information-flow Control:** Is my network exposing information that should be hidden?
  - competitive aspects

How Aggregation Affects Validity



"Over-aggressive" aggregation does not accurately reflect **progress** *to destination.*

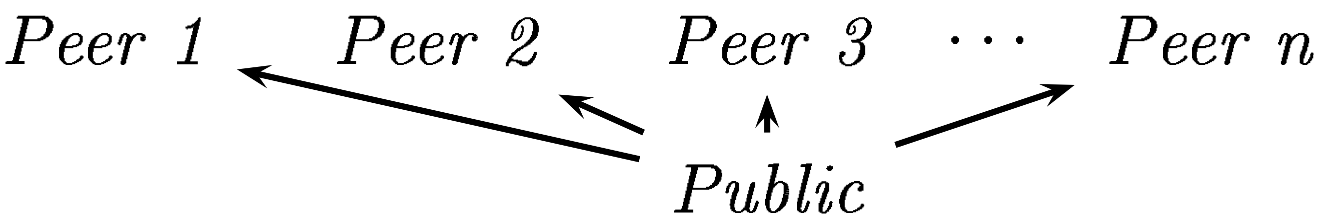
(Operator should care.)

Information-flow Control

Ensure that routing protocol doesn’t "leak" information.

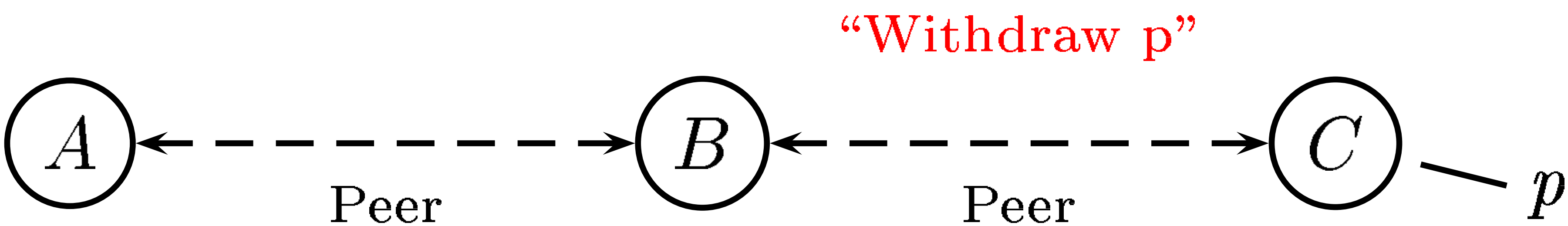
- Idea: Denning’s lattice model.

- Rule: "read access" goes down the lattice only
  - e.g., don’t advertise routes heard from one peer to another peer



Information-flow Control

Example: "stateless" BGP implementation
(phenomenon observed by Labovitz in 1997.)



A: *peer A*; prefixes from A: *customers*
C: *peer C*; prefixes from C: *customers*
D: *customers*; prefixes from D: *public*