Communication Networks

Spring 2017

Laurent Vanbever

www.vanbever.eu

ETH Zürich (D-ITET)

April, 24 2017

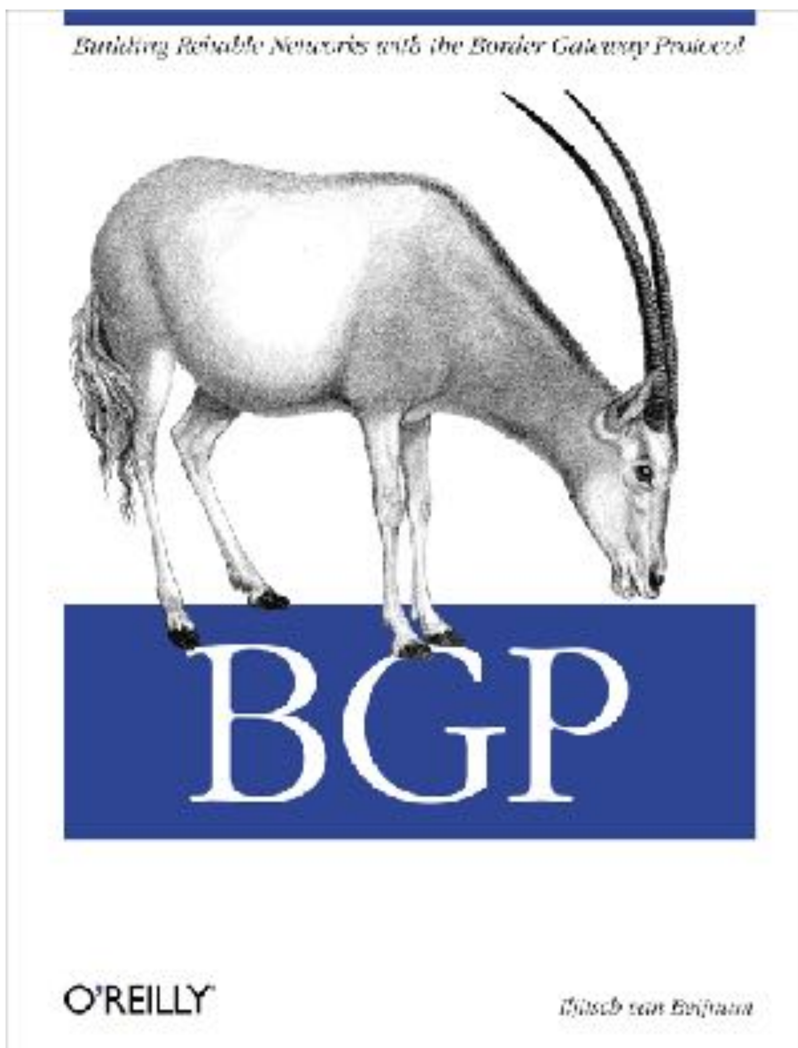
Material inspired from Scott Shenker & Jennifer Rexford

Two weeks ago on

Communication Networks

Border Gateway Protocol

policies and more



1 **BGP Policies**

Follow the Money

2 **Protocol**

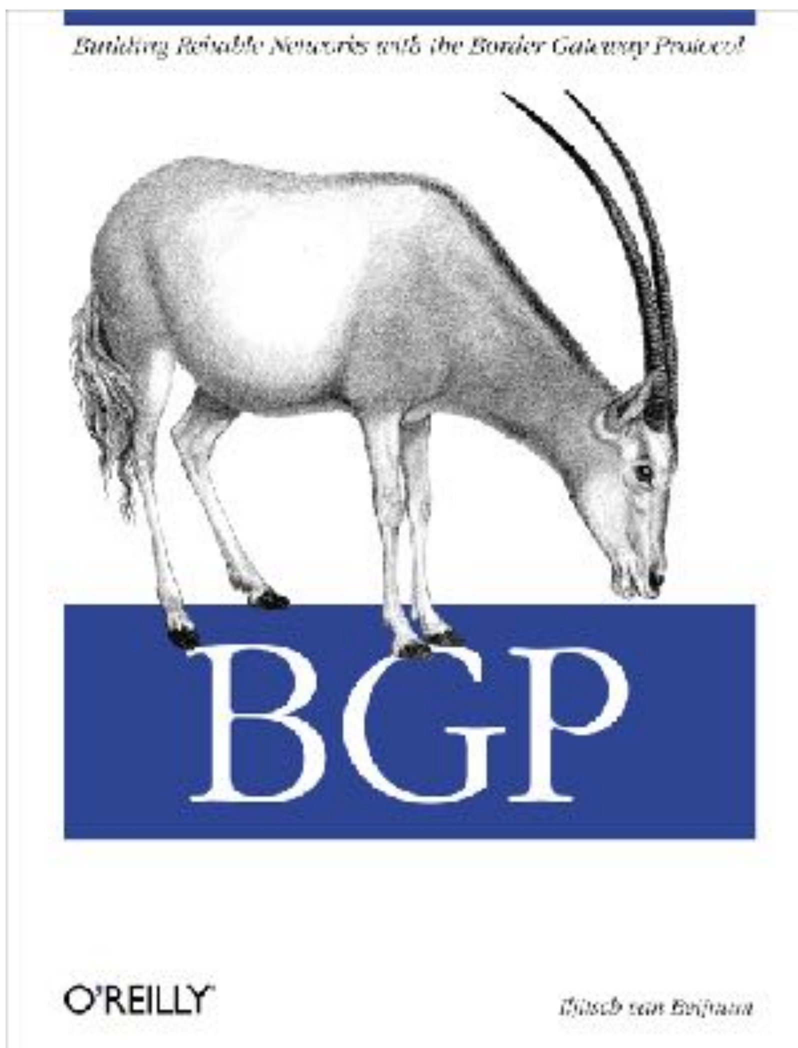
How does it work?

3 **Problems**

security, performance, …

Border Gateway Protocol

policies and more



1 **BGP Policies**

Follow the Money

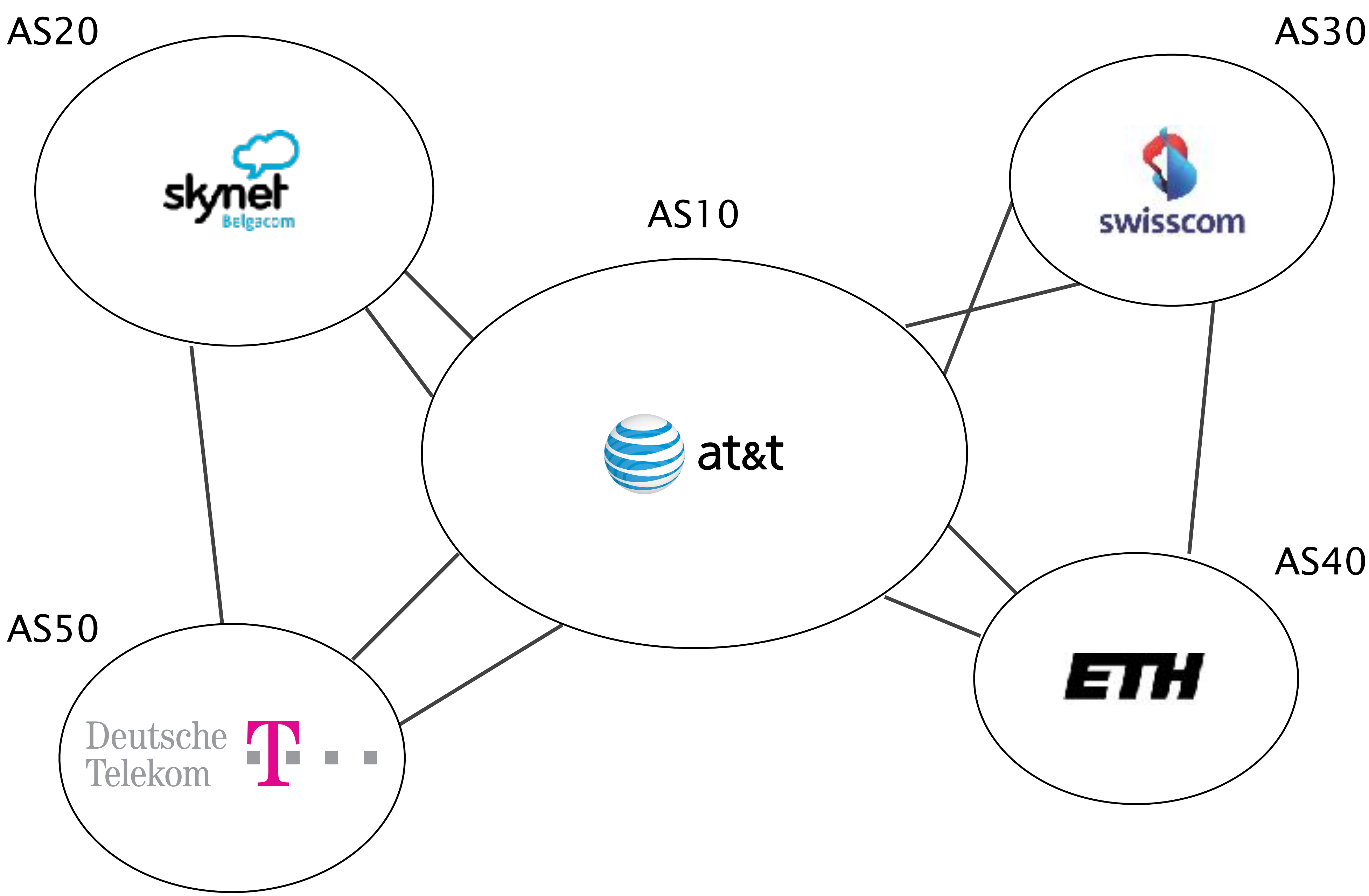
Protocol

How does it work?

Problems

security, performance, …

The Internet topology is shaped according to *business* relationships



There are 2 main business relationships today:

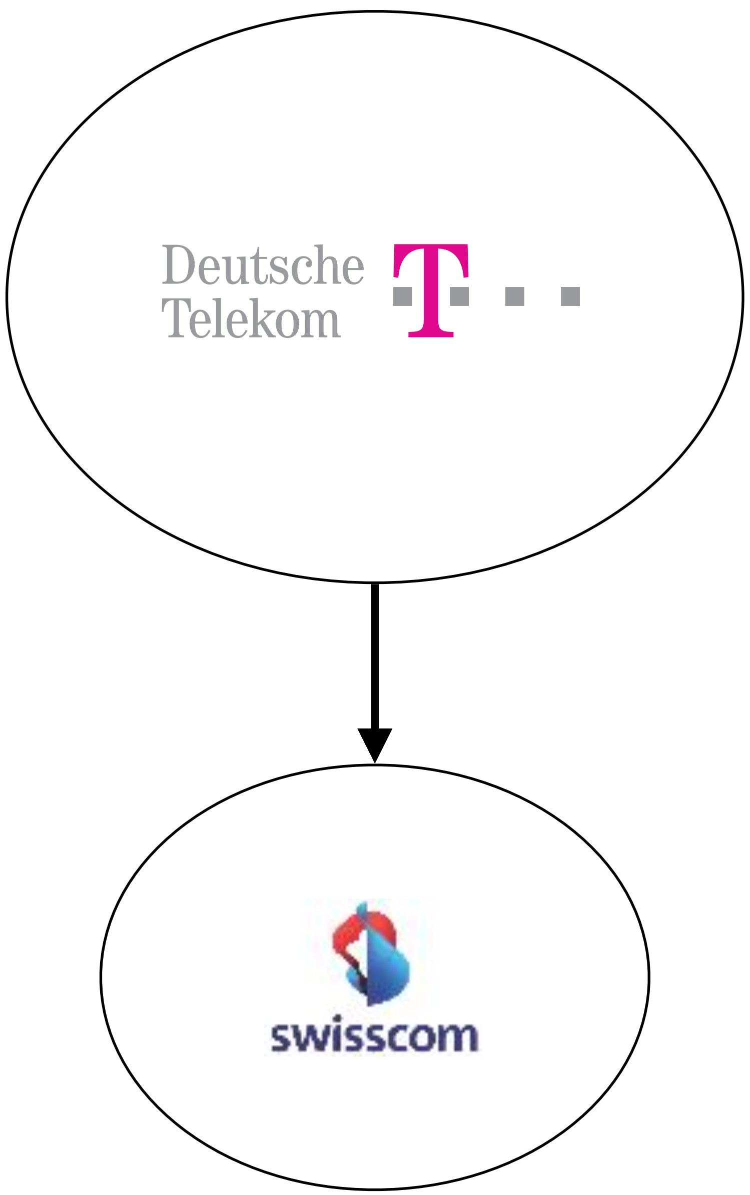
- customer/provider
- peer/peer

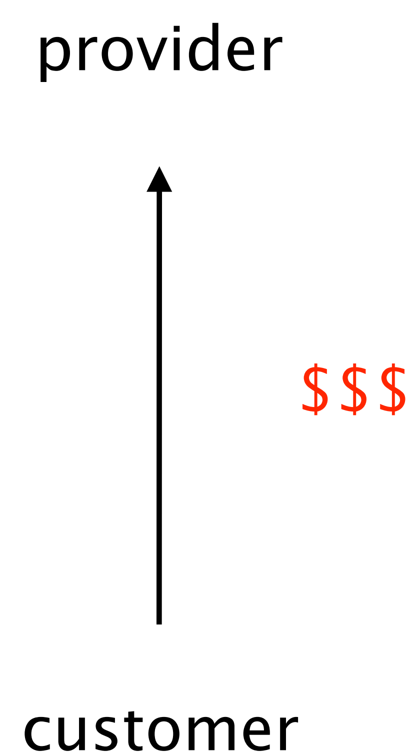
many less important ones (siblings, backups,…)

There are 2 main business relationships today:

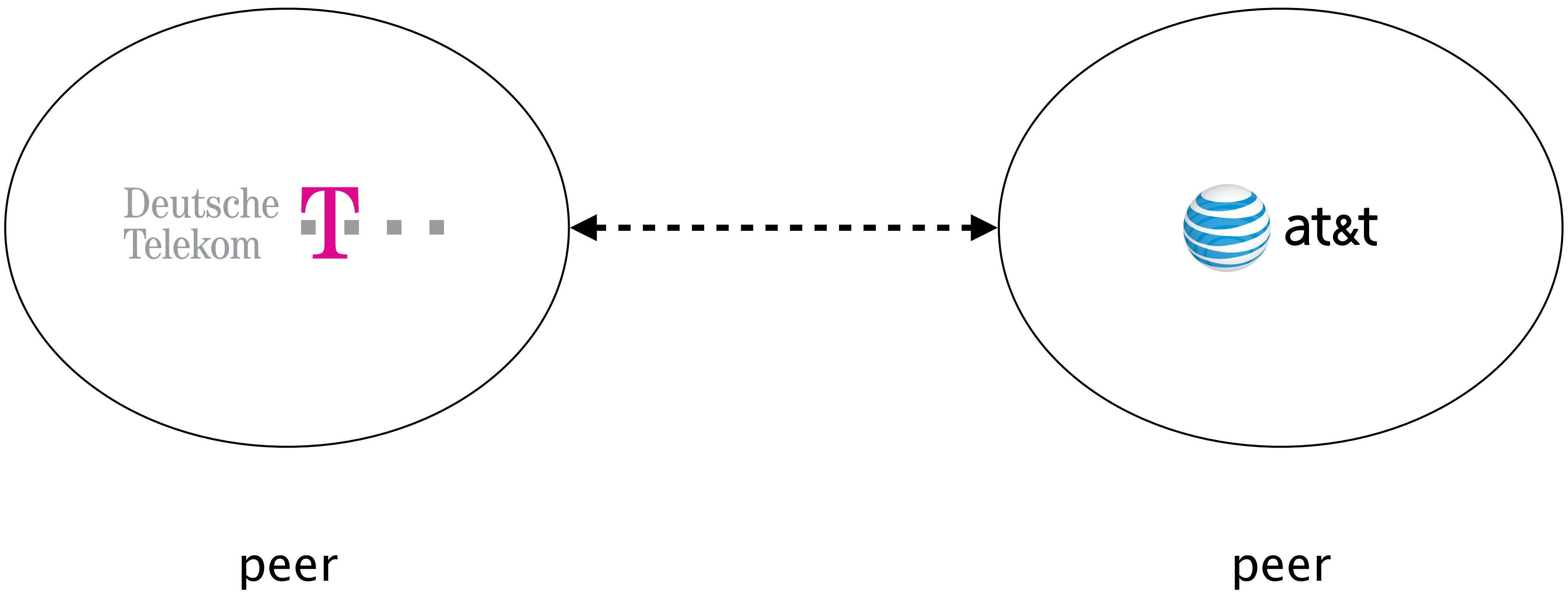
- customer/provider
- peer/peer

Customers pay providers
to get Internet connectivity





Peers don’t pay each other for connectivity, they do it *out of common interest*



DT and ATT exchange *tons* of traffic.

they save money by directly connecting to each other

Business relationships conditions

route selection

For a destination *p*, prefer routes coming from

- customers over
- peers over
- providers

↓ *route type*

Business relationships conditions

route exportation

		send to		
		customer	peer	provider
	customer			
from	peer			
	provider			

Routes coming from customers are propagated to everyone else

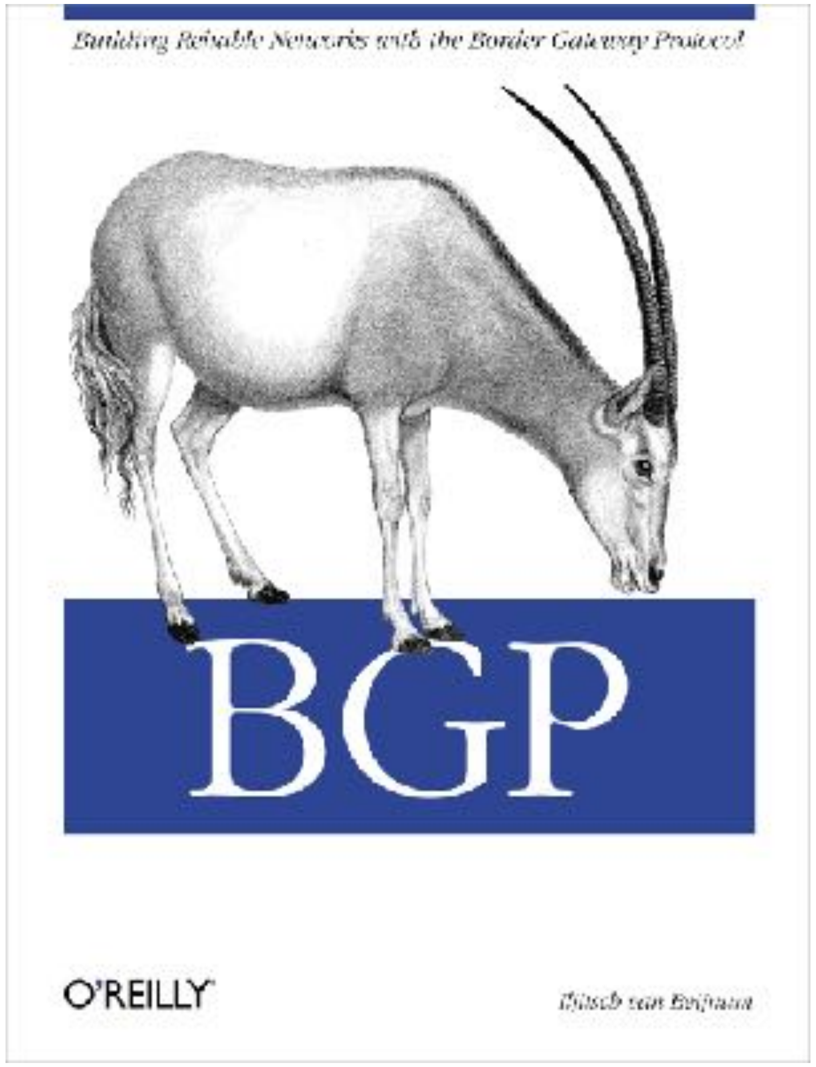
		send to		
		customer	peer	provider
	customer	✓	✓	✓
from	peer			
	provider			

Routes coming from peers and providers are only propagated to customers

		send to		
		customer	peer	provider
from	customer	✓	✓	✓
	peer	✓	-	-
	provider	✓	-	-

Border Gateway Protocol

policies and more



BGP Policies

Follow the Money

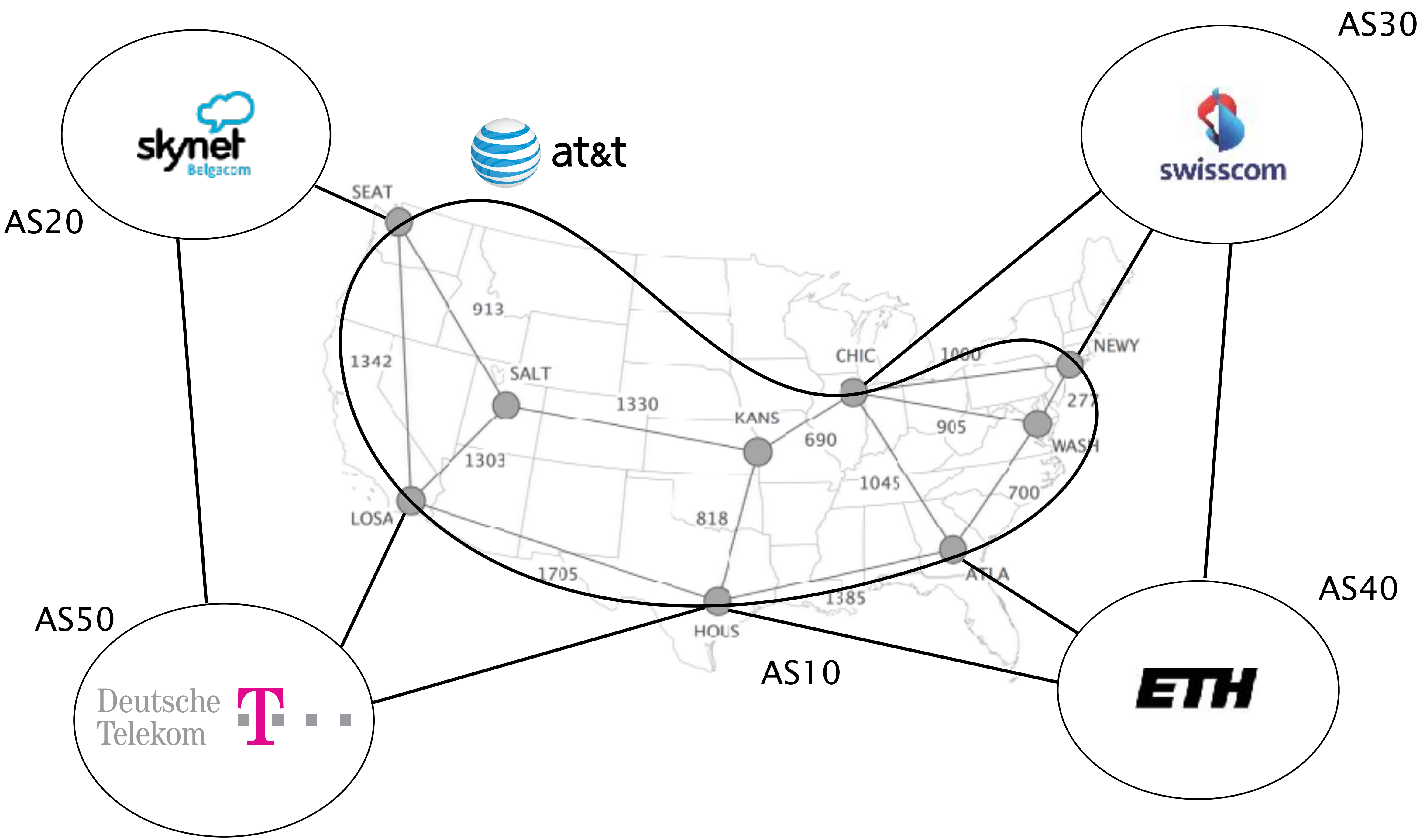
2 **Protocol**

How does it work?

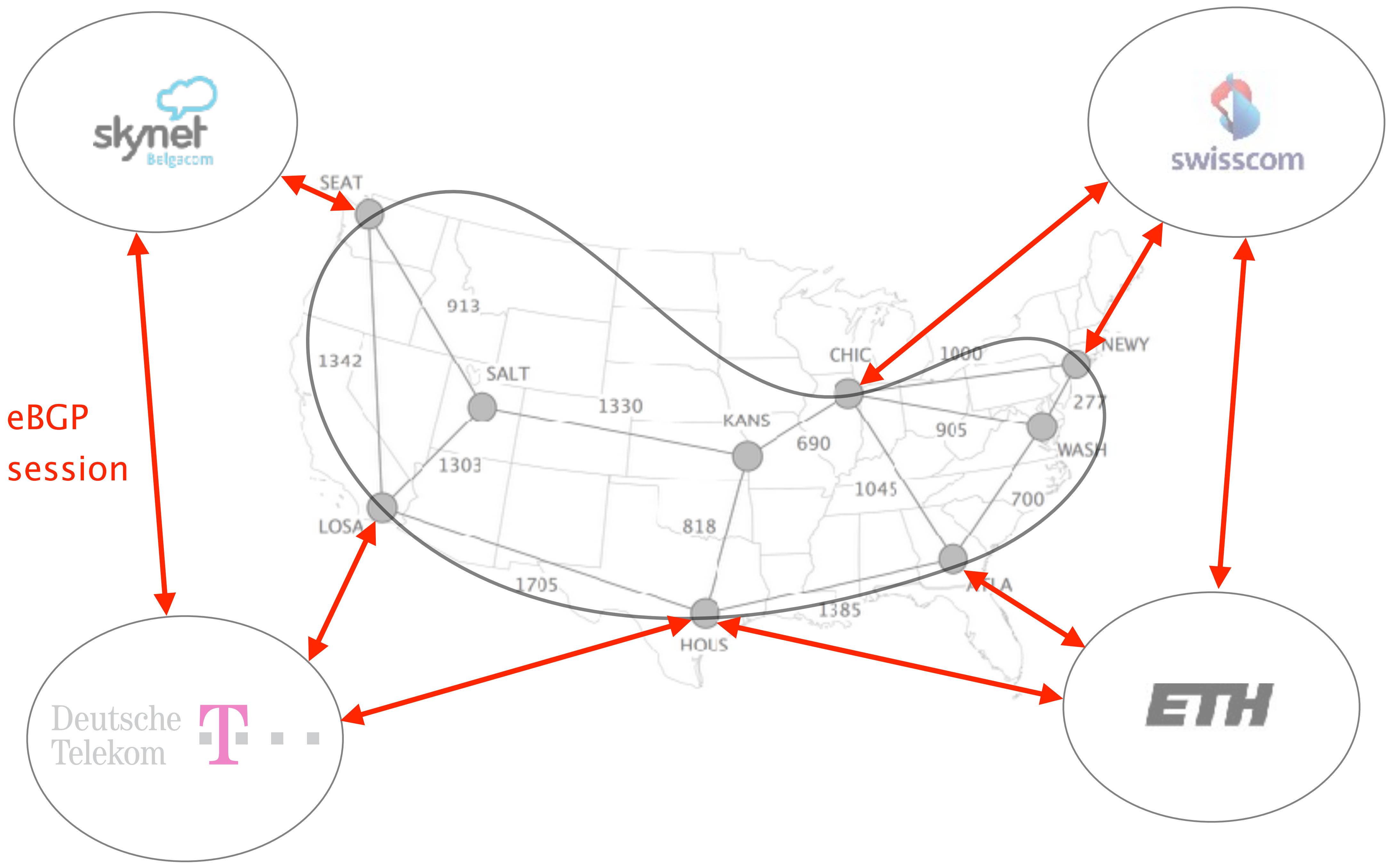
Problems

security, performance, …

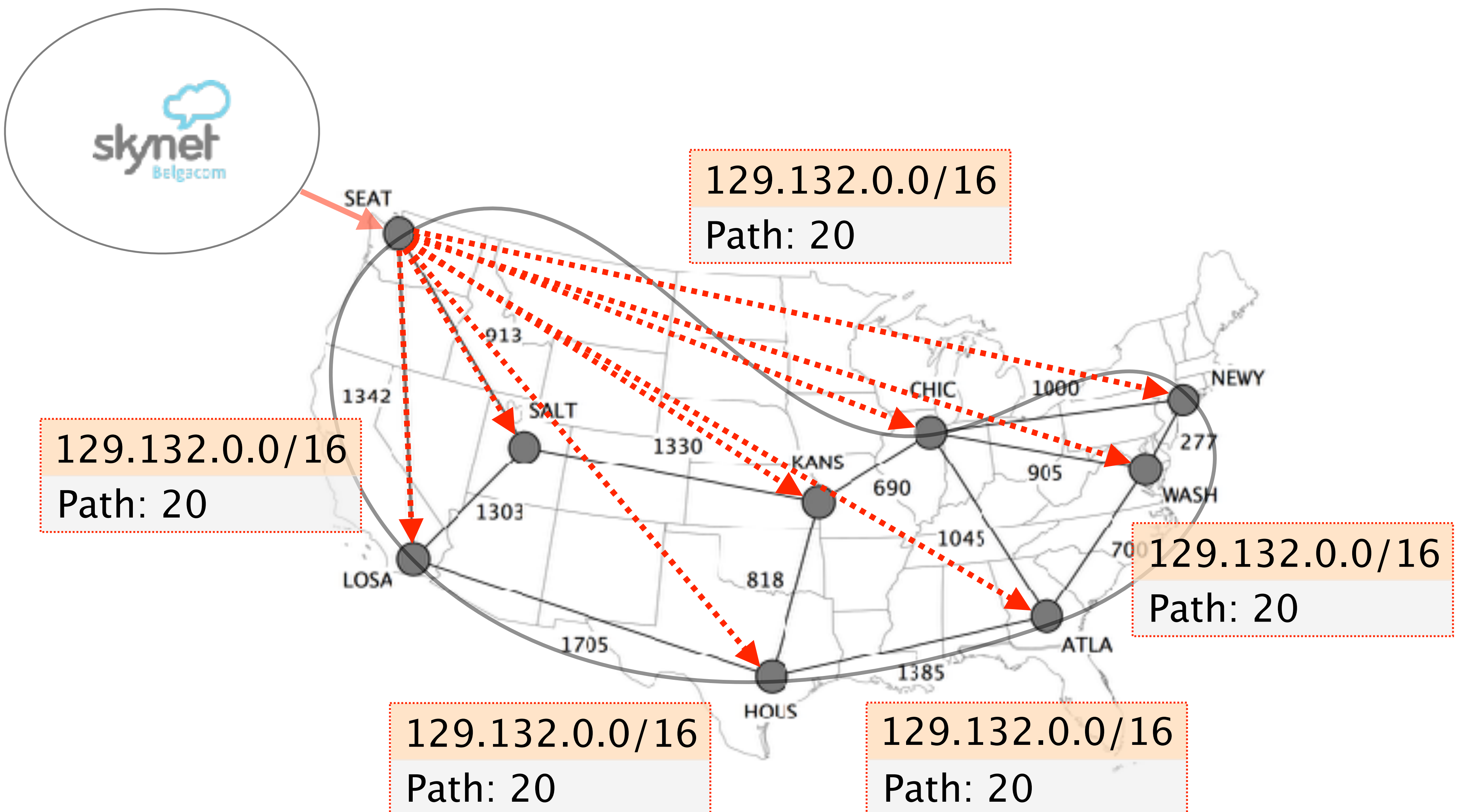
BGP sessions come in two flavors



external BGP (eBGP) sessions
connect border routers in different ASes



iBGP sessions are used to disseminate externally-learned routes internally



BGP UPDATEs carry an IP prefix together with a set of attributes

IP prefix

Attributes

Describe route properties

used in route selection/exportation decisions

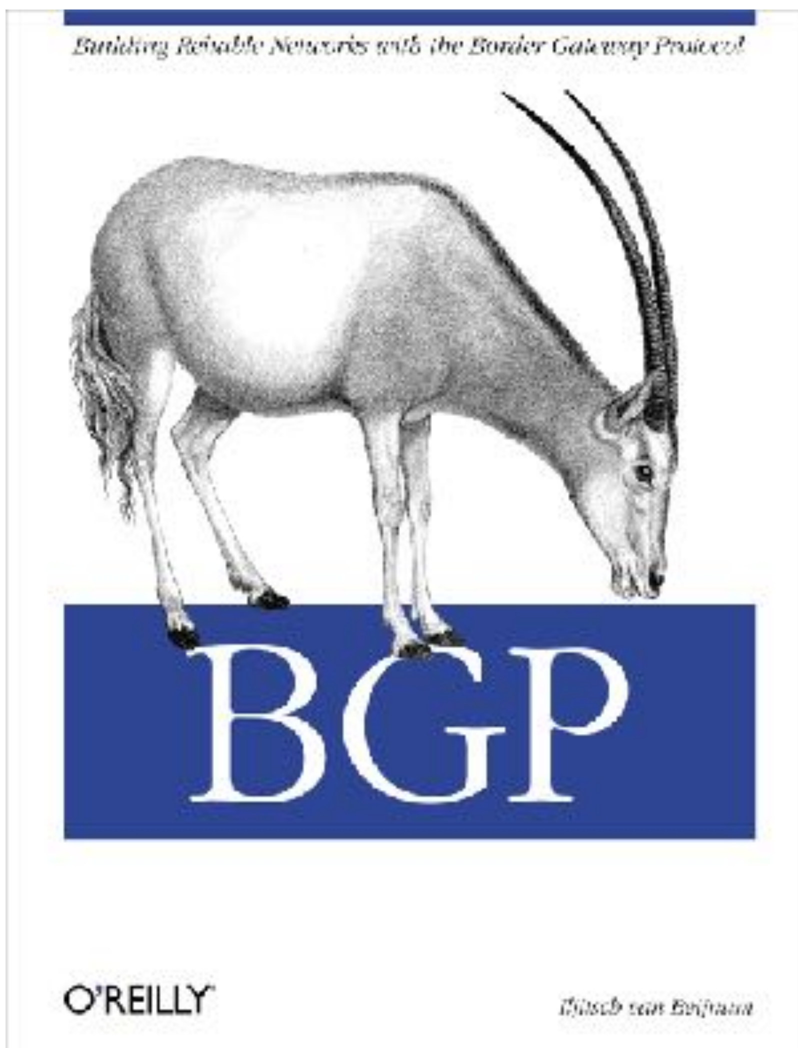
are either local (*only* seen on iBGP)

or global (seen on iBGP *and* eBGP)

Attributes	Usage
NEXT-HOP	egress point identification
AS-PATH	loop avoidance outbound traffic control inbound traffic control
LOCAL-PREF	outbound traffic control
MED	inbound traffic control

Border Gateway Protocol

policies and more



BGP Policies

Follow the Money

Protocol

How does it work?

3 **Problems**

security, performance, …

BGP suffers from many rampant problems

Problems

- Reachability
- Security
- Non-determinism
- Convergence
- Performance
- Anomalies
- Relevance

Problems

Reachability

Security

Non-determinism

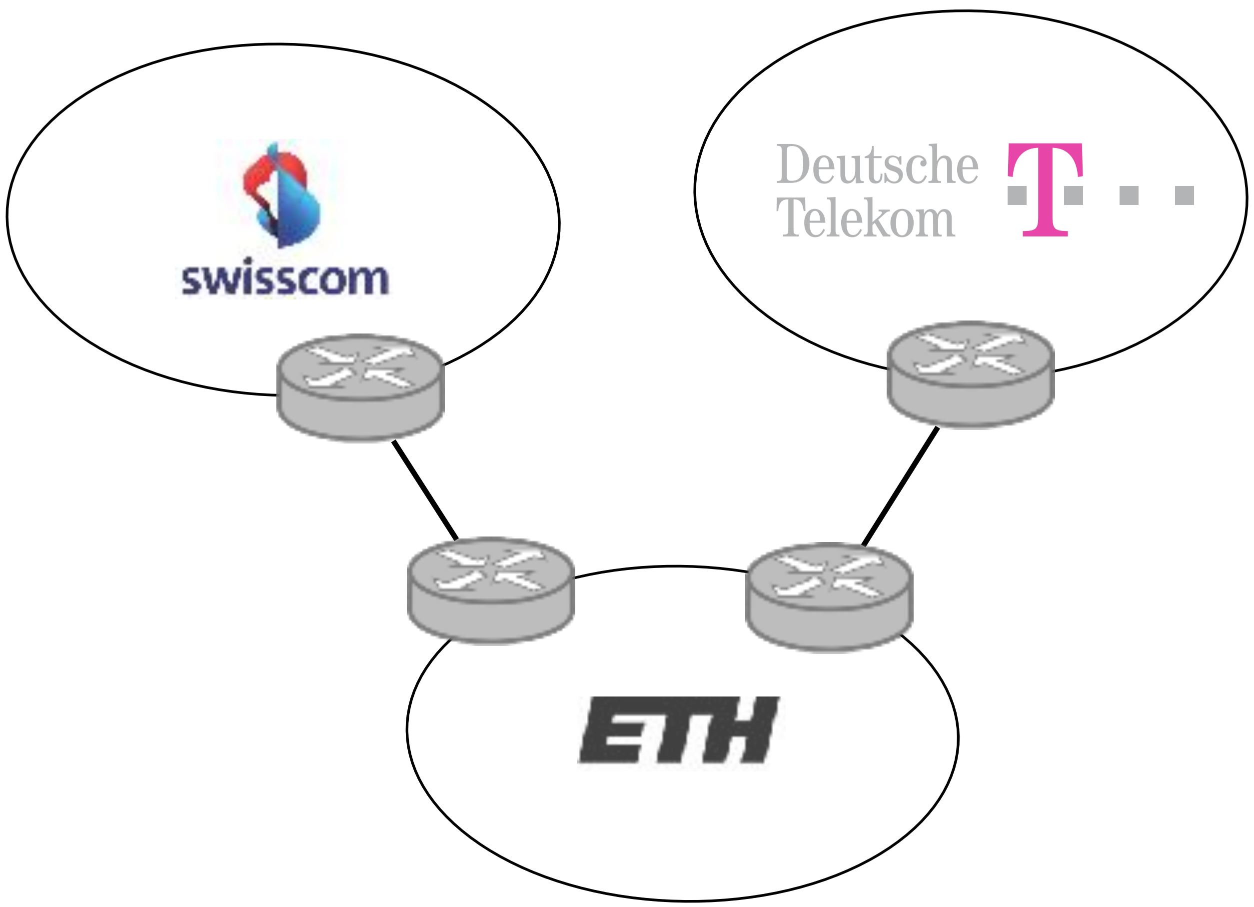
Convergence

Performance

Anomalies

Relevance

Unlike normal routing, policy routing does not guarantee reachability even if the graph is connected



Because of policies,

Swisscom cannot reach DT

even if the graph is connected

Problems

Reachability

Security

Non-determinism

Convergence

Performance

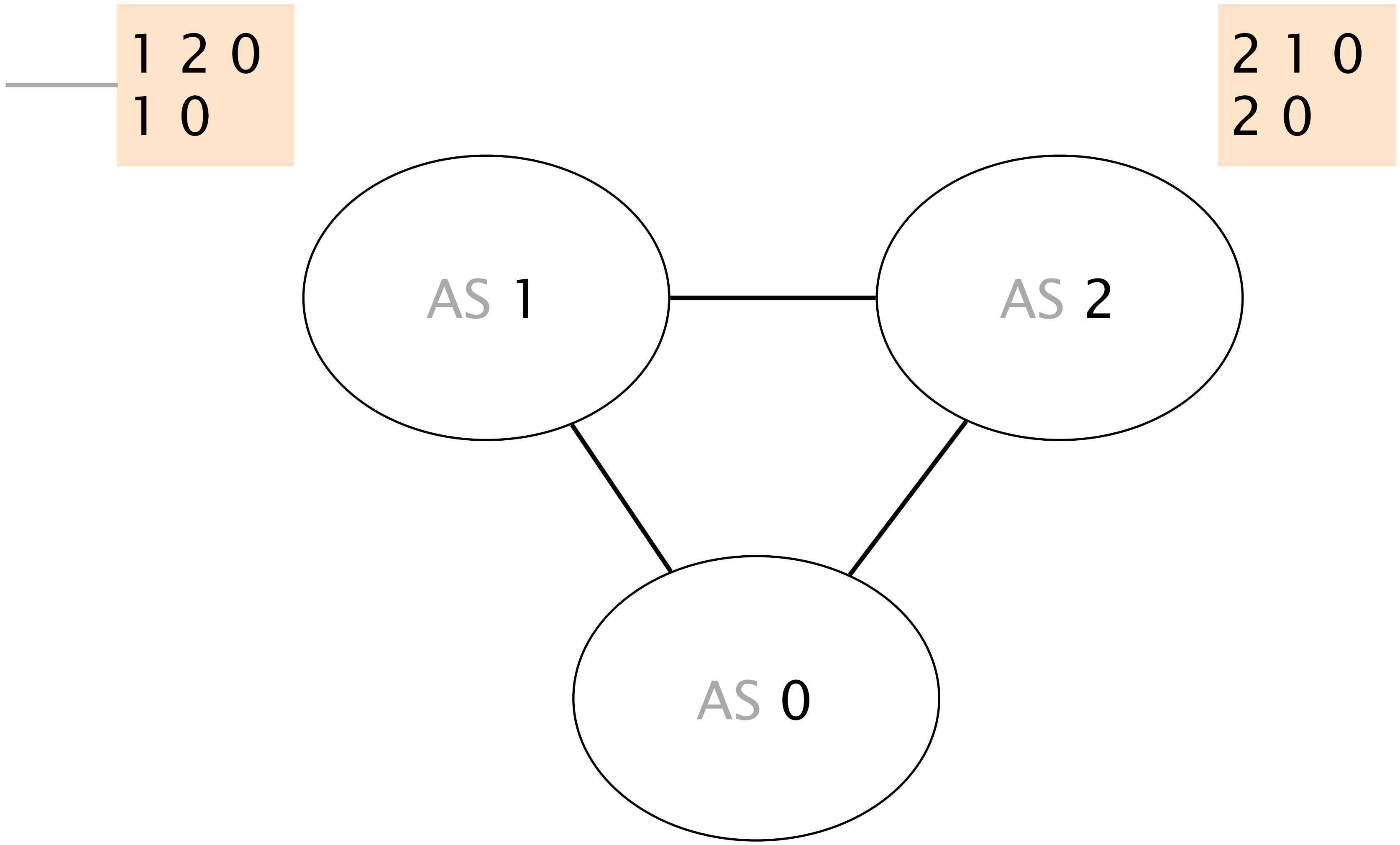
Anomalies

Relevance

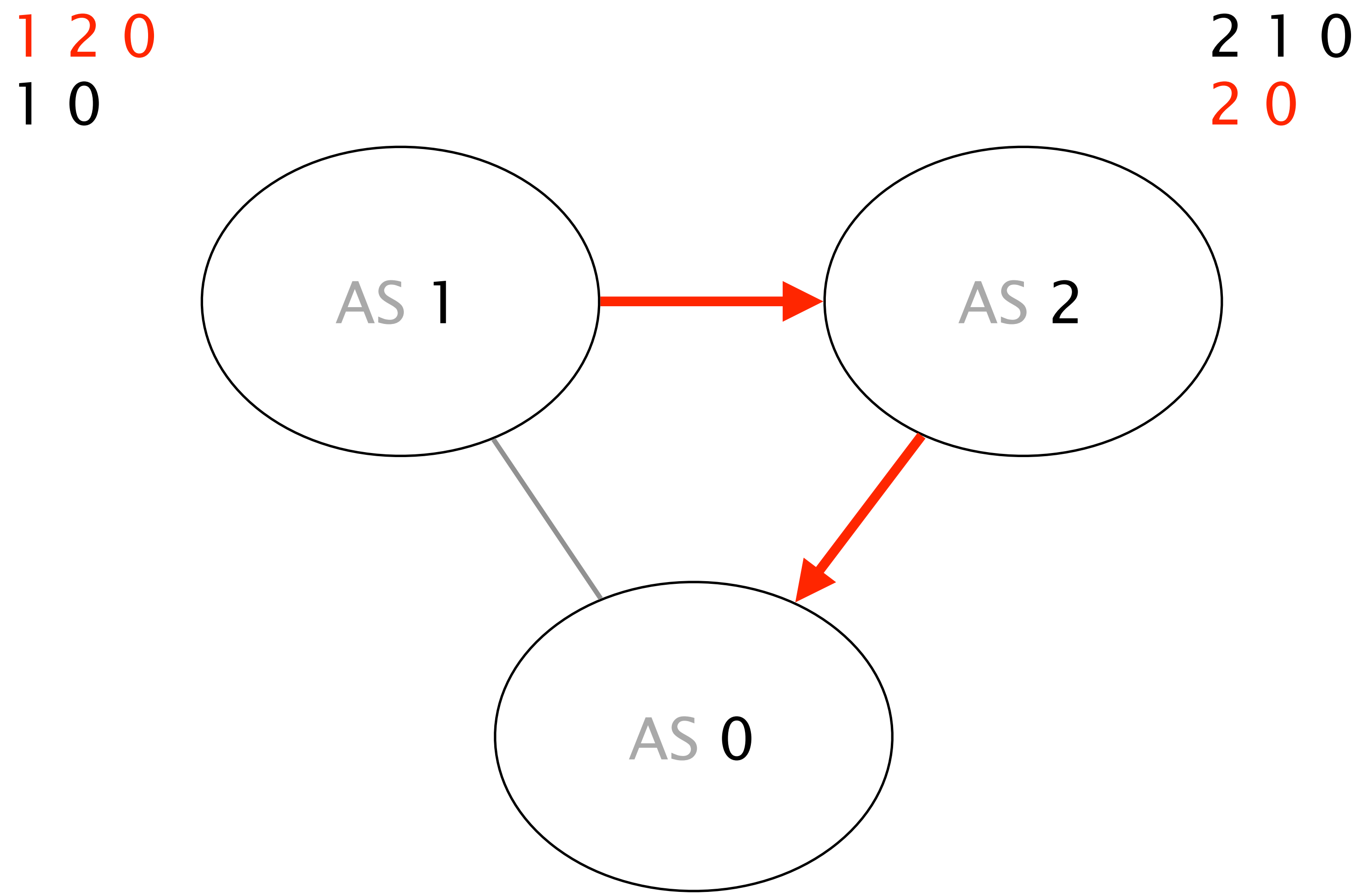
With arbitrary policies, BGP may have multiple stable states

preference list

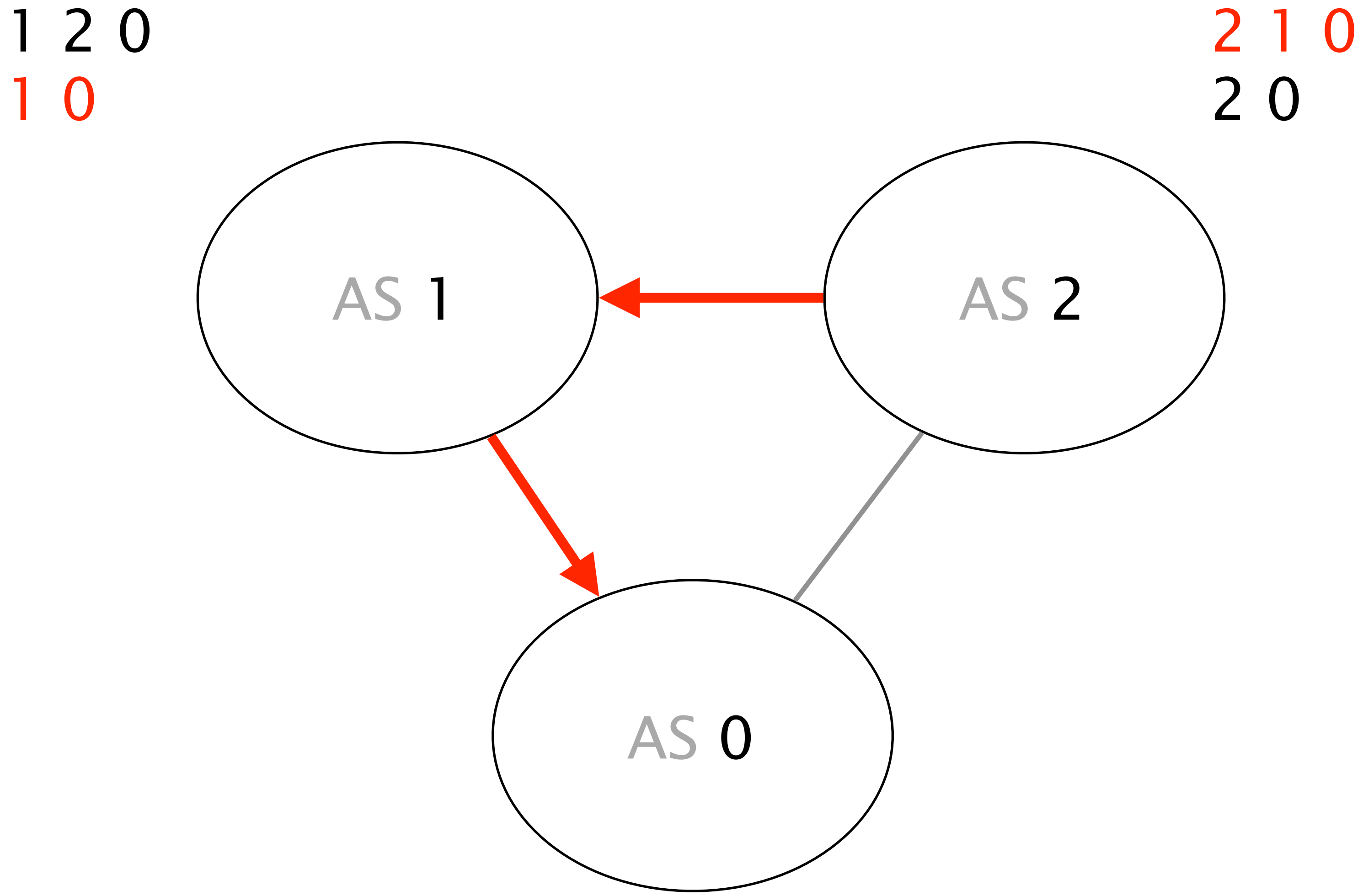
1 prefers to reach 0 via 2 rather than directly



If AS2 is the first to advertise 2 0,
the system stabilizes in a state where AS 1 is happy



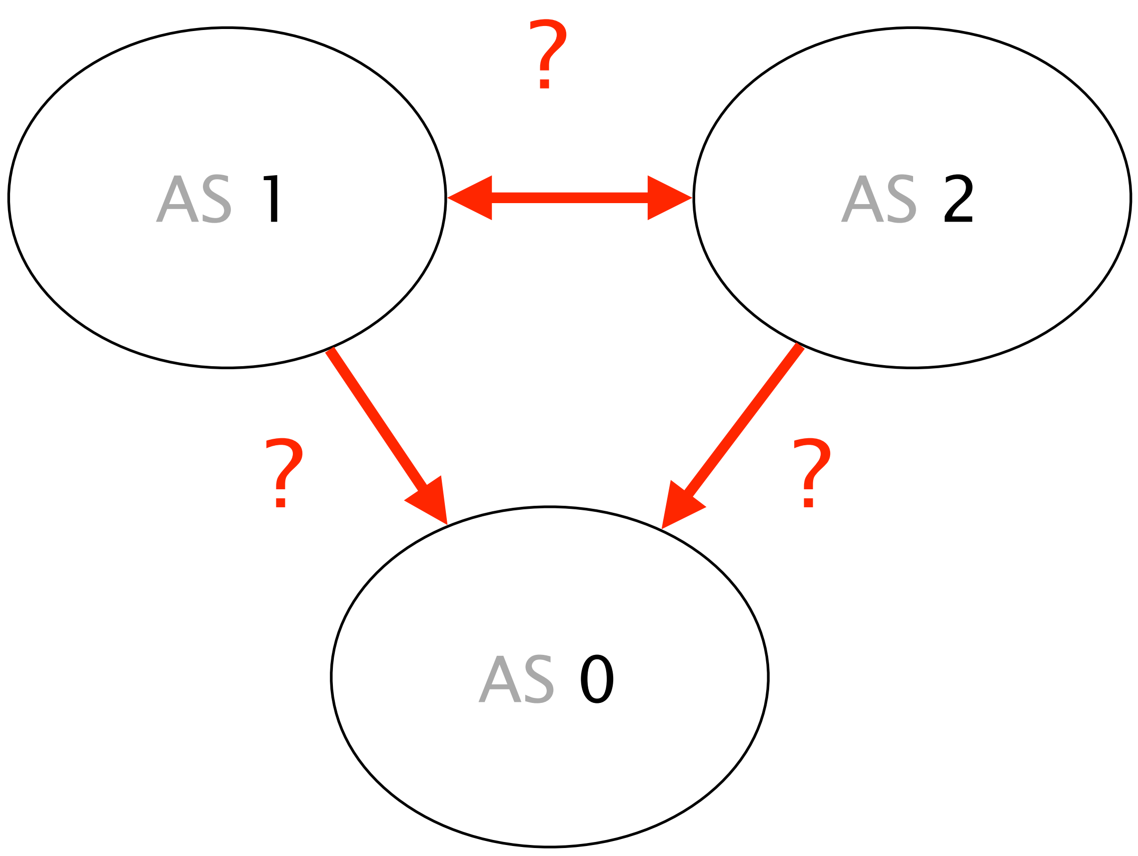
If AS1 is the first one to advertise 1 0,
the system stabilizes in a state where AS 2 is happy



The actual assignment depends on the ordering between the messages

Note that AS1/AS2 could change the outcome by manual intervention

… this is not always possible *



* https://www.nanog.org/meetings/nanog31/presentations/griffin.pdf

Problems

Reachability

Security

Non-determinism

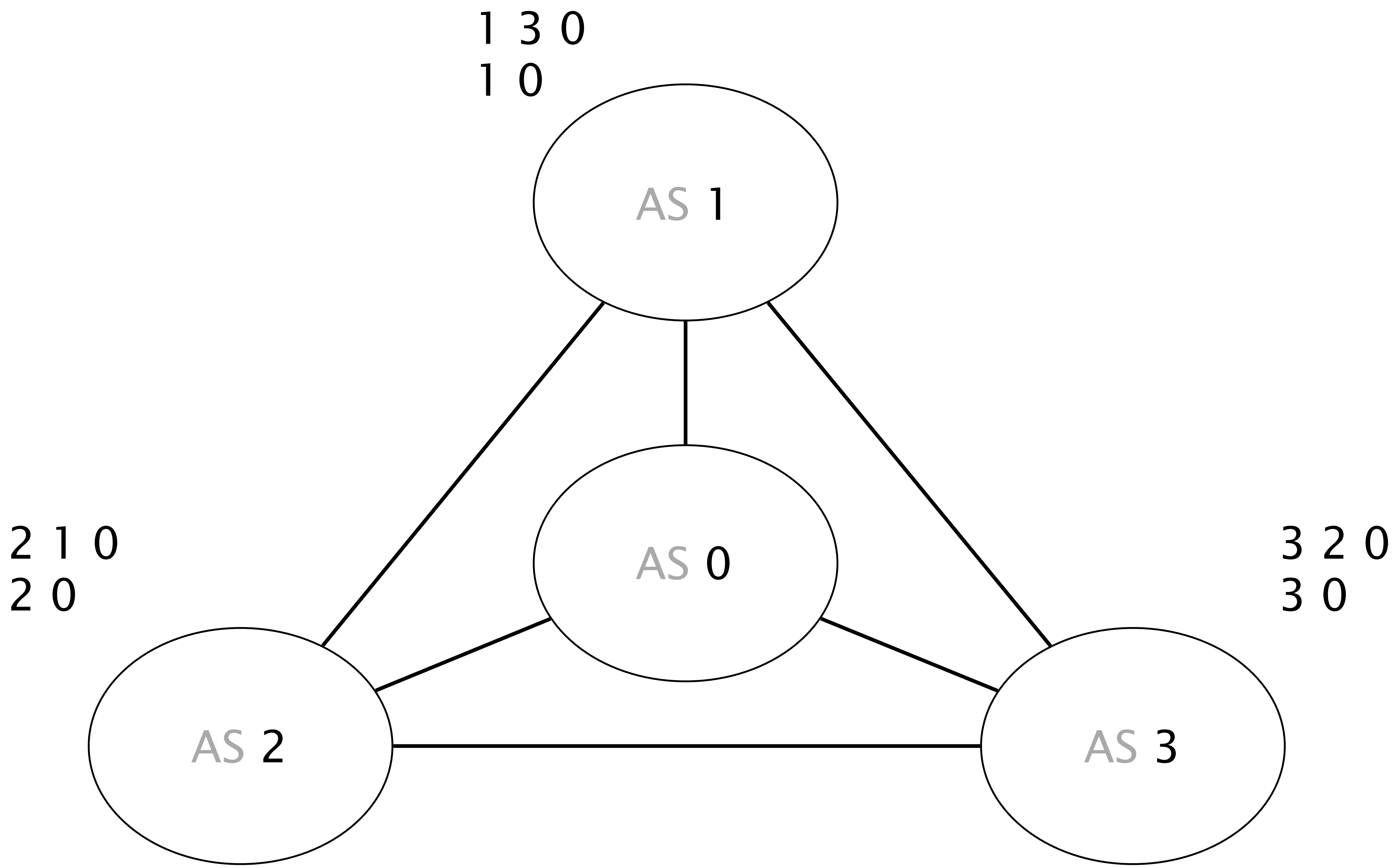
Convergence

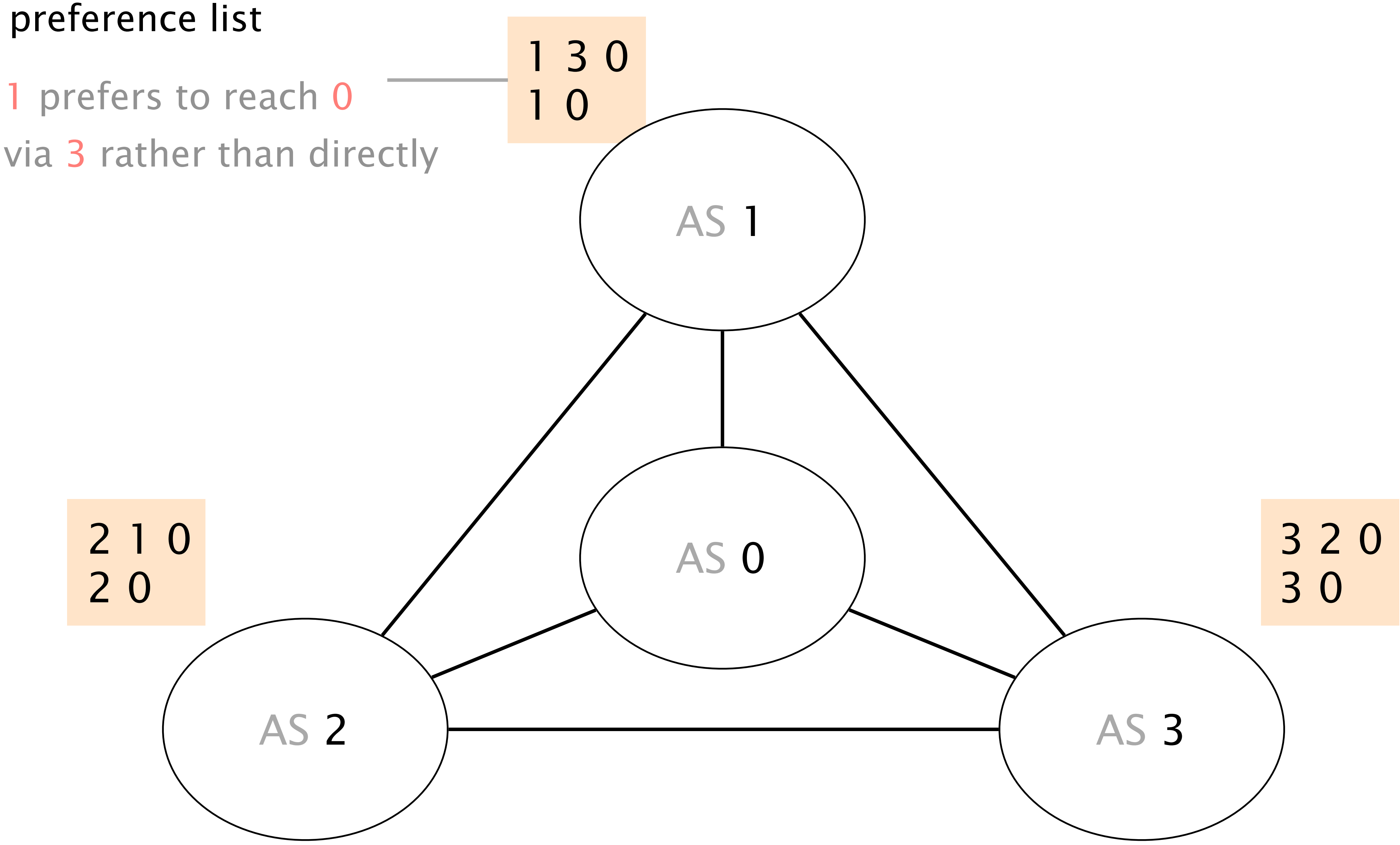
Performance

Anomalies

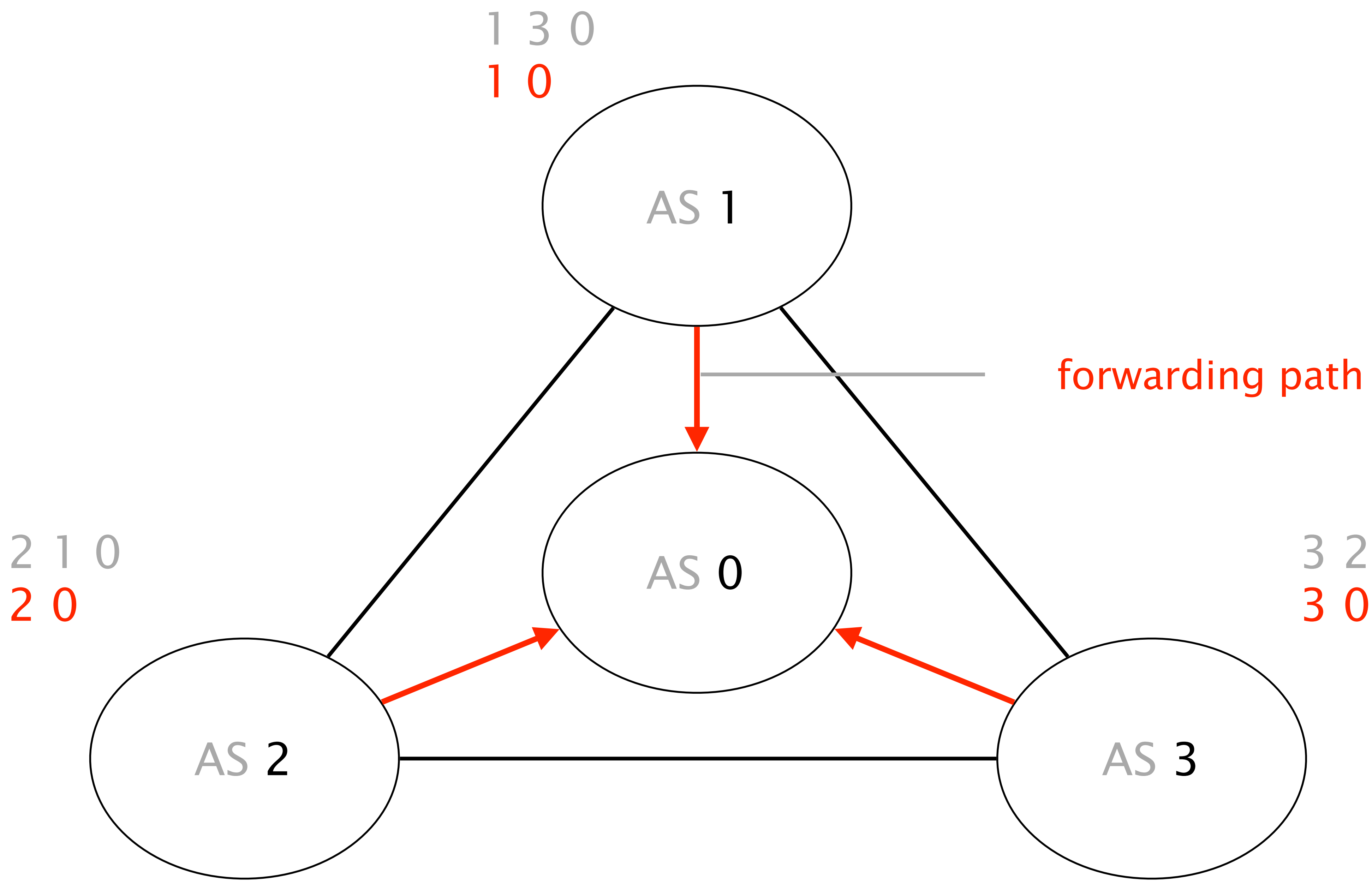
Relevance

With arbitrary policies,
BGP may fail to converge

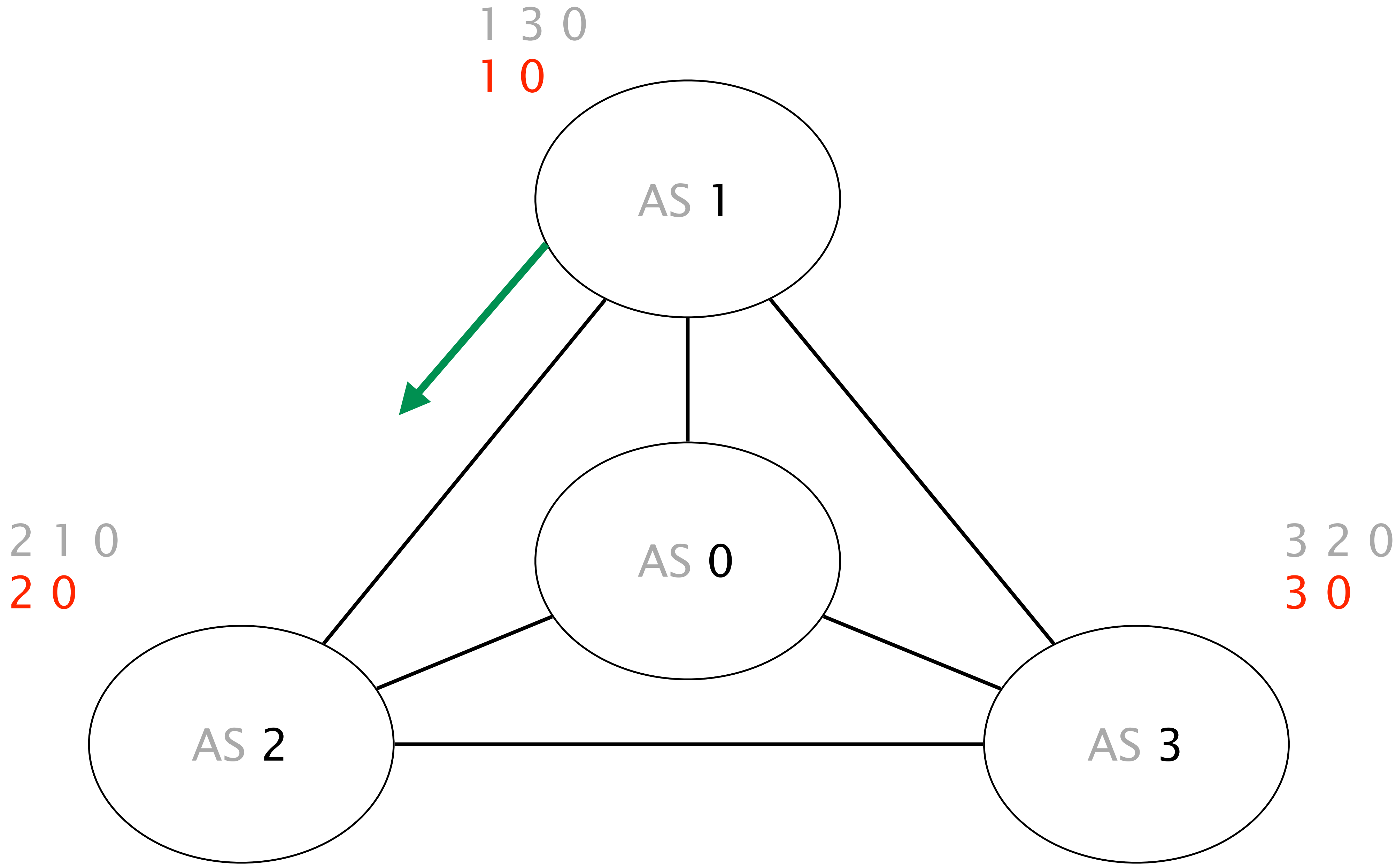

1 3 0
1 0
AS 1
2 1 0
2 0
AS 0
3 2 0
3 0
AS 2
AS 3


preference list
1 prefers to reach 0
via 3 rather than directly
1 3 0
1 0
AS 1
AS 0
2 1 0
2 0
3 2 0
3 0
AS 2
AS 3

Initially, all ASes only know the direct route to 0

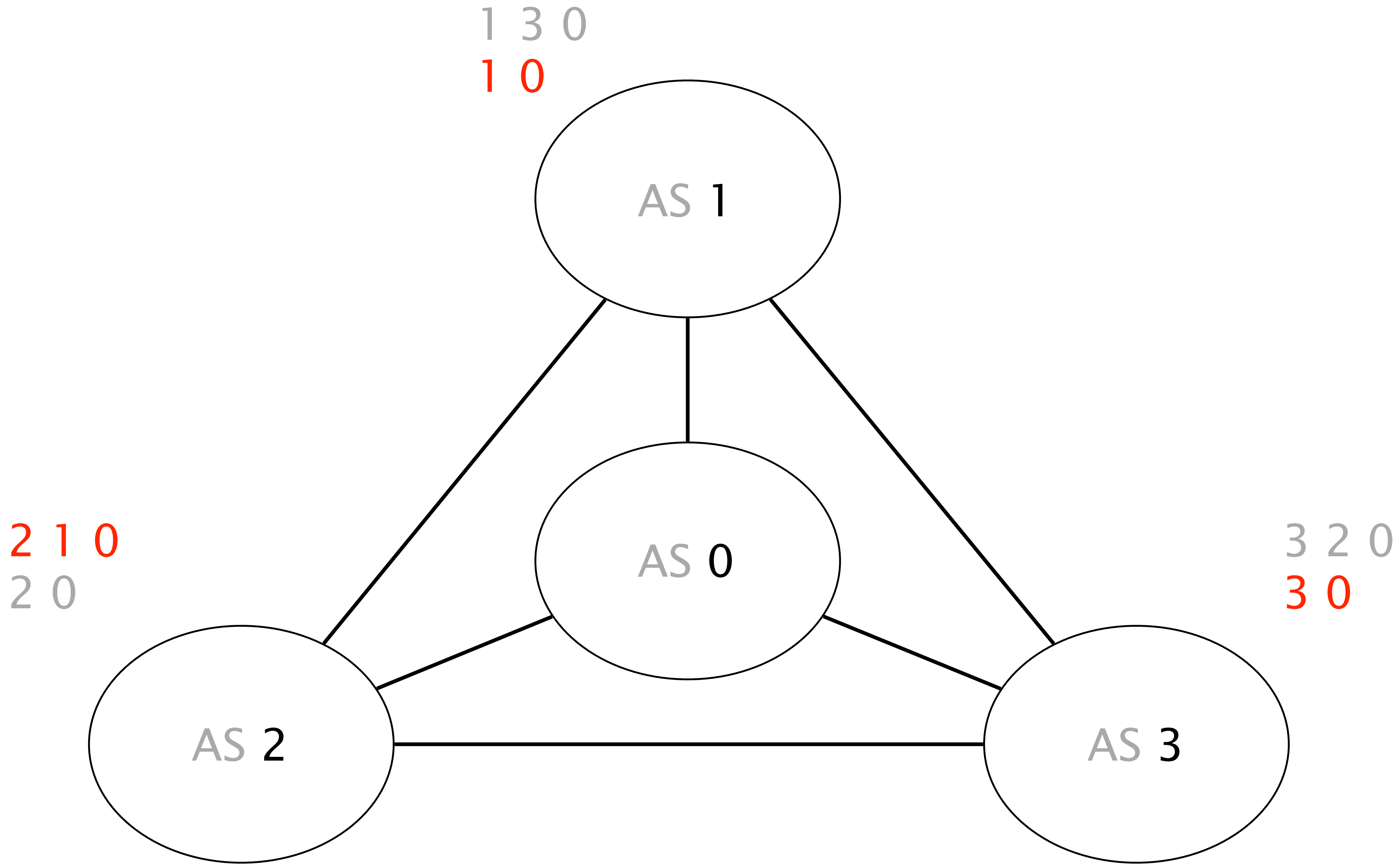


AS 1 advertises its path to AS 2

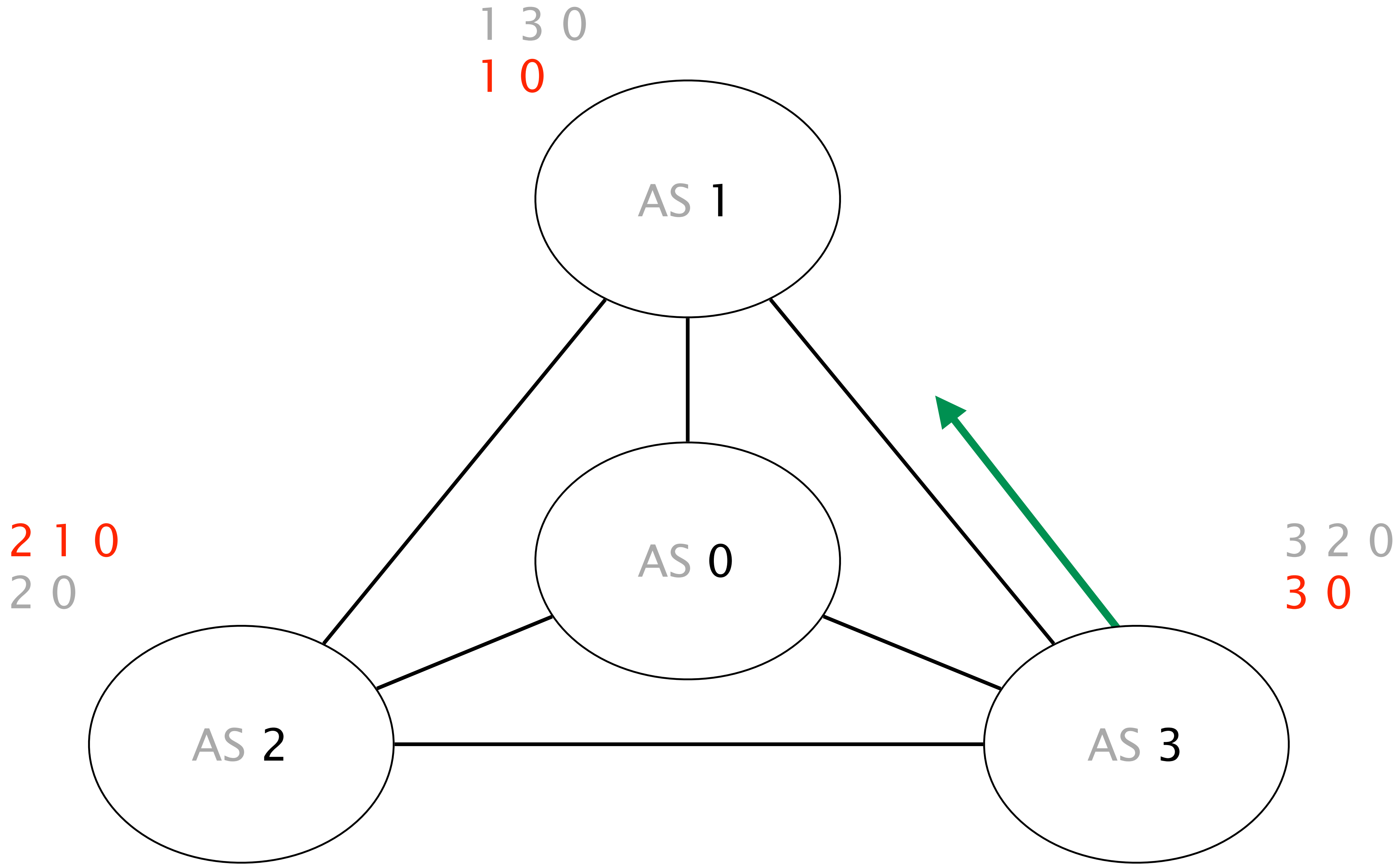


Upon reception,

AS 2 switches to 2 1 0 (preferred)

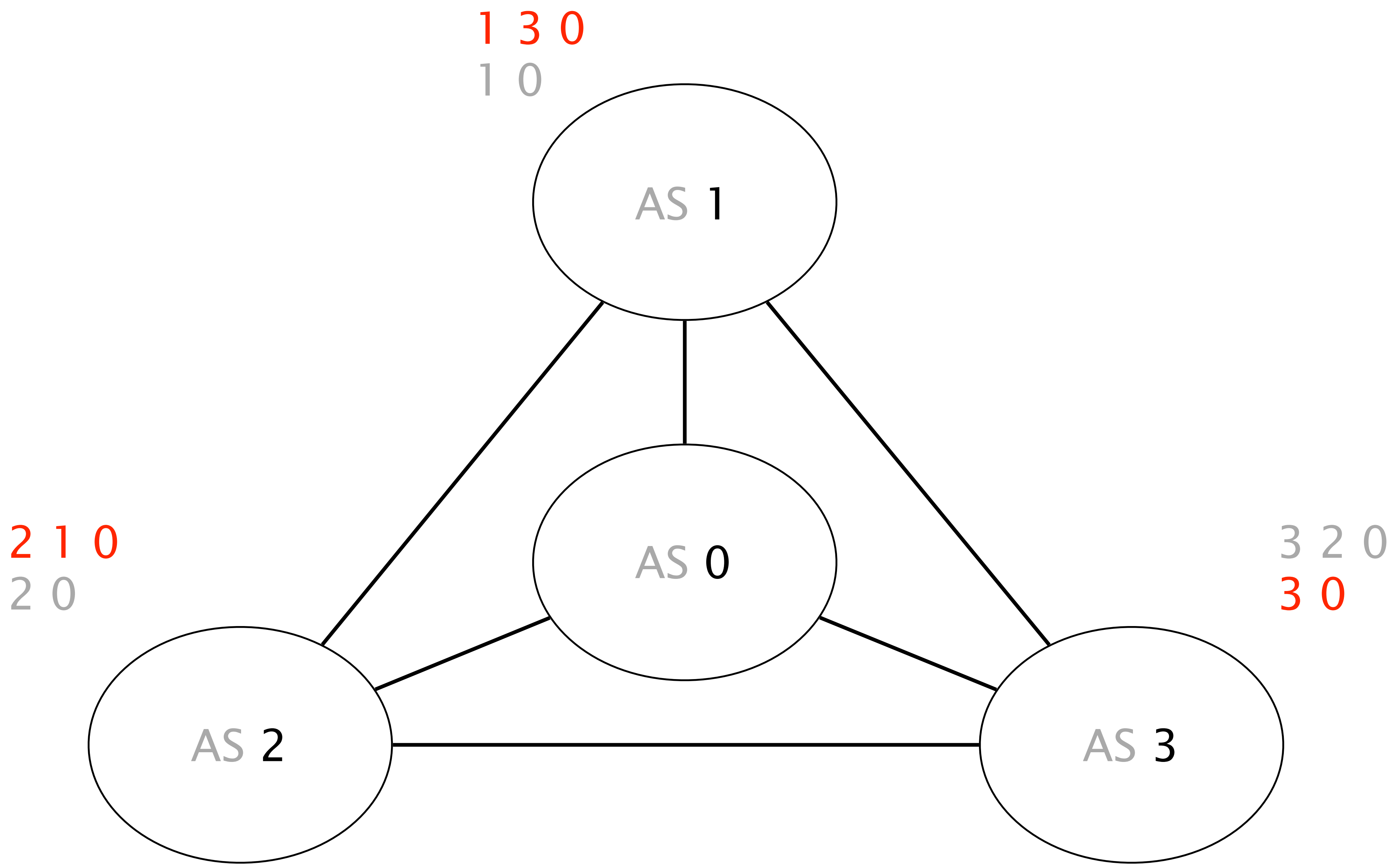


AS 3 advertises its path to AS 1

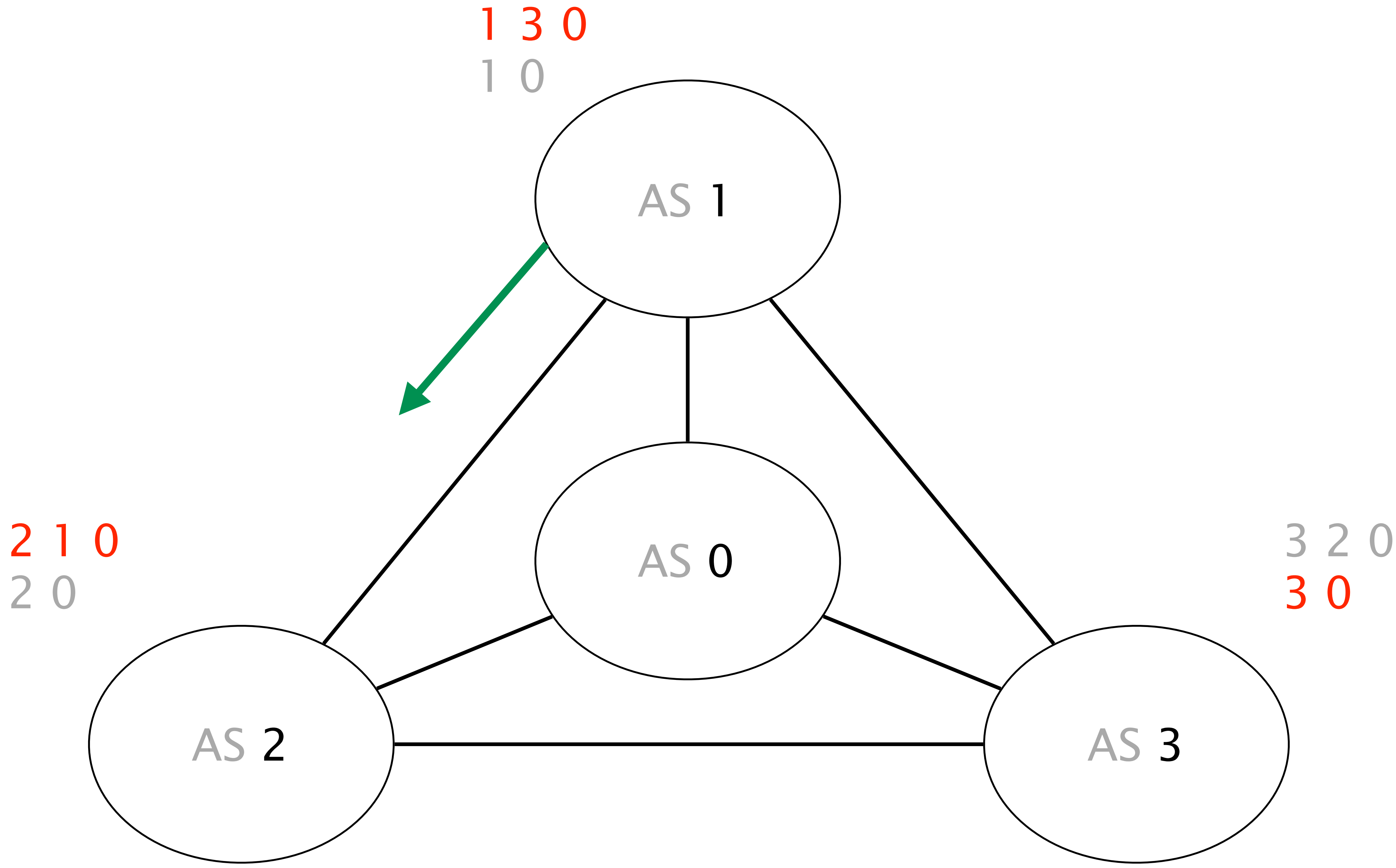


Upon reception,

AS 1 switches to 1 3 0 (preferred)

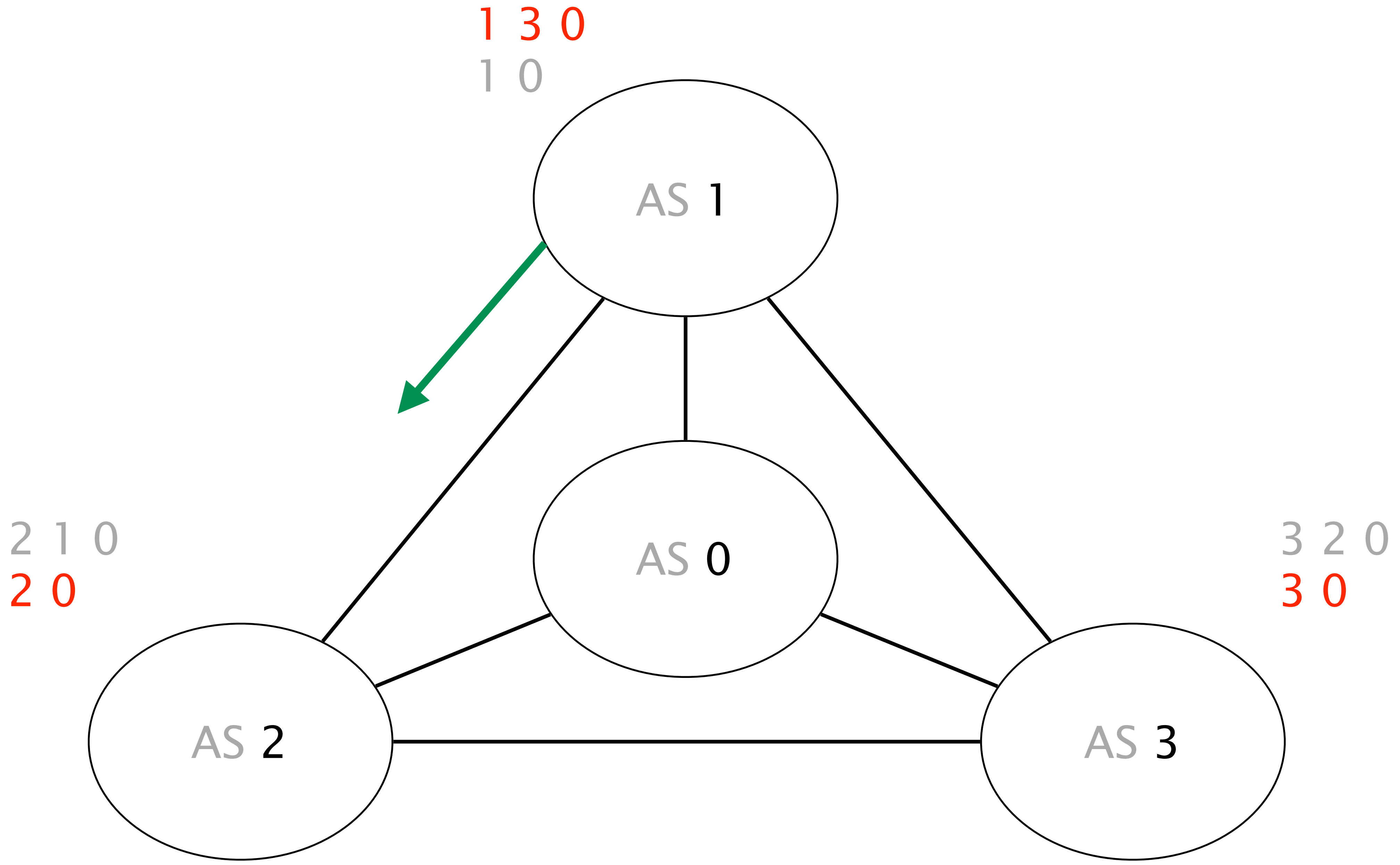


AS 1 advertises its new path 1 3 0 to AS 2

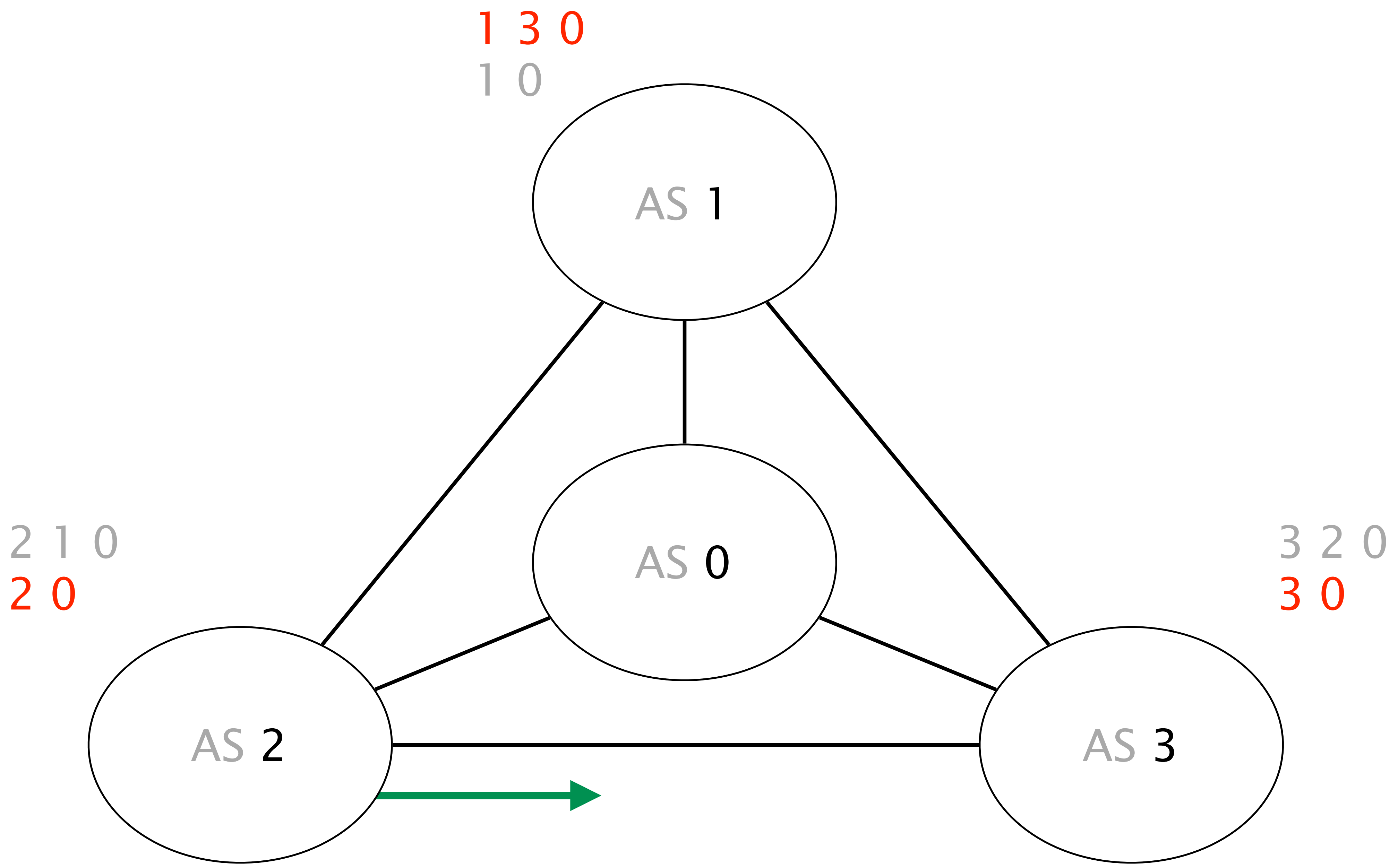


Upon reception,

AS 2 reverts back to its initial path 2 0

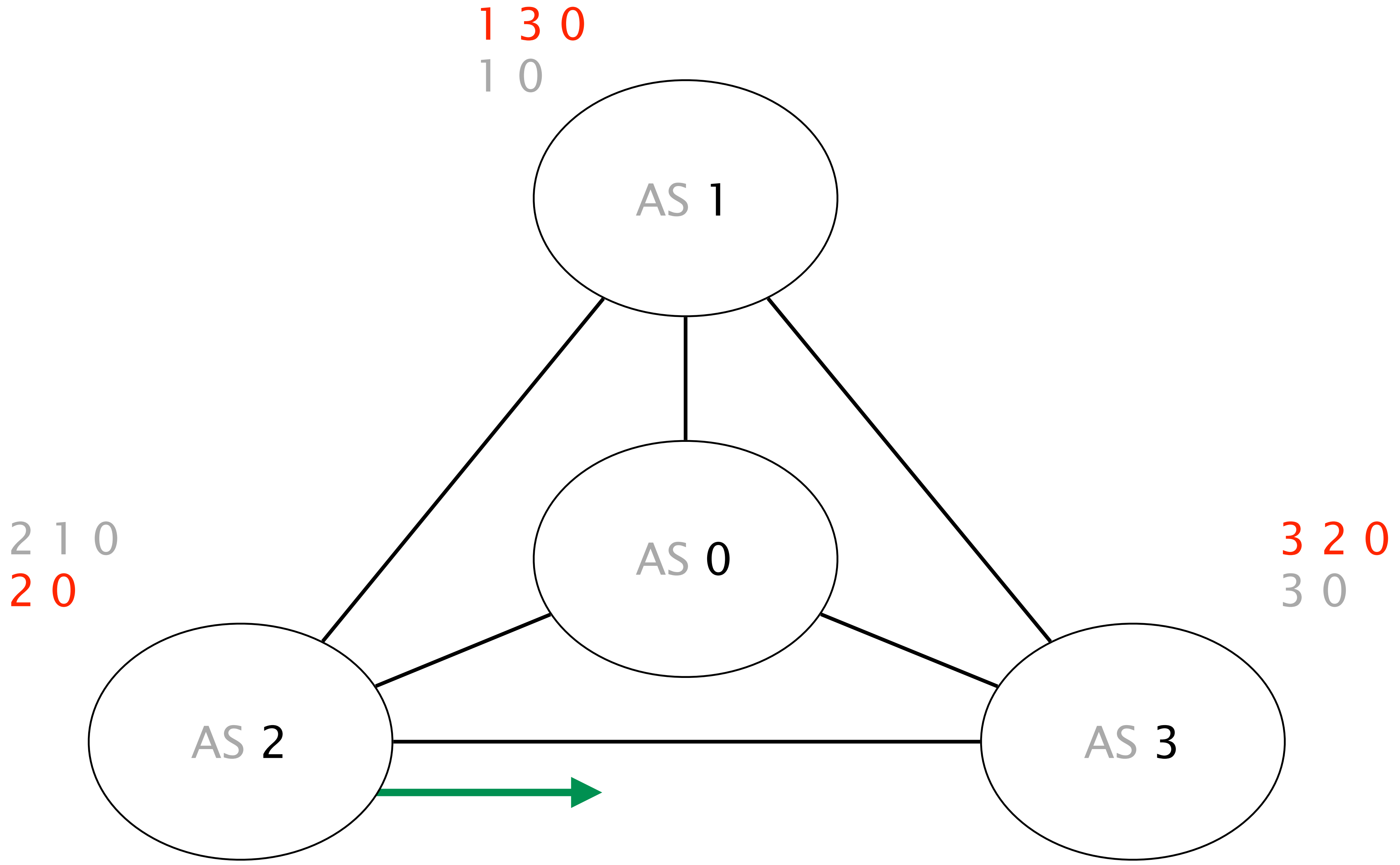


AS 2 advertises its path 2 0 to AS 3

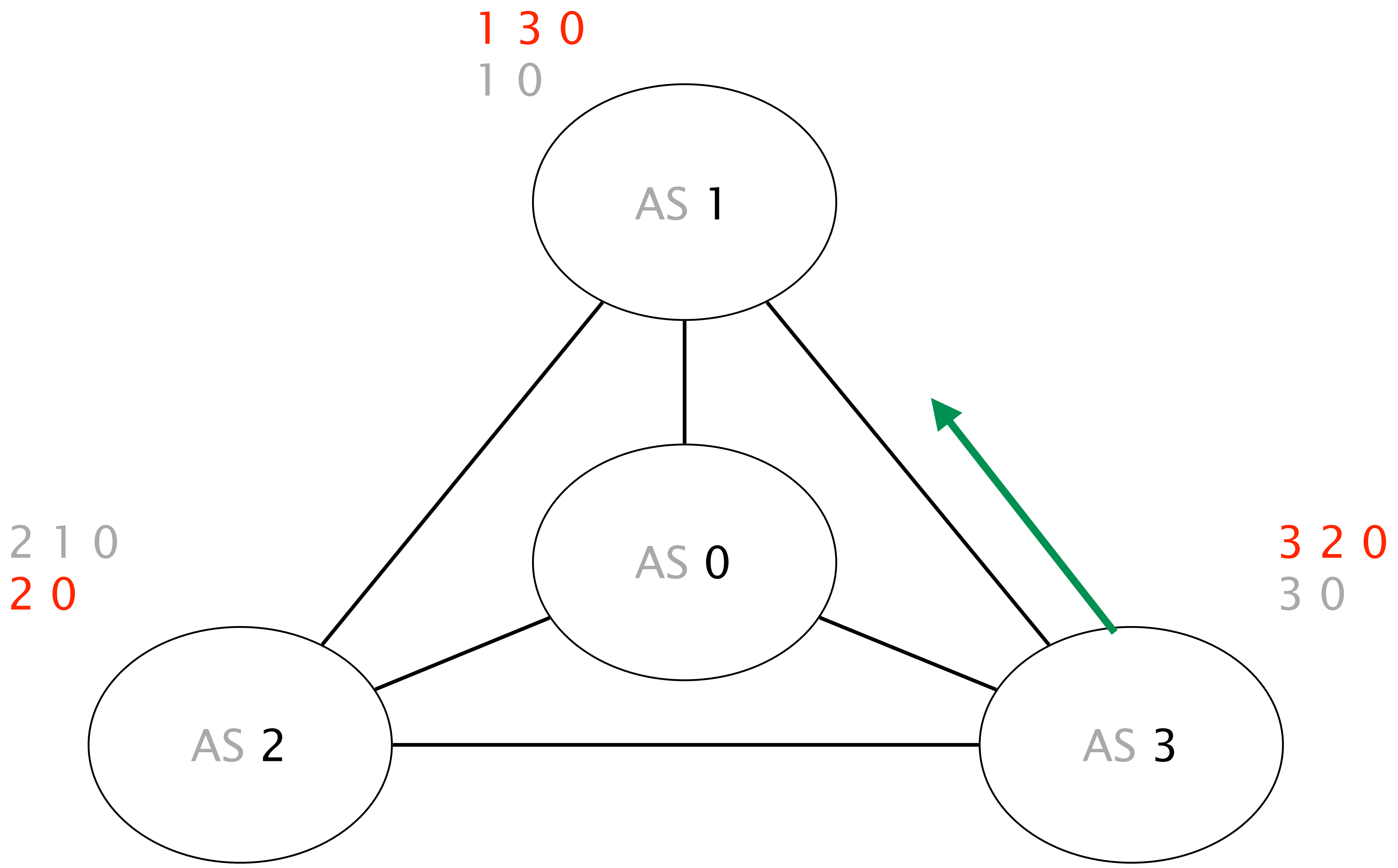


Upon reception,

AS 3 switches to 3 2 0 (preferred)

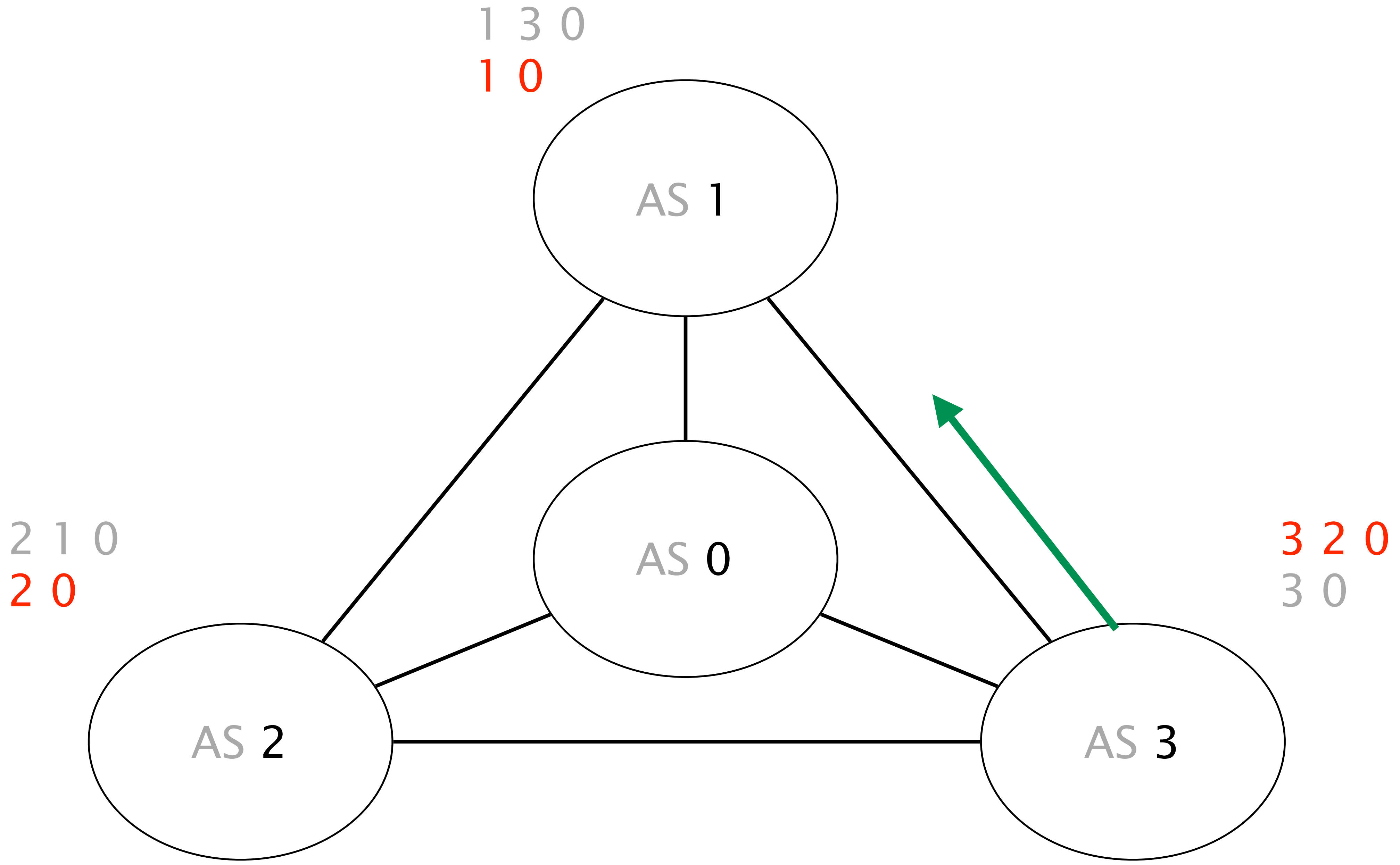


AS 3 advertises its new path 3 2 0 to AS 1

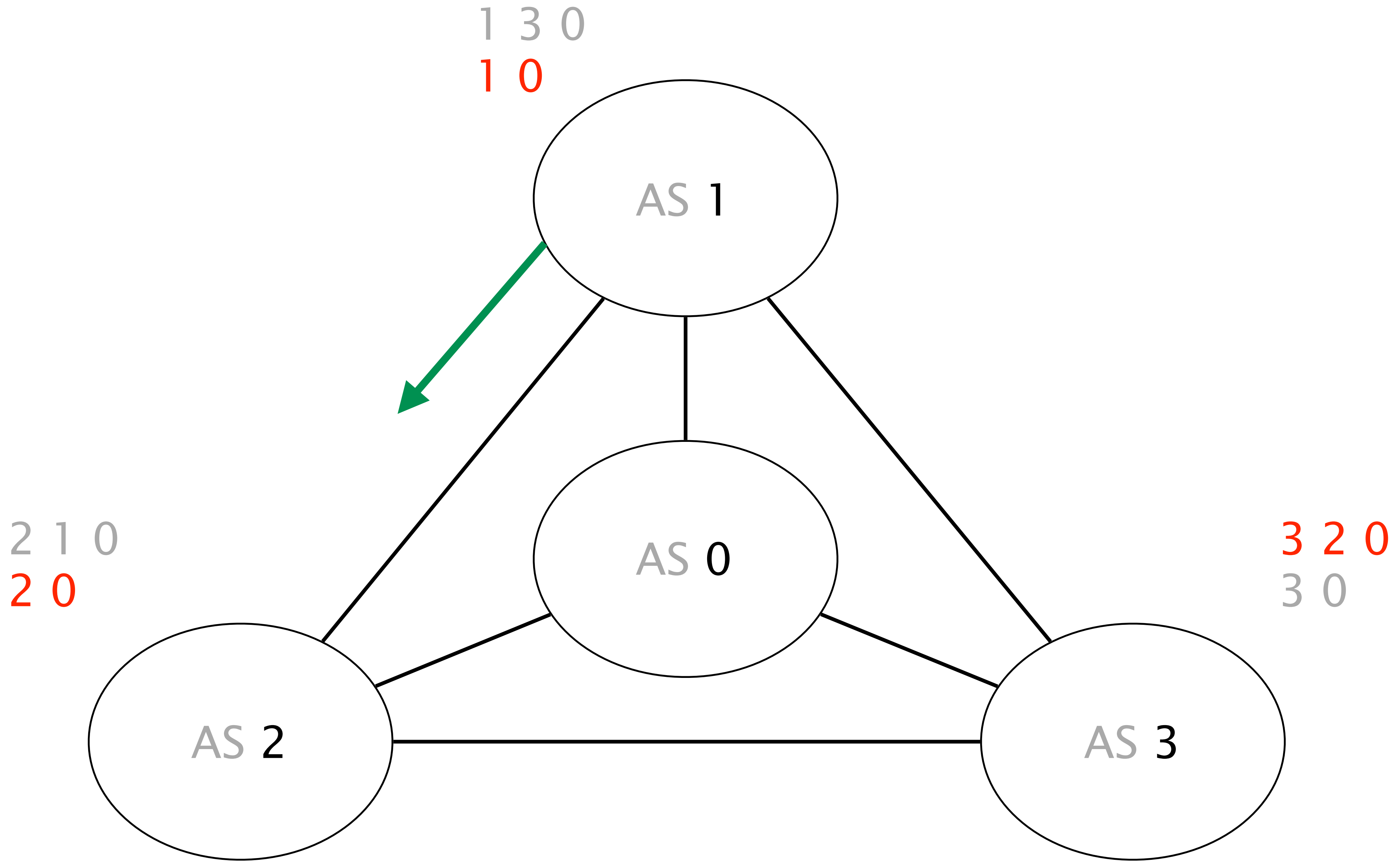


Upon reception,

AS 1 reverts back to 1 0 (initial path)

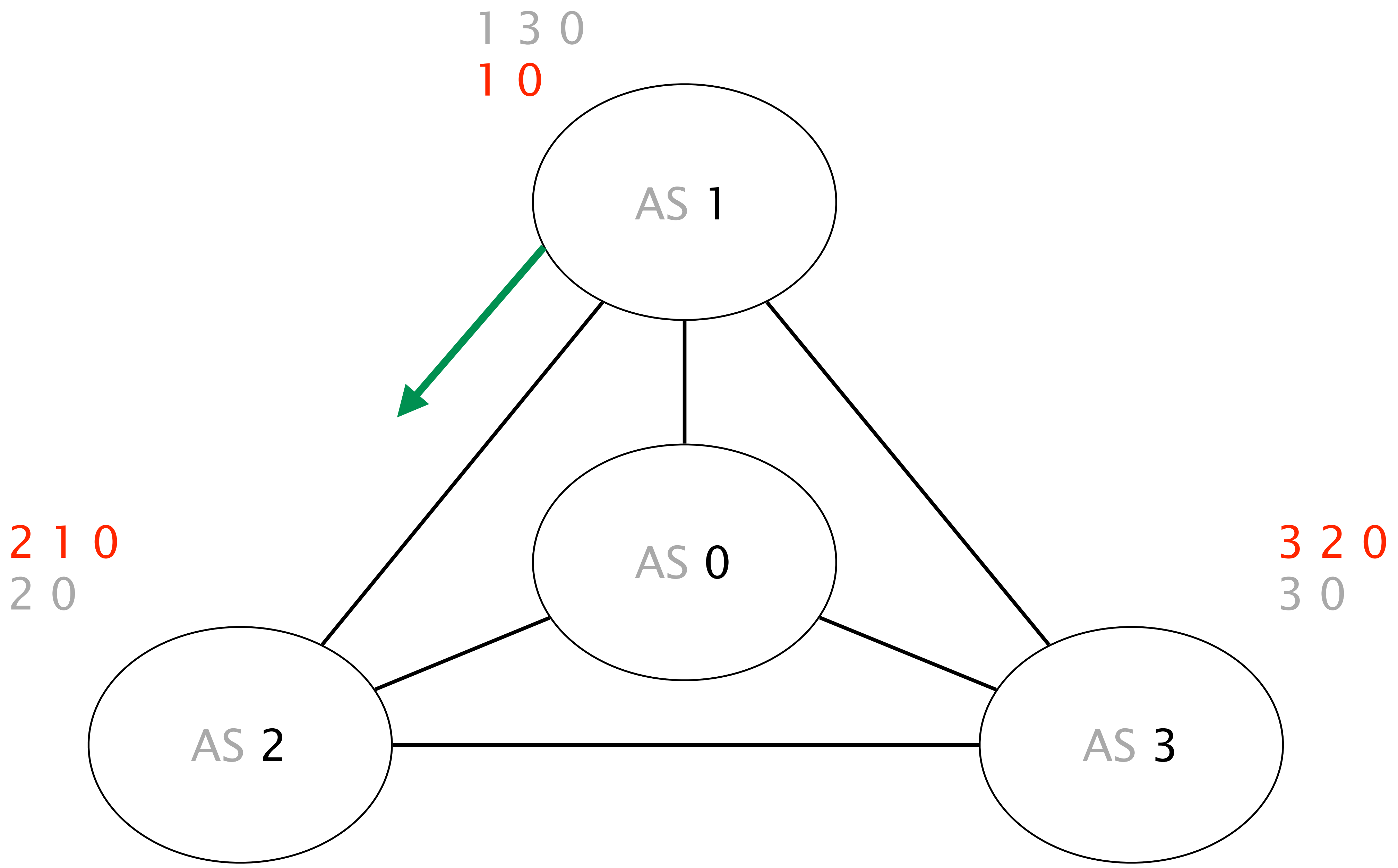


AS 1 advertises its new path 1 0 to AS 2

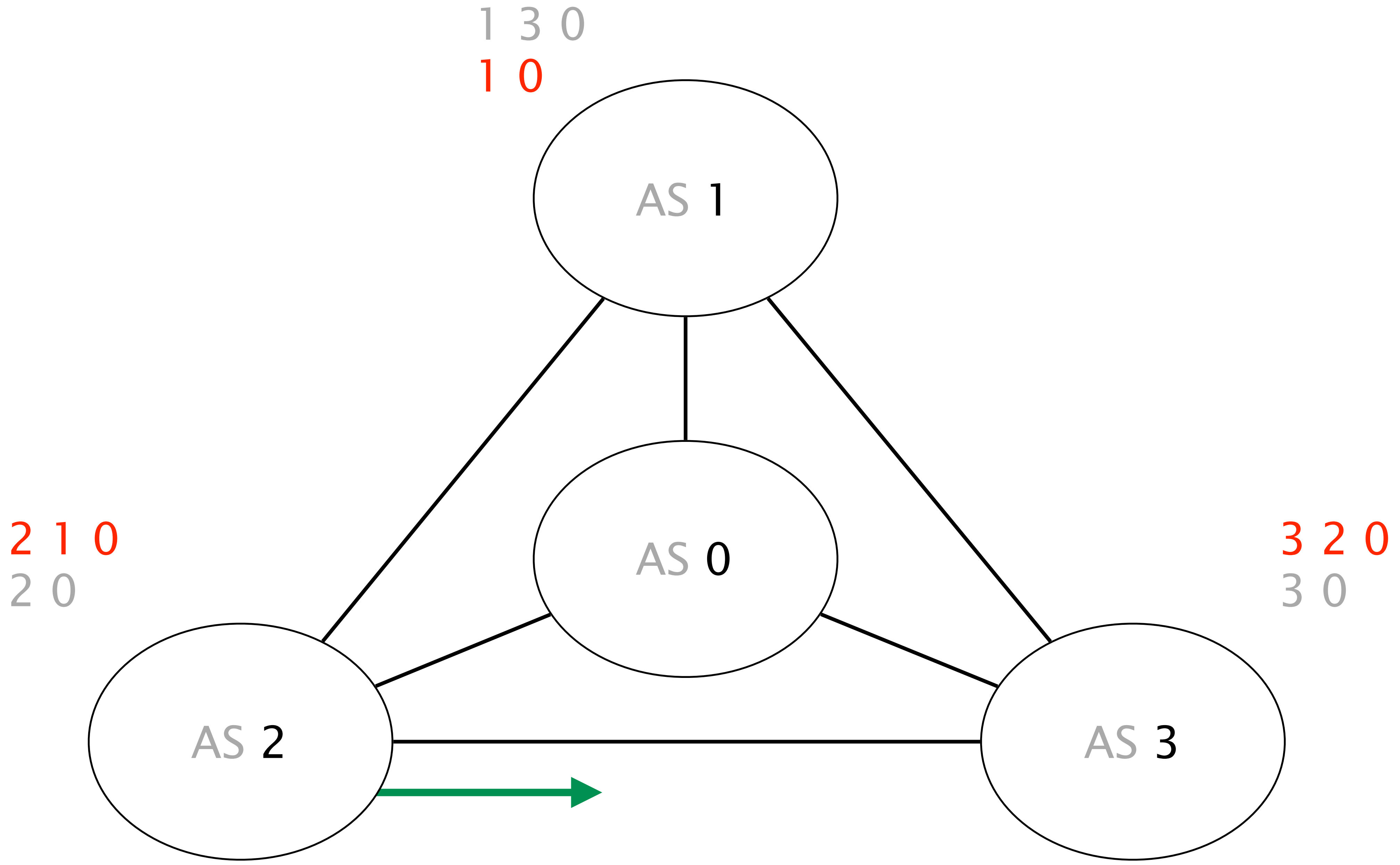


Upon reception,

AS 2 switches to 2 1 0 (preferred)

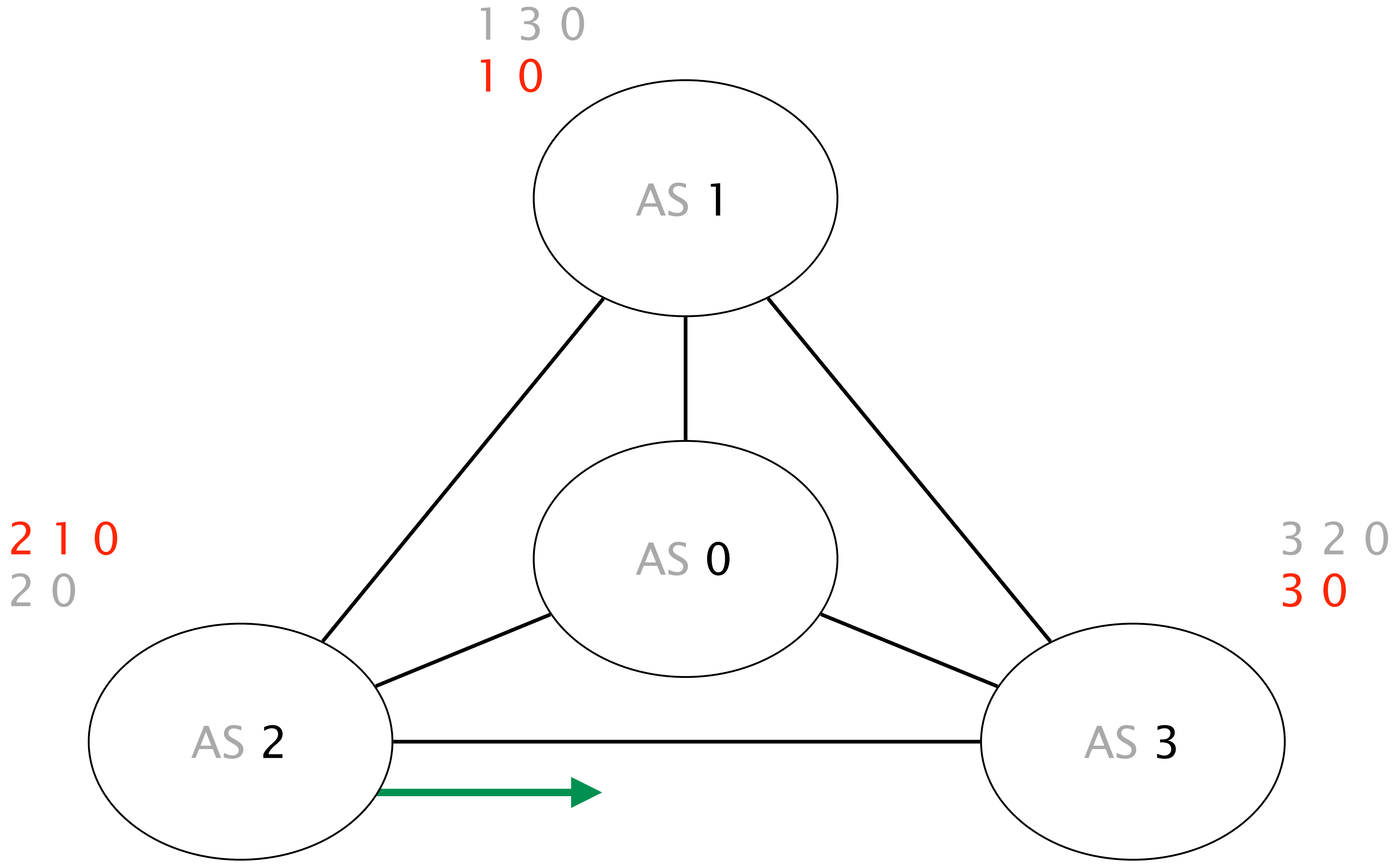


AS 2 advertises its new path 2 1 0 to AS 3

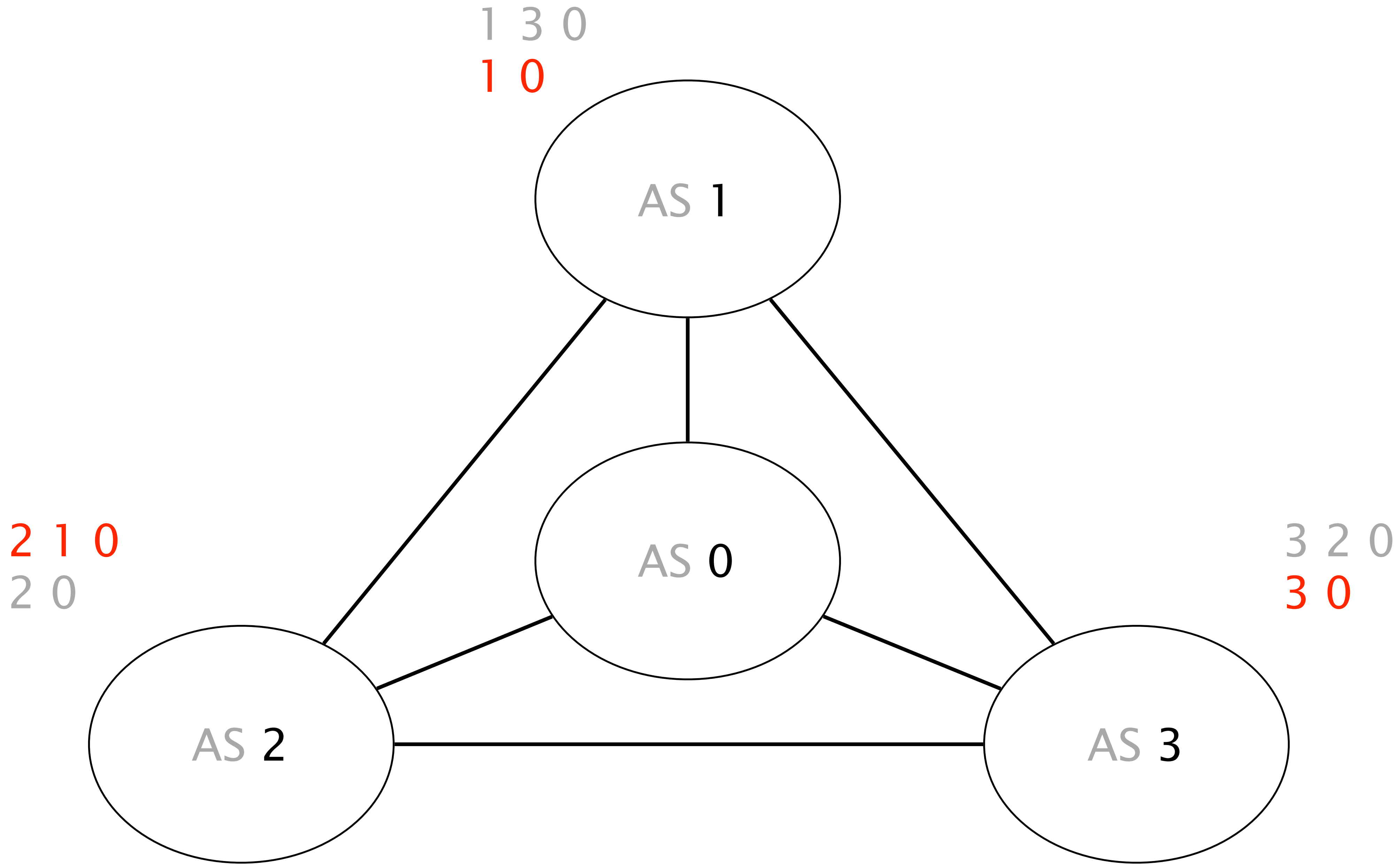


Upon reception,

AS 3 switches to its initial path 3 0



We are back where we started, from there on, the oscillation will continue forever



Ivanbever — R1 — R1 — ssh — 92×22

```
R1>
R1>
R1>
R1>
R1>
R1>
R1>
R1>
R1>
R1>
R1>
R1>
R1>
R1>
R1>
R1>
R1>
R1>
R1>
R1>
R1>
R1>
```

Ivanbever — R2 — R2 — ssh — 92×21

```
R2>
R2>
R2>
R2>
R2>
R2>
R2>
R2>
R2>
R2>
R2>
R2>
R2>
R2>
R2>
R2>
R2>
R2>
R2>
R2>
R2>
```

Ivanbever — RA — RA — ssh — 52×6

```
RA(config-router)#
RA(config-router)#
RA(config-router)#
RA(config-router)#
RA(config-router)#
RA(config-router)#distance 80
```

Ivanbever — RB — RB — ssh — 52×7

```
RB(config-router)#
RB(config-router)#
RB(config-router)#
RB(config-router)#
RB(config-router)#
RB(config-router)#
RB(config-router)#distance 80
```

Policy oscillations and multiple state states are a direct consequence of policy autonomy

ASes are free to chose and advertise any paths they want

network stability argues against this

Guaranteeing the absence of oscillations is hard

even when you know all the policies!

Guaranteeing the absence of oscillations is hard

even when you know all the policies!

How come?

Theorem

Computationally, a BGP network is as "powerful" as

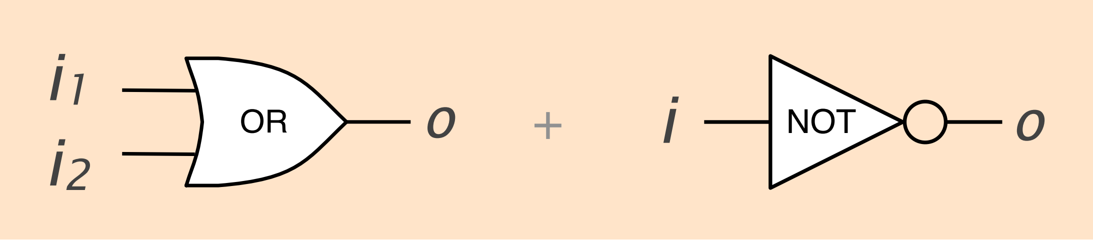
see "Using Routers to Build Logic Circuits: How Powerful is BGP?"

How do you prove such a thing?

How do you prove such a thing?

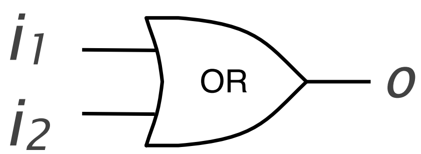
Easy, you build a computer using BGP…

Logic gates

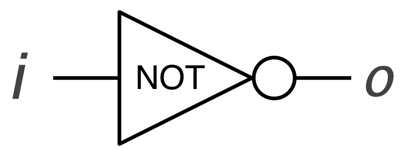


Logic gates

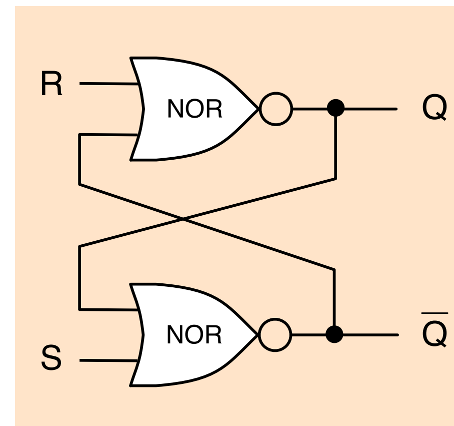
Memory



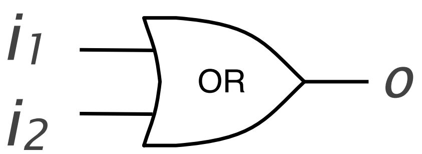
+



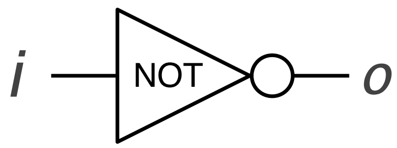
+



Logic gates

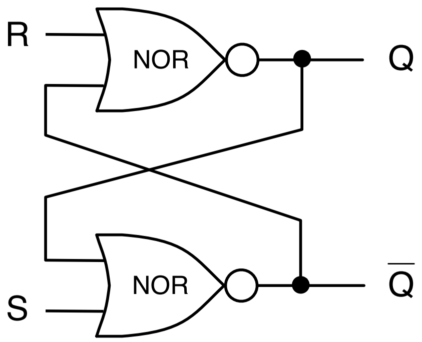


+



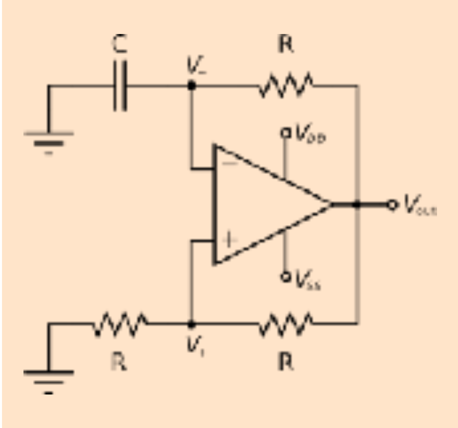
+

Memory

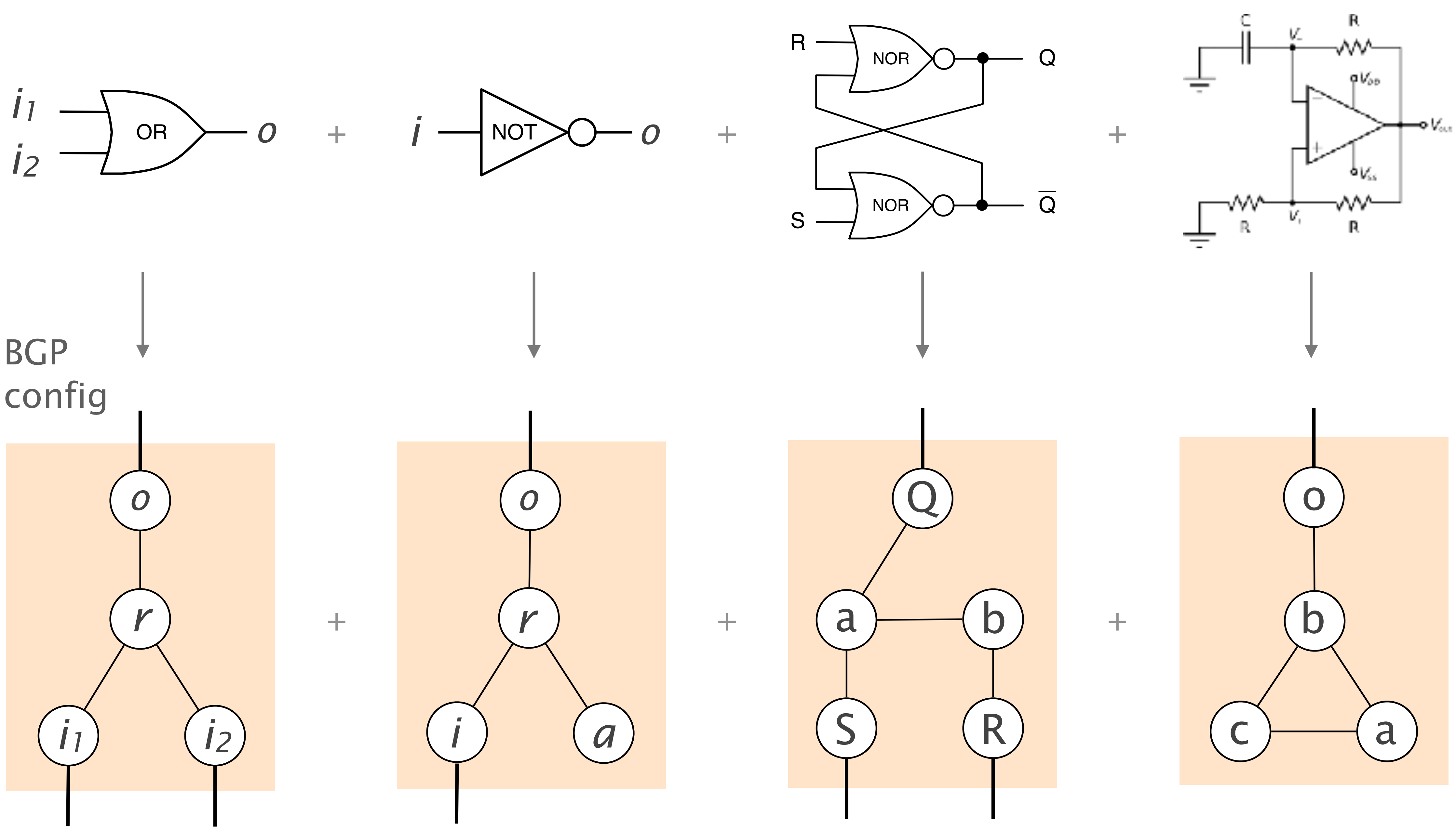


+

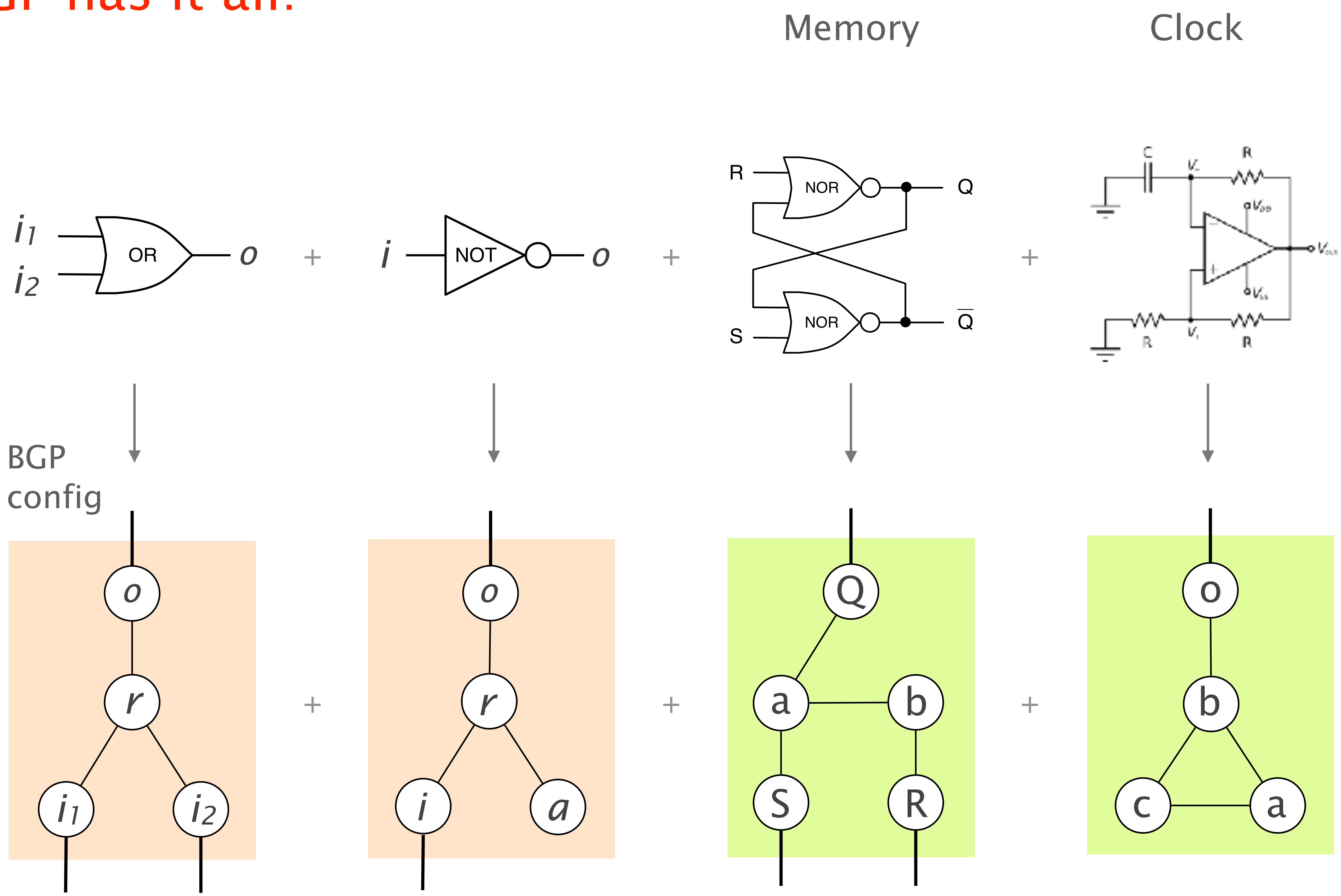
Clock



BGP has it all!



BGP has it all!

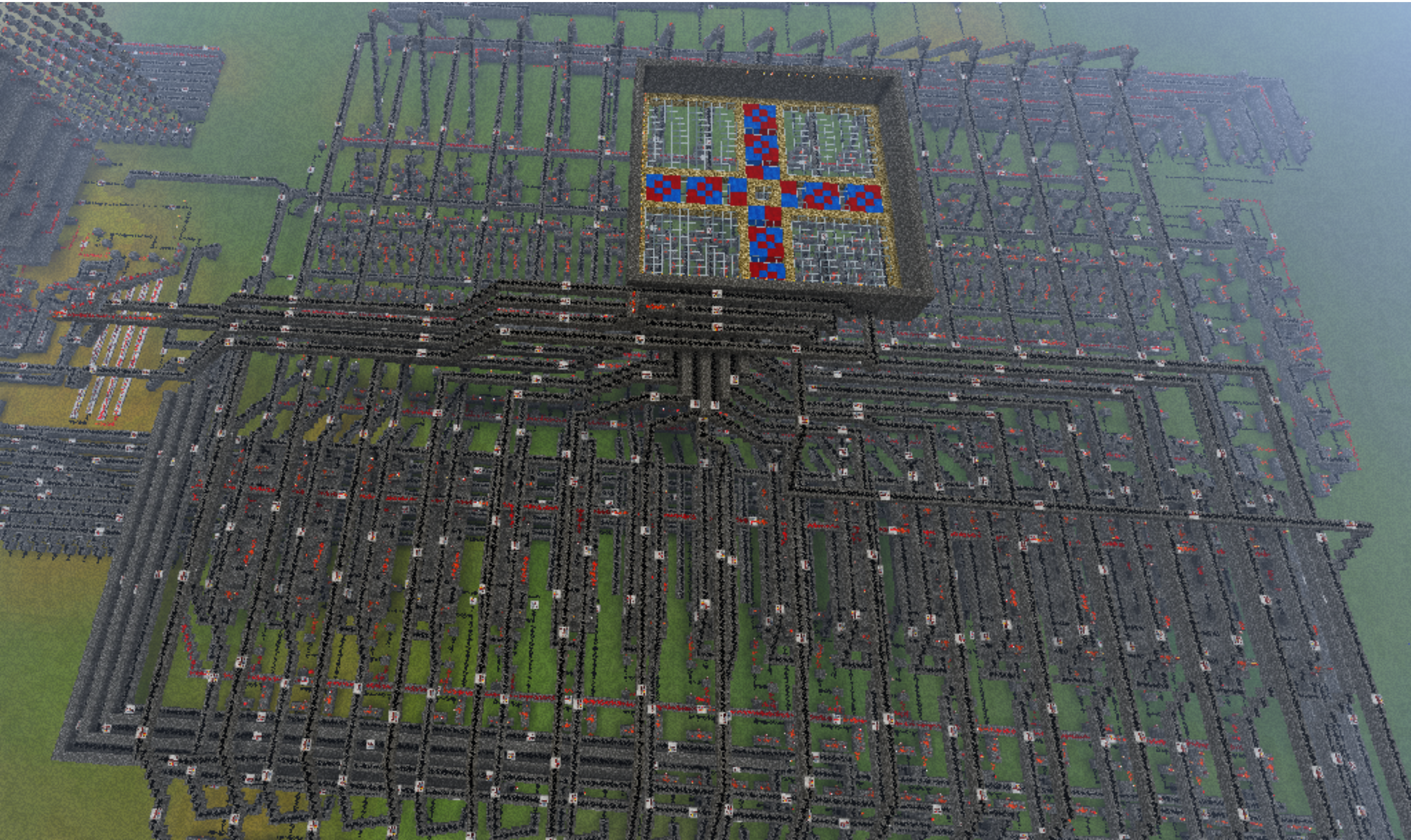


famous incorrect BGP configurations (Griffin et al.)

Instead of using Minecraft

for building a computer… use BGP!

Hack III, Minecraft's largest computer to date



Together, BGP routers form

the largest computer in the world!

Router-level view of the Internet, OPTE project

Checking BGP correctness is as hard as checking a general program

Theorem 1	Determining whether a finite BGP network converges is PSPACE-hard
Theorem 2	BGP has the same computing power as a Turing Machine

In practice though,

BGP does not oscillate “that” often

known as “Gao-Rexford” rules

Theorem

If all AS policies follow the cust/peer/provider rules,
BGP is guaranteed to converge

Intuition

Oscillations require “preferences cycles”
which make no economical sense

Problems

Reachability

Security

Non-determinism

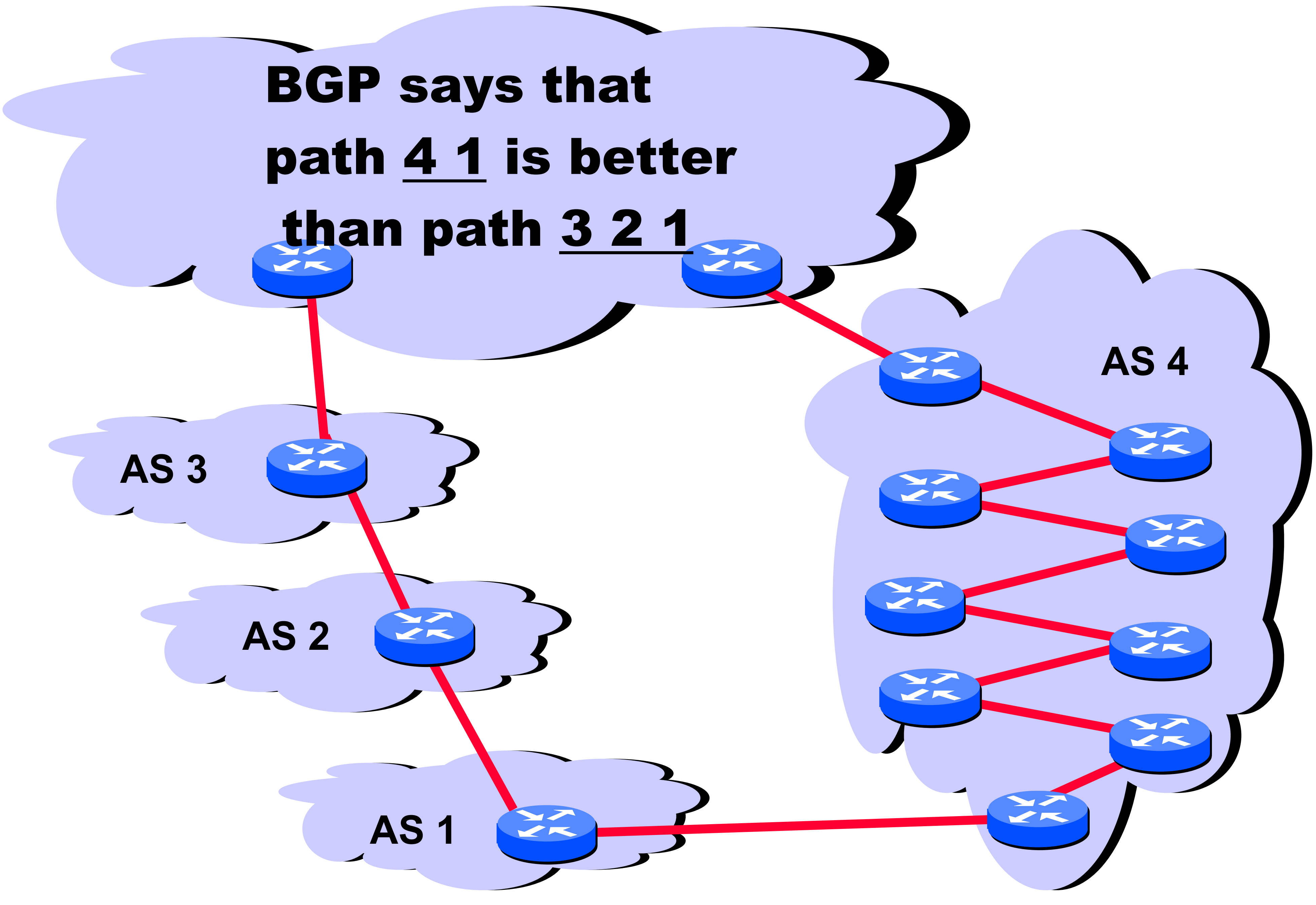
Convergence

Performance

Anomalies

Relevance

BGP path selection is mostly economical,
not based on accurate performance criteria



Problems

Reachability

Security

Non-determinism

Convergence

Performance

Anomalies

Relevance

BGP configuration is hard to get right,
you probably understand why already

BGP is both "bloated" and underspecified

lots of knobs and (sometimes, conflicting) interpretations

BGP is often manually configured

humans make mistakes, often

BGP abstraction is fundamentally flawed

disjoint, router-based configuration to effect AS-wide policy

“Human factors are responsible

for 50% to 80% of network outages”

Juniper Networks, *What’s Behind Network Downtime?*, 2008

Problems

Reachability

Security

Non-determinism

Convergence

Performance

Anomalies

Relevance

The world of BGP policies is rapidly changing

ISPs are now eyeballs talking to content networks

e.g., Swisscom and Netflix/Spotify/YouTube

Transit becomes less important and less profitable

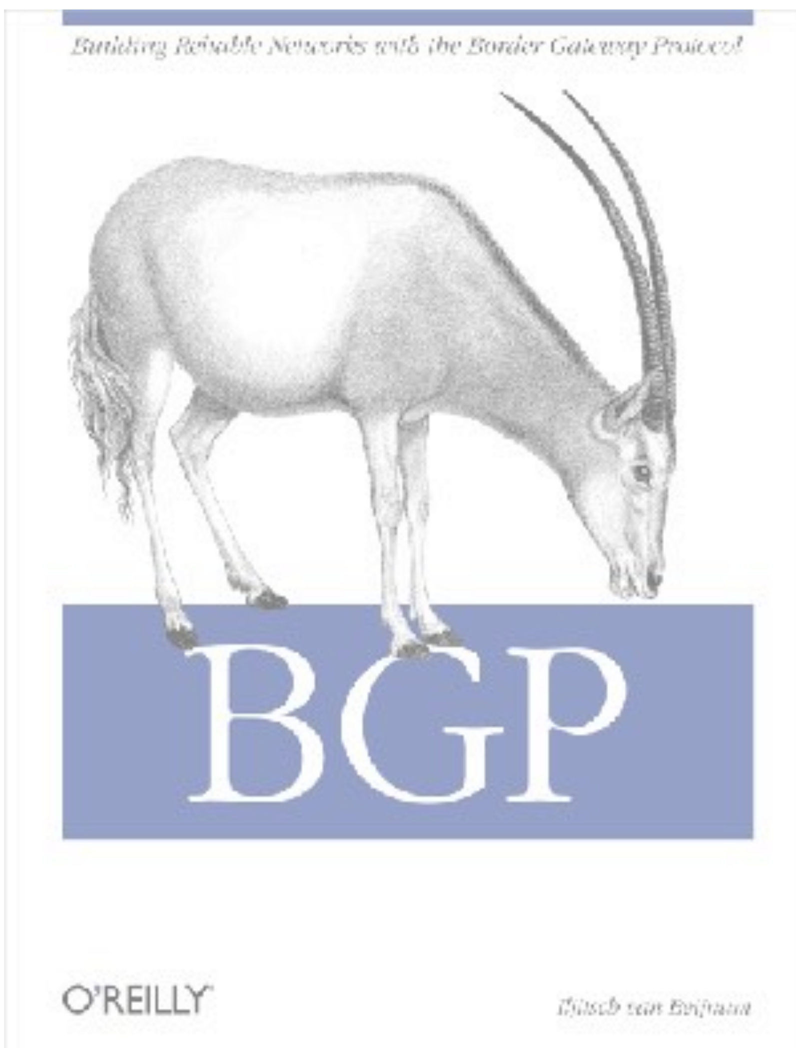
traffic move more and more to interconnection points

No systematic practices, yet

details of peering arrangements are private anyway

Border Gateway Protocol

policies and more



BGP Policies

Follow the Money

Protocol

How does it work?

Problems

security, performance, …

This week on

Communication Networks

Routing Security

[image source]

One can identify six basic security properties,

which also apply to routing security

confidentiality	concealment of information or resources
authenticity	identification & assurance of origin of info
integrity	trustworthiness of data in terms of unauthorized changes
availability	ability to use desired information or resource
non-repudiation	proof that a party indeed sent/receive info
access control	determine and enforce who is allowed to access to what resources (host, software, network…)

Routing security

attacks & mitigation

intra-domain routing	inter-domain routing
insider	in/outsider

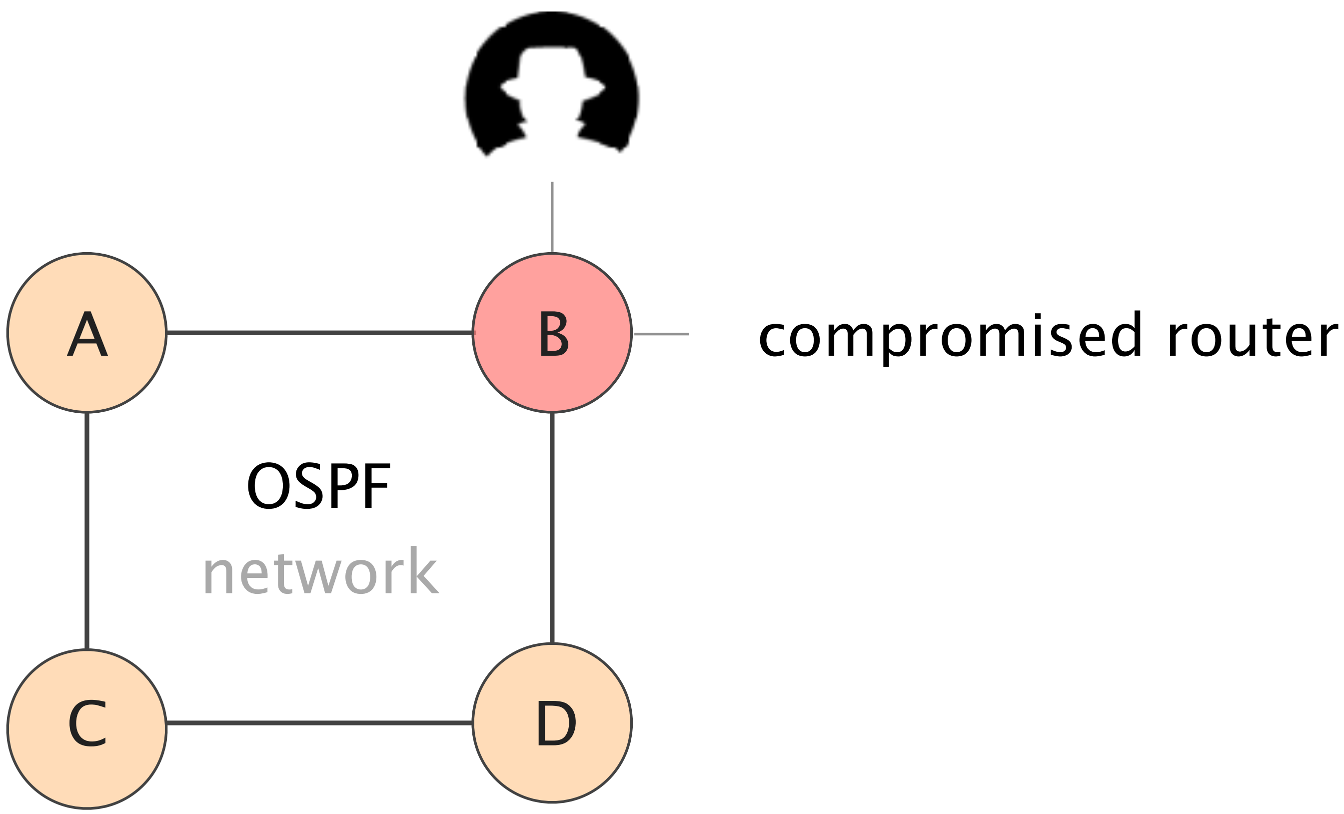
Routing security

attacks & mitigation

intra-domain routing	inter-domain routing
insider	in/outsider

To perform an attack on link-state protocols,

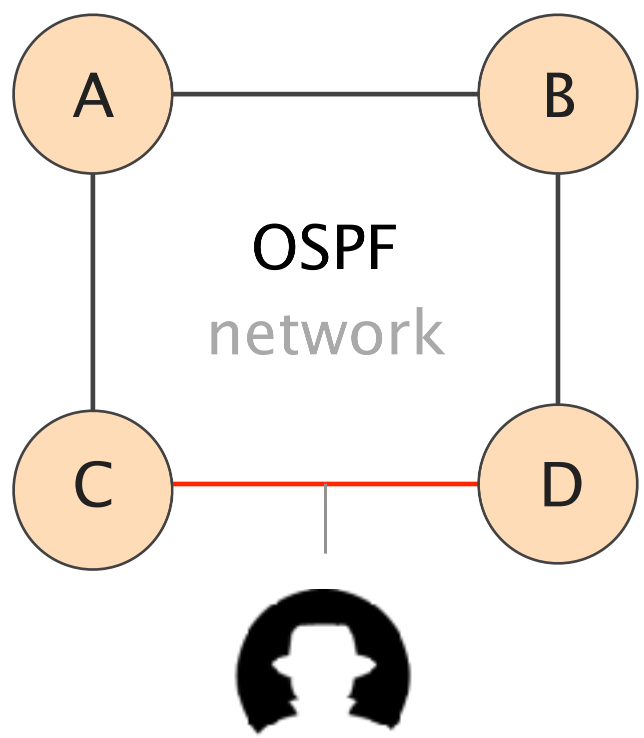
one only needs to compromise *one* router …



Why? Because link-state protocols rely on **flooding**

To perform an attack on link-state protocols,

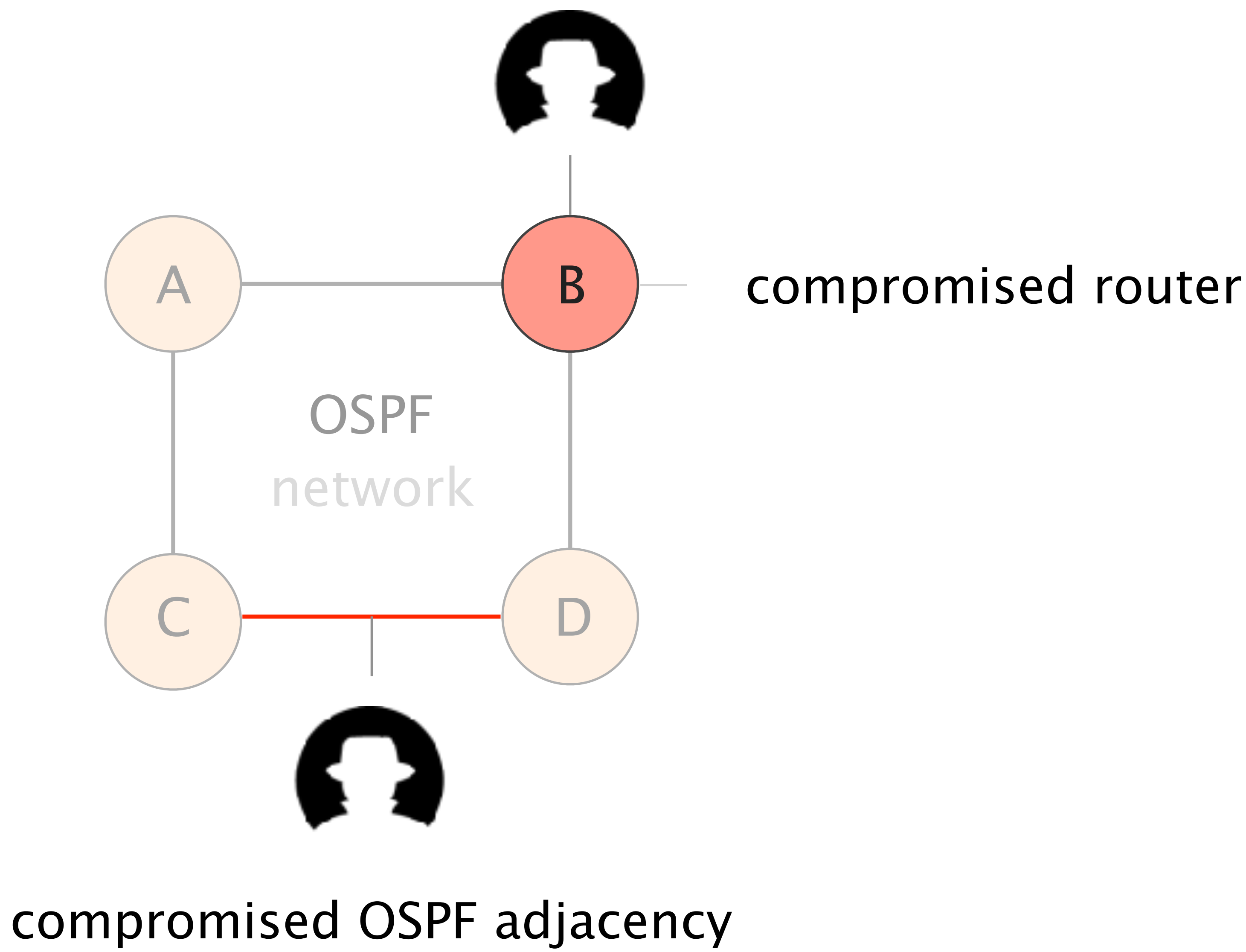
… or compromise one routing adjacency



compromised OSPF adjacency

attacker acts as a Man-in-the-Middle (MITM)

In both cases, the attacker obtains a complete network view & the ability to inject messages network-wide



Once you're owning the link-state protocol, what can you do? Unfortunately… plenty!

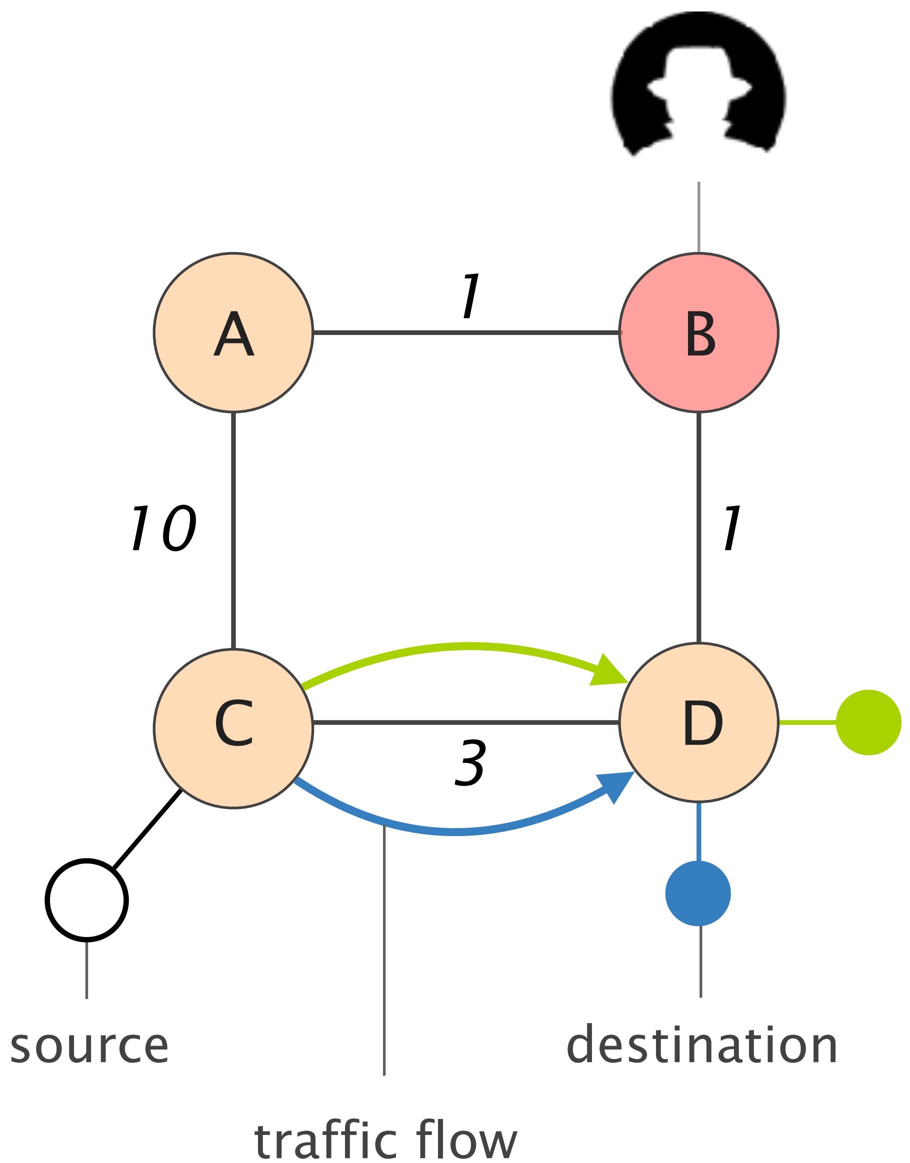
Most of the attacks on intra-domain routing aim at performing Denial-of-Service (DoS) or intercept traffic

Interception	eavesdrop on/drop/modify/inject/delay traffic steer traffic along paths controlled by the attacker
DoS	induce churn to overload the routers announce/withdraw at fast pace
	floods the routers link-state database inject thousands of prefixes
	induce congestion/higher delay steer traffic along fewer/low-throughput paths
	prevent reachability steer traffic along blackholes or loops

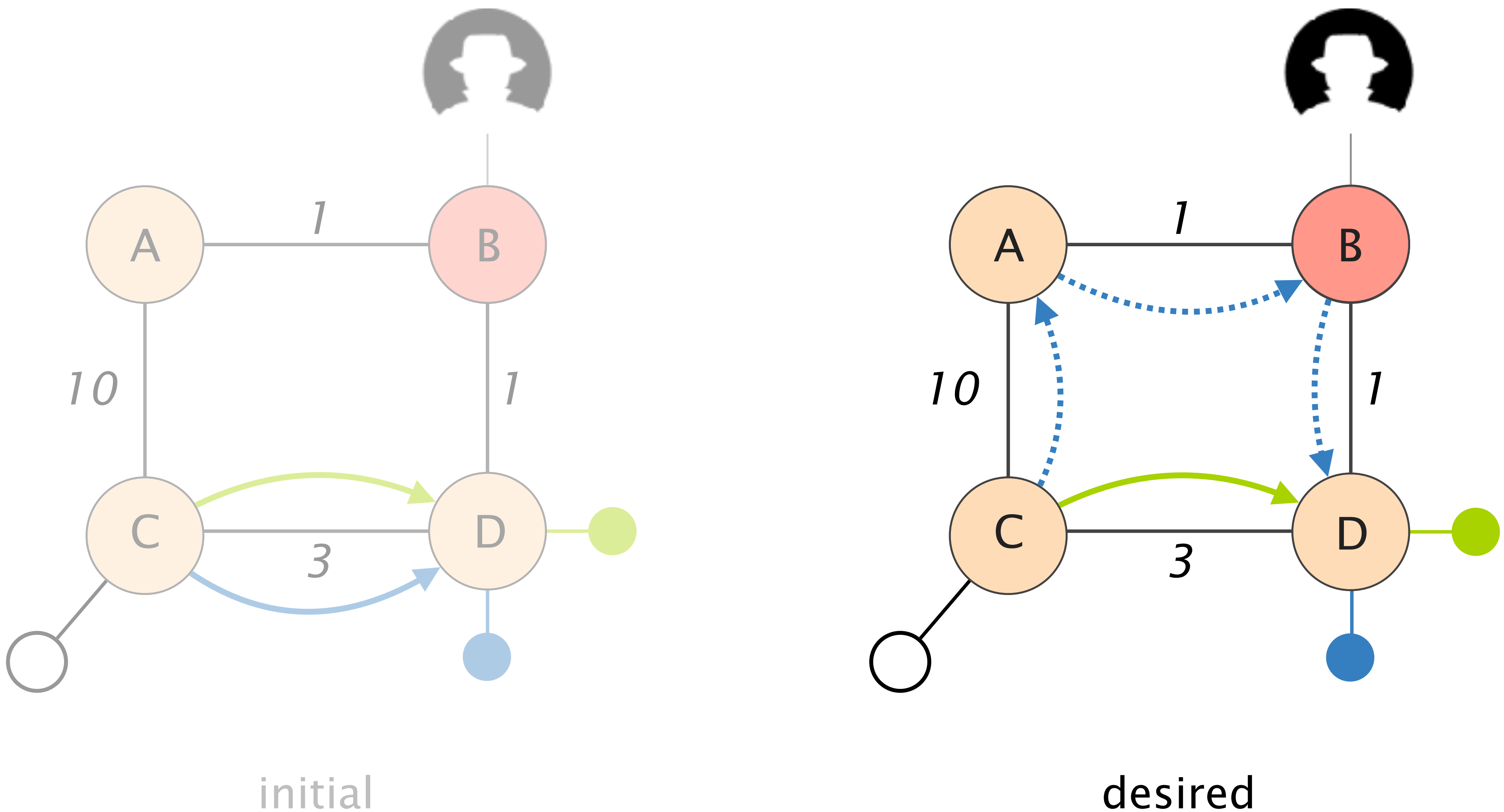
Most of the attacks on intra-domain routing aim at performing Denial-of-Service (DoS) or intercept traffic

Interception	eavesdrop on/drop/modify/inject/delay traffic steer traffic along paths controlled by the attacker
DoS	induce churn to overload the routers announce/withdraw at fast pace
	floods the routers link-state database inject thousands of prefixes
	induce congestion/higher delay steer traffic along fewer/low-throughput paths
	prevent reachability steer traffic along blackholes or loops

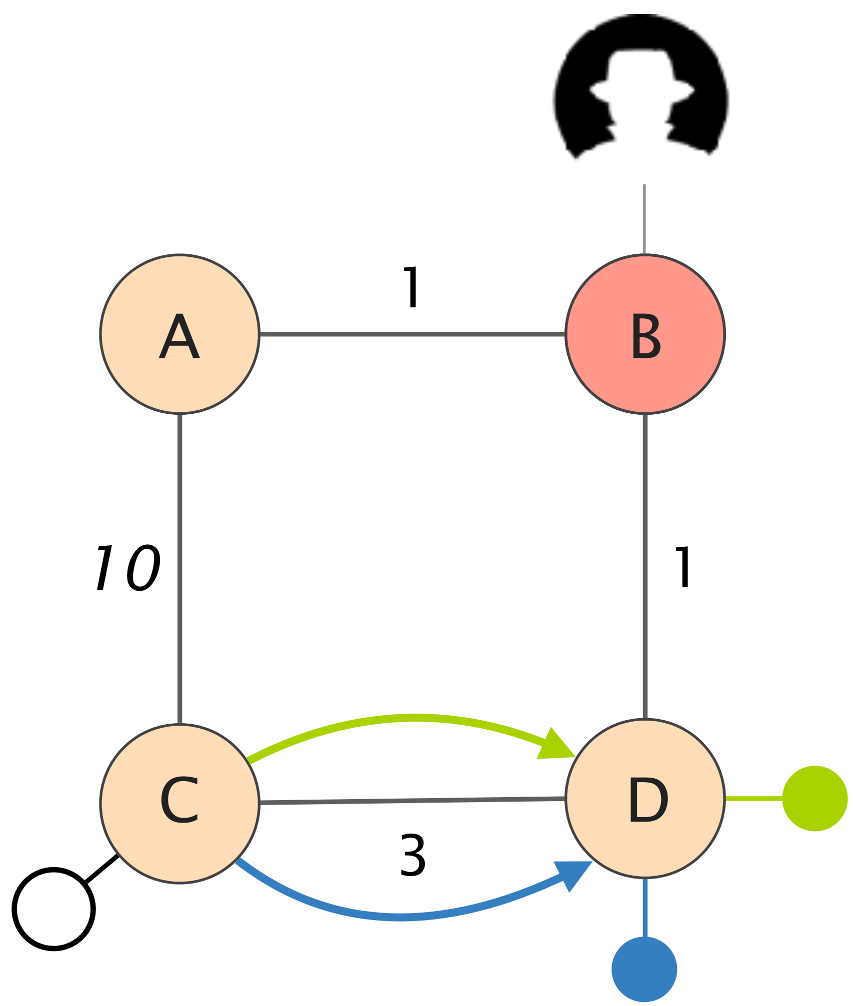
Consider a source connected to C that sends traffic to 2 destinations connected to D



The attacker wants to intercept traffic to the blue destination

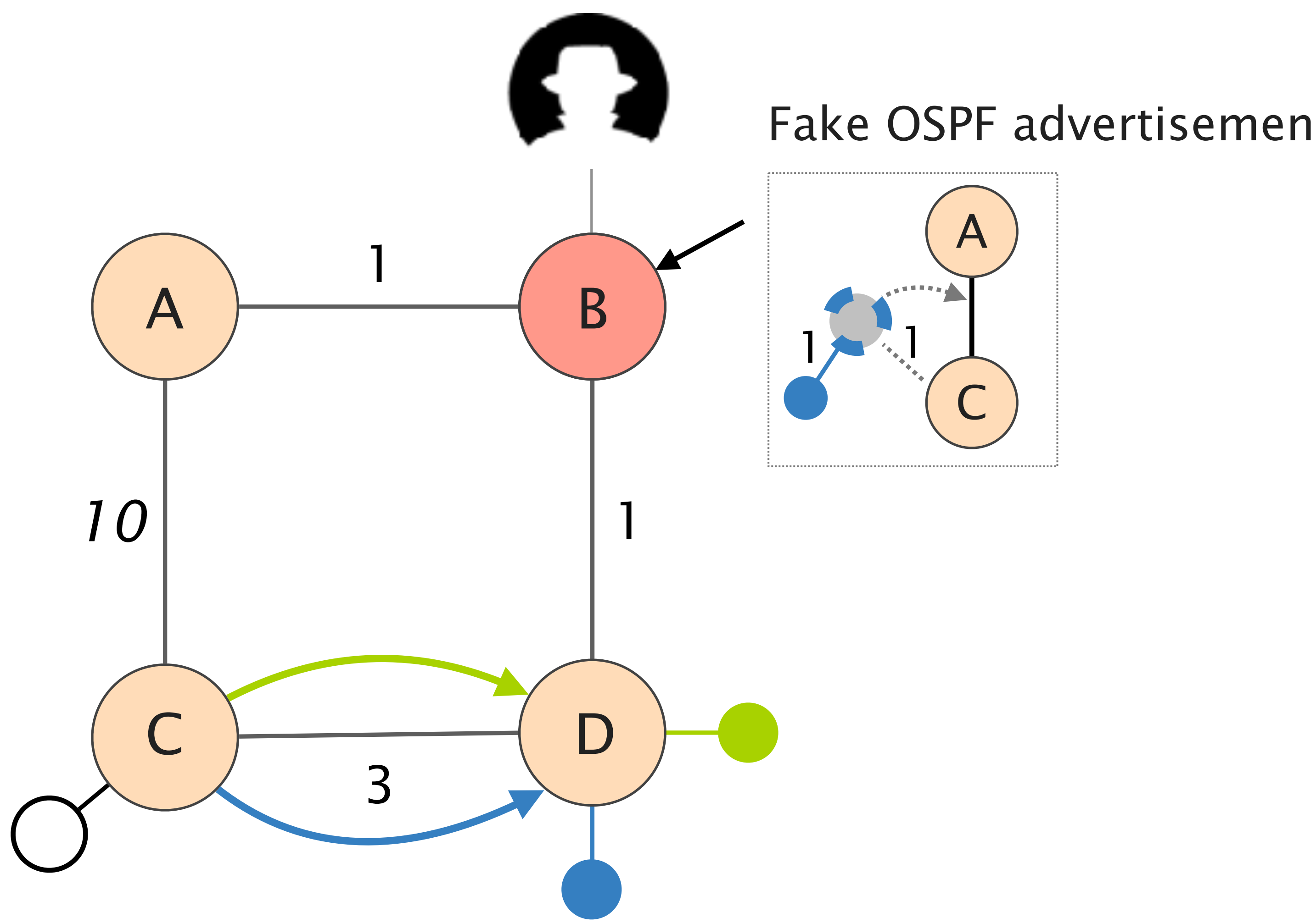


For that the attacker can “lie” to the routers

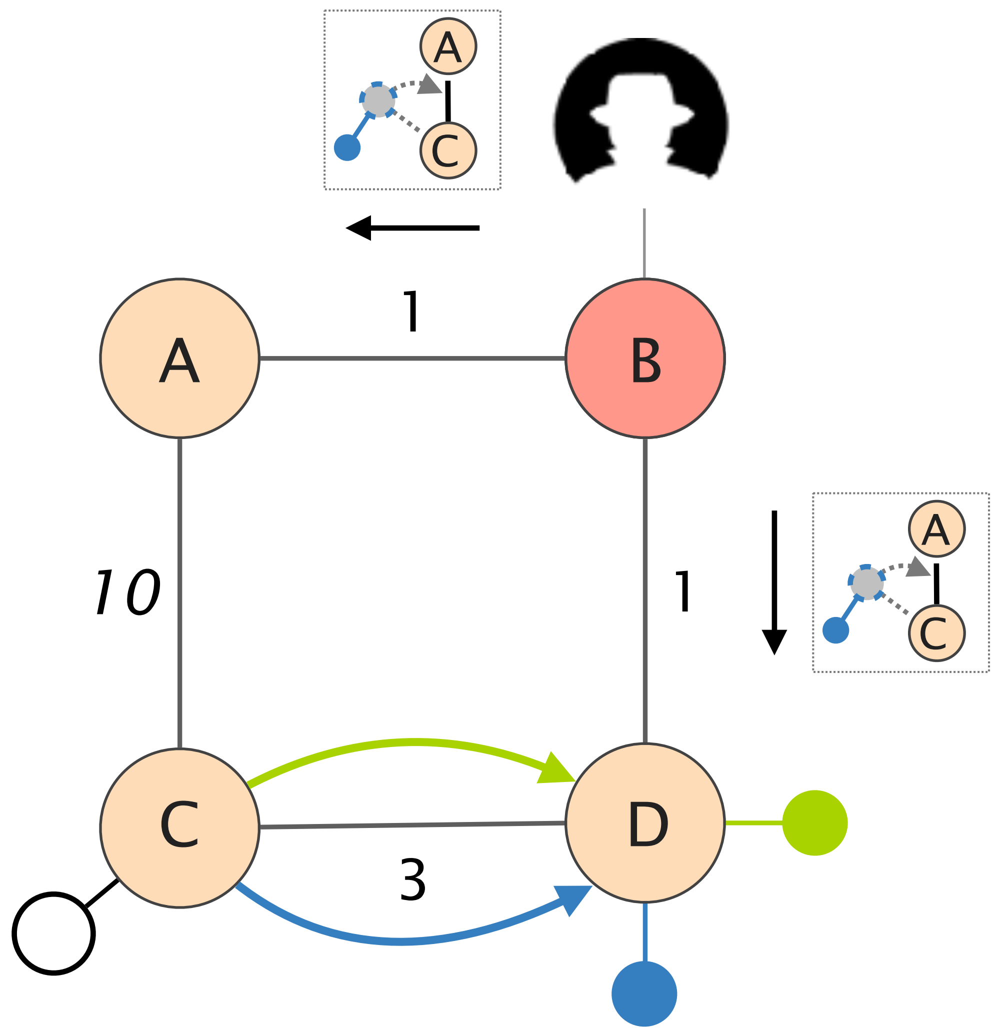


For that the attacker can “lie” to the routers

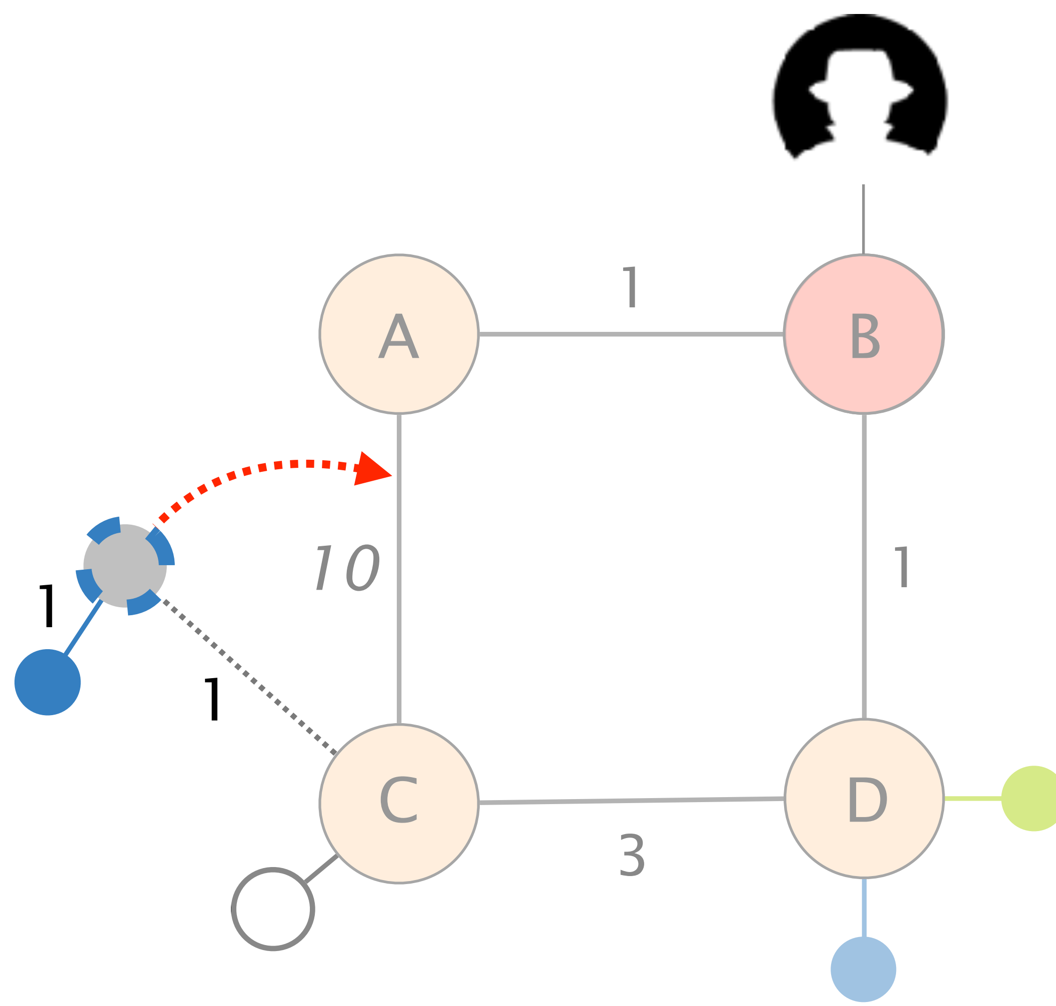
by injecting fake nodes, links and destinations in OSPF



Lies are propagated network-wide by the OSPF protocol

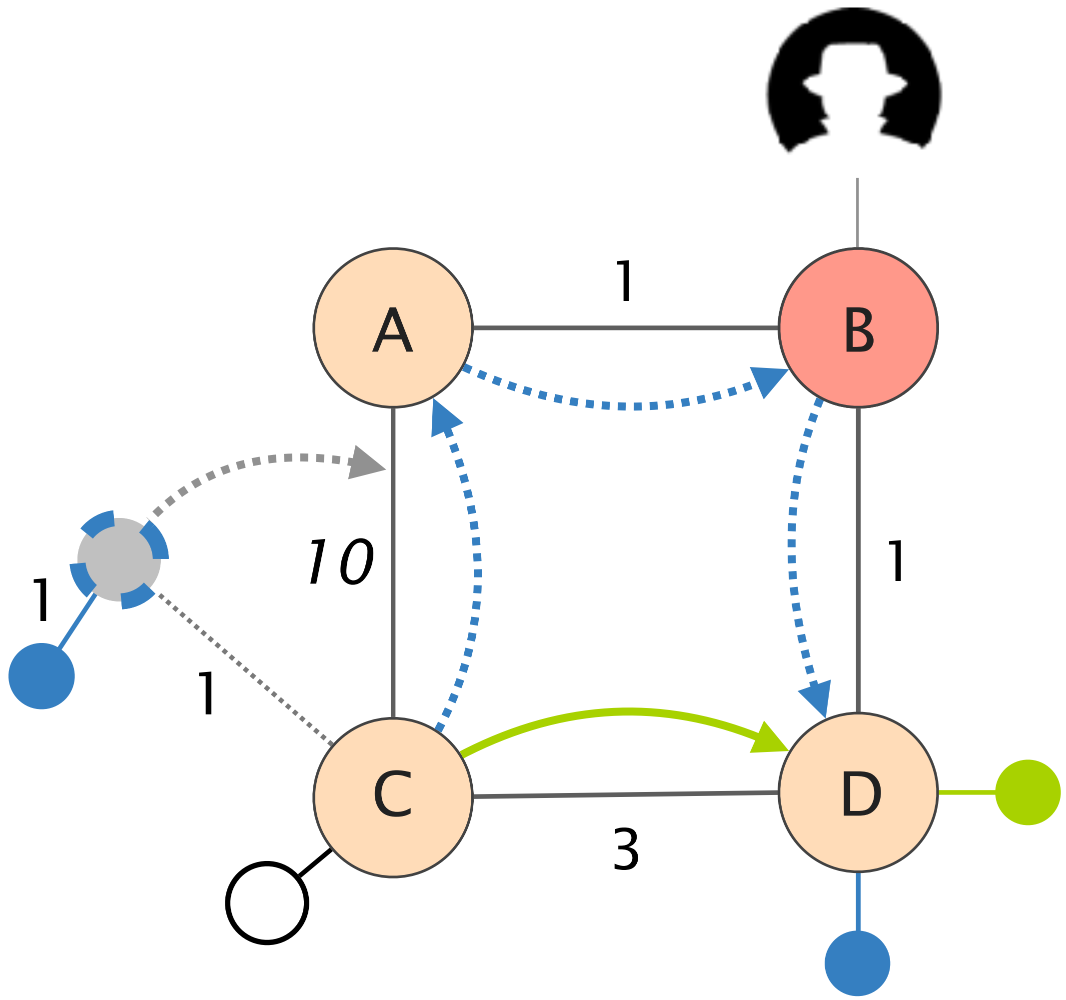


After the injection, this is the topology seen by all routers, on which they compute Dijkstra



Physical NH of the “lie” is A

C prefers A to reach the blue destination directing the traffic through the attacker



By injecting fake information into OSPF, the attacker can precisely control the network-wide behavior

Theorem

It is *always* possible to find fake OSPF messages forcing the routers to compute any forwarding tree

Observation

This gives us a way to program the network-wide behavior from a single location “à la SDN”, in existing networks

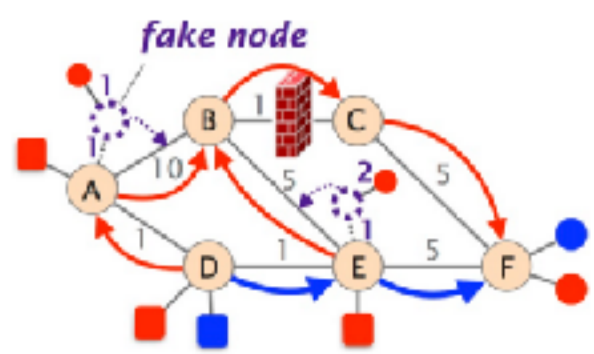
Check out our project

http://fibbing.net

Fibbing: Small Lies for Better Networks

Fibbing is an architecture that enables central control over distributed routing. This way, it combines the advantages of SDN (flexibility, expressivity, and manageability) and traditional (robustness, and scalability) approaches.

Fibbing introduces fake nodes and links into an underlying link-state routing protocol, so that routers compute their own forwarding tables based on the augmented topology. Fibbing is expressive, and readily supports flexible load balancing, traffic engineering, and backup routes. Fibbing works with any unmodified routers speaking OSPF.



Fibbing won the Best Paper Award at SIGCOMM 2015!

Read the papers | Look at the presentations

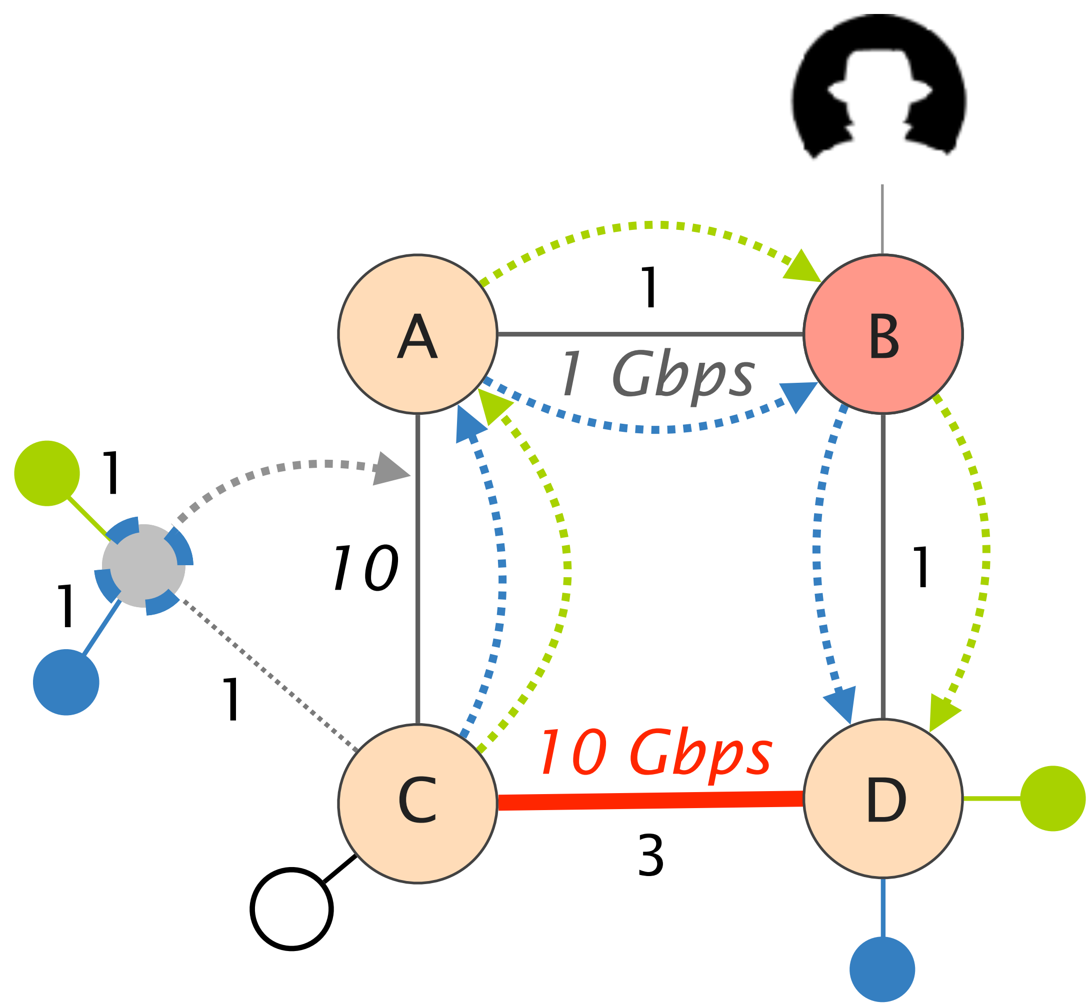
Watch the demo | Get the code

Most of the attacks on intra-domain routing aim at performing Denial-of-Service (DoS) or intercept traffic

Interception	eavesdrop on/drop/modify/inject/delay traffic steer traffic along paths controlled by the attacker
DoS	induce churn to overload the routers announce/withdraw at fast pace
	floods the routers link-state database inject thousands of prefixes
	induce congestion/higher delay steer traffic along fewer/low-throughput paths
	prevent reachability steer traffic along blackholes or loops

By steering traffic,

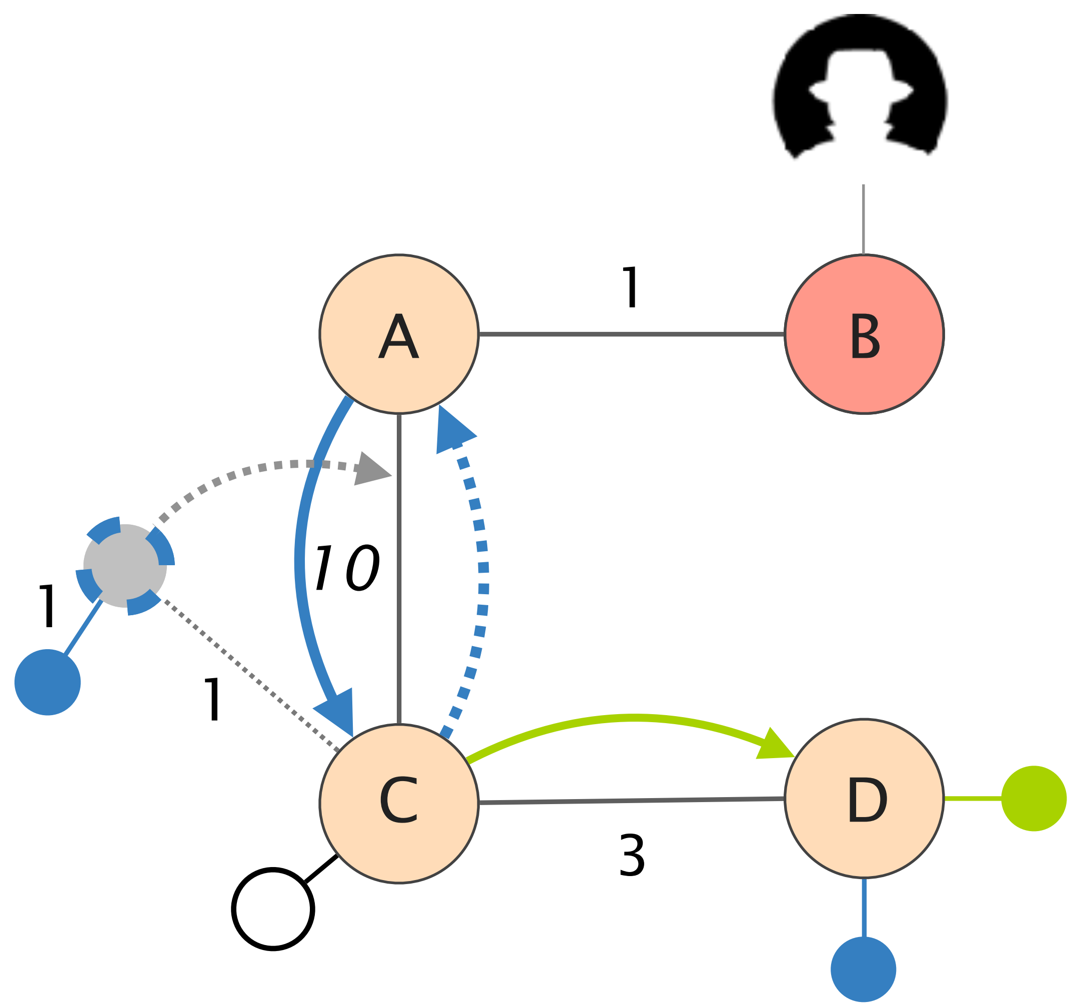
attackers can create congestion and increase delay



traffic flows along a low throughput path

By steering traffic,

attackers can create loops and induce blackholes



traffic is trapped in a forwarding loop between A and C

The solution is quite simple:

Rely on cryptography!

Problem	Bogus advertisements can be injected
	Legitimate advertisements can be tampered with
Solution 1 (light)	Use Cryptographic Authentication (header)
	integrity and authentication
Solution 2 (heavy)	Encrypt the entire advertisement (header/payload)
	integrity, authentication *and* confidentiality

usually using Internet Protocol Security (IPsec)

Solution 2

(heavy)

Encrypt the entire advertisement (header/payload)

integrity, authentication *and* confidentiality

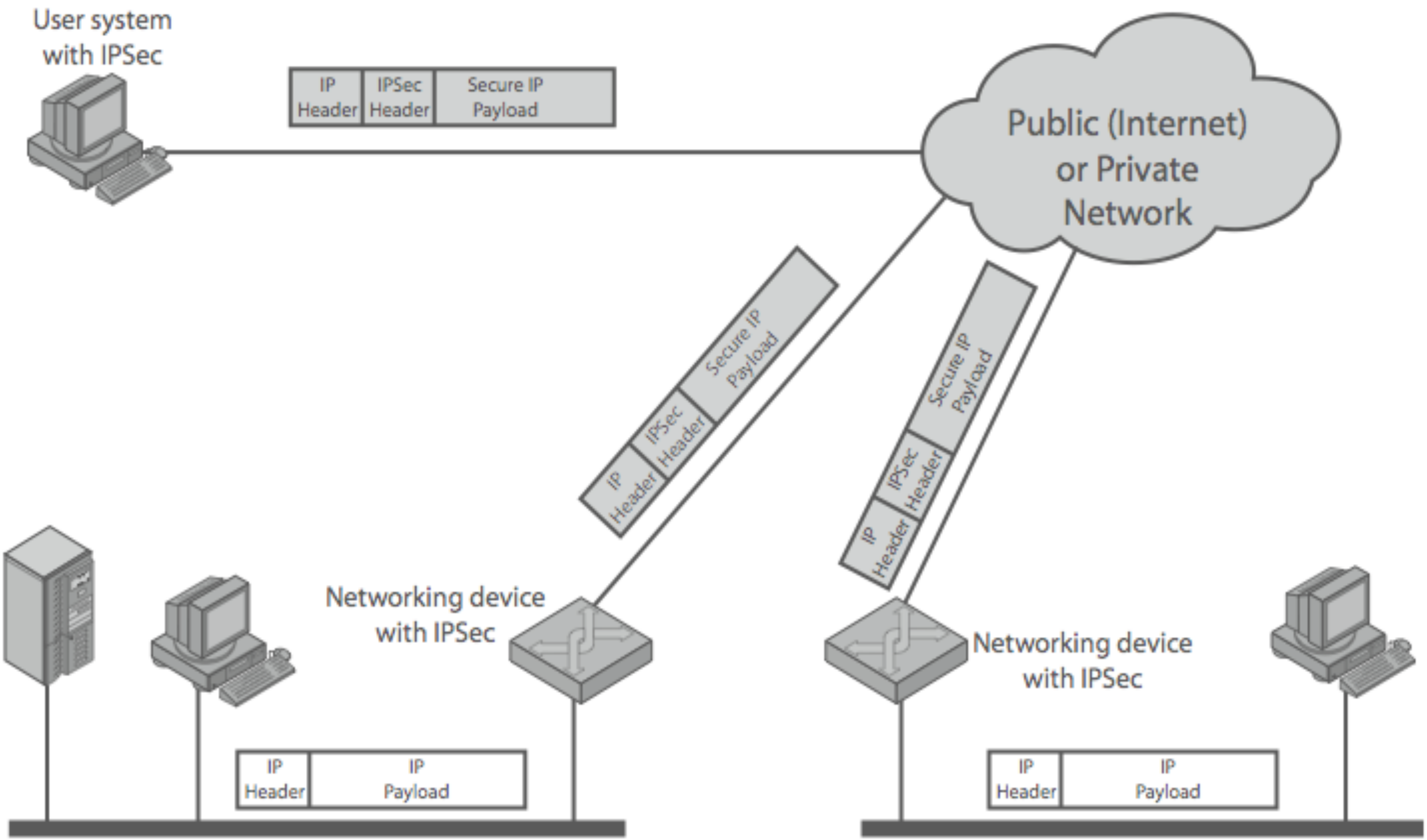
IPSec

- General IP Security framework

- Allows one to provide
  - Access control, integrity, authentication, originality, and confidentiality

- Applicable to different settings
  - Narrow streams: Specific TCP connections
  - Wide streams: All packets between two gateways

IPSec Uses


User system
with IPSec
IP Header
IPSec Header
Secure IP Payload
Public (Internet)
or Private
Network
IP Header
IPSec Header
Secure IP Payload
IP Header
IPSec Header
Secure IP Payload
Networking device
with IPSec
Networking device
with IPSec
IP Header
IP Payload
IP Header
IP Payload

IP Security Architecture

- Specification quite complex
  - Mandatory support in IPv6, optional in IPv4
- Two security header extensions:
  - Authentication Header (AH)
    - Connectionless integrity, origin authentication
      - MAC over most header fields and packet body
    - Anti-replay protection
  - Encapsulating Security Payload (ESP)
    - These properties, plus confidentiality

Routing security

attacks & mitigation

intra-domain
routing

inter-domain
routing

insider

in/outsider

BGP (lack of) security:
problems & solutions

#1	BGP does not validate the origin of advertisements
#2	BGP does not validate the content of advertisements
#3	Proposed Enhancements
#4	What about the data plane?
#5	What's the Internet to do anyway?

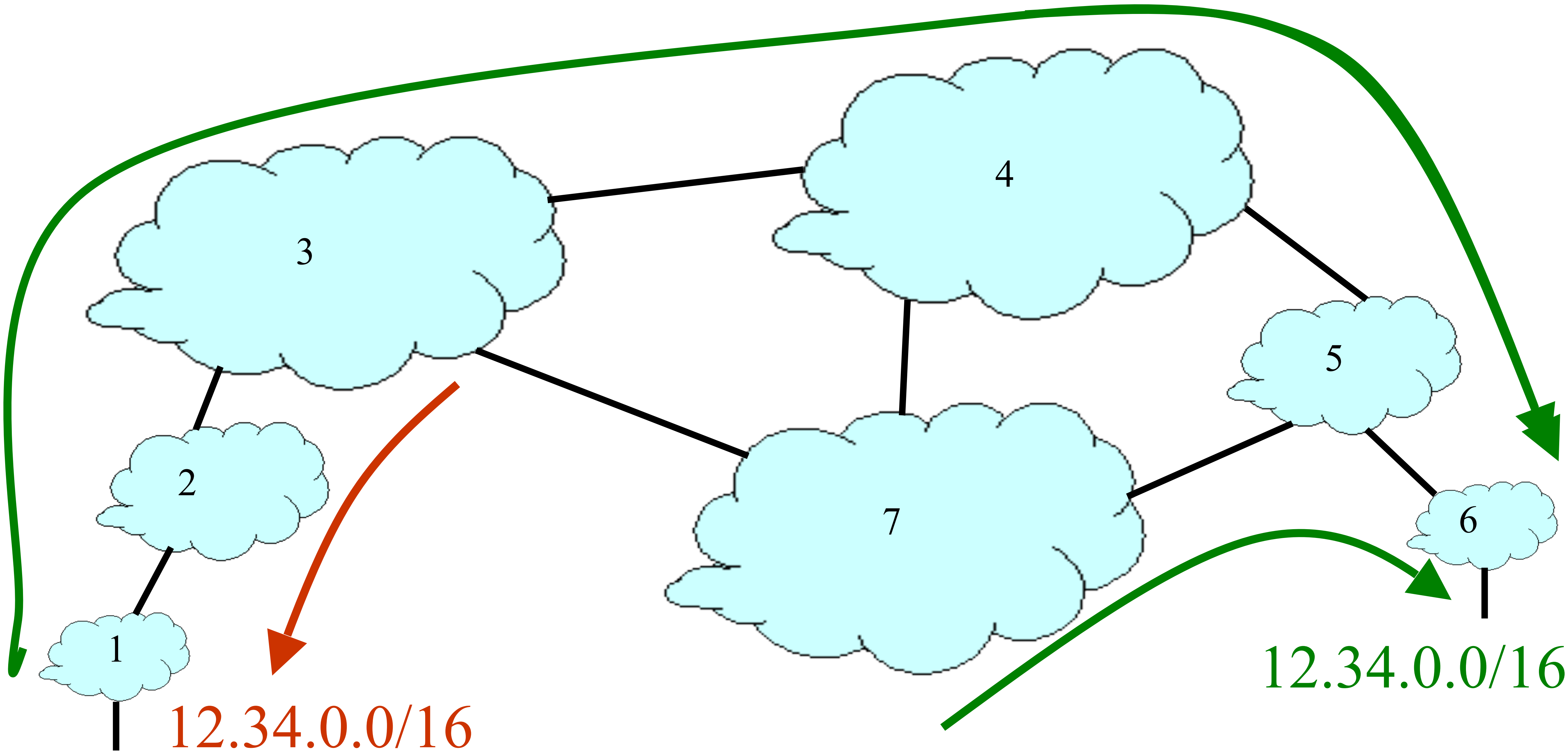
BGP (lack of) security:
problems & solutions

#1	BGP does not validate the origin of advertisements
#2	BGP does not validate the content of advertisements
#3	Proposed Enhancements
#4	What about the data plane?
#5	What's the Internet to do anyway?

IP Address Ownership and Hijacking

- IP address block assignment
  - Regional Internet Registries (ARIN, RIPE, APNIC)
  - Internet Service Providers
- Proper origination of a prefix into BGP
  - By the AS who owns the prefix
  - … or, by its upstream provider(s) in its behalf
- However, what’s to stop someone else?
  - Prefix hijacking: another AS originates the prefix
  - BGP does not verify that the AS is authorized
  - Registries of prefix ownership are inaccurate

Prefix Hijacking

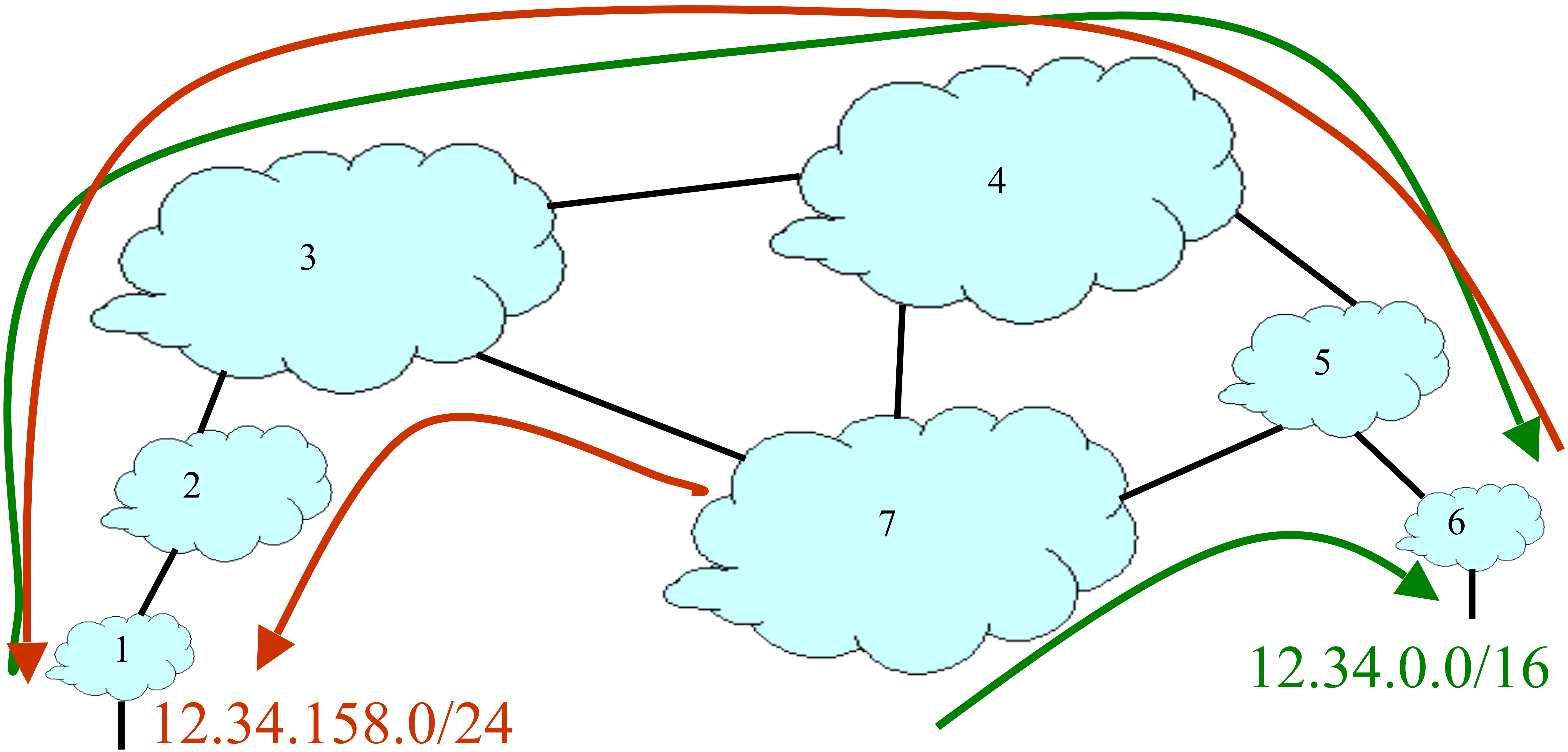


- Blackhole: data traffic is discarded
- Snooping: data traffic is inspected, then redirected
- Impersonation: traffic sent to bogus destinations

Hijacking is Hard to Debug

- The victim AS doesn't see the problem
  - Picks its own route, might not learn the bogus route
- May not cause loss of connectivity
  - Snooping, with minor performance degradation
- Or, loss of connectivity is isolated
  - E.g., only for sources in parts of the Internet
- Diagnosing prefix hijacking
  - Analyzing updates from many vantage points
  - Launching traceroute from many vantage points

Sub-Prefix Hijacking



- Originating a more-specific prefix
  - Every AS picks the bogus route for that prefix
  - Traffic follows the longest matching prefix

How to Hijack a Prefix

- The hijacking AS has
  - Router with BGP session(s)
  - Configured to originate the prefix
- Getting access to the router
  - Network operator makes configuration mistake
  - Disgruntled operator launches an attack
  - Outsider breaks in to the router and reconfigures
- Getting other ASes to believe bogus route
  - Neighbor ASes do not discard the bogus route
  - E.g., not doing protective filtering

YouTube Outage on Feb 24, 2008

- YouTube (AS 36561)
  - Web site www.youtube.com (208.65.152.0/22)
- Pakistan Telecom (AS 17557)
  - Government order to block access to YouTube
  - Announces 208.65.153.0/24 to PCCW (AS 3491)
  - All packets to YouTube get dropped on the floor
- Mistakes were made
  - AS 17557: announce to everyone, not just customers
  - AS 3491: not filtering routes announced by AS 17557
- Lasted 100 minutes for some, 2 hours for others

Timeline (UTC Time)

- 18:47:45
  - First evidence of hijacked /24 route in Asia
- 18:48:00
  - Several big trans-Pacific providers carrying the route
- 18:49:30
  - Bogus route fully propagated
- 20:07:25
  - YouTube starts advertising /24 to attract traffic back
- 20:08:30
  - Many (but not all) providers are using valid route

Timeline (UTC Time)

- 20:18:43
  – YouTube announces two more-specific /25 routes
- 20:19:37
  – Some more providers start using the /25 routes
- 20:50:59
  – AS 17557 starts prepending (“3491 17557 17557”)
- 20:59:39
  – AS 3491 disconnects AS 17557
- 21:00:00
  – Videos of cats flushing toilets are available again!

Another Example: Spammers

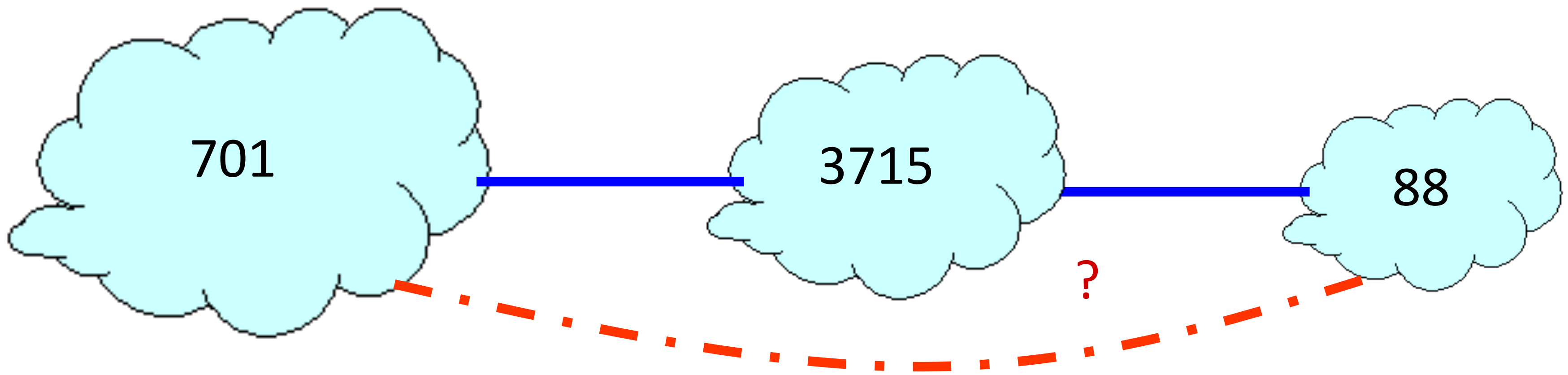
- Spammers sending spam
  - Form a (bidirectional) TCP connection to mail server
  - Send a bunch of spam e-mail, then disconnect
- But, best not to use your real IP address
  - Relatively easy to trace back to you
- Could hijack someone's address space
  - But you might not receive all the (TCP) return traffic
- How to evade detection
  - Hijack unused (i.e., unallocated) address block
  - Temporarily use the IP addresses to send your spam

BGP (lack of) security:
problems & solutions

#1	BGP does not validate the origin of advertisements
#2	BGP does not validate the content of advertisements
#3	Proposed Enhancements
#4	What about the data plane?
#5	What's the Internet to do anyway?

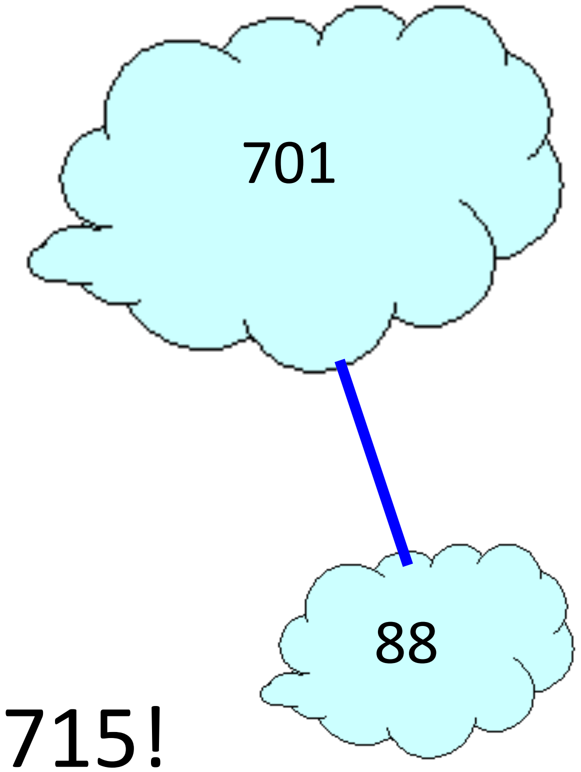
Bogus AS Paths

- Remove ASes from the AS path
  - E.g., turn "701 3715 88" into "701 88"
- Motivations
  - Attract sources that normally try to avoid AS 3715
  - Help AS 88 look like it is closer to the Internet's core
- Who can tell that this AS path is a lie?
  - Maybe AS 88 *does* connect to AS 701 directly



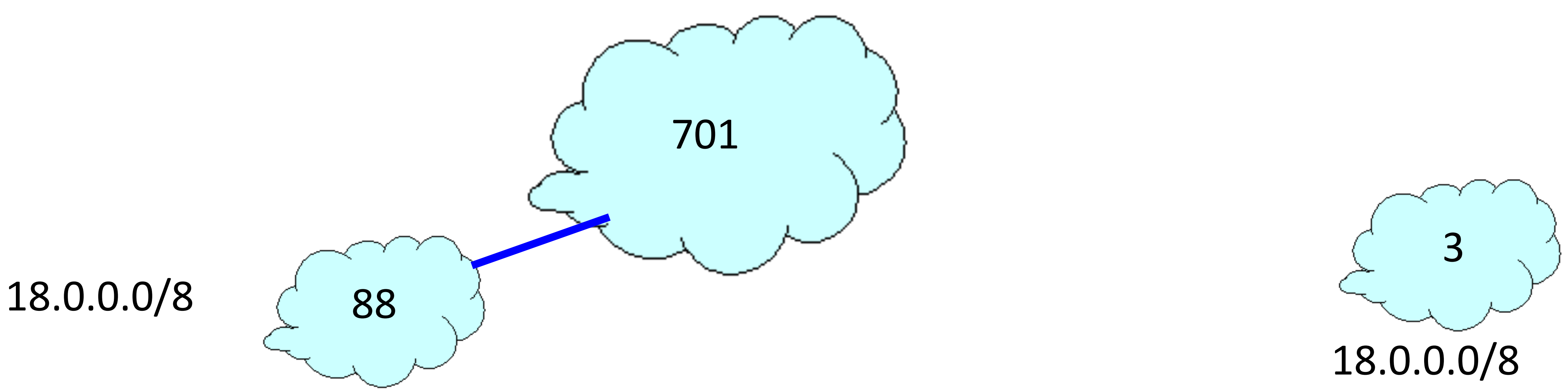
Bogus AS Paths

- Add ASes to the path
  - E.g., turn “701 88” into “701 3715 88”
- Motivations
  - Trigger loop detection in AS 3715
    - Denial-of-service attack on AS 3715
    - Or, blocking unwanted traffic coming from AS 3715!
  - Make your AS look like is has richer connectivity
- Who can tell the AS path is a lie?
  - AS 3715 could, if it could see the route
  - AS 88 could, but would it really care?



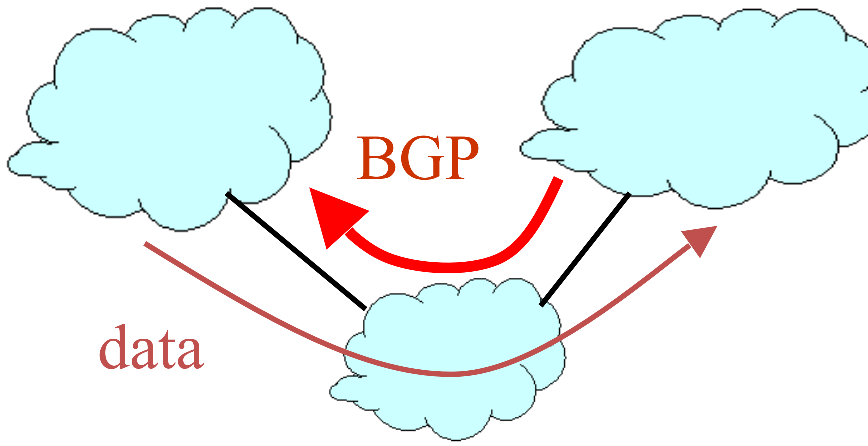
Bogus AS Paths

- Adds AS hop(s) at the end of the path
  - E.g., turns "701 88" into "701 88 3"
- Motivations
  - Evade detection for a bogus route
  - E.g., by adding the legitimate AS to the end
- Hard to tell that the AS path is bogus…
  - Even if other ASes filter based on prefix ownership



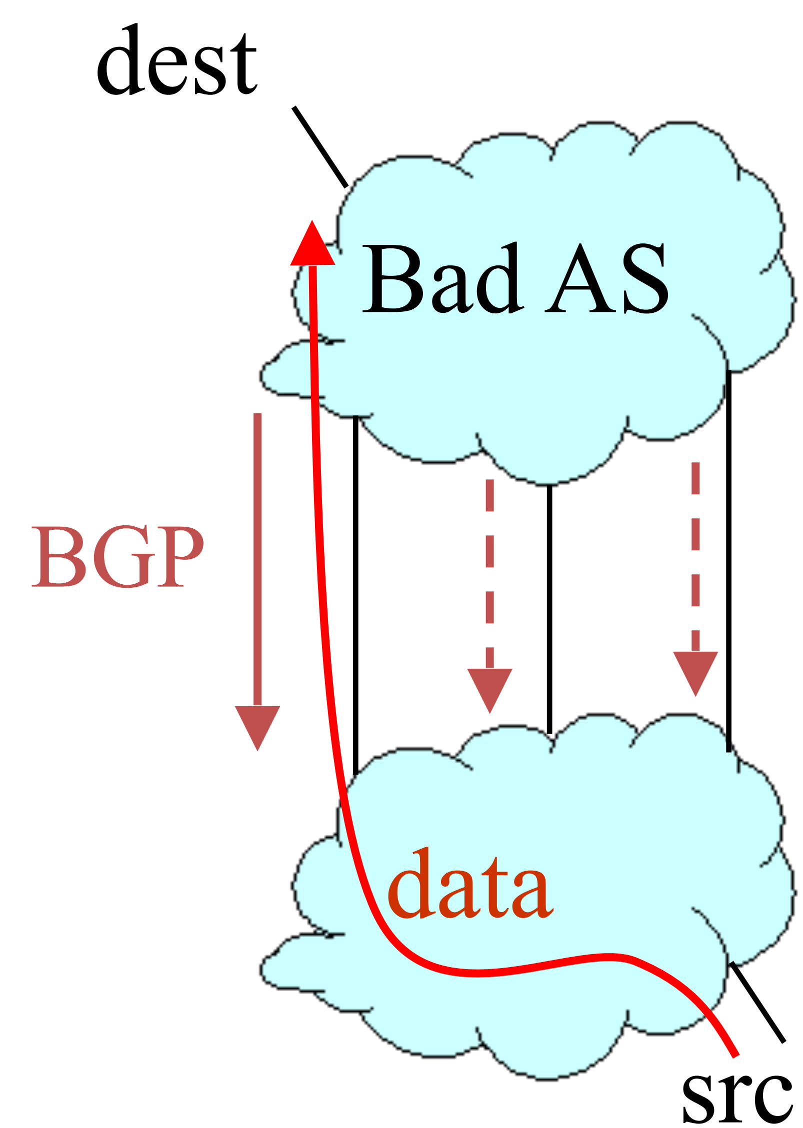
Invalid Paths

- AS exports a route it shouldn't
  – AS path is a valid sequence, but violated policy
- Example: customer misconfiguration
  – Exports routes from one provider to another
- Interacts with provider policy
  – Provider prefers customer routes
  – Directing all traffic through customer
- Main defense
  – Filtering routes based on prefixes and AS path



Missing/Inconsistent Routes

- Peers require consistent export
  - Prefix advertised at all peering points
  - Prefix advertised with same AS path length
- Reasons for violating the policy
  - Trick neighbor into “cold potato”
  - Configuration mistake
- Main defense
  - Analyzing BGP updates, or traffic,
  - … for signs of inconsistency



BGP Security Today

- Applying best common practices (BCPs)
  – Securing the session (authentication, encryption)
  – Filtering routes by prefix and AS path
  – Packet filters to block unexpected control traffic
- This is not good enough
  – Depends on vigilant application of BCPs
  – Doesn’t address fundamental problems
    - Can’t tell who owns the IP address block
    - Can’t tell if the AS path is bogus or invalid
    - Can’t be sure the data packets follow the chosen route

Routing attacks can be used to de-anonymize Tor users

RAPTOR: Routing Attacks on Privacy in Tor

Yixin Sun, *Princeton University*
Anne Edmundson, *Princeton University*
Laurent Vanbever, *ETH Zurich*
Oscar Li, *Princeton University*
Jennifer Rexford, *Princeton University*
Mung Chiang, *Princeton University*
Prateek Mittal, *Princeton University*

Abstract

The Tor network is a widely used system for anonymous communication. However, Tor is known to be vulnerable to attackers who can observe traffic at both ends of the communication path. In this paper, we show that prior attacks are just the tip of the iceberg. We present a suite of new attacks, called Raptor, that can be launched by Autonomous Systems (ASes) to compromise user anonymity. First, AS-level adversaries can exploit the asymmetric nature of Internet routing to increase the chance of observing at least one direction of user traffic at both ends of the communication. Second, AS-level adversaries can exploit natural churn in Internet routing to lie on the BGP paths for more users over

specific Tor guard nodes) and interceptions (to perform traffic analysis). We demonstrate the feasibility of Ran-

journalists, businesses and ordinary citizens concerned about the privacy of their online communications [9].

Along with anonymity, Tor aims to provide low latency and, as such, does not obfuscate packet timings or sizes. Consequently, an adversary who is able to observe traffic on both segments of the Tor communication channel (*i.e.*, between the server and the Tor network, and between the Tor network and the client) can correlate packet sizes and packet timings to deanonymize Tor clients [45, 46].

There are essentially two ways for an adversary to gain visibility into Tor traffic, either by compromising (or owning enough) Tor relays or by manipulating the underlying network communications so as to put herself on the forwarding path for Tor traffic. Regarding net-

a portion of all links, and observe any unencrypted infor-

See http://vanbever.eu/pdfs/vanbever_raptor_usenix_security_2015.pdf

Routing attacks can be used to partition the Bitcoin network

Hijacking Bitcoin: Routing Attacks on Cryptocurrencies

https://btc-hijack.ethz.ch

Maria Apostolaki
ETH Zürich
apmaria@ethz.ch

Aviv Zohar
The Hebrew University
avivz@cs.huji.ac.il

Laurent Vanbever
ETH Zürich
lvanbever@ethz.ch

***Abstract*—As the most successful cryptocurrency to date, Bitcoin constitutes a target of choice for attackers. While many attack vectors have already been uncovered, one important vector has been left out though: attacking the currency via the Internet routing infrastructure itself. Indeed, by manipulating routing advertisements (BGP hijacks) or by naturally intercepting traffic, Autonomous Systems (ASes) can intercept and manipulate a large fraction of Bitcoin traffic.**

This paper presents the first taxonomy of routing attacks and their impact on Bitcoin, considering both small-scale attacks, targeting individual nodes, and large-scale attacks, targeting the network as a whole. While challenging, we show that two key properties make routing attacks practical: *(i)* the efficiency of routing manipulation; and *(ii)* the significant centralization of Bitcoin in terms of mining and routing. Specifically, we find that any network attacker can hijack few (<100) BGP prefixes to isolate ~50% of the mining power—even when considering that mining pools are heavily multi-homed. We also show that on-path network attackers can considerably slow down block propagation by interfering with few key Bitcoin messages.

We demonstrate the feasibility of each attack against the deployed Bitcoin software. We also quantify their effectiveness on ... m a Bitcoin

See https://btc-hijack.ethz.ch

The potential damage to Bitcoin is worrying. By isolating parts of the network or delaying block propagation, attackers can cause

One important attack vector has been overlooked though: attacking Bitcoin via the Internet infrastructure using *routing attacks*. As Bitcoin connections are routed over the Internet—in clear text and without integrity checks—any third-party on the forwarding path can eavesdrop, drop, modify, inject, or delay Bitcoin messages such as blocks or transactions. Detecting such attackers is challenging as it requires inferring the exact forwarding paths taken by the Bitcoin traffic using measurements (e.g., traceroute) or routing data (BGP announcements), both of which can be forged [41]. Even ignoring detectability, mitigating network attacks is also hard as it is essentially a human-driven process consisting of filtering, routing around or disconnecting the attacker. As an illustration, it took Youtube close to 3 hours to locate and resolve rogue BGP announcements targeting its infrastructure in 2008 [6]. More recent examples of routing attacks such as [51] (resp. [52]) took 9 (resp. 2) hours to resolve in November (resp. June) 2015.

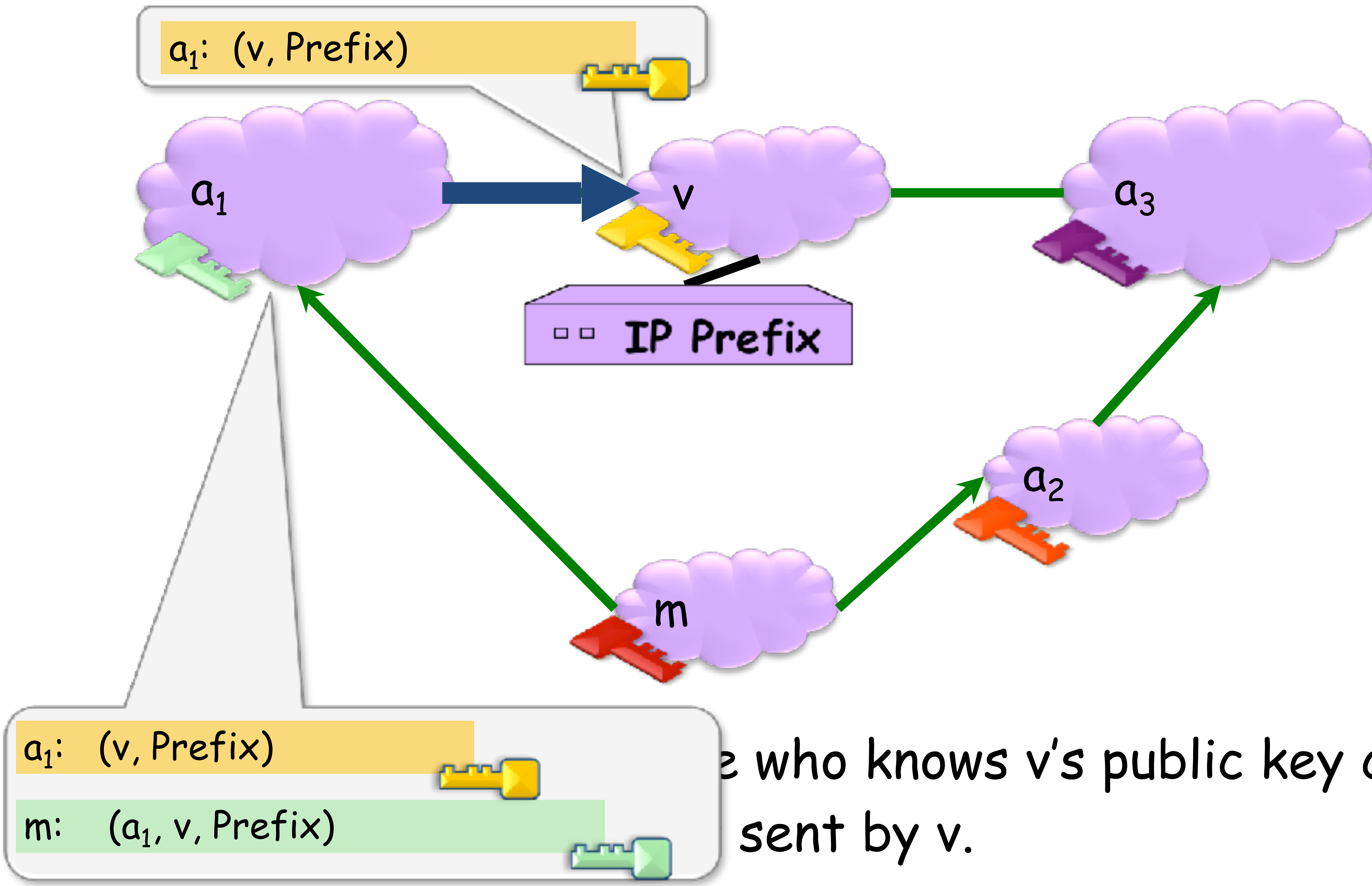
One of the reasons why routing attacks have been overlooked in Bitcoin is that they are often considered too challenging to be practical. Indeed, perturbing a vast peer-to-peer

BGP (lack of) security:
problems & solutions

#1	BGP does not validate the origin of advertisements
#2	BGP does not validate the content of advertisements
#3	Proposed Enhancements
#4	What about the data plane?
#5	What's the Internet to do anyway?

Secure BGP

Origin Authentication + cryptographic signatures



e who knows v's public key can

sent by v.

S-BGP Secure Version of BGP

- Address attestations
  - Claim the right to originate a prefix
  - Signed and distributed out-of-band
  - Checked through delegation chain from ICANN
- Route attestations
  - Distributed as an attribute in BGP update message
  - Signed by each AS as route traverses the network
- S-BGP can validate
  - AS path indicates the order ASes were traversed
  - No intermediate ASes were added or removed

S-BGP Deployment Challenges

- Complete, accurate registries of prefix "owner"
- Public Key Infrastructure
  - To know the public key for any given AS
- Cryptographic operations
  - E.g., digital signatures on BGP messages
- Need to perform operations quickly
  - To avoid delaying response to routing changes
- Difficulty of incremental deployment
  - Hard to have a "flag day" to deploy S-BGP

Incrementally Deployable Solutions?

- Backwards compatible
  - No changes to router hardware or software
  - No cooperation from other ASes
- Incentives for early adopters
  - Security benefits for ASes that deploy the solution
  - … and further incentives for others to deploy
- What kind of solutions are possible?
  - Detecting suspicious routes
  - … and then filtering or depreferencing them

Detecting Suspicious Routes

- Monitoring BGP update messages
  - Use past history as an implicit registry
- E.g., AS that announces each address block
  - Prefix 18.0.0.0/8 usually originated by AS 3
- E.g., AS-level edges and paths
  - Never seen the subpath "7018 88 1785"
- Out-of-band detection mechanism
  - Generate reports and alerts
  - Internet Alert Registry: http://iar.cs.unm.edu/
  - Prefix Hijack Alert System: http://phas.netsec.colostate.edu/

Avoiding Suspicious Routes

- Soft response to suspicious routes
  - Prefer routes that agree with the past
  - Delay adoption of unfamiliar routes when possible
- Why is this good enough?
  - Some attacks will go away on their own
  - Let someone else be the victim instead of you
  - Give network operators time to investigate
- How well would it work?
  - If top ~40 largest ASes applied the technique
  - … most other ASes are protected, too

BGP (lack of) security:

problems & solutions

#1 BGP does not validate the origin of advertisements

#2 BGP does not validate the content of advertisements

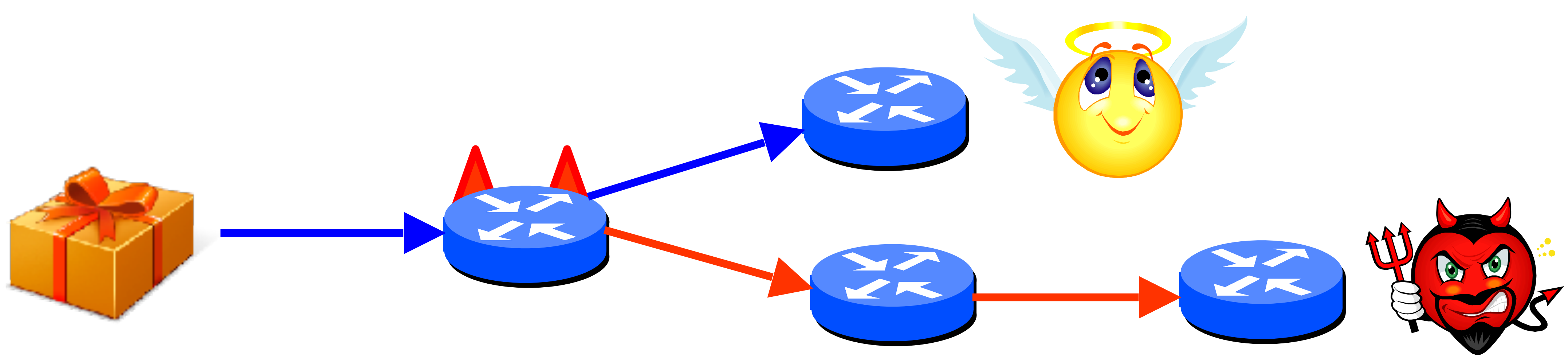
#3 Proposed Enhancements

#4 What about the data plane?

#5 What's the Internet to do anyway?

Control Plane vs. Data Plane

- Control plane
  - BGP security concerns validity of routing messages
  - I.e., did the BGP message follow the sequence of ASes listed in the AS-path attribute
- Data plane
  - Routers forward data packets
  - Supposedly along path chosen in the control plane
  - But what ensures that this is true?



Data-Plane Attacks, Part 1

- Drop packets in the data plane
  - While still sending the routing announcements
- Easier to evade detection
  - Especially if you only drop some packets
  - Like, oh, say, BitTorrent or Skype traffic
- Even easier if you just slow down some traffic
  - How different are normal congestion and an attack?
  - Especially if you let traceroute packets through?

Data-Plane Attacks, Part 2

- Send packets in a different direction
  - Disagreeing with the routing announcements
- Direct packets to a different destination
  - E.g., one the adversary controls
- What to do at that bogus destination?
  - Impersonate the legitimate destination
  - Snoop on traffic and forward along to real destination
- How to detect?
  - Traceroute? Longer than usual delays?
  - End-to-end checks, like site certificate or encryption?

Data-Plane Attacks are Harder

- Adversary must control a router along the path
  - So that the traffic flows through him
- How to get control a router
  - Buy access to a compromised router online
  - Guess the password, exploit router vulnerabilities
  - Insider attack (disgruntled network operator)
- Malice vs. greed
  - Malice: gain control of someone else's router
  - Greed: Verizon DSL blocks Skype to encourage me to use (Verizon) landline phone

BGP (lack of) security:
problems & solutions

#1	BGP does not validate the origin of advertisements
#2	BGP does not validate the content of advertisements
#3	Proposed Enhancements
#4	What about the data plane?
#5	What's the Internet to do anyway?

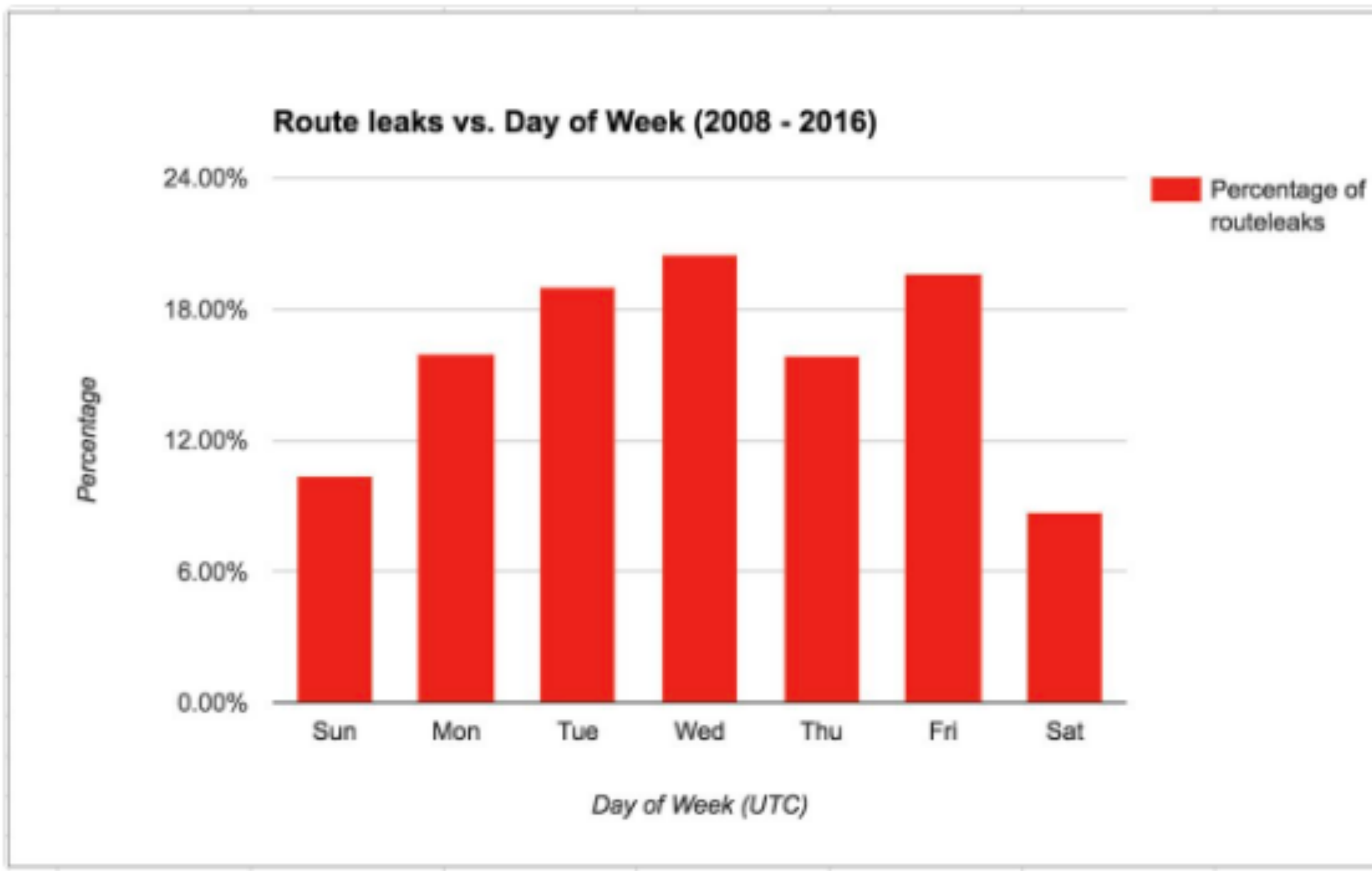
BGP is Sooo Vulnerable

- Several high-profile outages
  - http://merit.edu/mail.archives/nanog/1997-04/msg00380.html
  - http://www.renesys.com/blog/2005/12/internetwide_nearcatastrophela.shtml
  - http://www.renesys.com/blog/2006/01/coned_steals_the_net.shtml
  - http://www.renesys.com/blog/2008/02/pakistan_hijacks_youtube_1.shtml
  - http://www.theregister.co.uk/2010/04/09/china_bgp_interweb_snafu/
- Many smaller examples
  - Blackholing a single destination prefix
  - Hijacking unallocated addresses to send spam
- Why isn't it an even bigger deal?
  - Really, most big outages are configuration errors
  - Most bad guys want the Internet to stay up

Job Snijders
@JobSnijders

Follow

Fun fact: most BGP route leaks happen on Wednesdays, but in the weekend us humans collectively take a break! :-)



BGP is Sooo Hard to Fix

- Complex system
  - Large, with around 60,000 ASes
  - Decentralized control among competitive Ases
- Hard to reach agreement on the right solution
  - S-BGP with PKI, registries, and crypto?
  - Who should be in charge of running PKI & registries?
  - Worry about data-plane attacks or just control plane?
- Hard to deploy the solution once you pick it
  - Hard enough to get ASes to apply route filters
  - Now you want them to upgrade to a new protocol

Conclusions

- Internet protocols designed based on trust
  - Insiders are good guys, bad guys on the outside
- Border Gateway Protocol is very vulnerable
  - Glue that holds the Internet together
  - Hard for an AS to locally identify bogus routes
  - Attacks can have very serious global consequences
- Proposed solutions/approaches
  - Secure variants of the Border Gateway Protocol
  - Anomaly detection, with automated response
  - Broader focus on data-plane availability

Communication Networks

Spring 2017

Laurent Vanbever

www.vanbever.eu

ETH Zürich (D-ITET)

April, 24 2017