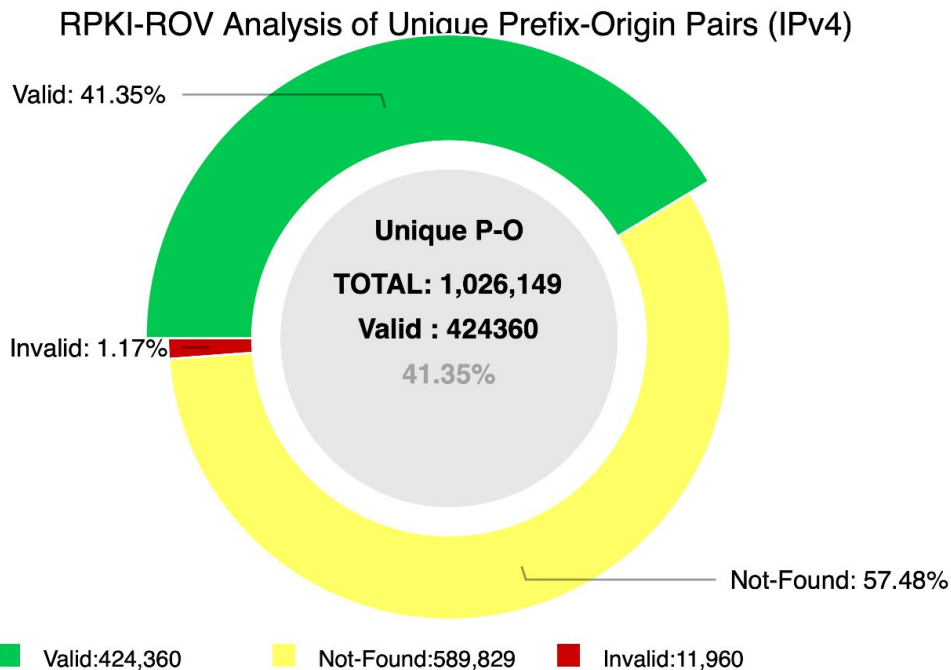# BGP security today

- Resource Public Key Infrastructure (RPKI)
  - A framework to support improved BGP security:
    - A secure way to map AS numbers to IP prefixes.
    - A distributed repository system for storing and disseminating the mappings.
- RPKI operations
  - RPKI relies on cryptographic certificates (X.509)
  - The certificate infrastructure mimics the way IP prefixes are distributed: from IANA, to Regional Internet Registries (RIR), to end-customers.
  - A Route Origination Authorization (ROA) states which AS is authorized to originate certain IP prefixes.

BGP security today - long way to go

RPKI-ROV Analysis of Unique Prefix-Origin Pairs (IPv4)

Valid: 41.35%

Invalid: 1.17%

Unique P-O

TOTAL: 1,026,149

Valid : 424360

41.35%

Not-Found: 57.48%

■ Valid:424,360   ■ Not-Found:589,829   ■ Invalid:11,960

NIST RPKI Monitor: RPKI-ROV Analysis   **Protocol:** IPv4   **RIR:** All   **Date:** 2023-03-02 06:00

https://rpki-monitor.antd.nist.gov

BGP is extremely vulnerable

- Several high-profile outages
- 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
  - … so they can send unwanted traffic (e.g., spam, identity theft, denial-of-service attacks, port scans, …)

BGP is hard to fix

- Complex system
    - Large, with around 50,000 ASes
    - Decentralized control among competitive ASes
    - Core infrastructure that forms the Internet
- Hard to reach agreement on the right solution
    - S-BGP with public key infrastructure, registries, crypto?
    - Who should be in charge of running PKI and 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
    - … all at the exact same moment? A "flag day"

Because BGP is based on policy - it is not guaranteed to converge

- 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

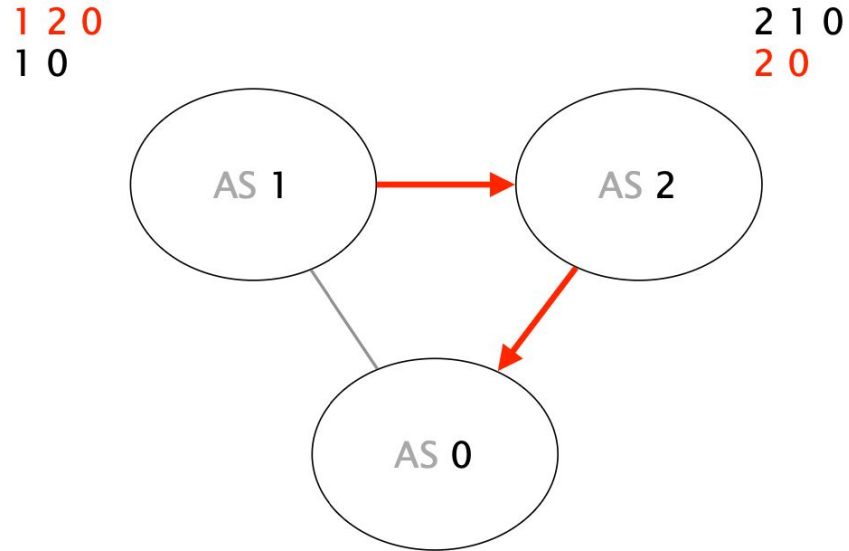# Because of policy, BGP can have multiple stable states

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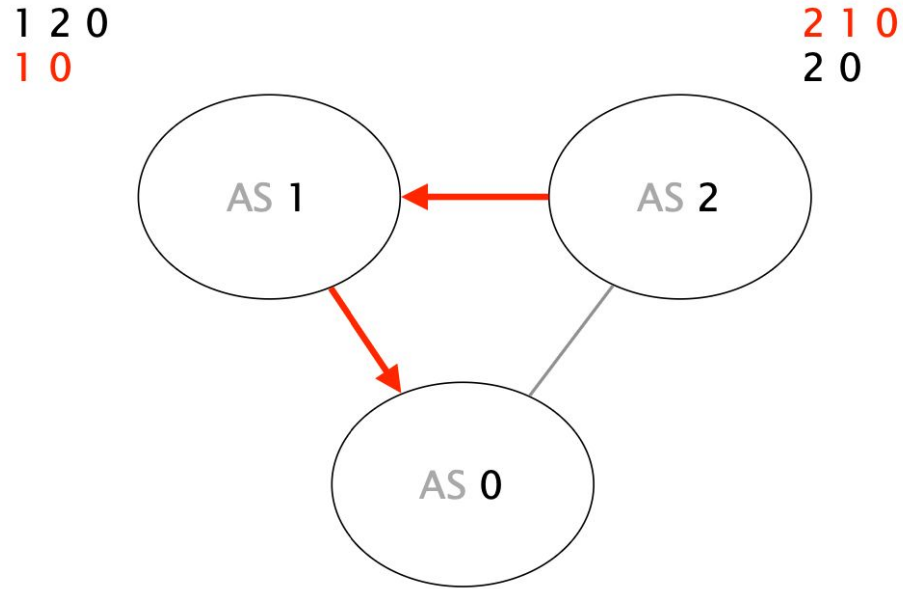
1 prefers to reach 0
via 2 rather than directly

```
1 2 0
1 0
```

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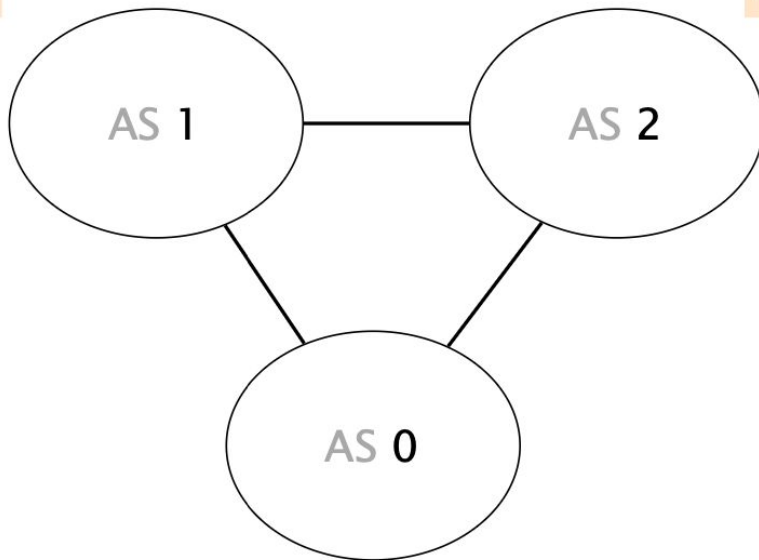
If AS2 is the first to advertise [2 0],
the system stabilizes in a state where AS 1 is happy

1 2 0
1 0

2 1 0
2 0

AS 1

AS 2

AS 0

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
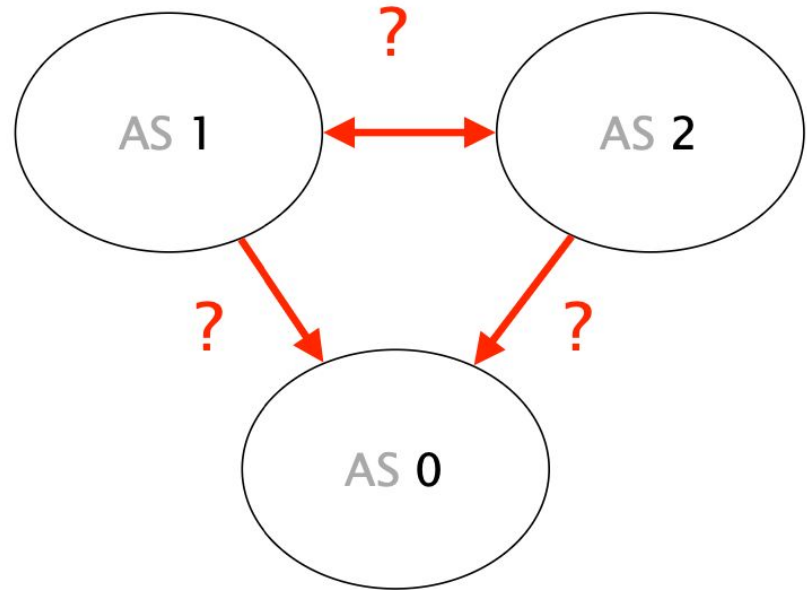
With arbitrary policies, BGP may fail to converge
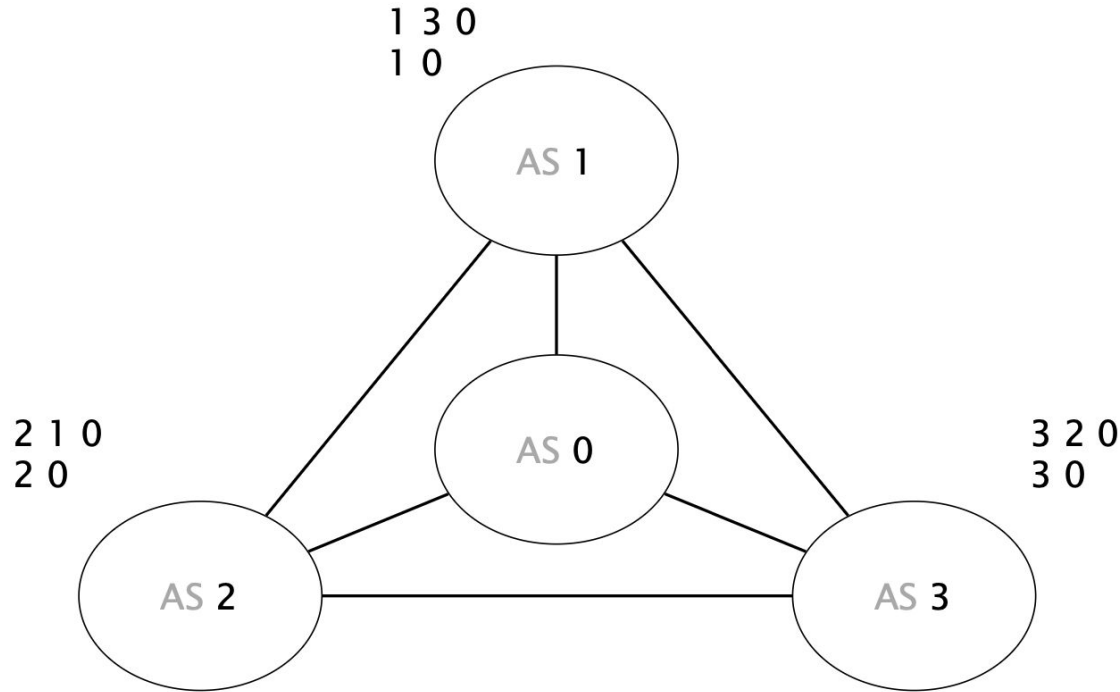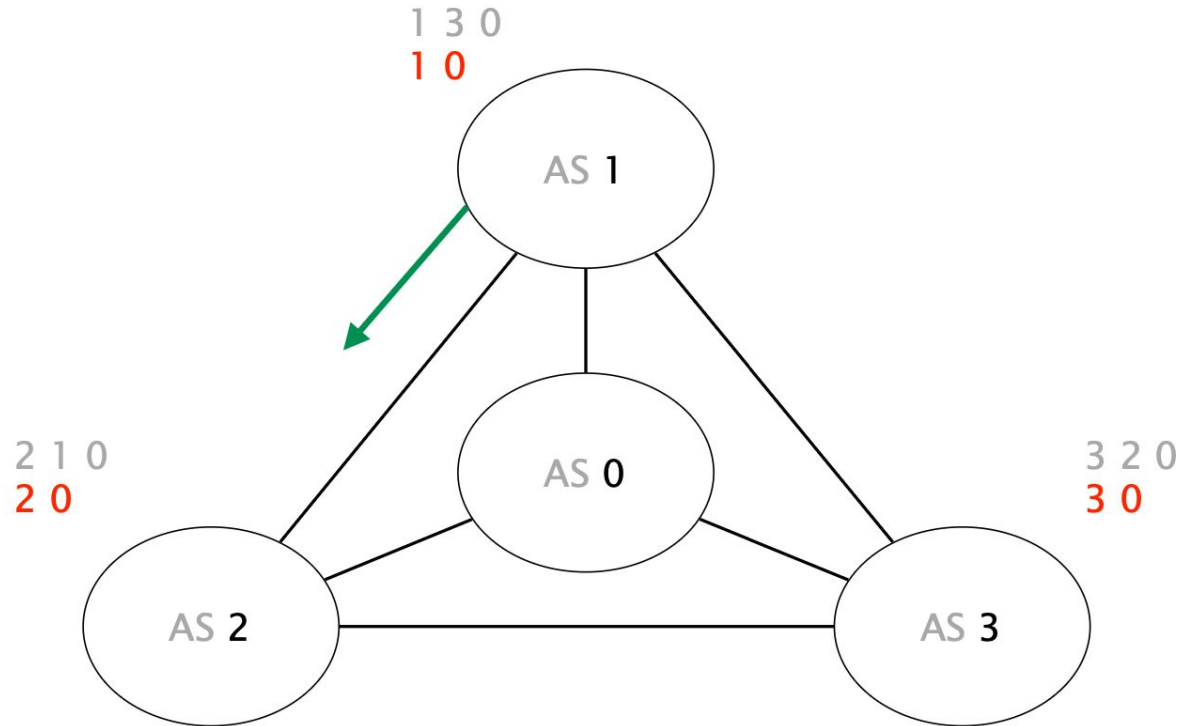
With arbitrary policies, BGP may fail to converge

preference list

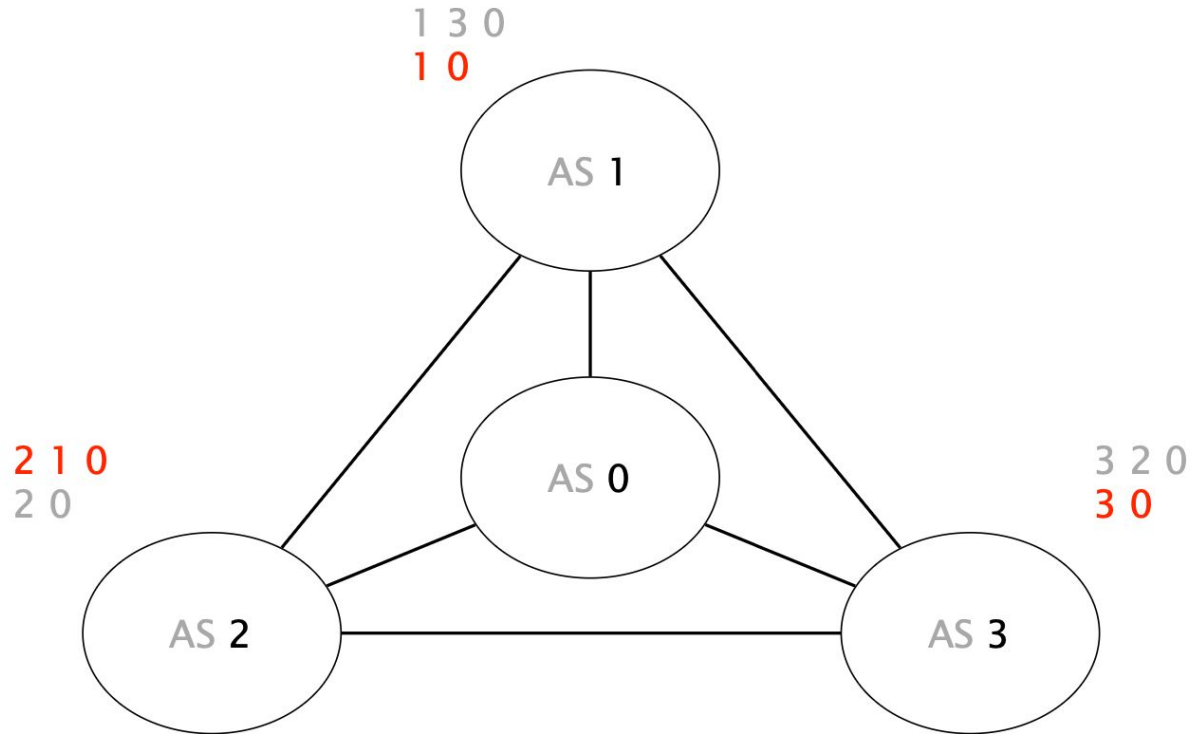1 prefers to reach 0
via 3 rather than directly

1 3 0
1 0

AS 1

2 1 0
2 0

AS 0

3 2 0
3 0

AS 2

AS 3
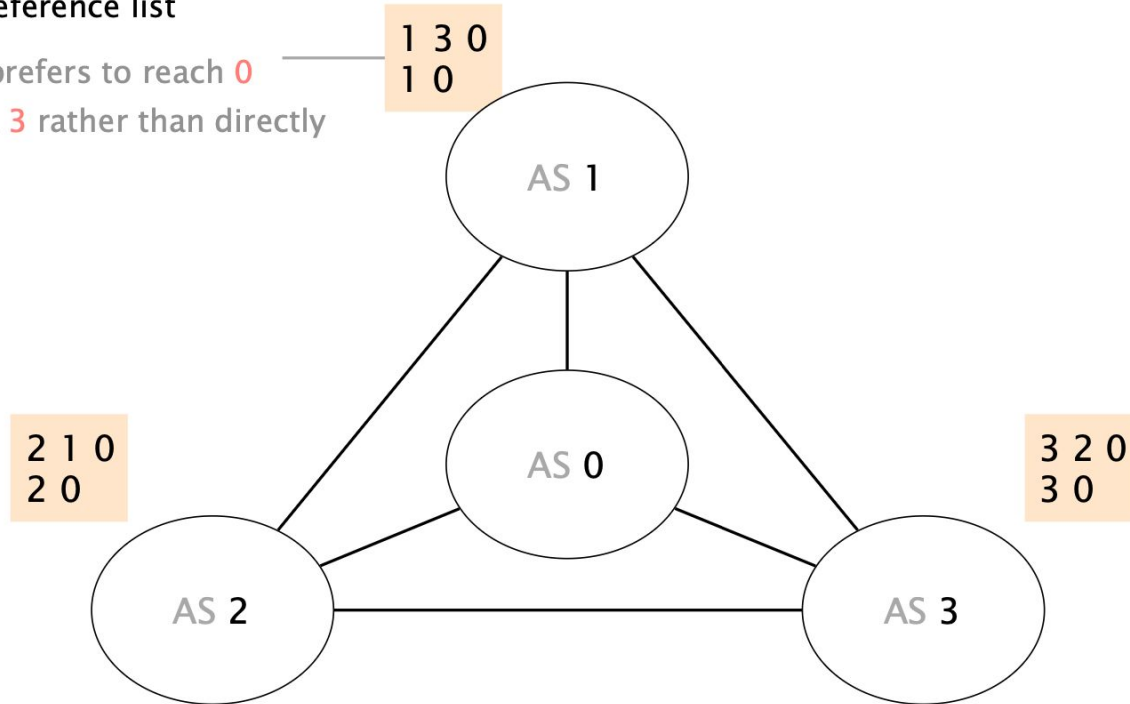
Initially, all ASes only know the direct route to 0
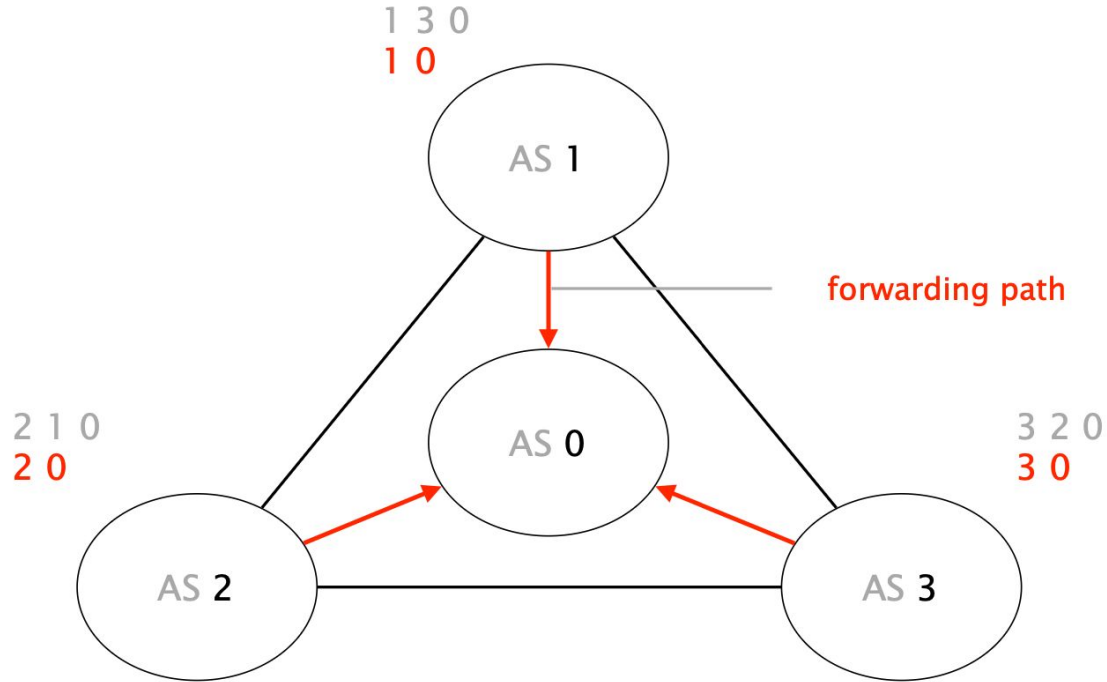
AS 1 advertises its path to AS 2

Upon reception, AS 2 switches to 2 1 0 (preferred)

1 3 0
1 0

AS 1

2 1 0
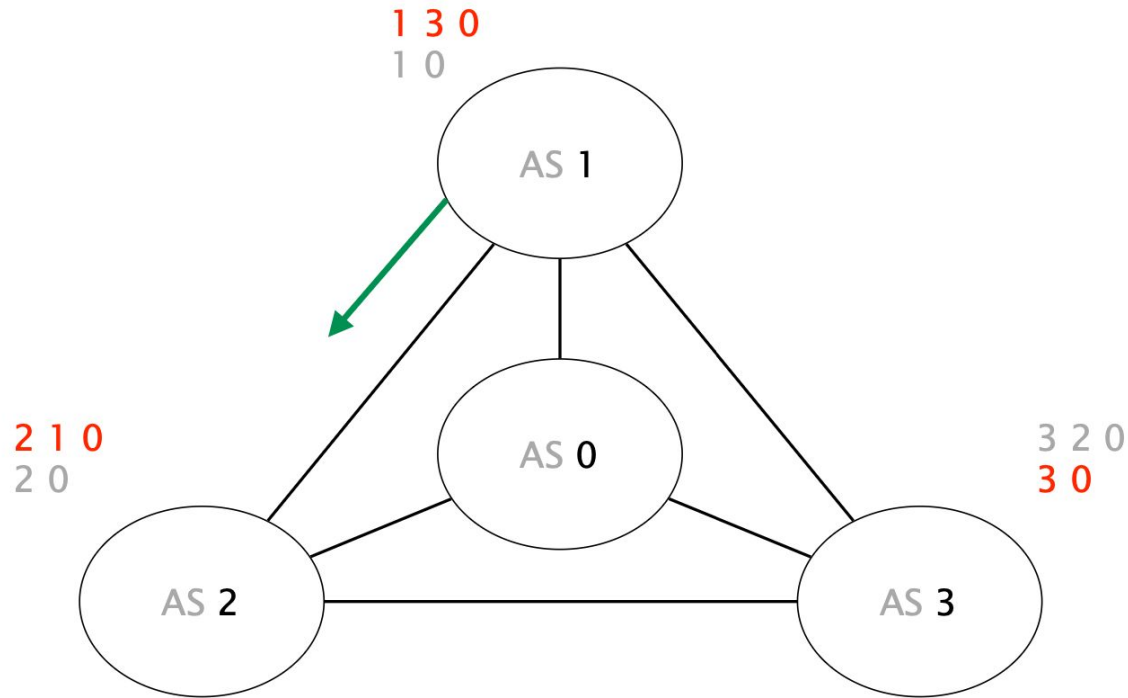2 0

AS 0

3 2 0
3 0

AS 2
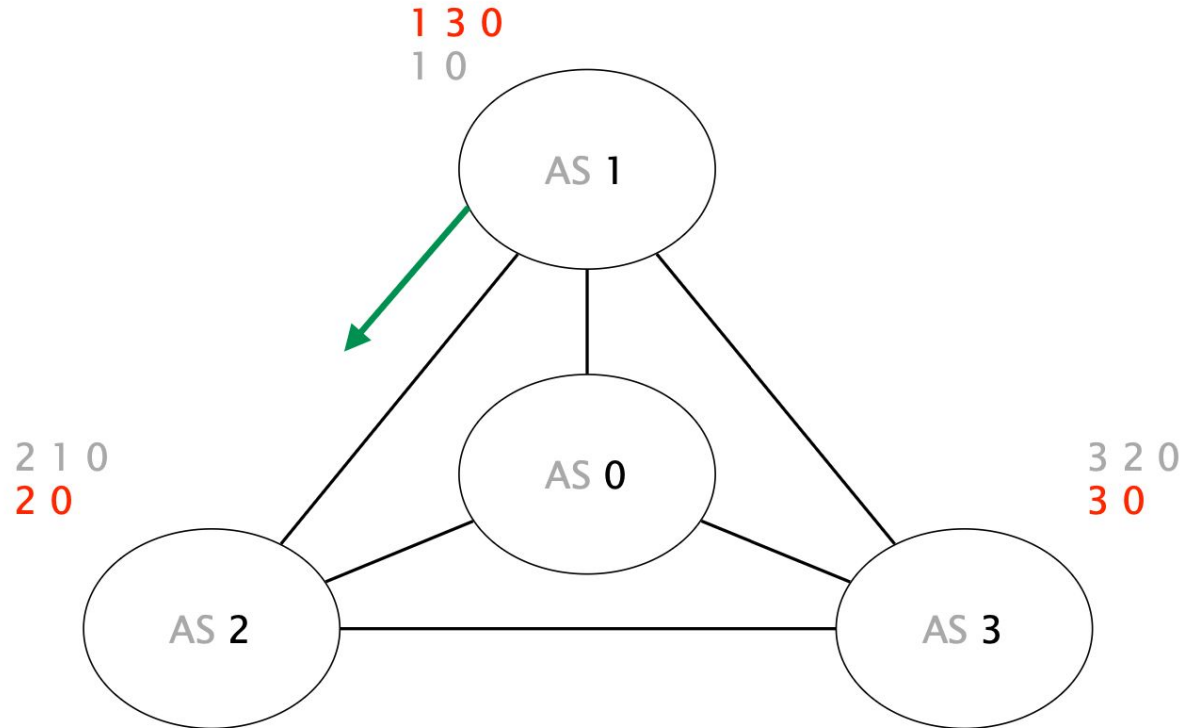
AS 3

AS 3 advertises its path to AS 1

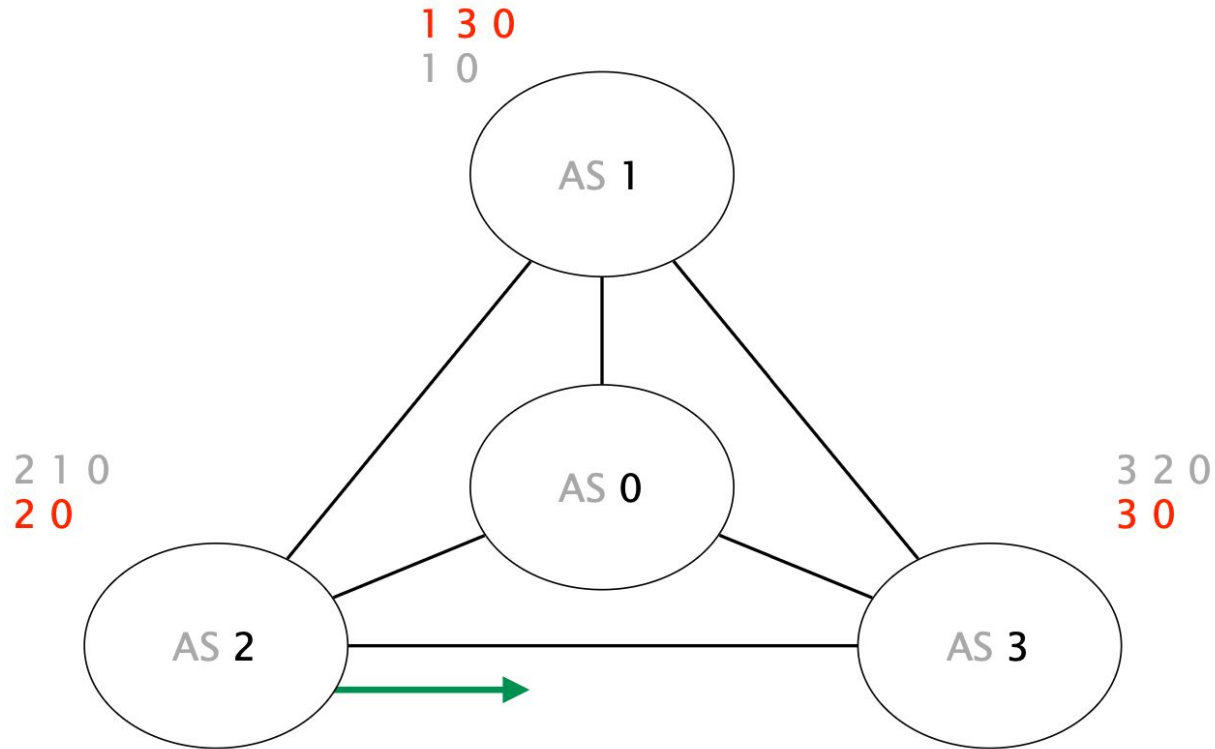# Upon reception, AS 1 switches to 1 3 0 (preferred)
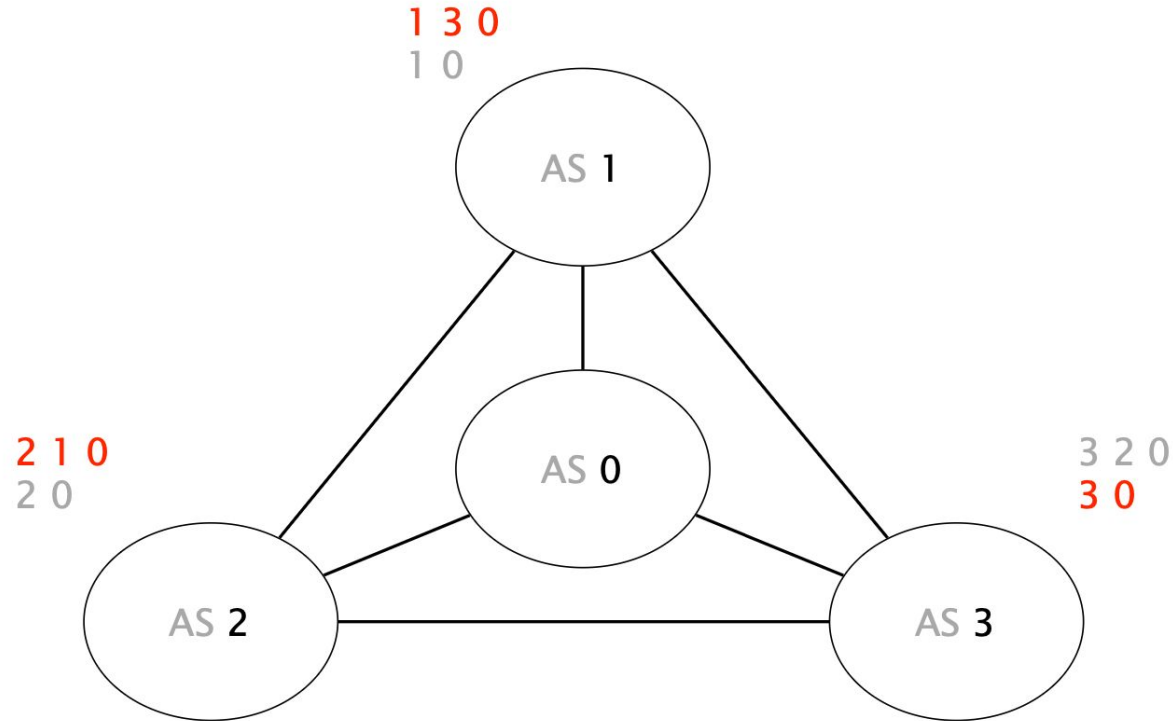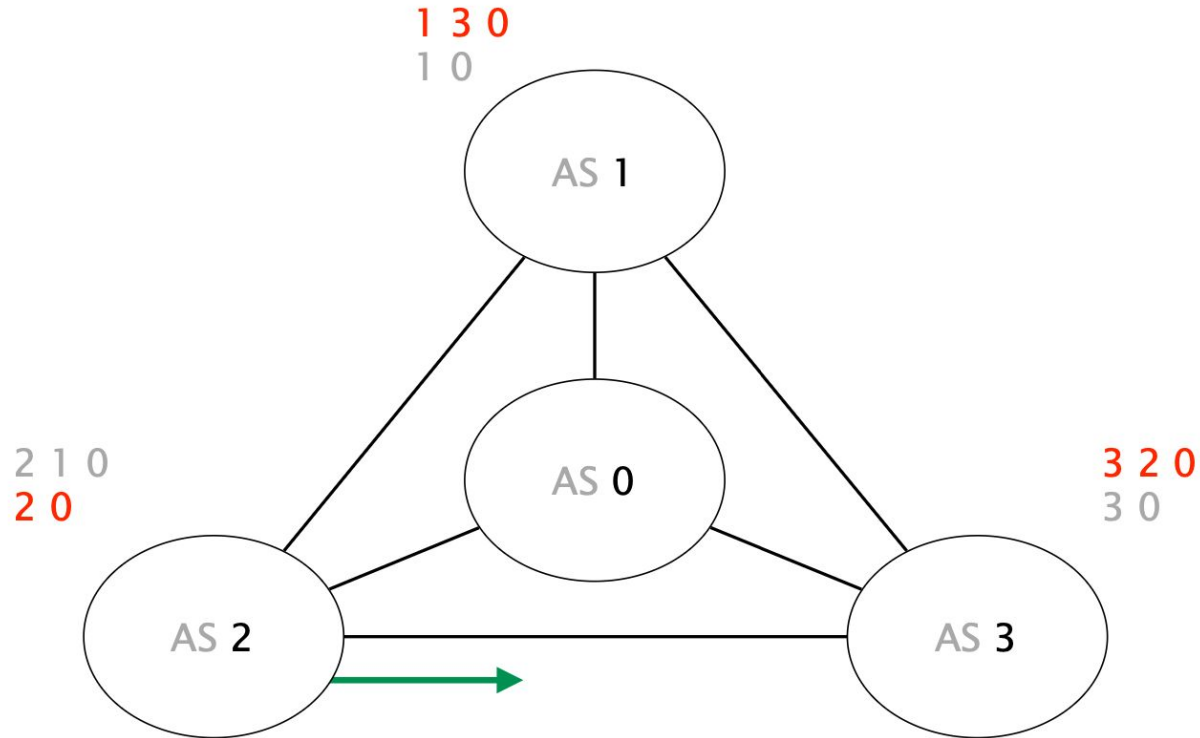
AS 1 advertises its new path 1 3 0 to AS 2

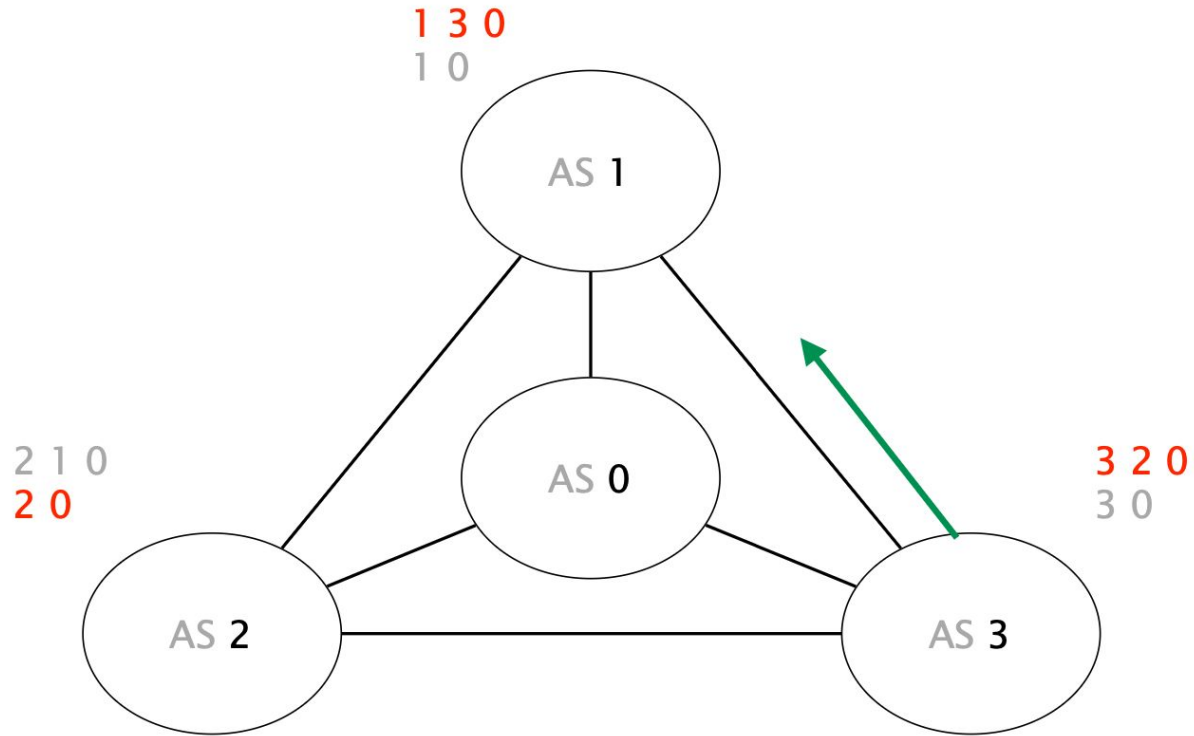# Upon reception, AS 2 reverts back to its initial path 2 0
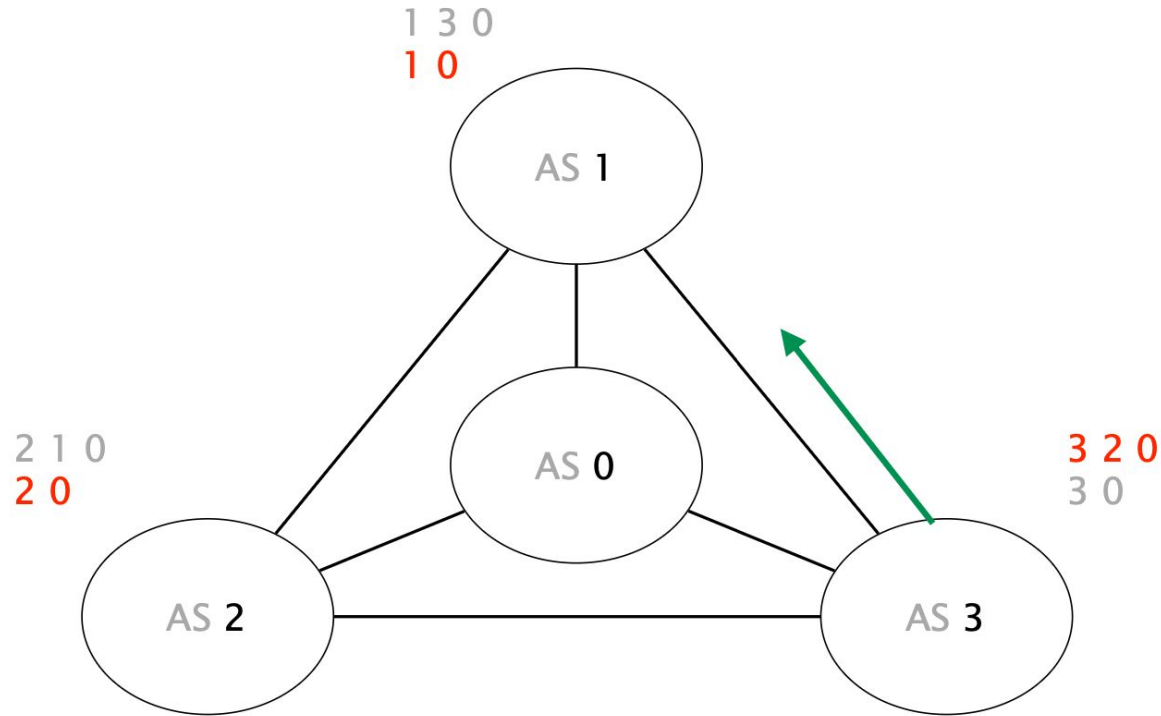
AS 2 advertises its path 2 0 to AS 3

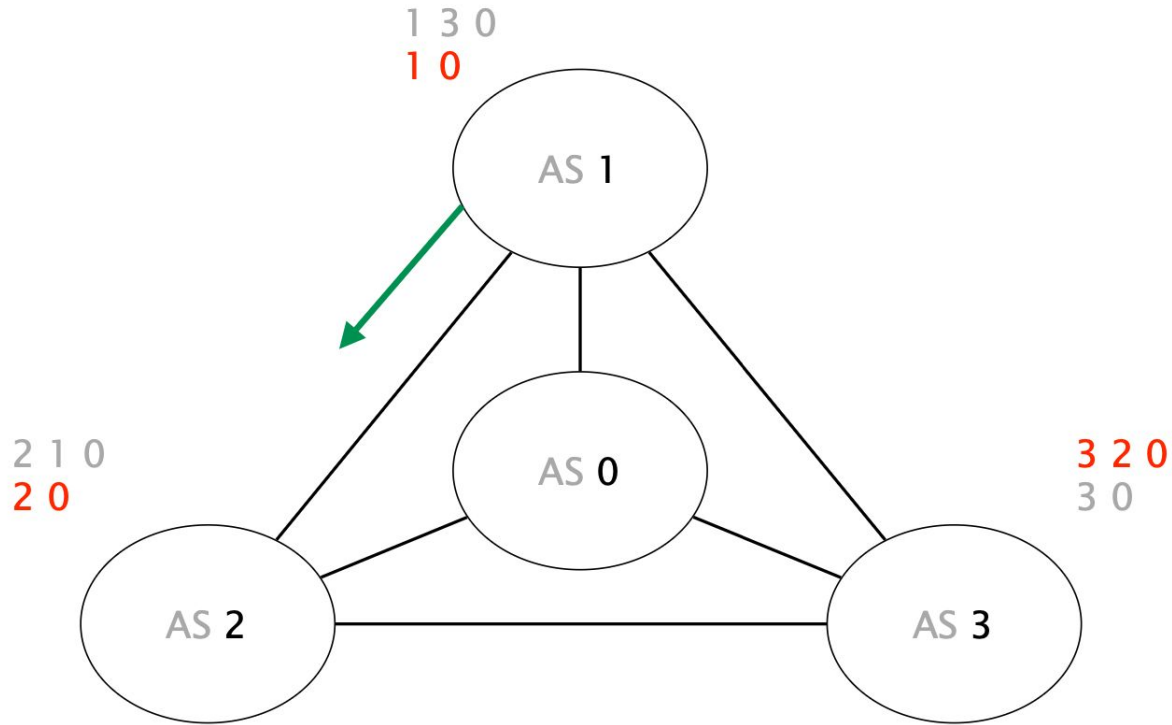# Upon reception, AS 3 switches to 3 2 0 (preferred)
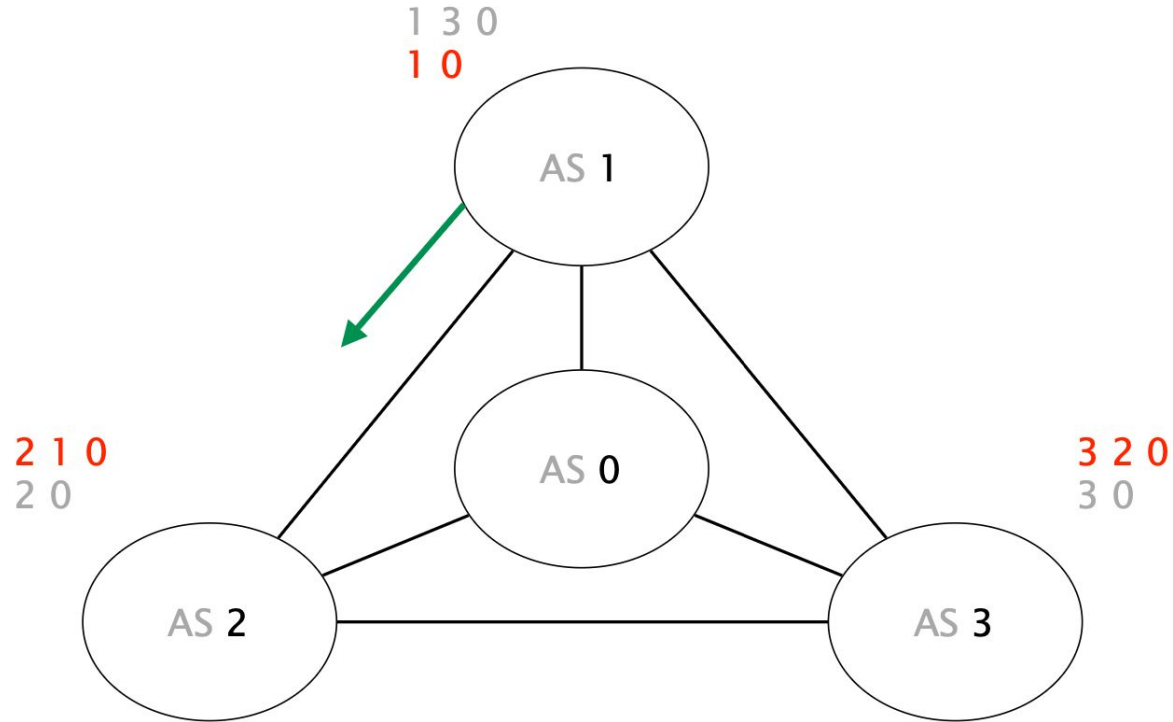
AS 3 advertises its new path 3 2 0 to AS 1

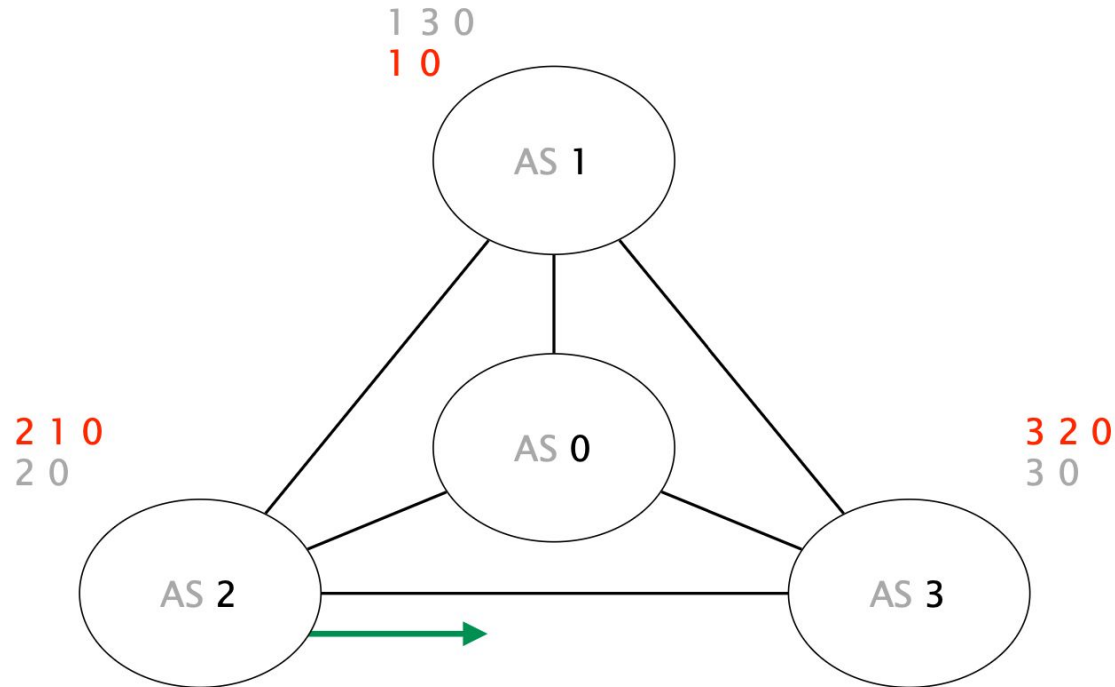# Upon reception, AS 1 reverts back to 1 0 (initial path)

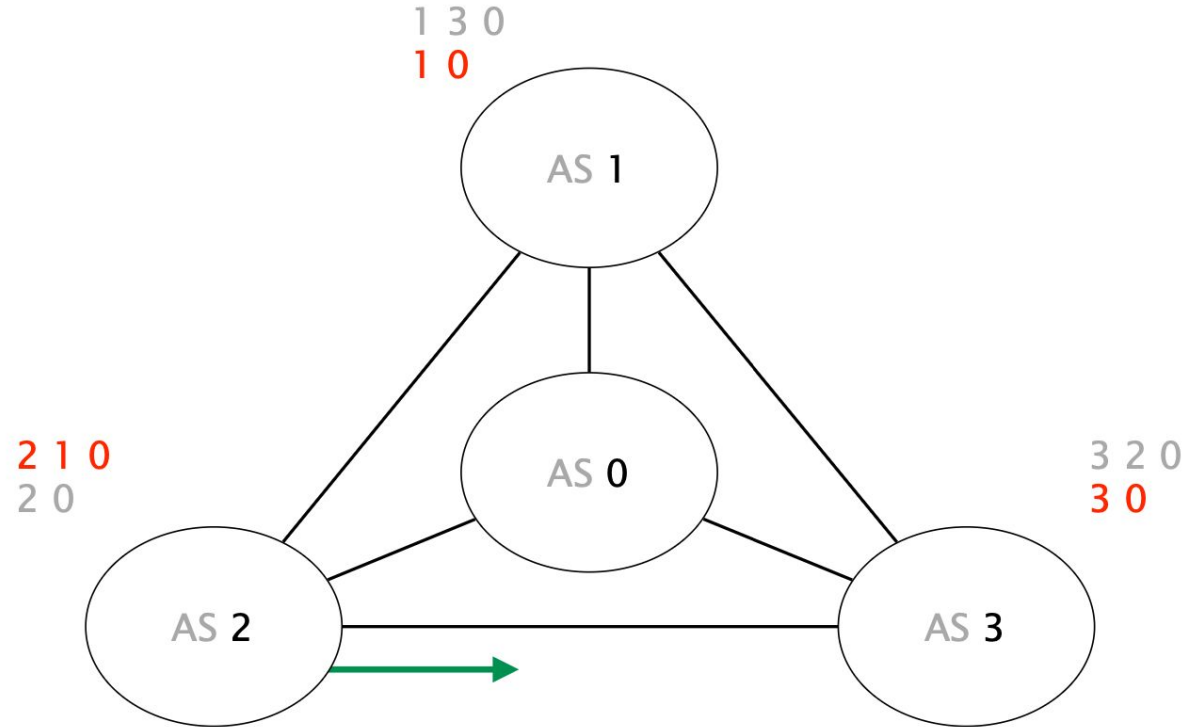AS 1 advertises its new path 1 0 to AS 2

Upon reception, AS 2 switches to 2 1 0 (preferred)
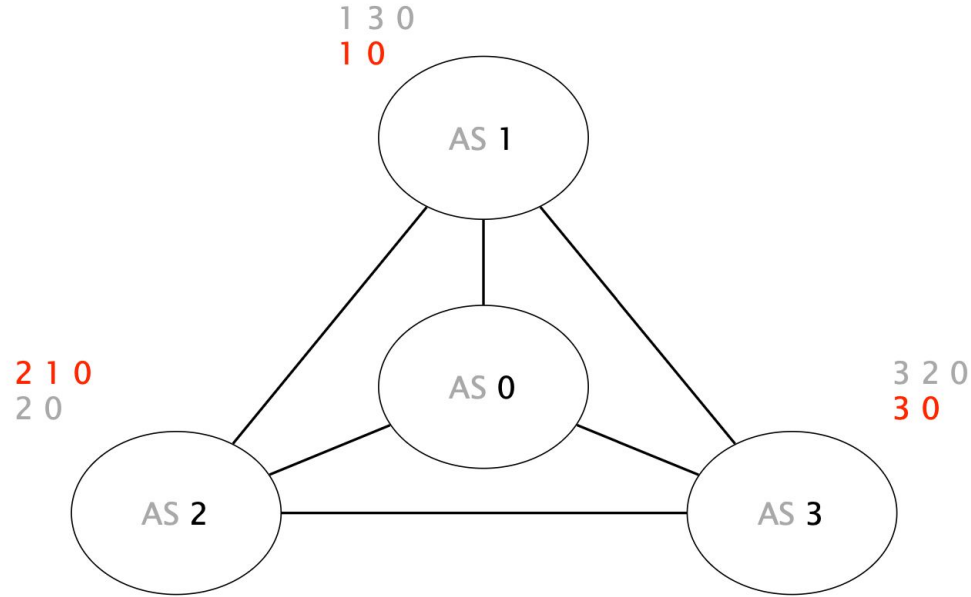
AS 2 advertises its new path 2 1 0 to AS 3

Upon reception, AS 3 switches to its initial path 3 0

1 3 0
1 0

2 1 0
2 0

3 2 0
3 0

AS 1

AS 0

AS 2

AS 3

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

Policy oscillations 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!

In practice, BGP does not oscillate "that" often

known as "Gao-Rexford" rules

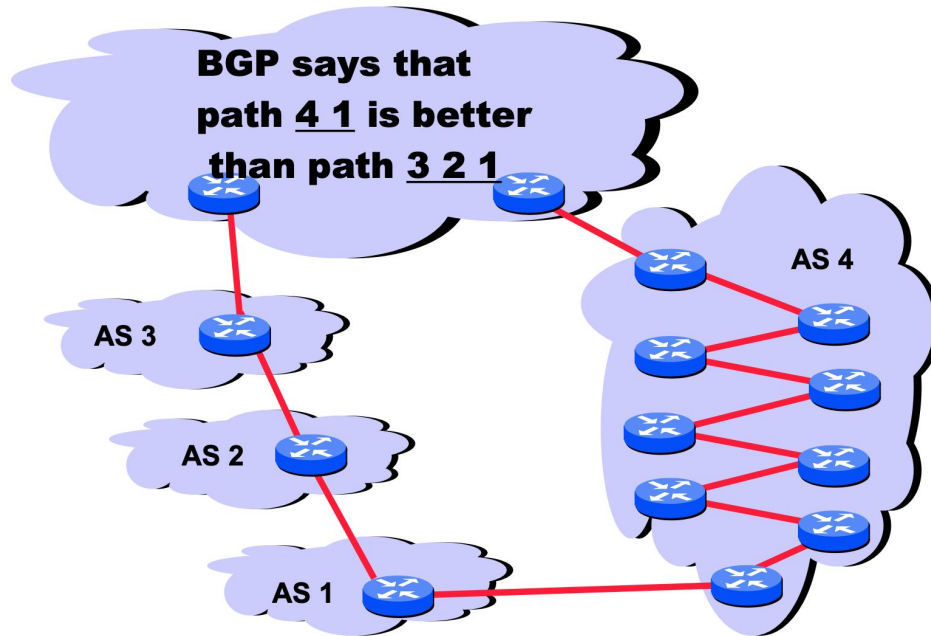**Theorem**   If all AS policies follow the cust/peer/provider rules,

BGP is guaranteed to converge

Intuition   Oscillations require "preferences cycles"

which make no economical sense

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

BGP is fragile

BGP is both "bloated" and underspecified

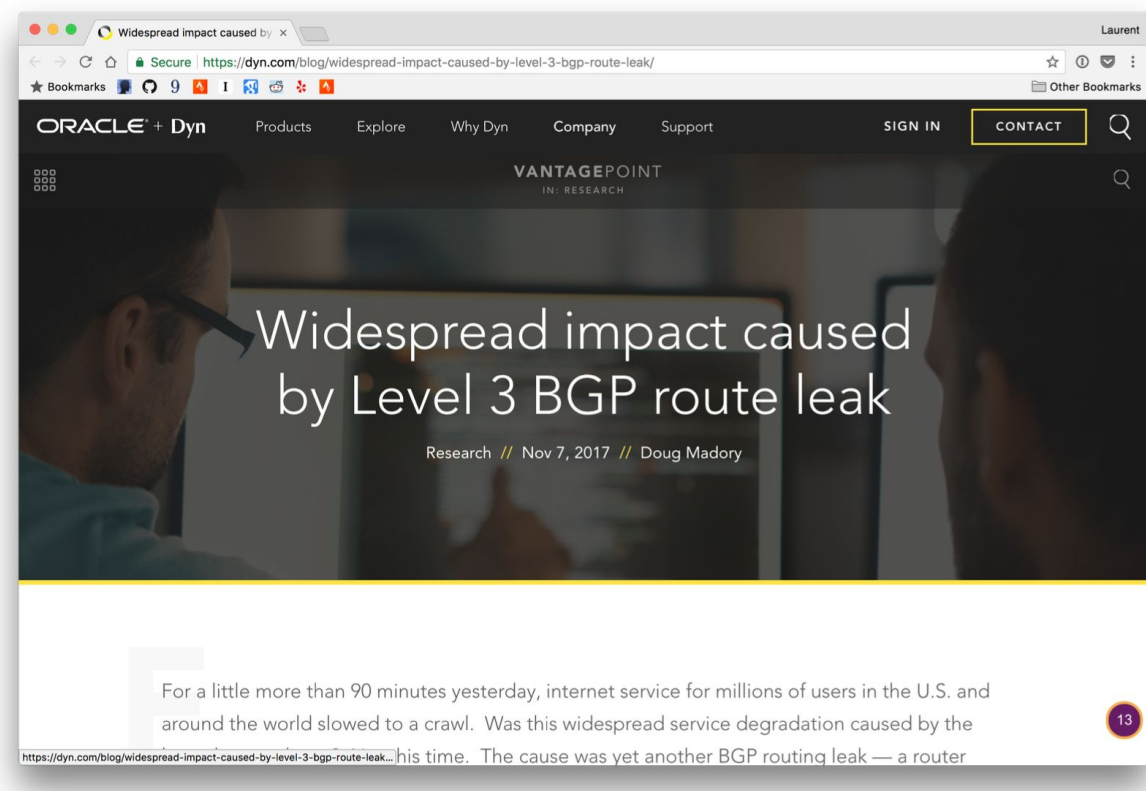lots of knobs and (sometimes, conflicting) interpretations

BGP is often manually configured

humans make mistakes, often

BGP abstraction is fundamentally flawed

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

Many outages

Widespread impact caused by Level 3 BGP route leak

Research // Nov 7, 2017 // Doug Madory

For a little more than 90 minutes yesterday, internet service for millions of users in the U.S. and around the world slowed to a crawl. Was this widespread service degradation caused by the ... his time. The cause was yet another BGP routing leak — a router

Many outages

For a little more than 90 minutes [...],

Internet service for millions of users in the U.S. and around the world slowed to a crawl.
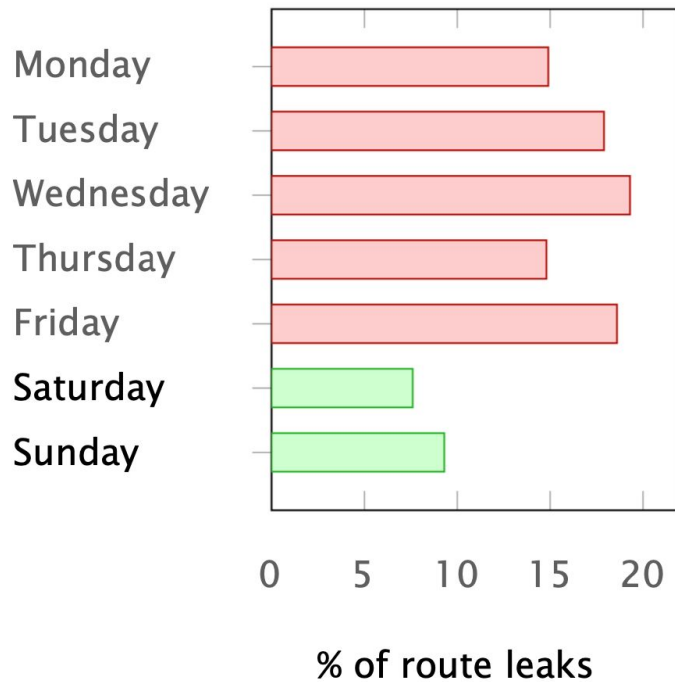
The cause was yet another BGP routing leak, a router misconfiguration directing Internet traffic from its intended path to somewhere else.

Many outages

"Human factors are responsible

for **50%** to **80%** of network outages"

Ironically, this means the Internet works better on the weekends

% of route leaks

source: Job Snijders (NTT)

BGP continues to have many problems

- Instability
  - Route flapping (network x.y/z goes down... tell everyone)
  - Not guaranteed to converge, NP-hard to tell if it does
- Scalability still a problem
  - ~485,000 network prefixes in default-free table today
  - Tension: Want to manage traffic to very specific networks (eg. multihomed content providers) but also want to aggregate information.
- Performance
  - Non-optimal, doesn't balance load across paths

The world of BGP is changing

- "Flattening" of the Internet
- ISPs are now eyeballs talking to content networks
  - e.g., Spectrum 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

Review

An Ideal Network Topology Fulfills Three Requirements

- Resilient to failures
    - \>1 path should exist between each node

- Allow sharing (to be feasible and cost-effective)
    - The number of links should be kept low

- Provide adequate capacity
    - Links should not be too small

Two Approaches to Shared Links

- Reservation (Circuit switching)
    - Reserve what you need in advance

- On-demand (Packet switching)
    - Send data when needed

Pros and Cons of Circuit Switching

Pros

- Predictable performance

- Simple and fast switching once circuit established

Cons

- Inefficient if traffic is bursty or short lived

- Complex circuit setup / teardown

- Requires new circuit upon failure

# Pros and Cons of Packet Switching

Pros

- Efficient use of resources

- Simple to implement

- Route around problems

Cons

- Unpredictable performance

- Requires buffer management and congestion control
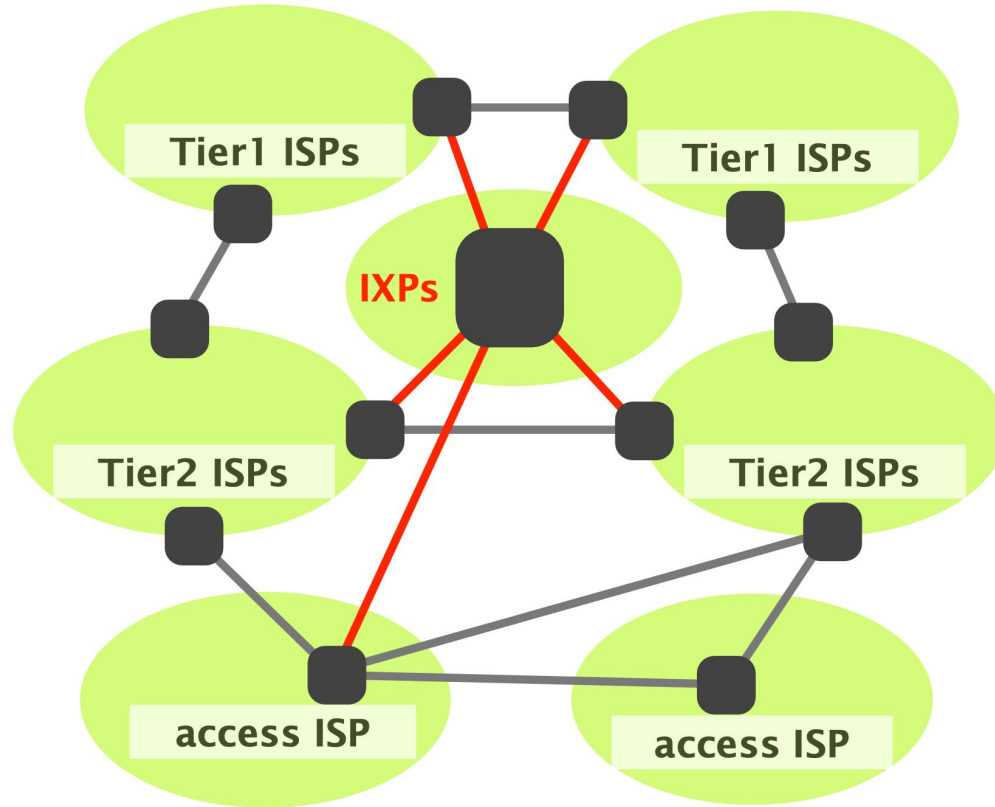
The Internet Hierarchy

- Tier-1 (international)
  - Have no provider

- Tier-2 (national)
  - Provide transit to tier-3s
  - Have at least one provider

- Tier-3 (local)
  - Do not provide any transit
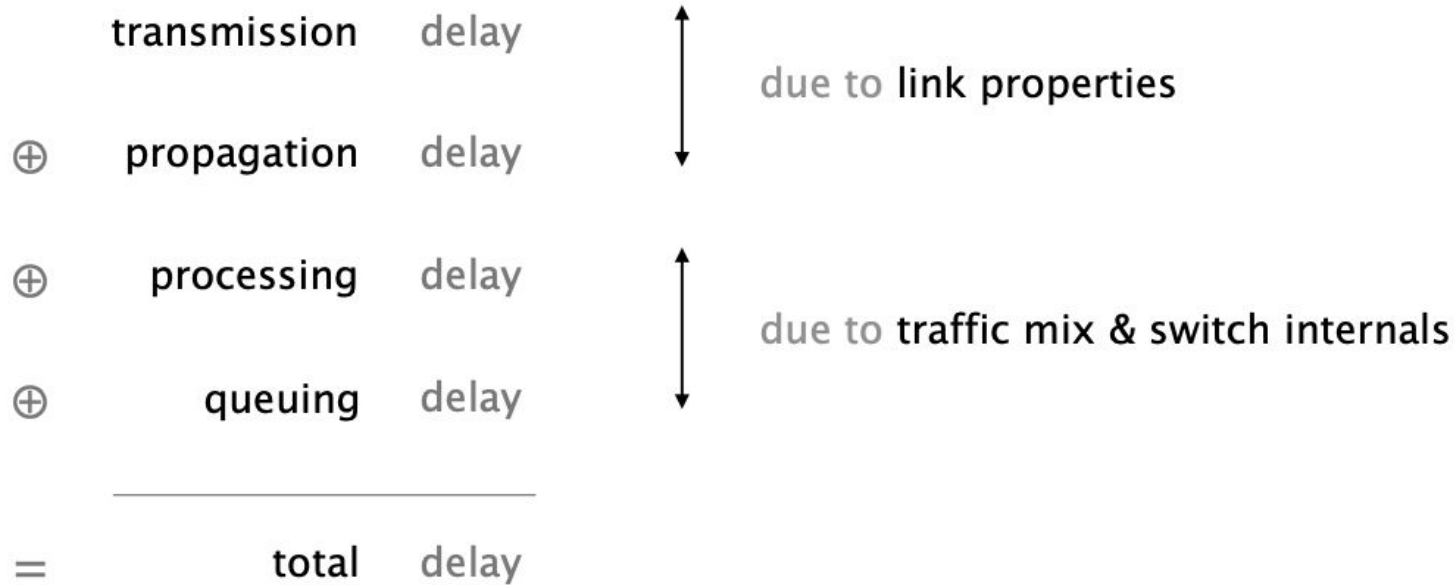 - Have at least one provider
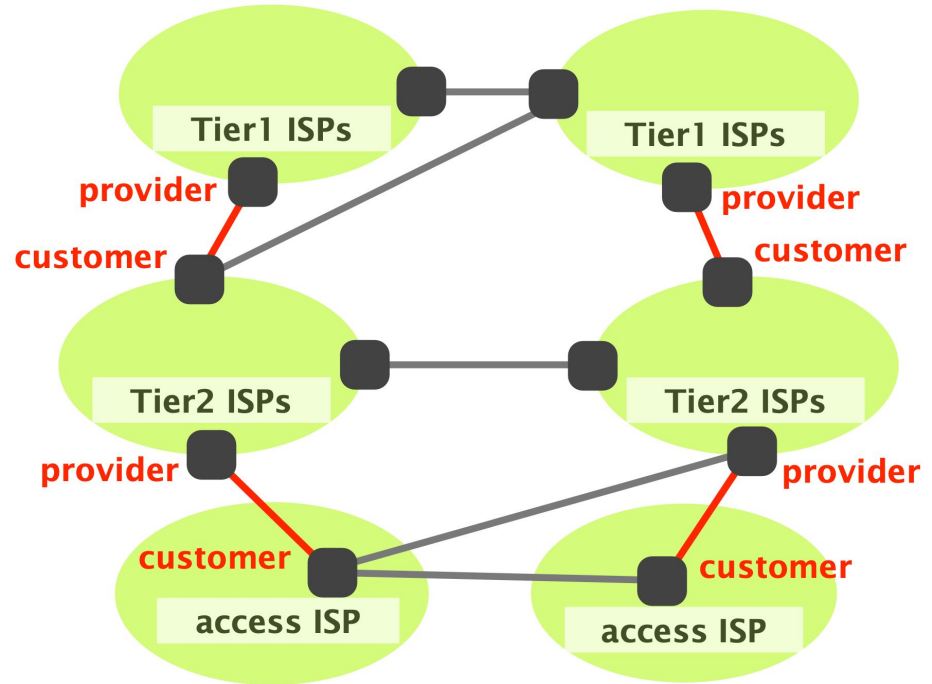
Some Networks Connect Directly to One Another to Save $$$
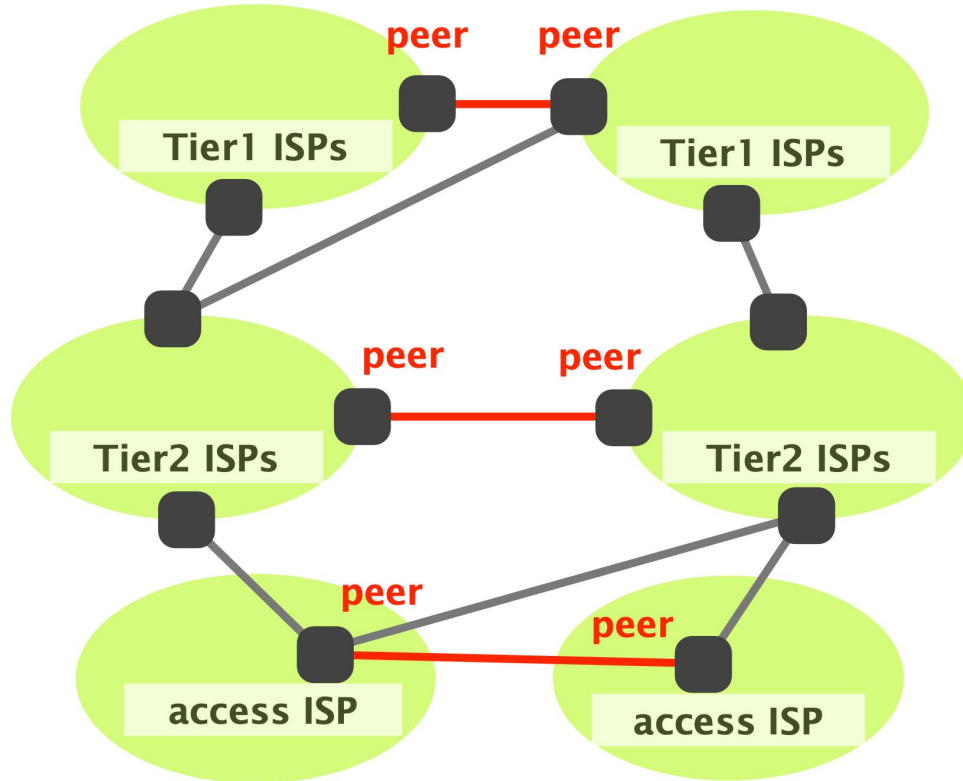
Interconnecting One-by-One is not Efficient / Cost Effective

Every Packet Suffers from Several Types of Delay _at Each Node_

transmission   delay

⊕   propagation   delay

due to **link properties**

⊕   processing   delay

⊕   queuing   delay

due to **traffic mix & switch internals**

=   total   delay

Transmission Delay is the Time Required to Push all Bits onto the Link

Transmission delay = $\dfrac{\text{packet size} \quad [\#bits]}{\text{link bandwidth} \quad [\#bits/sec]}$

[sec]

Example $\qquad \dfrac{1000 \text{ bits}}{100 \text{ Gbps}} \qquad$ 10 ns

Propagation Delay is the Time Required a Bit to Travel to the End of the Link

Propagation delay $=$ $\dfrac{\text{link length}}{\text{propagation speed}}$ $\dfrac{\text{[m]}}{\text{[m/sec]}}$

[sec]

(fraction of speed of light)

Example $\qquad \dfrac{30\,000\ m}{2\times10^{8}\ m/sec}$ $\qquad$ 150 µsec

(speed of light in fiber)

Queueing Delay is the Amount of Time a Packet Waits in a Buffer to be Transmitted

- Hardest to evaluate
  - Varies from packet to packet

- Characterized by statistical measures
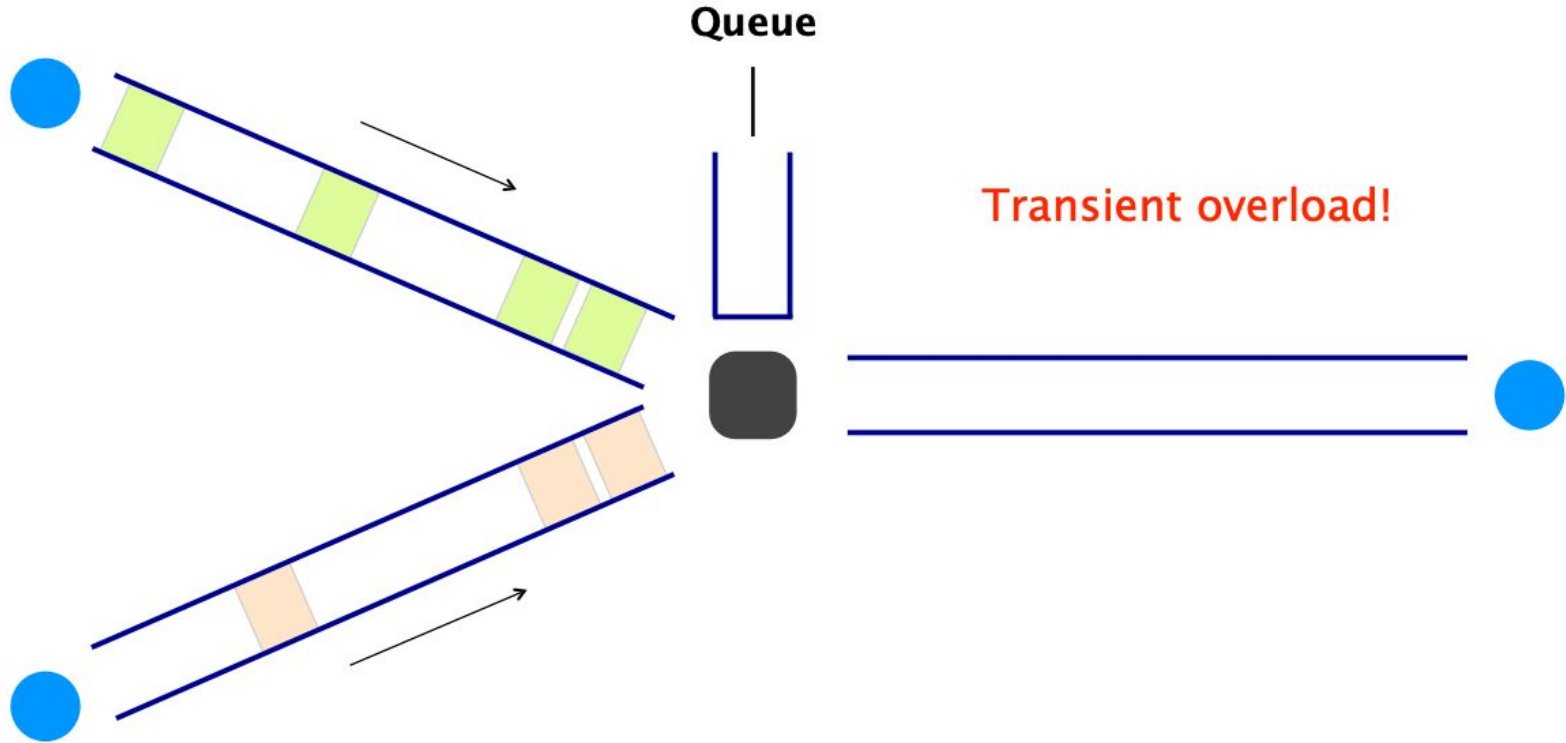 - E.g., average delay, variance, probability of exceeding x

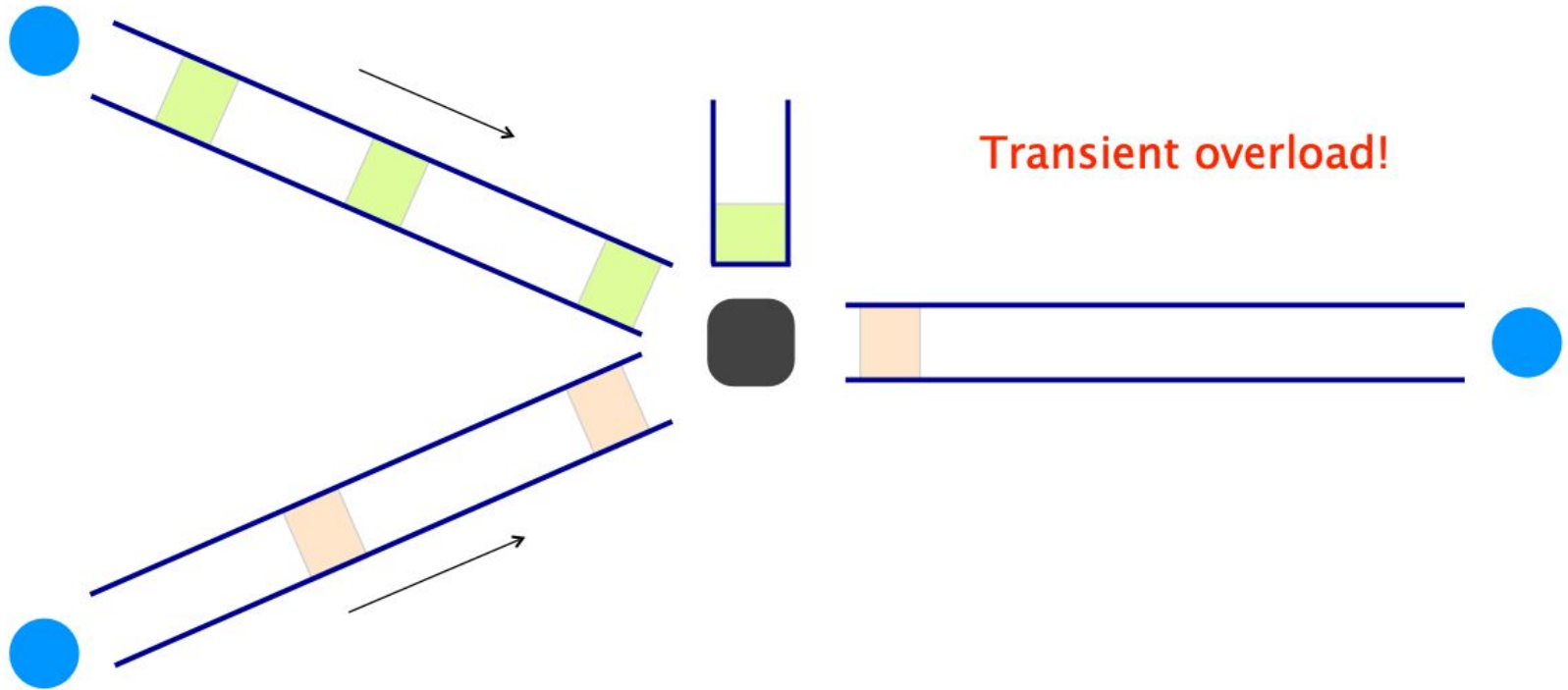# Queueing Delay Depends on the Traffic Pattern
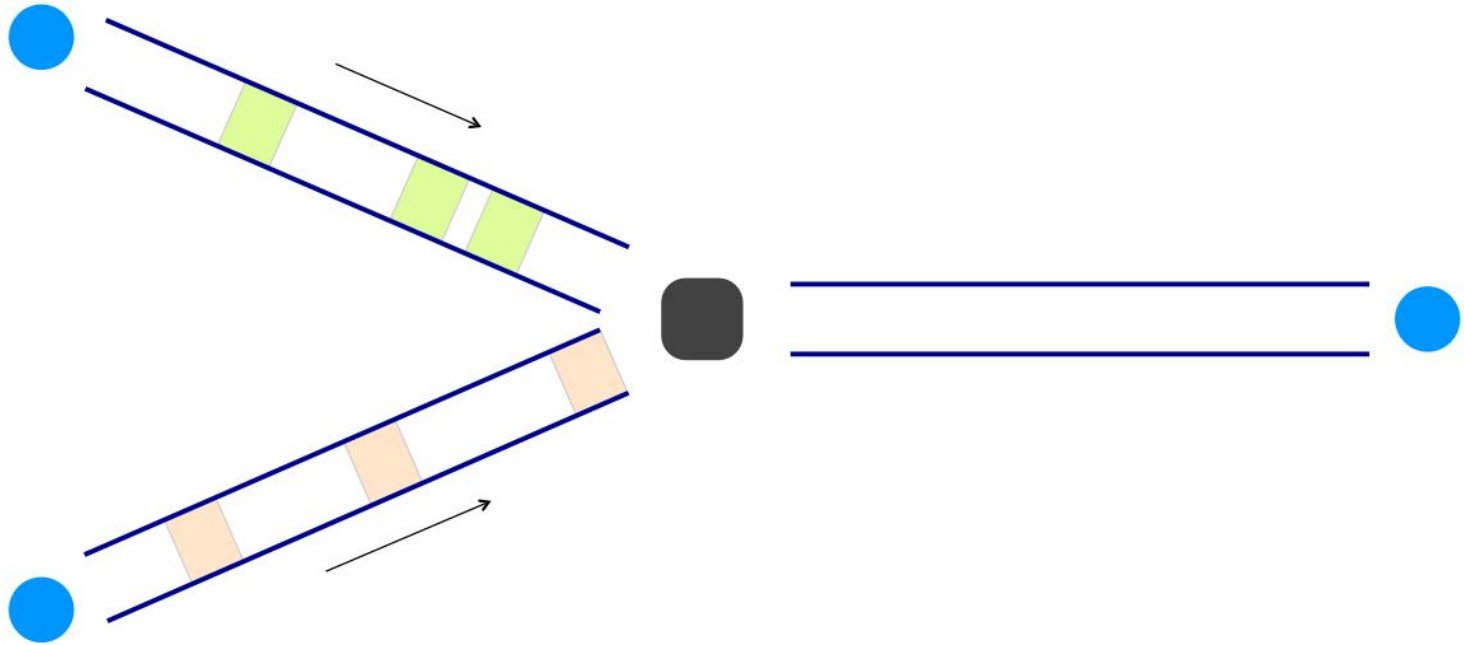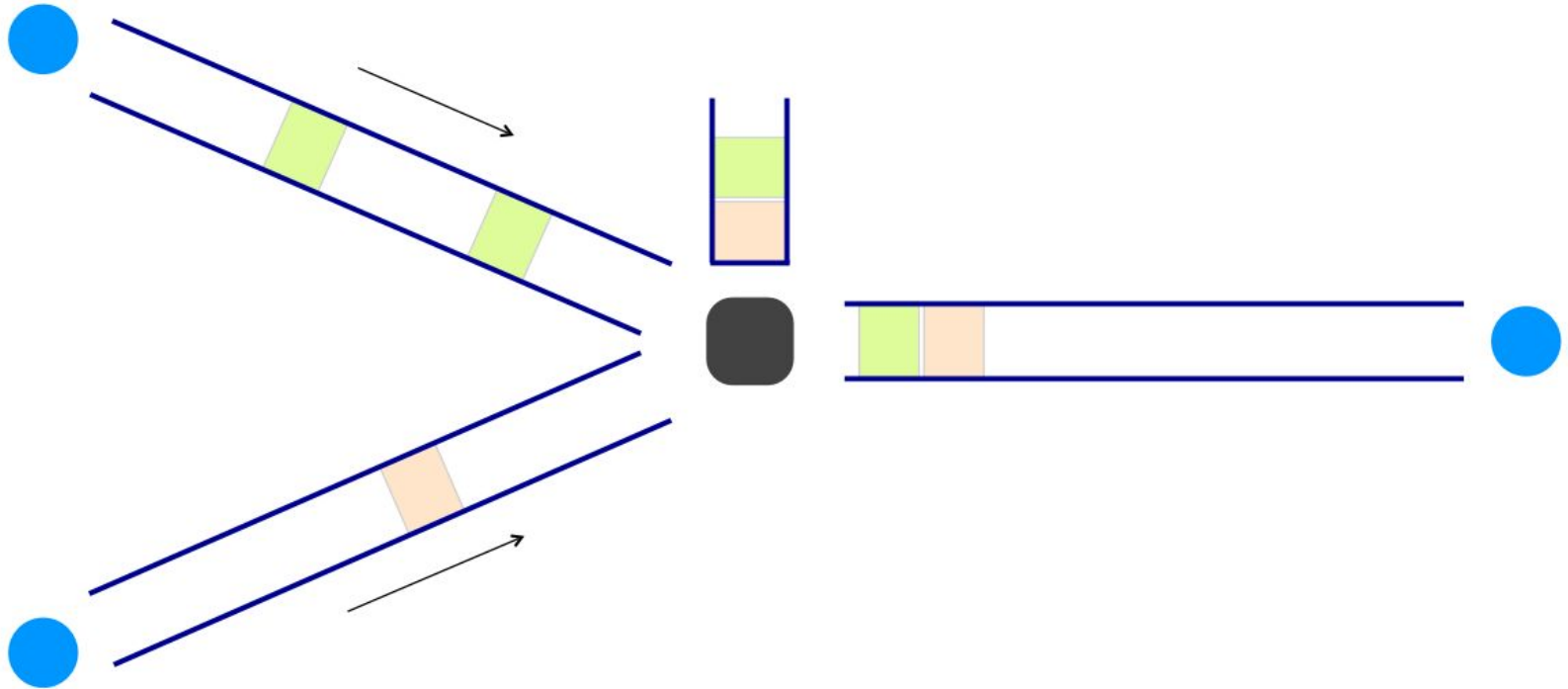
Queueing Delay Depends on the Traffic Pattern

No overload

Queueing Delay Depends on the Traffic Pattern

Queueing Delay Depends on the Traffic Pattern

Transient overload!
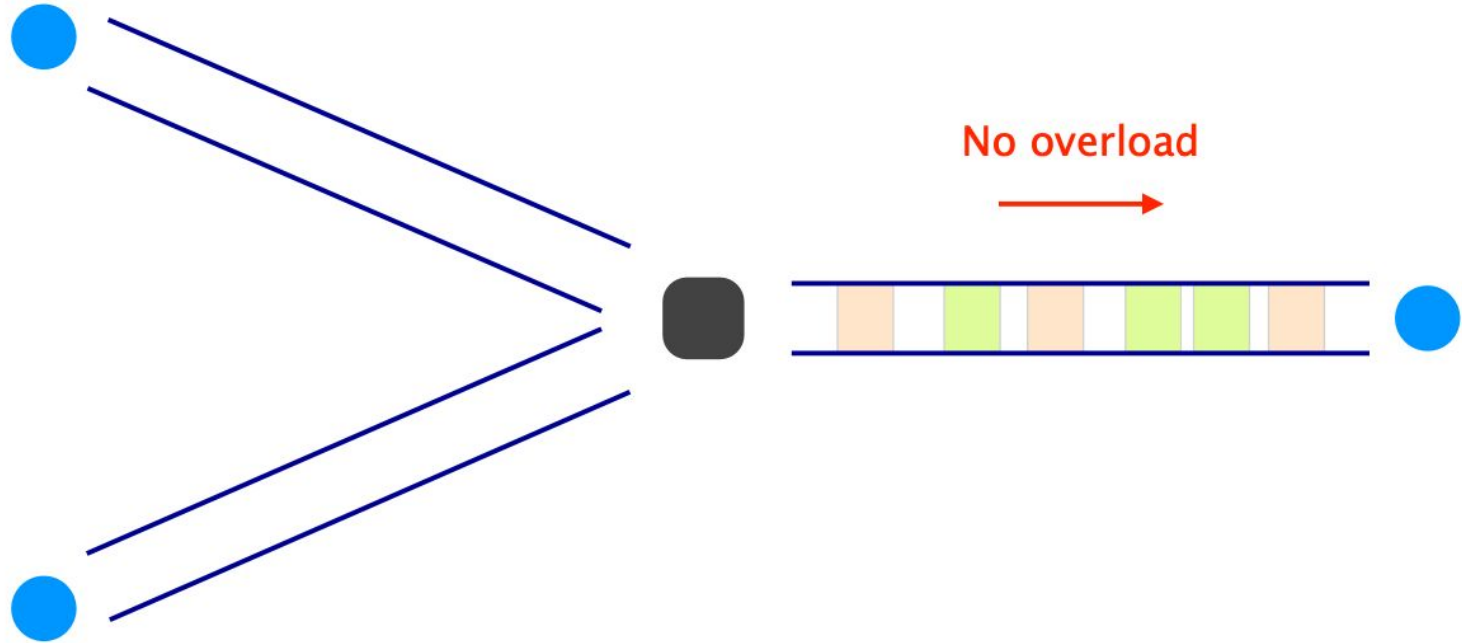
Queueing Delay Depends on the Traffic Pattern

Queueing Delay is Dependent on the Traffic Pattern

- Arrival rate at the queue
- Transmission rate of outgoing link
- Traffic burstiness

average packet arrival rate	a	[packet/sec]
transmission rate of outgoing link	R	[bit/sec]
fixed packets length	L	[bit]
average bits arrival rate	La	[bit/sec]
traffic intensity	La/R	