Communications Network Design

lecture 20

Matthew Roughan
<matthew.roughan@adelaide.edu.au>

Discipline of Applied Mathematics
School of Mathematical Sciences
University of Adelaide

March 2, 2009

BGP

BGP (the Border Gateway Protocol) version 4 is the defacto inter-domain routing protocol.

BGP

- Border Gateway Protocol [1]
- BGP has to support all of this “policy” stuff
  - generically called **policy** based routing
  - I will use the term **path-vector** routing
- incredibly flexible
- large, complex dynamic system
  - hard to understand
  - hard to predict
  - hard to optimize

Path Vector

- similar procedure to distance vector
  - transmission of updates is similar
  - nodes select best route to transmit to neighbours
  - metric for choosing paths is not purely distance based
  - added loop detection
- **choice is based on policy**
- distance vector is a special case
  - metric is distance
  - simple uniform policy (shortest paths)
  - guaranteed convergence
- unlike distance vector, path vector is not guaranteed to converge

BGP means

- RFC 1771
- optional extensions:
  - RFC 1997 BGP Communities Attribute
  - RFC 2439 BGP Route Flap Damping
  - RFC 2796 BGP Route Reflection
  - RFC 3065 AS Confederations for BGP
- implementation details
  - timers, proprietary extensions (`WEIGHT`), ...
- routing policy configuration languages
  - vendor specific
- current practises in management of inter-domain routing (e.g. RFC 1772, RFC 2270, ...)

How BGP works

Messages sent between "peers"

- note **peer** just means two routers that communicate
  - not ISP "peers"
- note BGP peers don't have to be adjacent!
- **hard-state** protocol (no periodic updates)
  - scalability requirement

Types of message

- open: establish peering session
- keep alive: handshake at regular intervals
- notification: shuts down peering session
- **update:** announcing or withdrawing routes
  - route to a prefix

BGP attributes

- route announcements = prefix + attributes
  - not all attributes needed for all announcements
- BGP gives **attributes** to routes it distributes
- important attributes

2: AS-path (primarily to avoid loops)

3: next hop

4: Multi-Exit Discriminator (MED)

5: local pref

8: community

9: originator ID

How BGP works

Policy is implemented by a set of rules

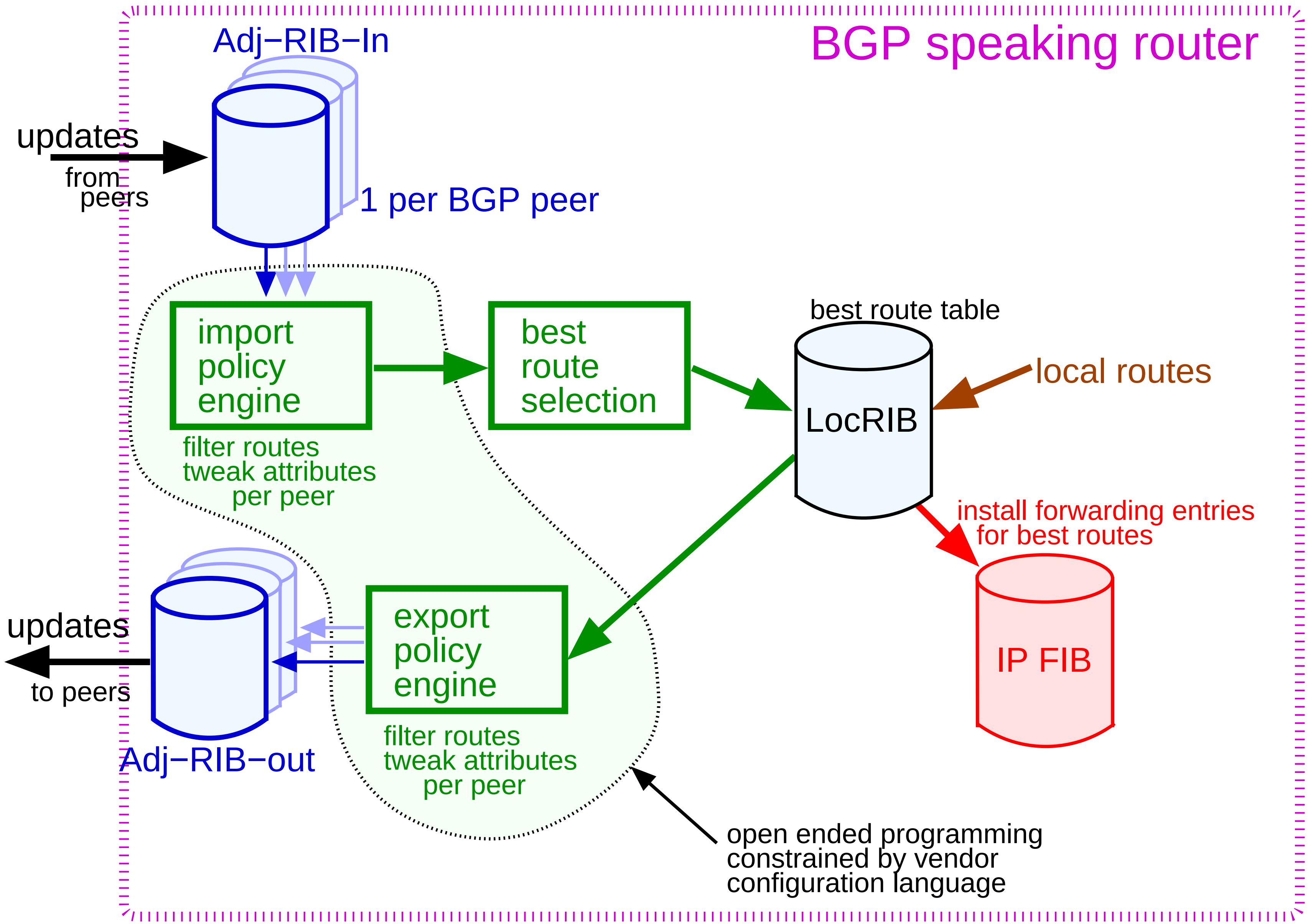
- import rules
  - can ignore routes by filtering them on input
  - changing route attributes
    - make the route appear more attractive
- export rules
  - can prevent customer from using a route by filtering export of rules
    - don't tell someone about a route, and they can't use it
  - by changing route attributes on export, we can make a route appear less attractive

BGP decision process

To know how to change routes to be more or less attractive, we need to know how BGP makes decisions. A simplified (ignoring vendor specific bits) version of that process follows (in order of precedence).

- don't select paths with inaccessible next hops
- prefer the path with higher local preference
- prefer the path the shortest AS-path
- prefer the route with the lowest MED
- prefer the route that can be reached through the closest IGP neighbour (hot potato)
- prefer the route that has been around the longest
- tie-break: prefer the path with the lowest IP address, as specified by the BGP router ID.

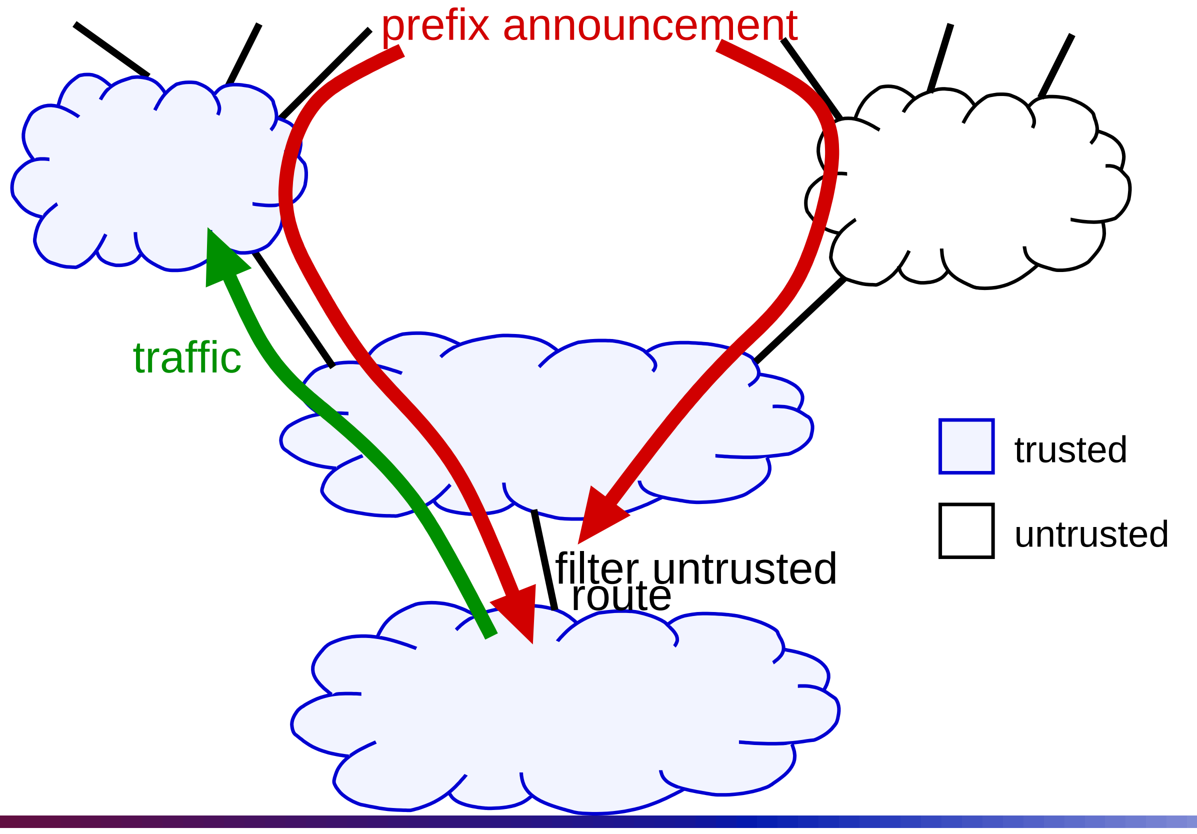
How BGP works



Example 1

Filtering inputs

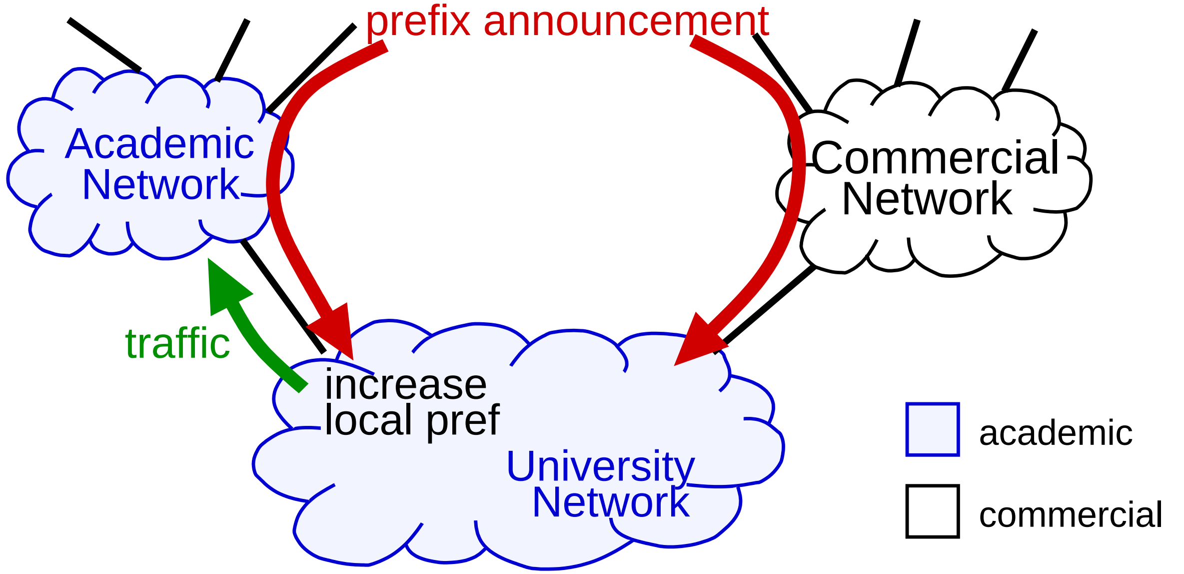
- we don't use "untrusted" networks
    - filter out any routes that cross untrusted networks



Example 2

Changing route attributes (on input)

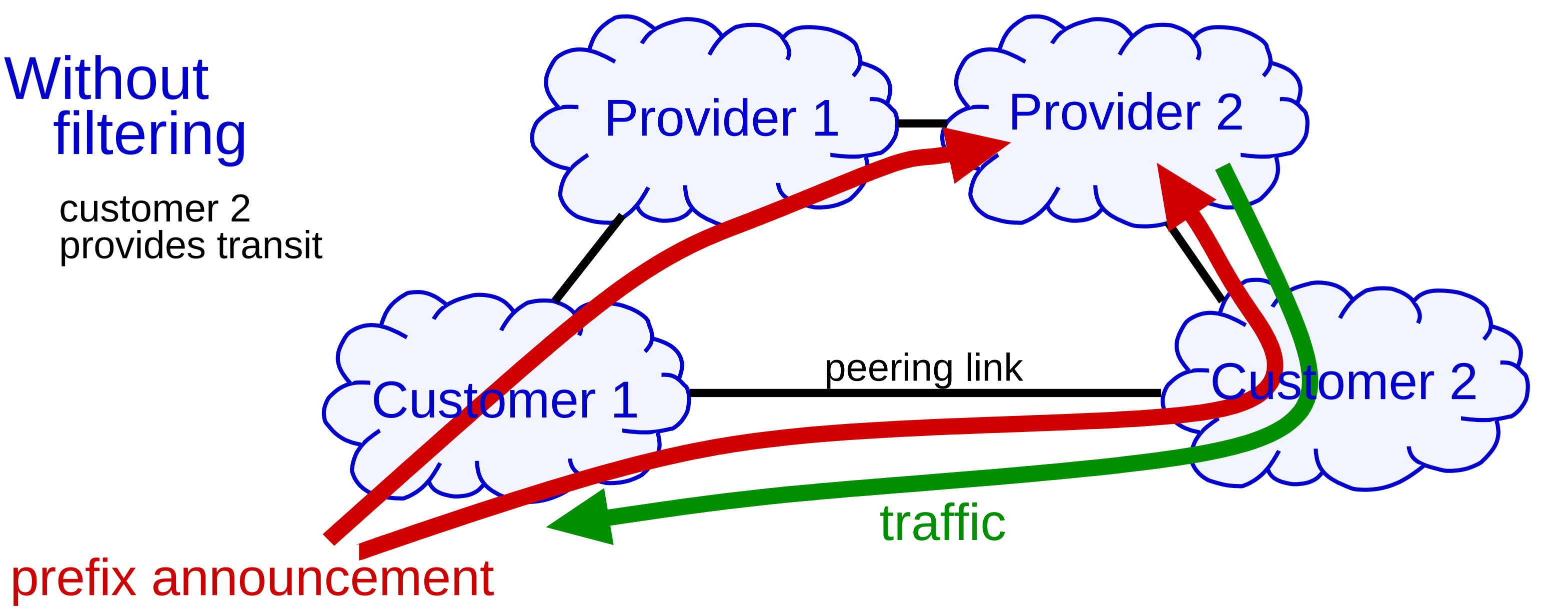
- university network prefers academic network routes to commercial provider
  - when academic network route is input give it a high local pref.
  - we prefer routes with high local pref



Example 3

Filtering of outputs

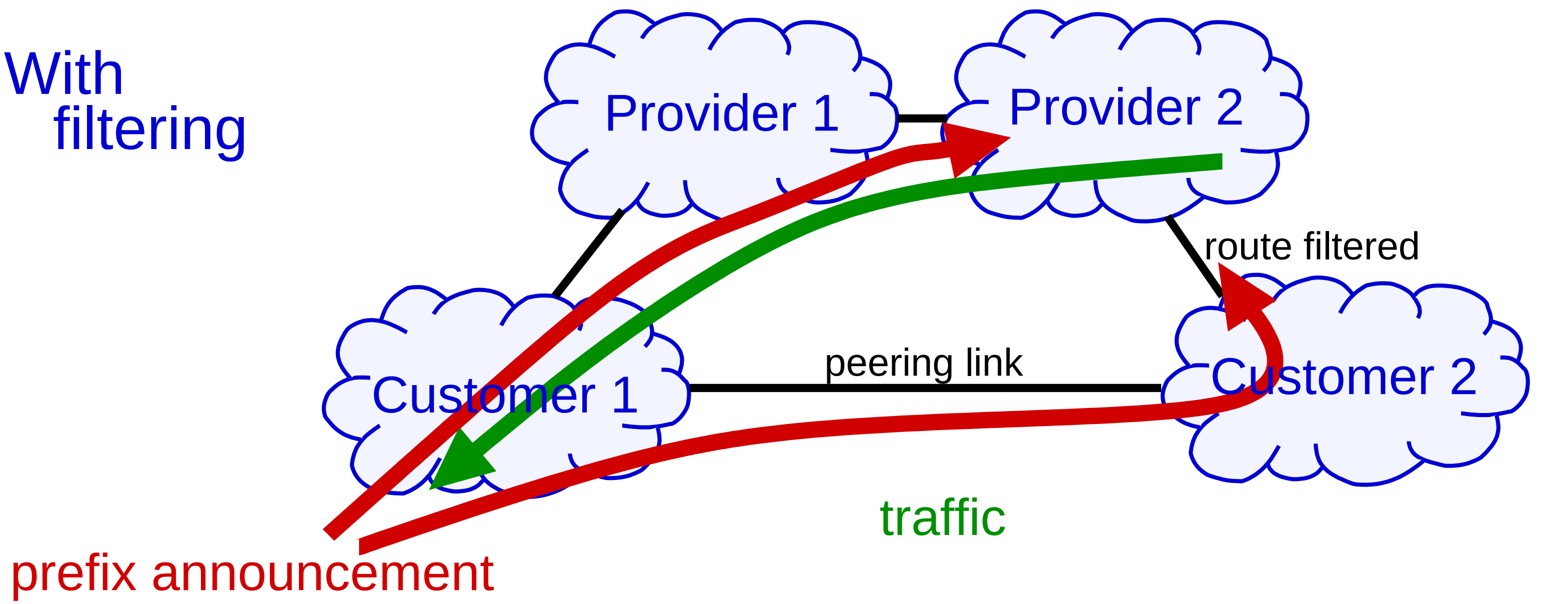
- an ISP doesn't provide transit to peers
  - don't send routes learnt from peers, or providers to our peers or providers
  - only send customer routes to peers, so they will only route traffic to our customers through us



Example 3

Filtering of outputs

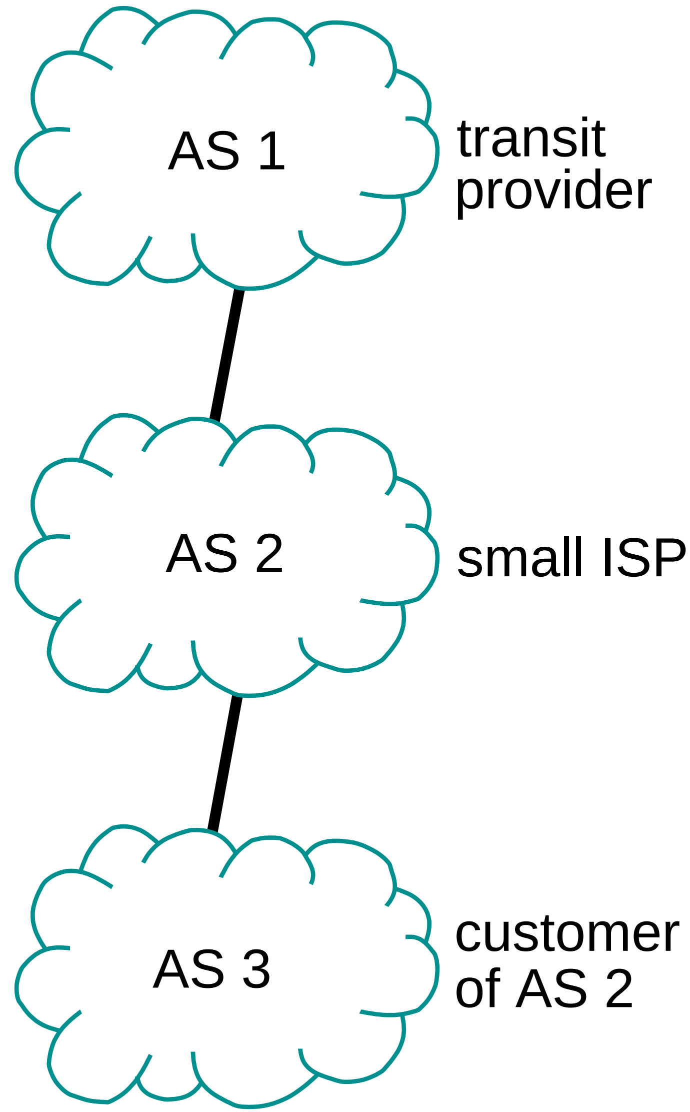
- an ISP doesn't provide transit to peers
  - don't send routes learnt from peers, or providers to our peers or providers
  - only send customer routes to peers, so they will only route traffic to our customers through us



Examples

Example (from RFC 2650) of policy for ***AS2***

```
aut-num:   AS2
as-name:   CAT-NET
descr:     Catatonic State University
import:    from AS1 accept ANY
import:    from AS3 accept <^AS3+$>
export:    to AS3 announce ANY
export:    to AS1 announce AS2 AS3
admin-c:   AO36-RIPE
tech-c:    CO19-RIPE
mnt-by:    OPS4-RIPE
changed:   orangeripe.net
source:    RIPE
```



Does BGP solve SPF?

- what is SPF here?
  - prefer routes with shorter AS-path
  - but AS path doesn't have **anything** to do with physical distance
- policies may prefer longer AS-paths explicitly
  - e.g. prefer cheaper transit charges
- all else being equal we prefer shorter IGP distances
  - hot potato routing
  - does do some distance minimization, but not SPF

No, BGP does not solve SPF

- except in limited situations

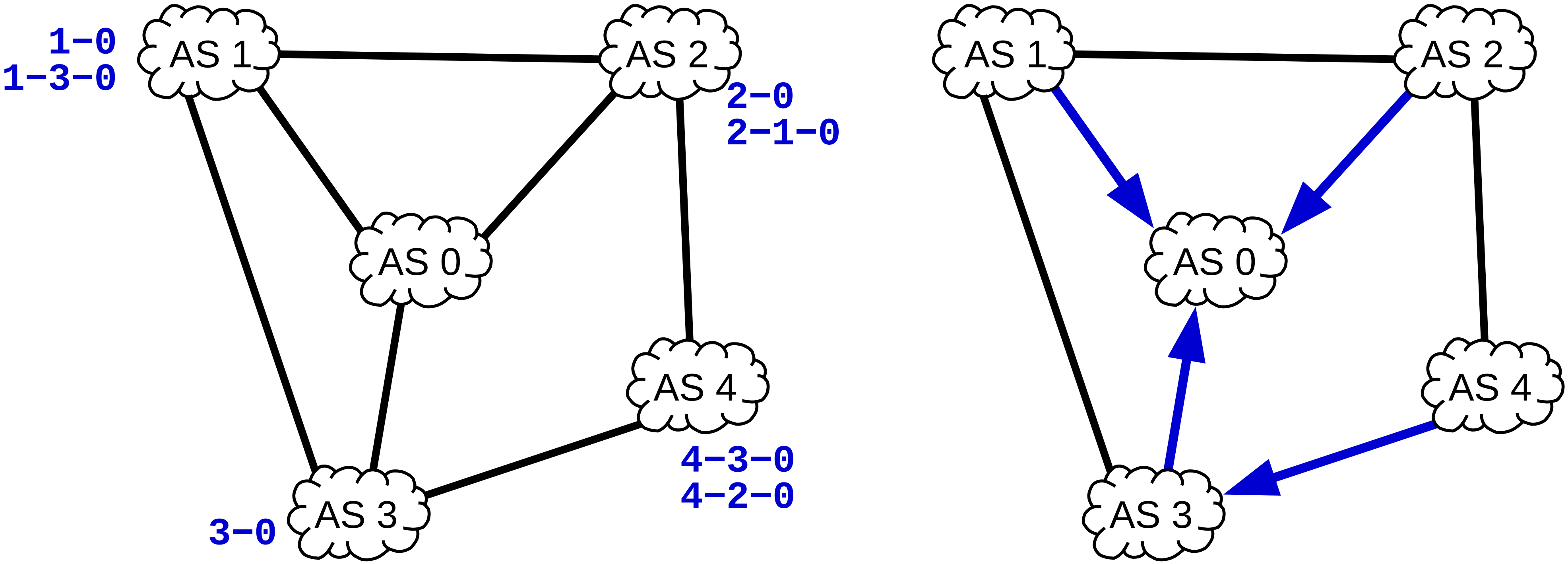
Does BGP try to optimize?

- BGP is trying to satisfy policy
- what is policy?
  - a bunch of rules
  - usually these rules are related to an optimization objective
    - e.g. reduce load (and congestion) on our network
    - e.g. reduce transit costs
- so BGP is solving an optimization problem
  - many individuals (ASes)
  - each has its own different optimization objectives, and constraints
  - objectives are all coupled
- maybe the largest distributed computations on the planet.

Stable Paths Problem

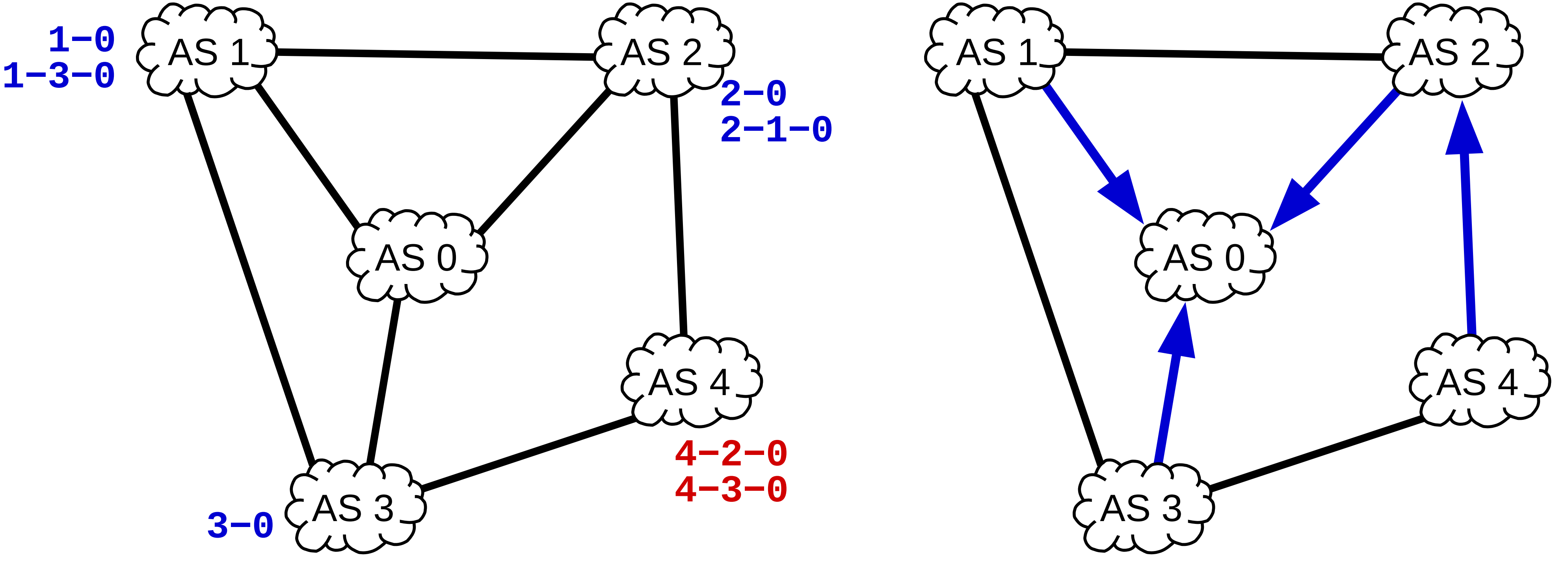
- we call this optimization problem
  - the **stable-paths problem** [2, 3]
  - looking for a set of stable paths which match policies
  - should still be a sink tree
- let's abstract the implementation (BGP)
  - abstract metric for paths $f(p,d)$
    - p is the path, d is the destination
    - better paths have smaller metric
  - each AS
    - chooses the path with the smallest metric
    - changes the metric
    - sends the path to its neighbours
    - they do not decrease the metric at each hop
    - the change can depend on the neighbour

Simple Example 1



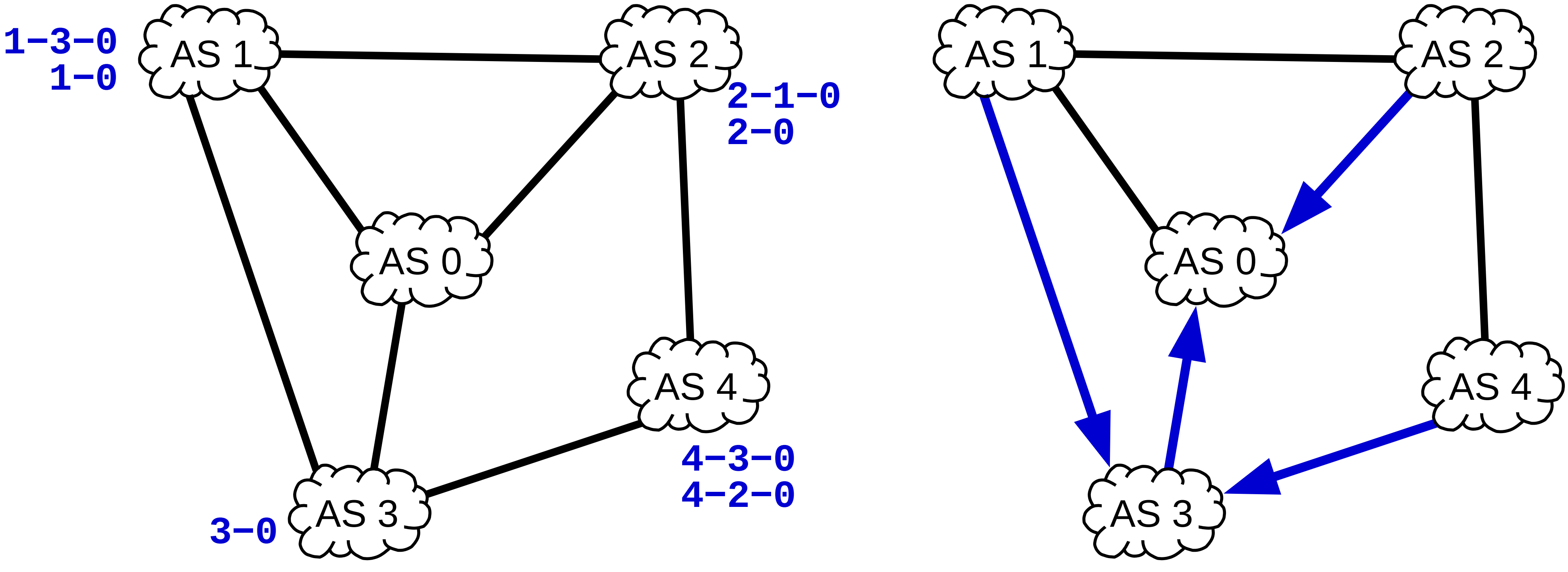
- destination is AS 0, arrows show traffic's route
- tables show acceptable routes in order of preferences
- result is a shortest-path tree

Simple Example 1



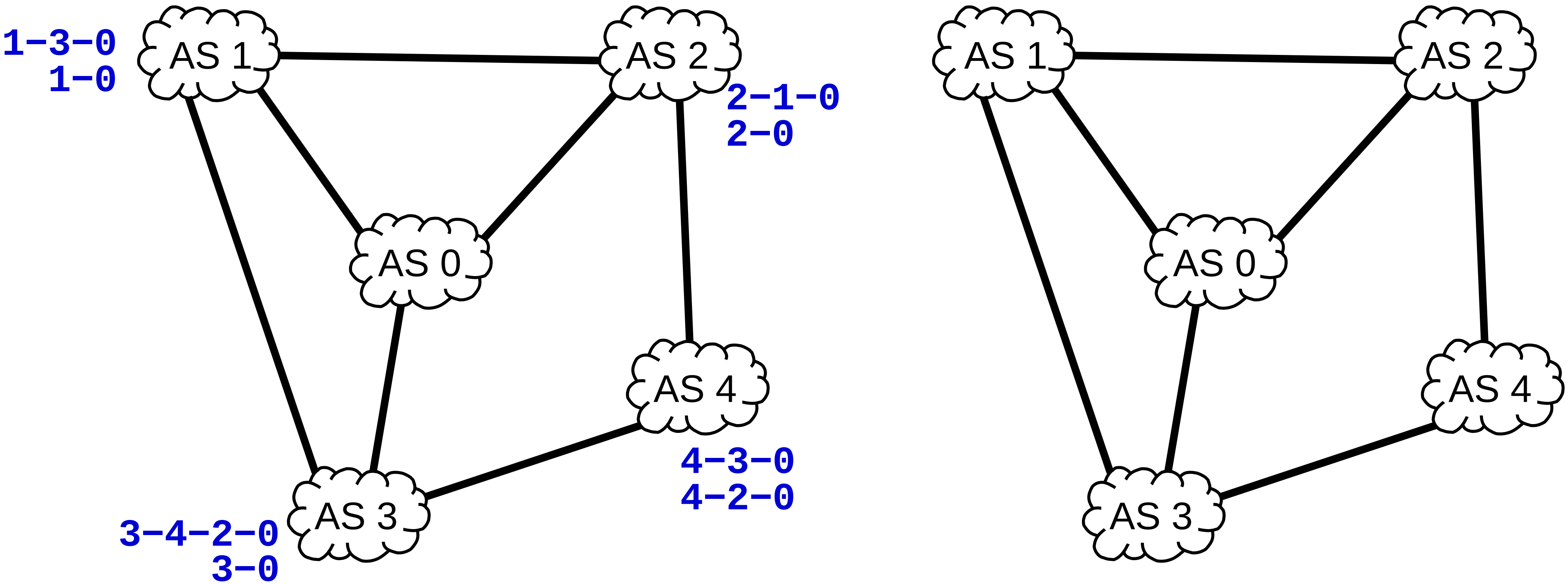
- destination is AS 0, arrows show traffic's route
- tables show acceptable routes in order of preferences
- result is a shortest-path tree

Simple Example 2



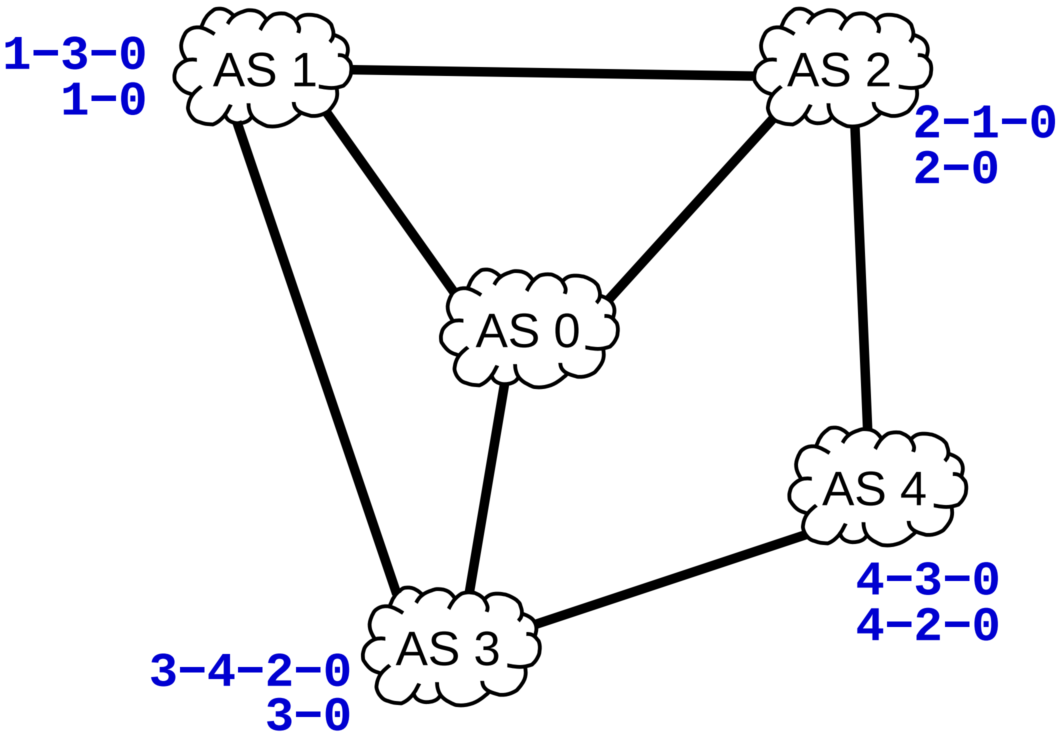
- result is not a SPF tree

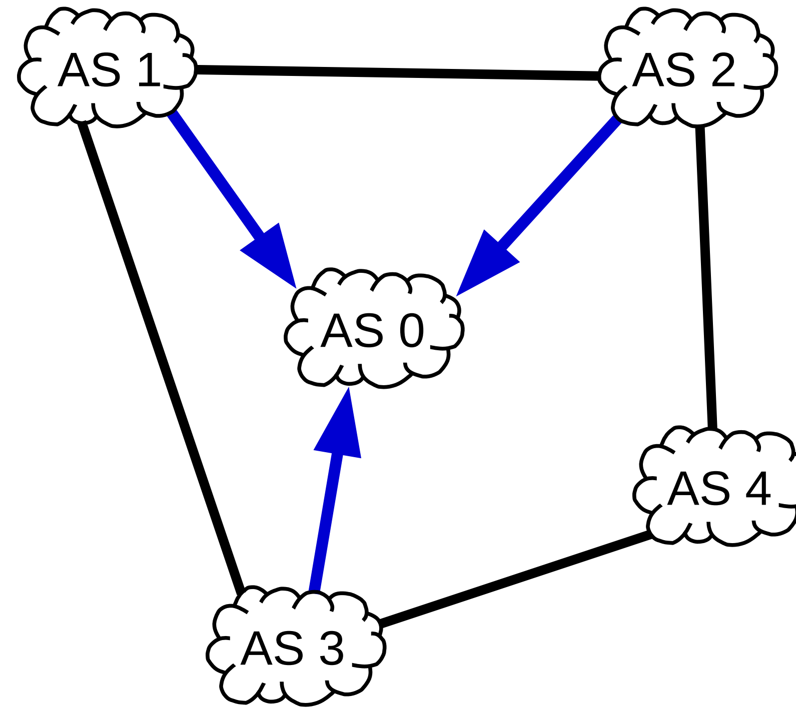
Simple Example 3



- show the process of obtaining the solution
- in this case a unique solution

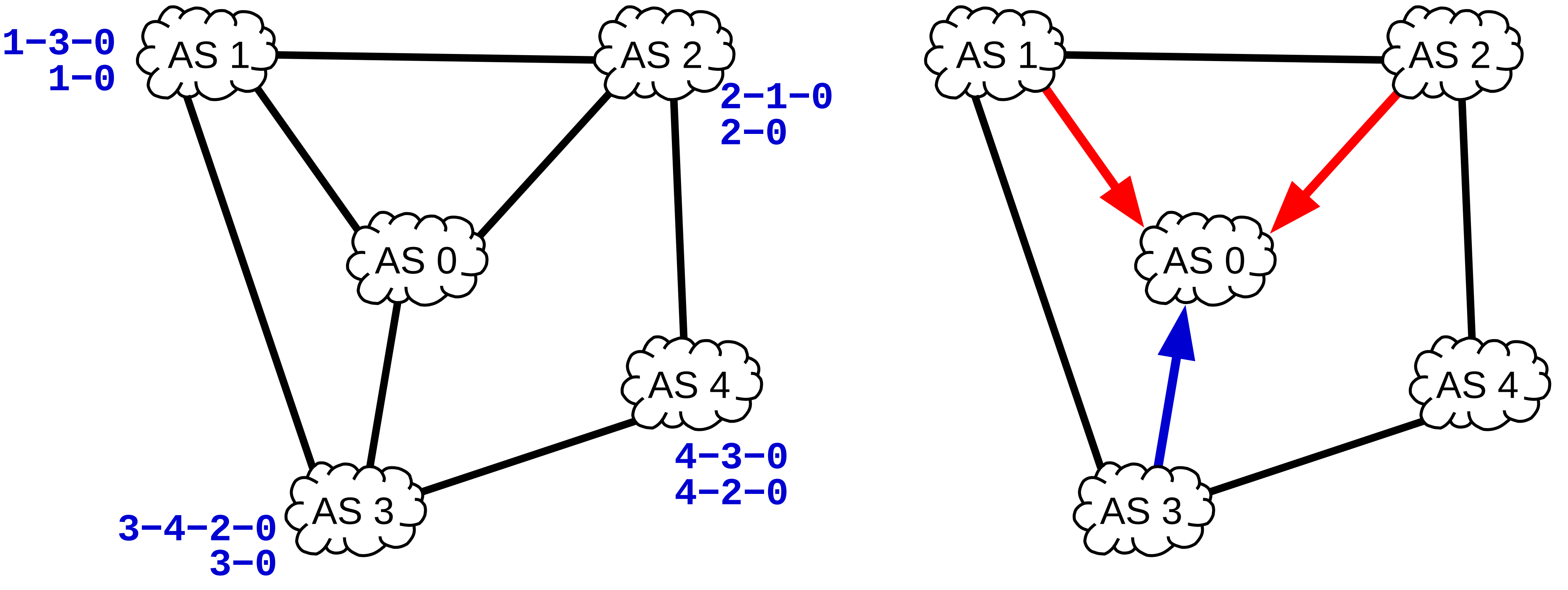
Simple Example 3





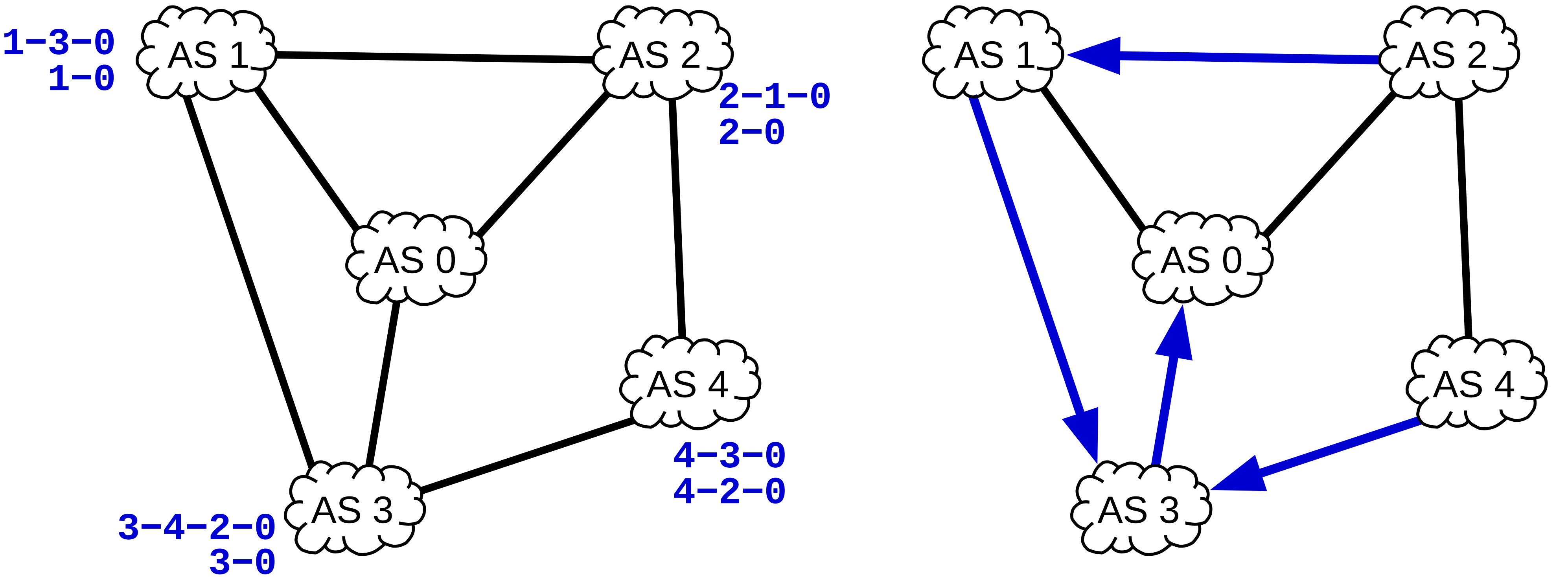
- show the process of obtaining the solution
- in this case a unique solution

Simple Example 3



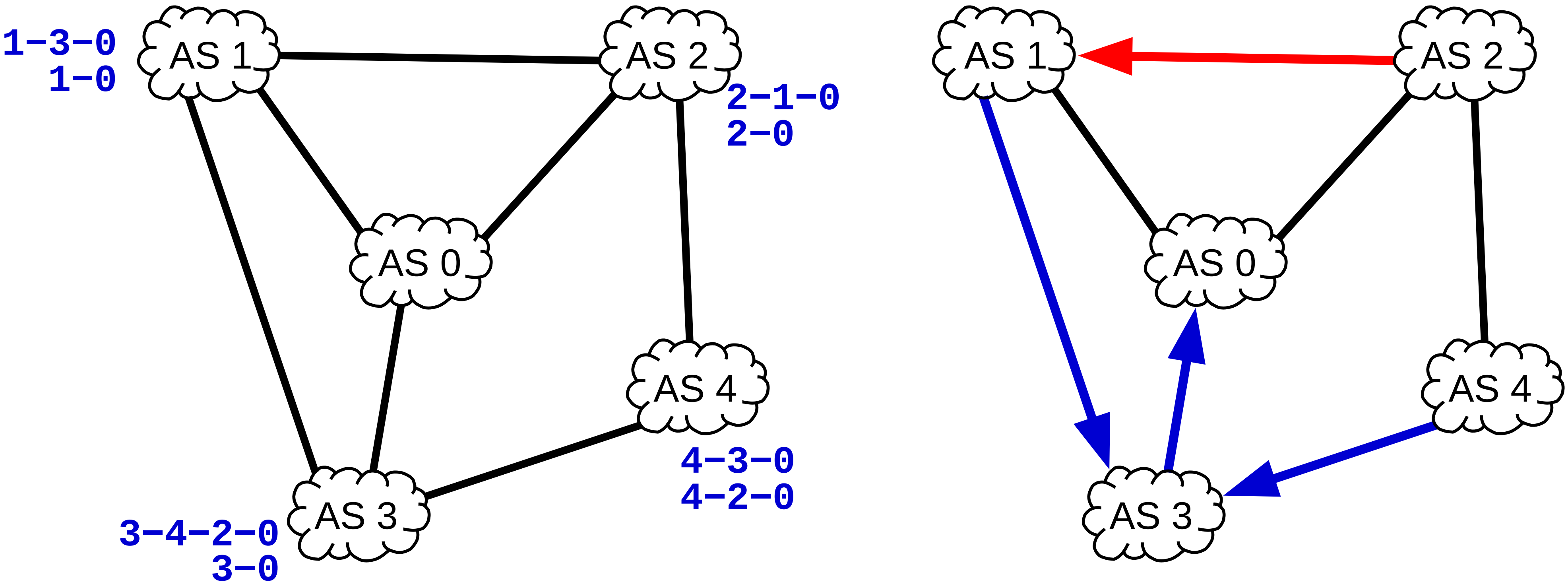
- show the process of obtaining the solution
- in this case a unique solution

Simple Example 3



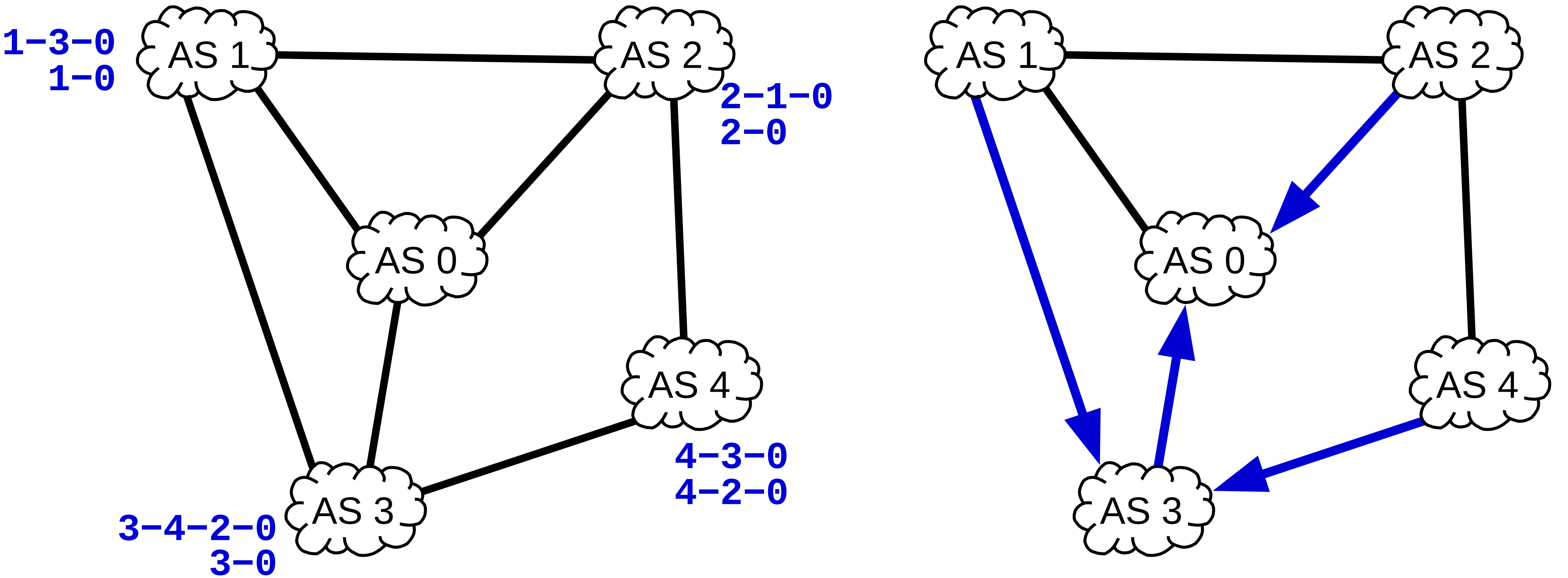
- show the process of obtaining the solution
- in this case a unique solution

Simple Example 3



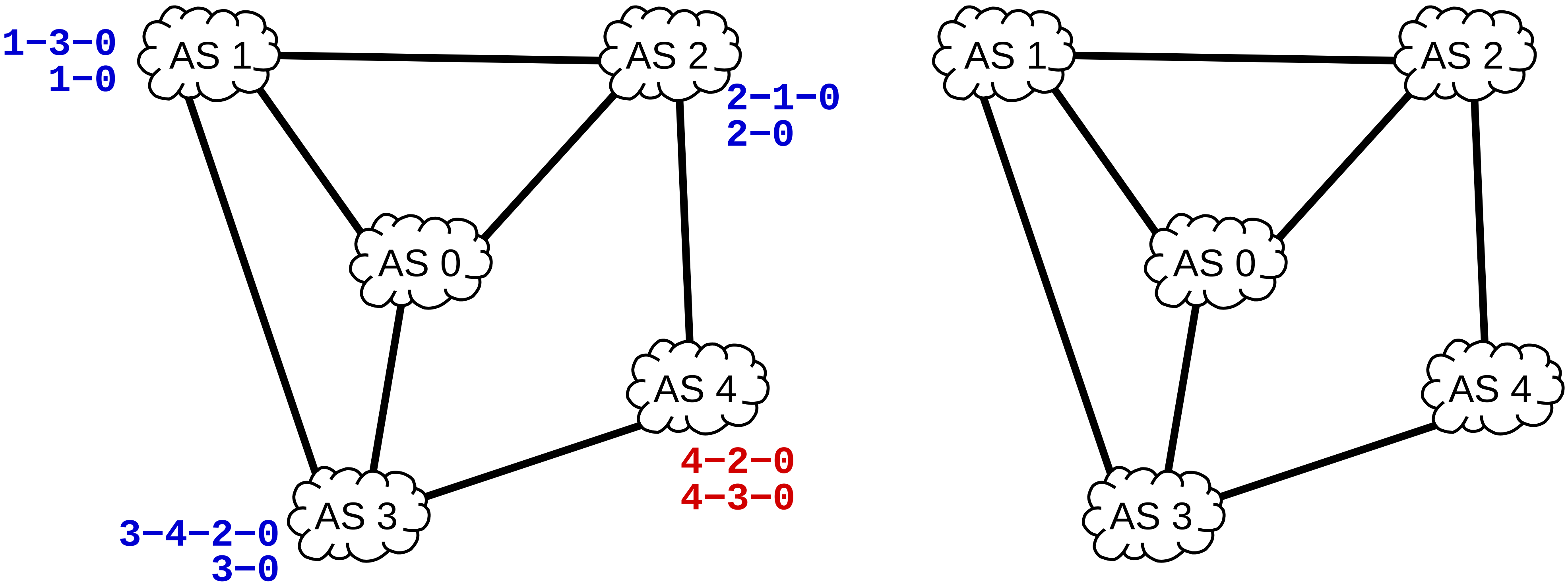
- show the process of obtaining the solution
- in this case a unique solution

Simple Example 3



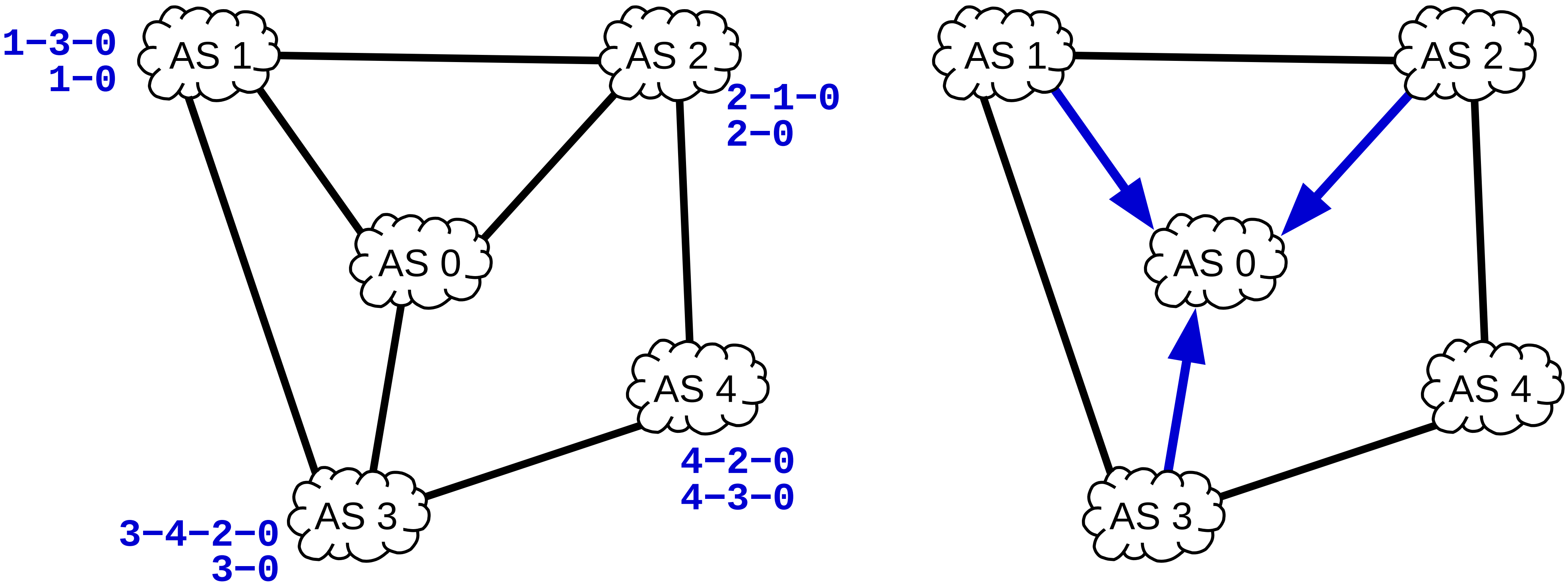
- show the process of obtaining the solution
- in this case a unique solution

Bad Widget



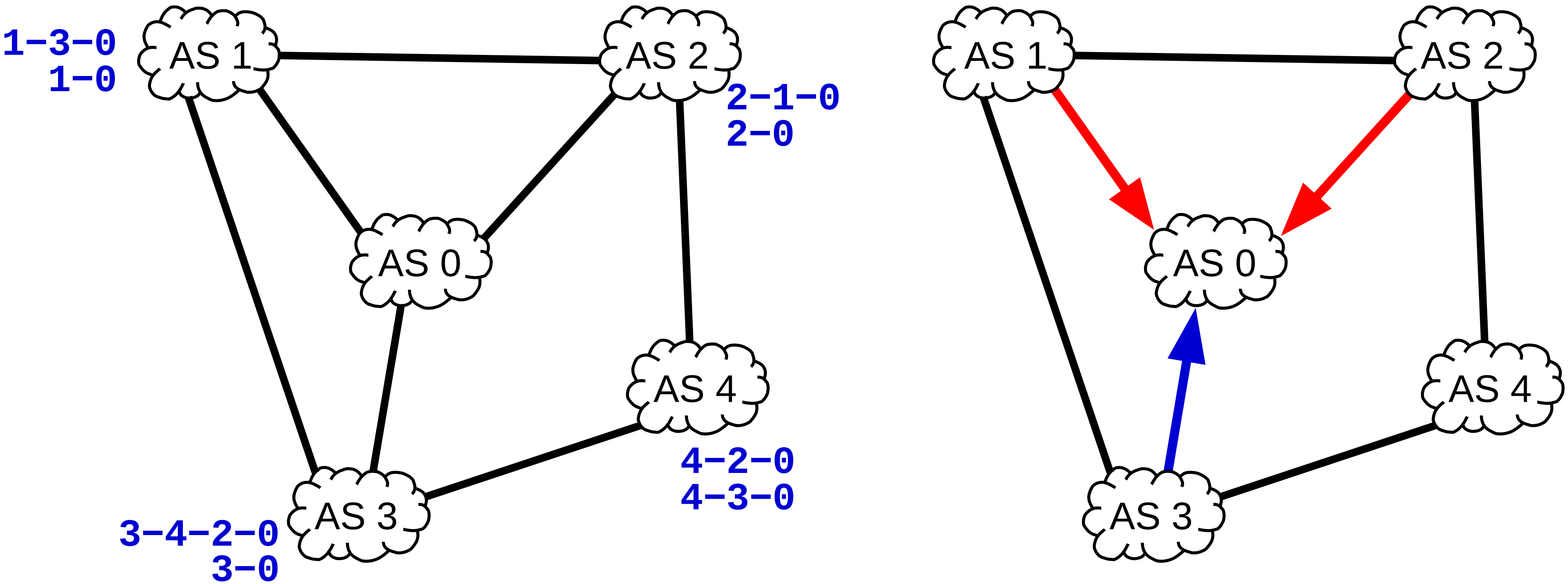
- simple change to policy at nodes 4
- no solution
- endless oscillation

Bad Widget



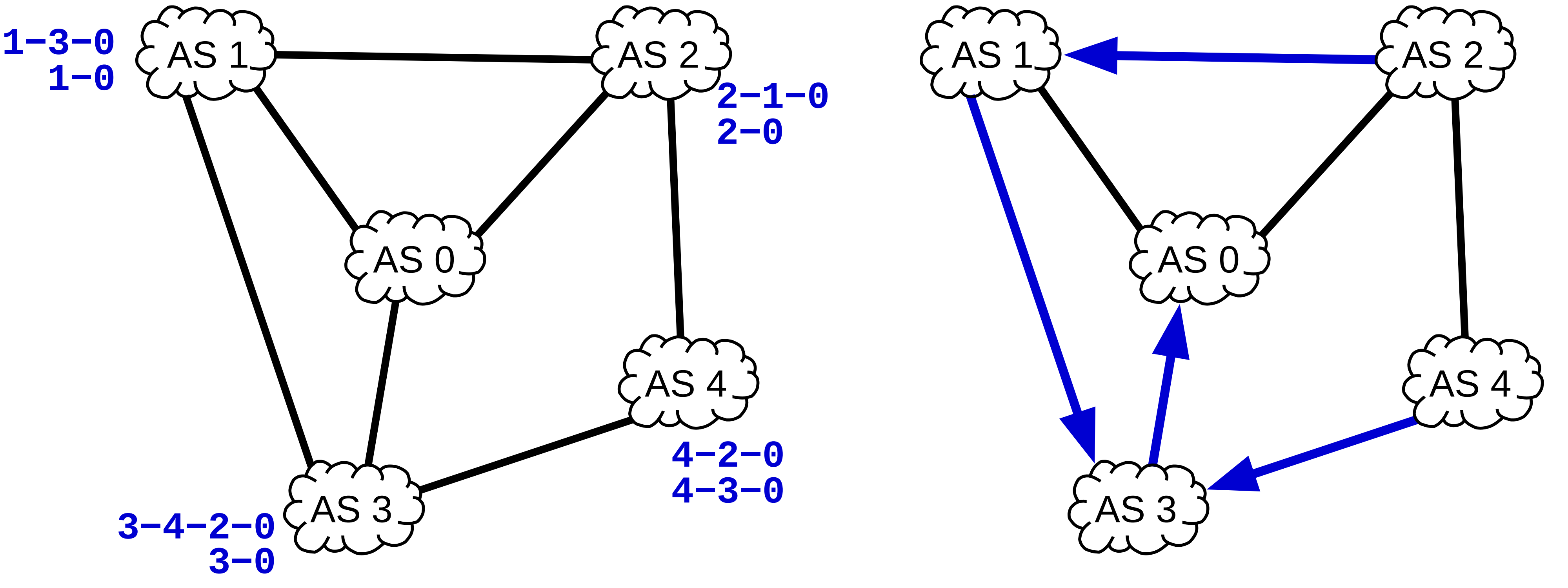
- simple change to policy at nodes 4
- no solution
- endless oscillation

Bad Widget



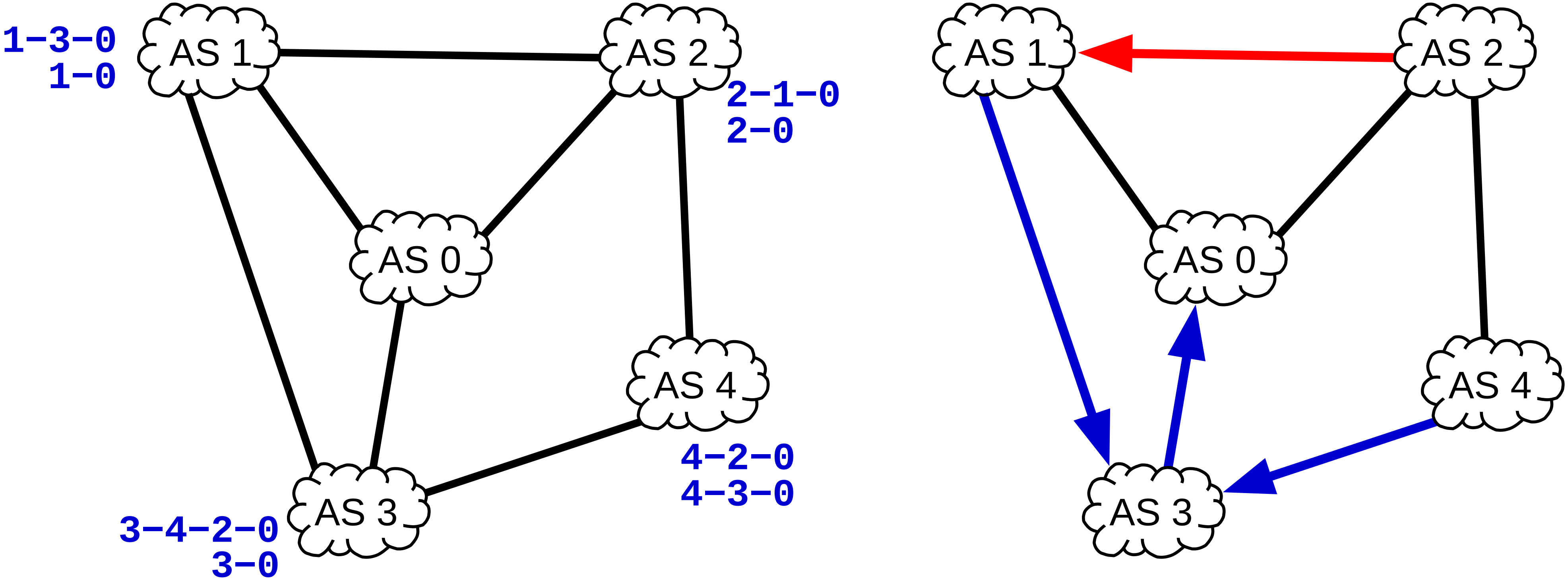
- simple change to policy at nodes 4
- no solution
- endless oscillation

Bad Widget



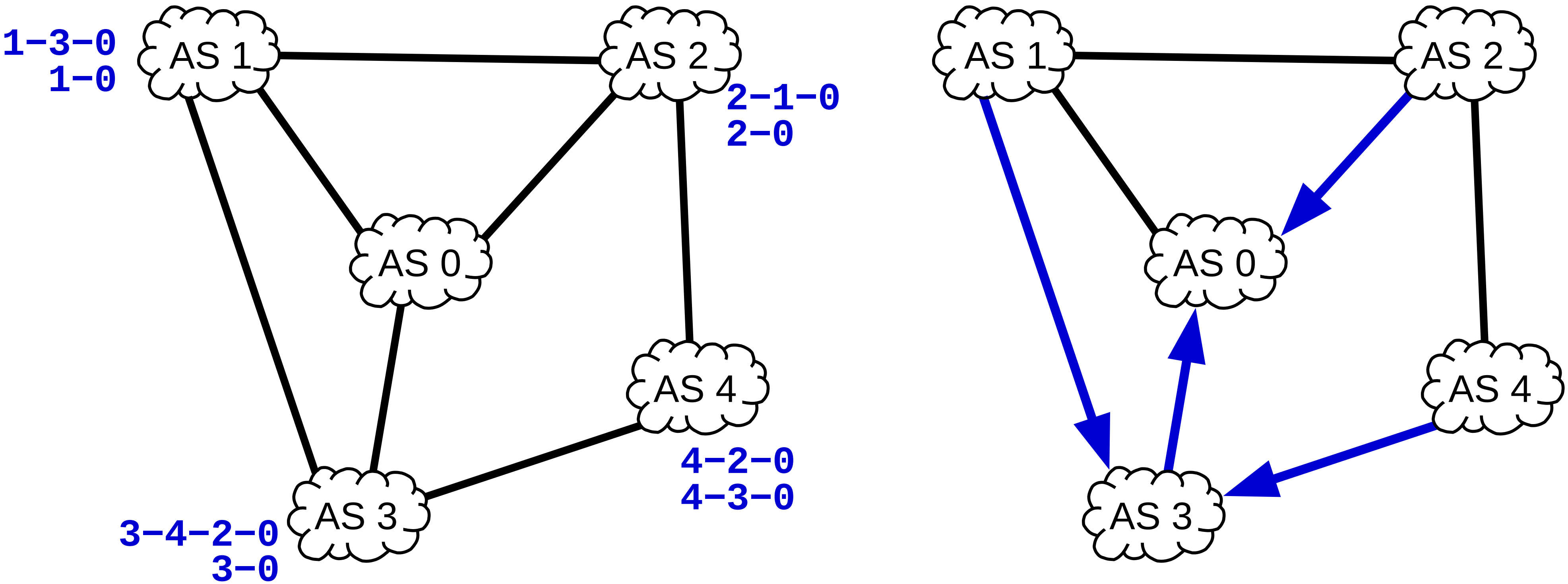
- simple change to policy at nodes 4
- no solution
- endless oscillation

Bad Widget



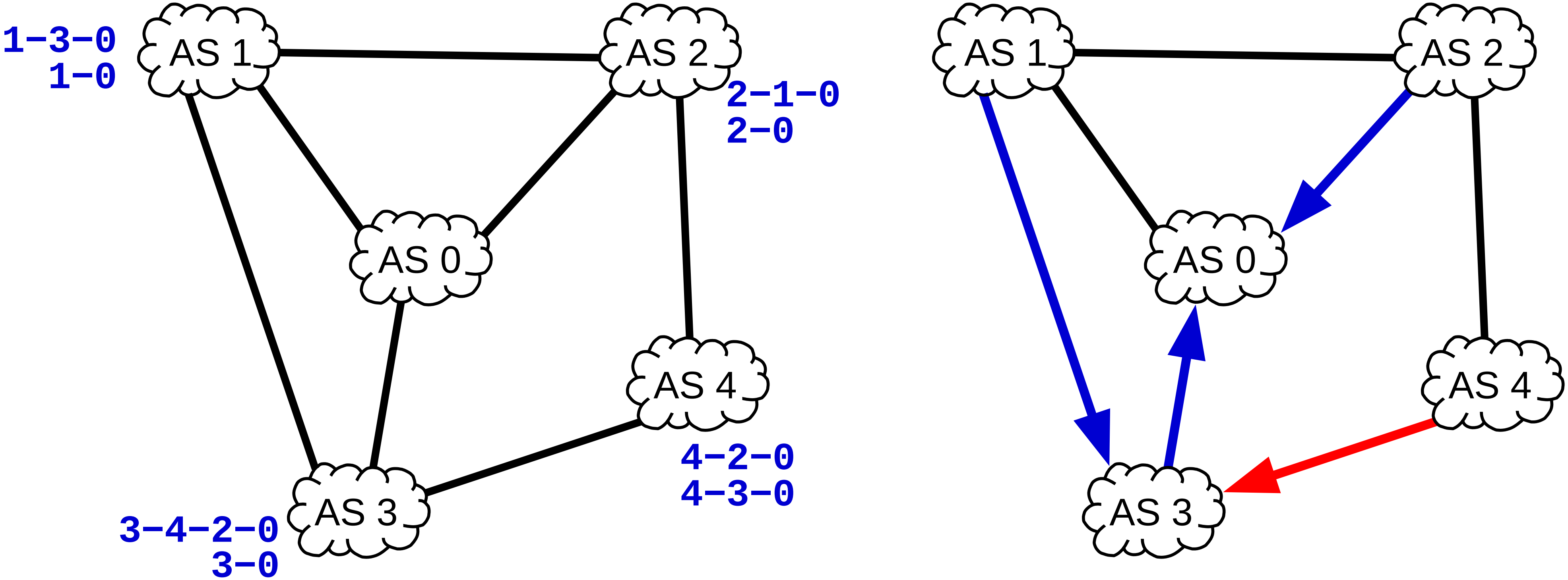
- simple change to policy at nodes 4
- no solution
- endless oscillation

Bad Widget



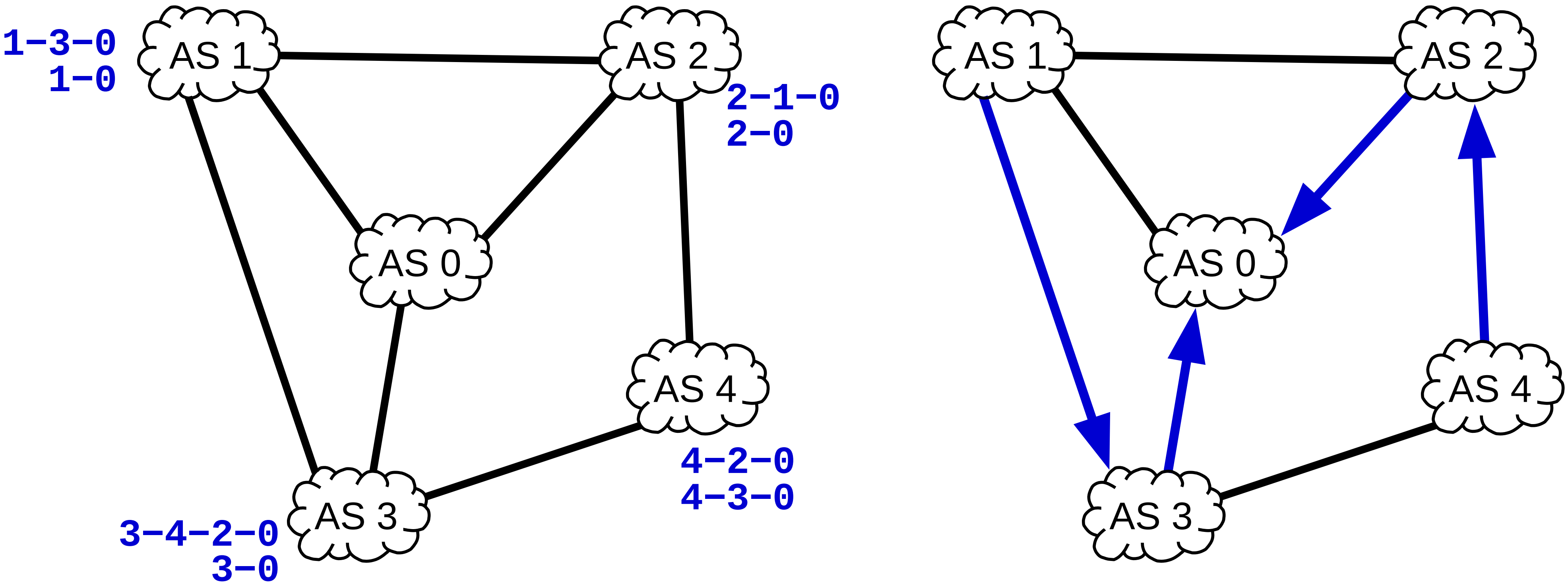
- simple change to policy at nodes 4
- no solution
- endless oscillation

Bad Widget



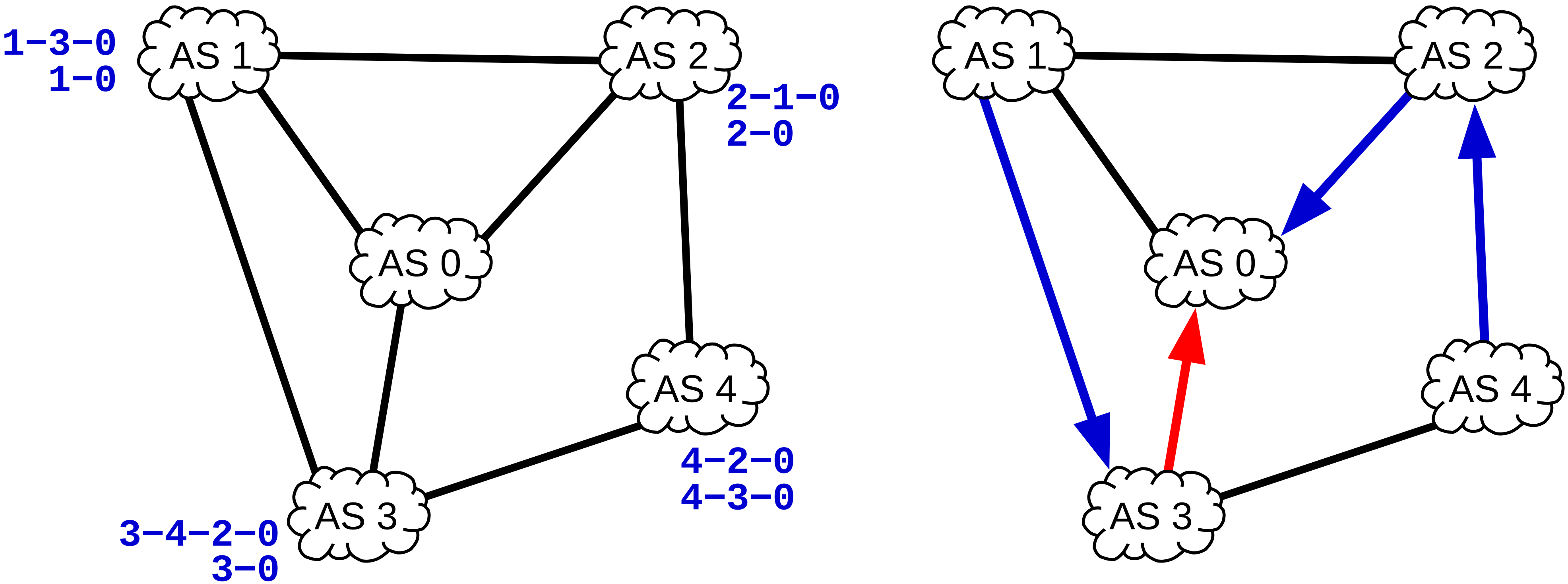
- simple change to policy at nodes 4
- no solution
- endless oscillation

Bad Widget



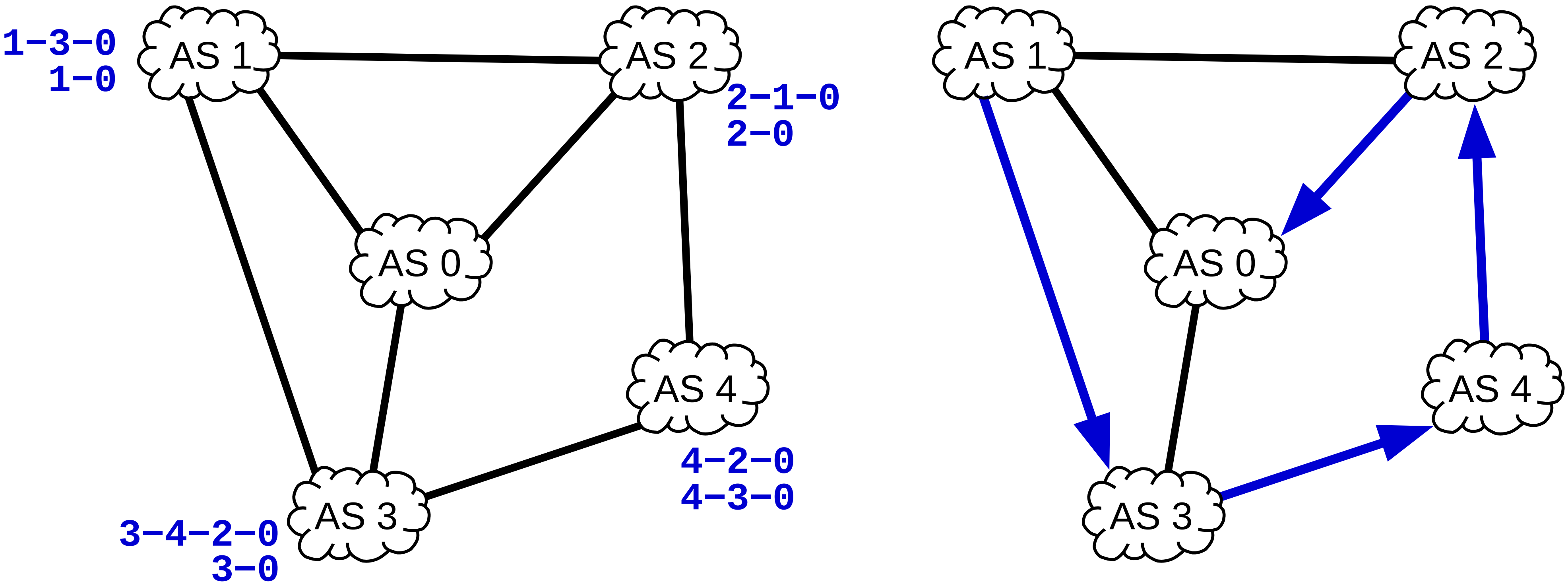
- simple change to policy at nodes 4
- no solution
- endless oscillation

Bad Widget



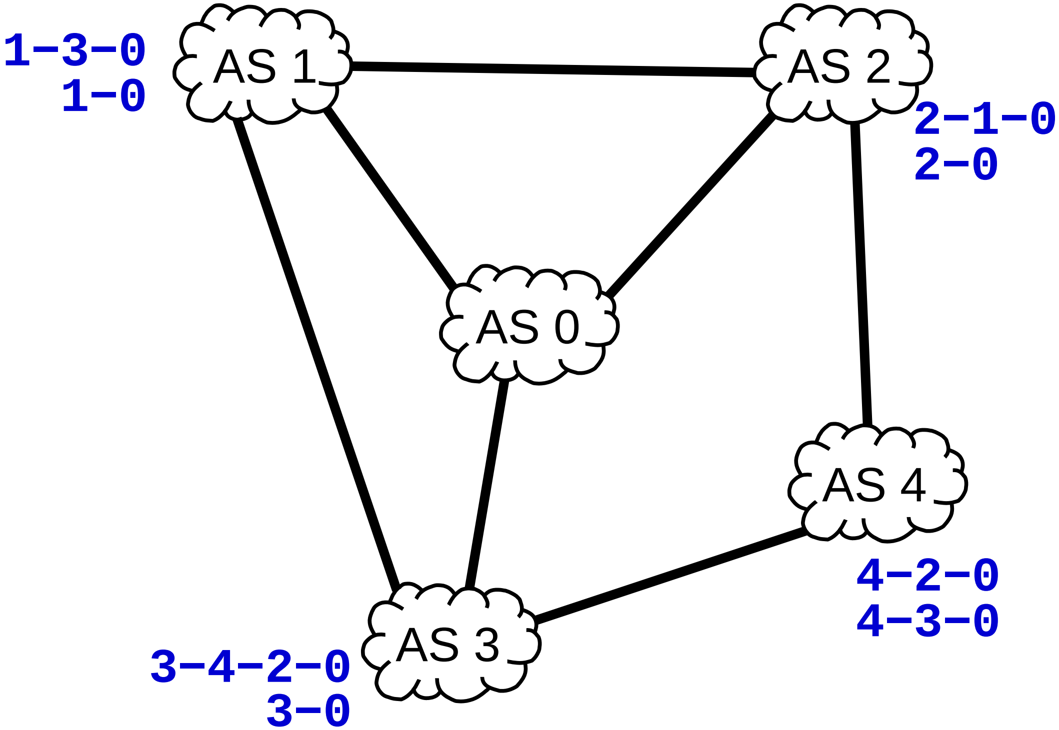
- simple change to policy at nodes 4
- no solution
- endless oscillation

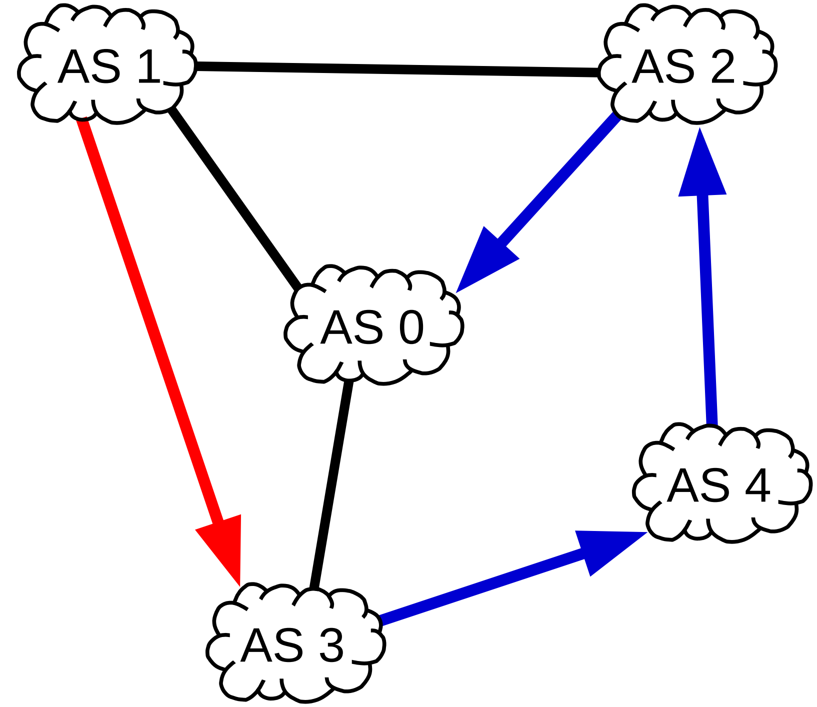
Bad Widget



- simple change to policy at nodes 4
- no solution
- endless oscillation

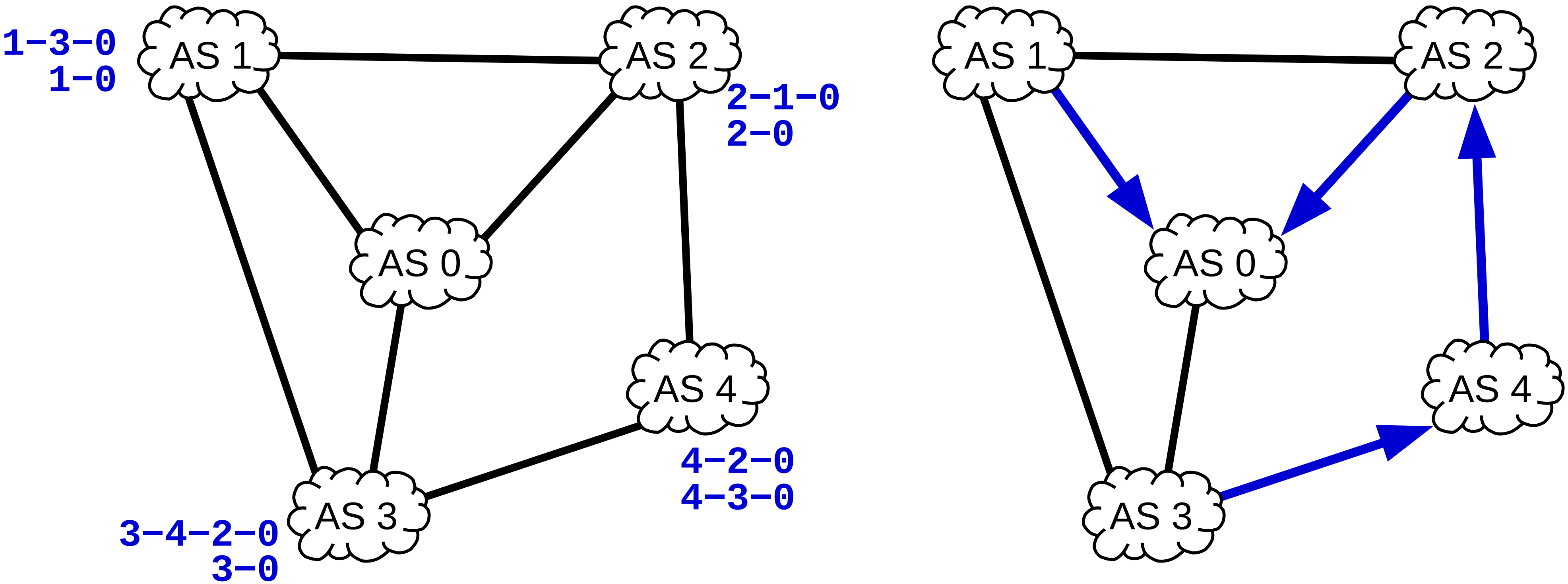
Bad Widget





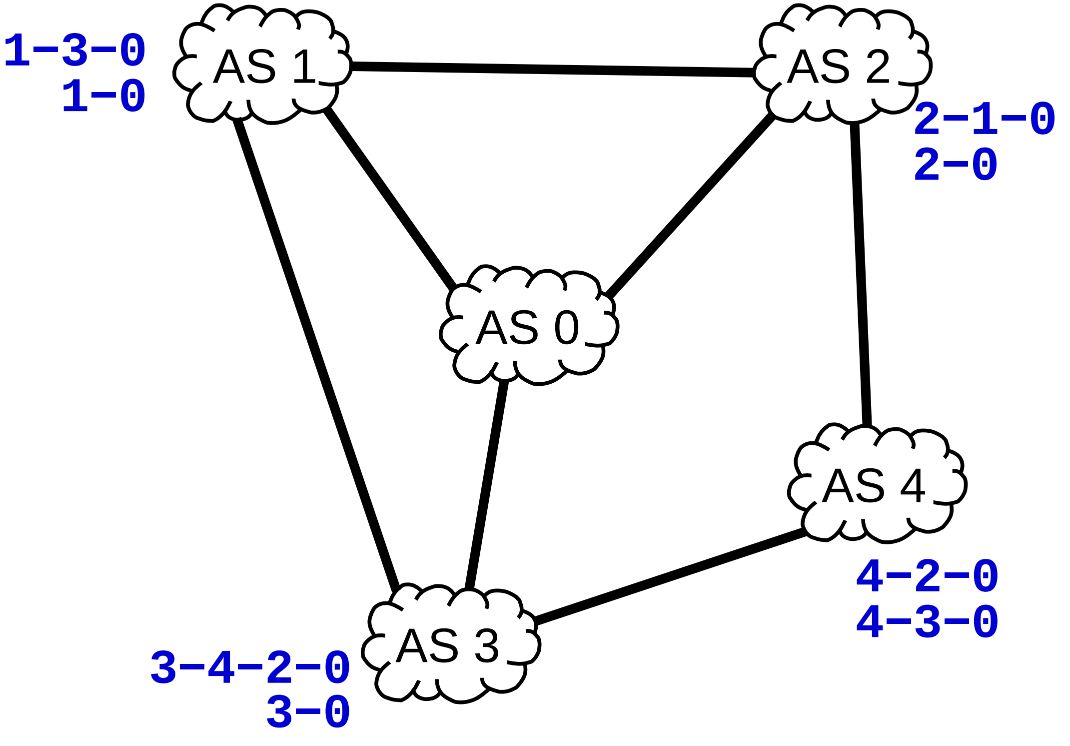
- simple change to policy at nodes 4
- no solution
- endless oscillation

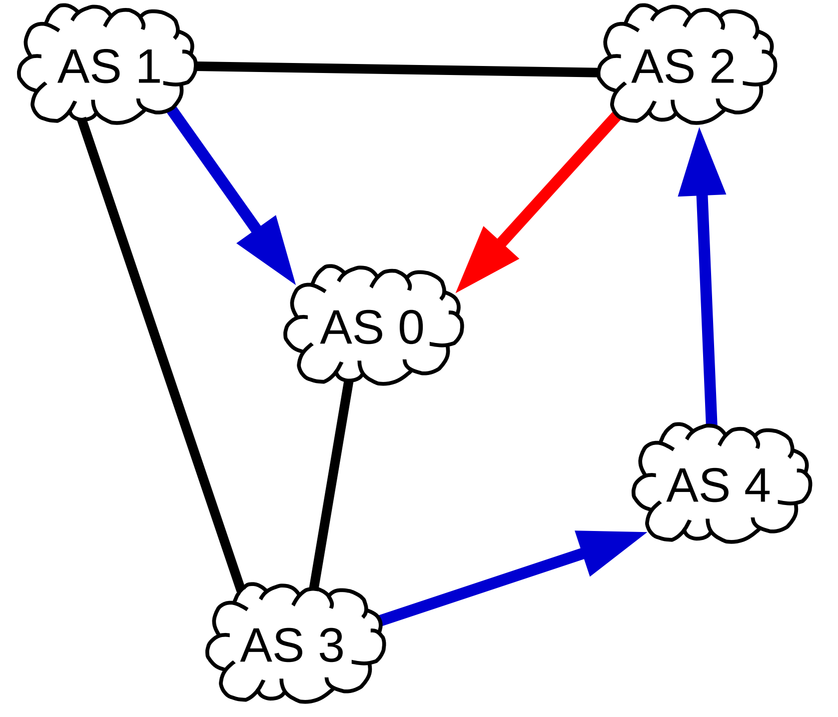
Bad Widget



- simple change to policy at nodes 4
- no solution
- endless oscillation

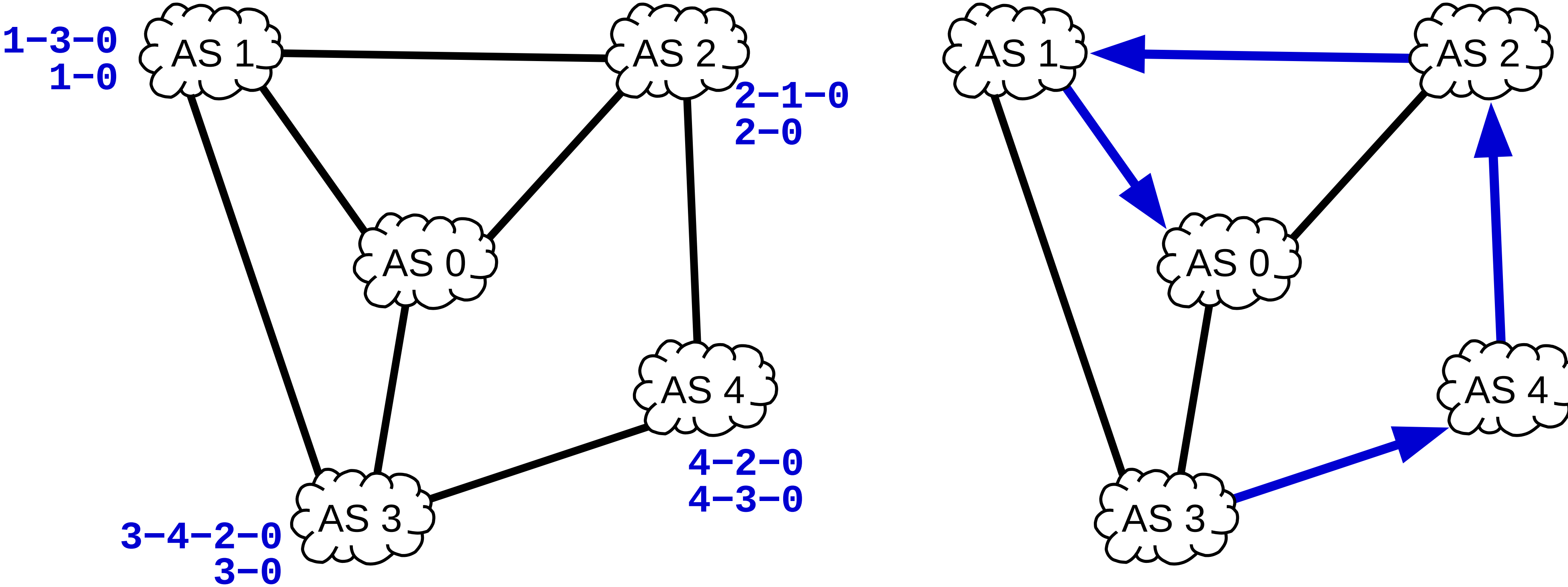
Bad Widget





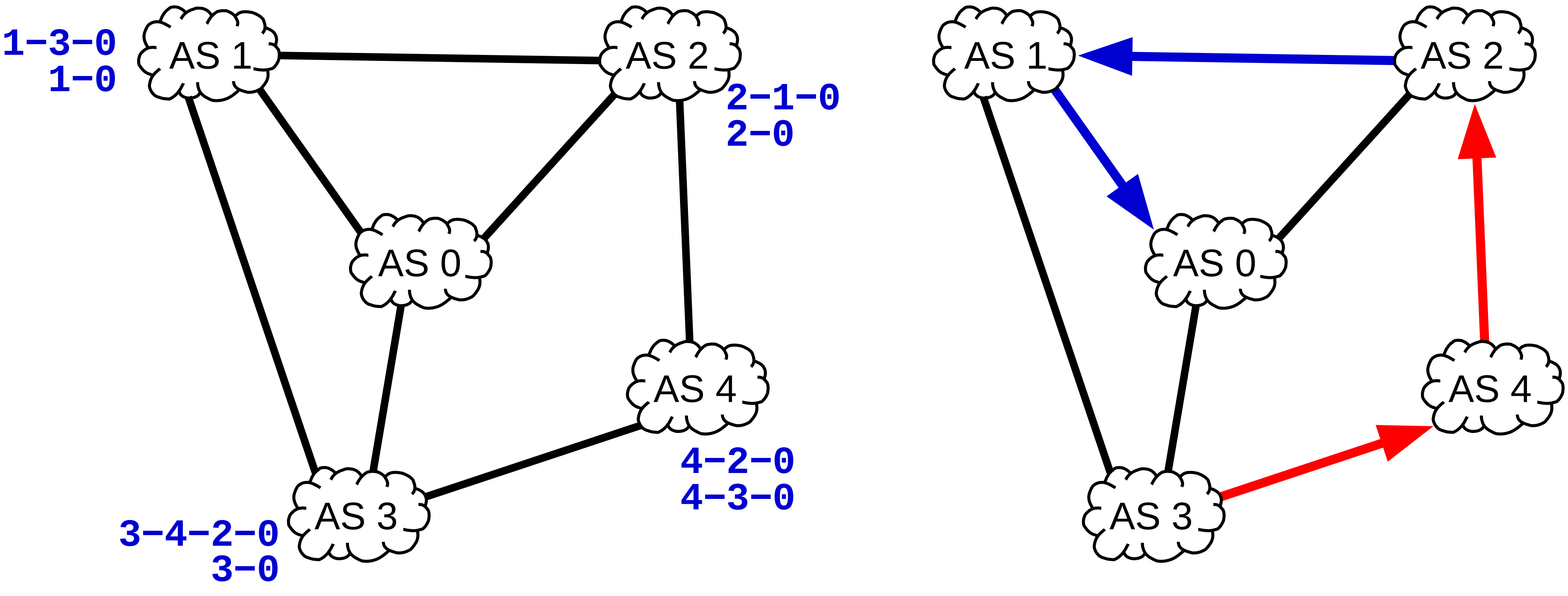
- simple change to policy at nodes 4
- no solution
- endless oscillation

Bad Widget



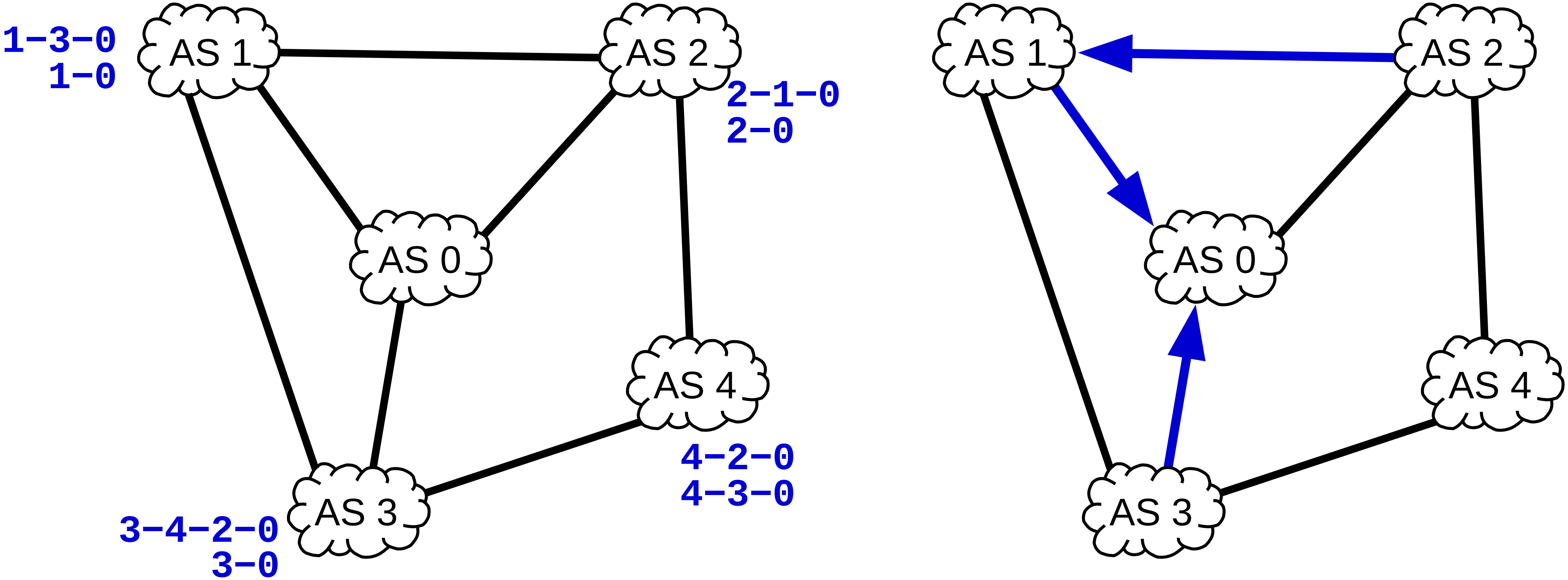
- simple change to policy at nodes 4
- no solution
- endless oscillation

Bad Widget



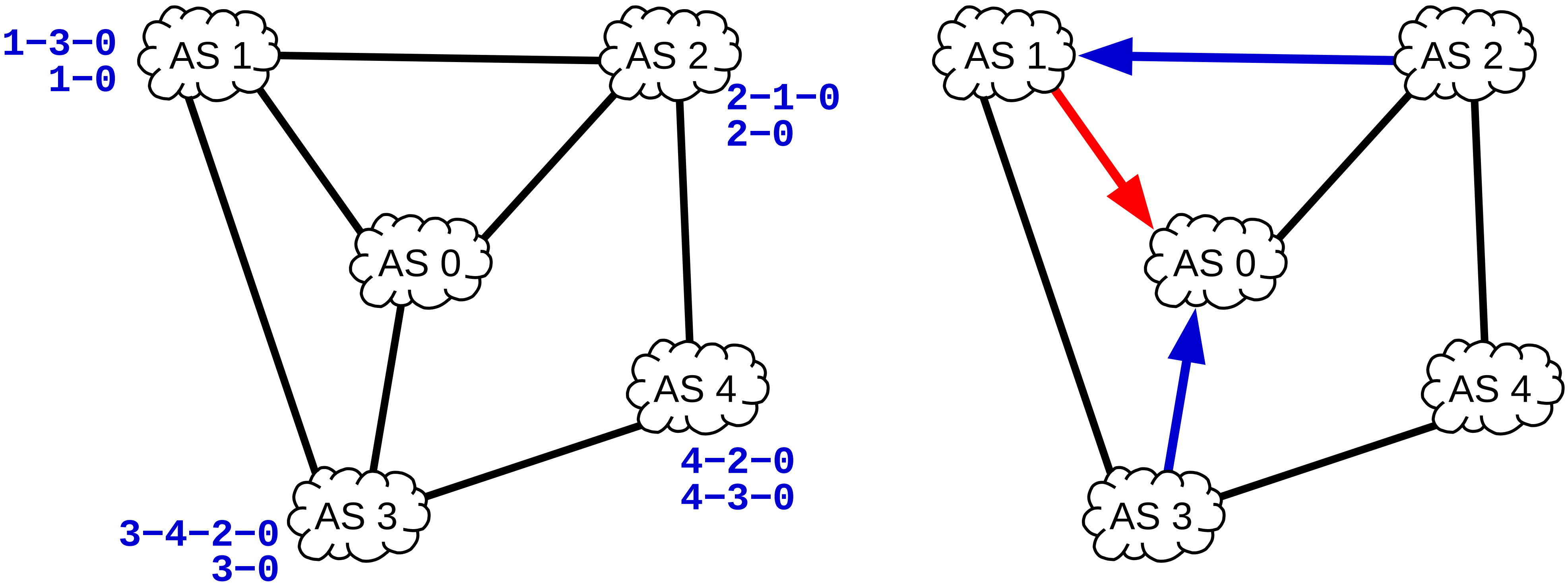
- simple change to policy at nodes 4
- no solution
- endless oscillation

Bad Widget



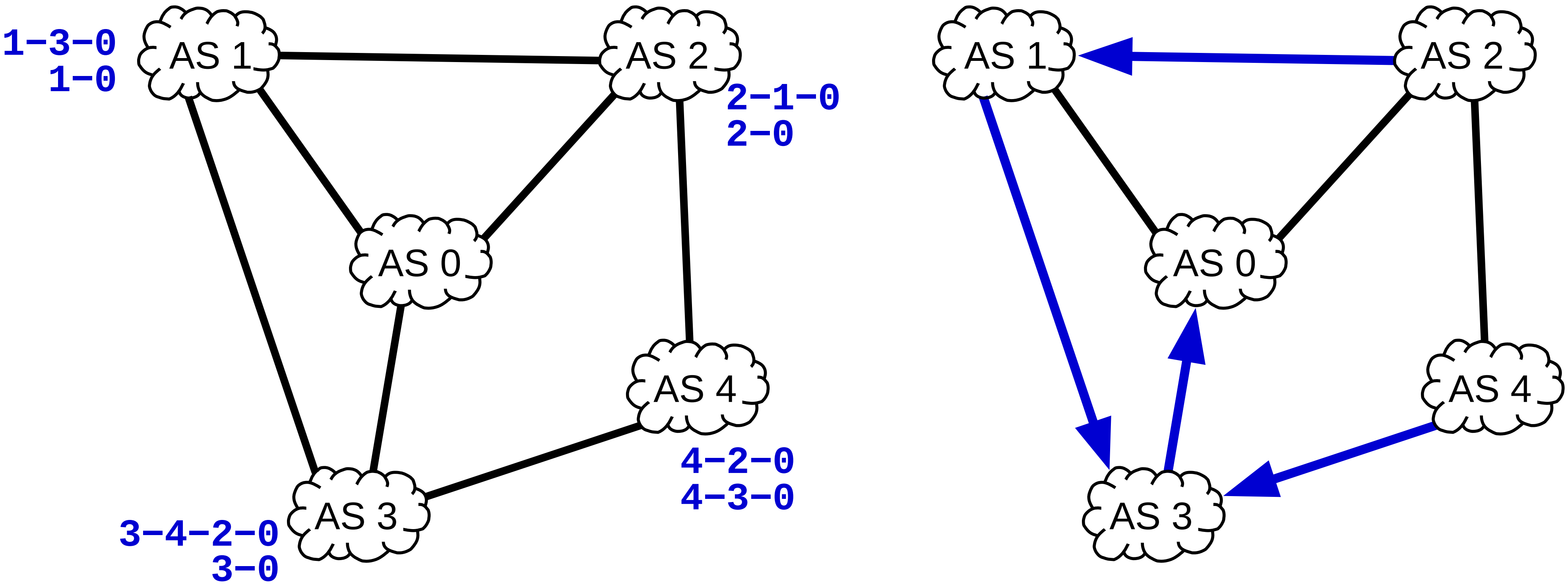
- simple change to policy at nodes 4
- no solution
- endless oscillation

Bad Widget



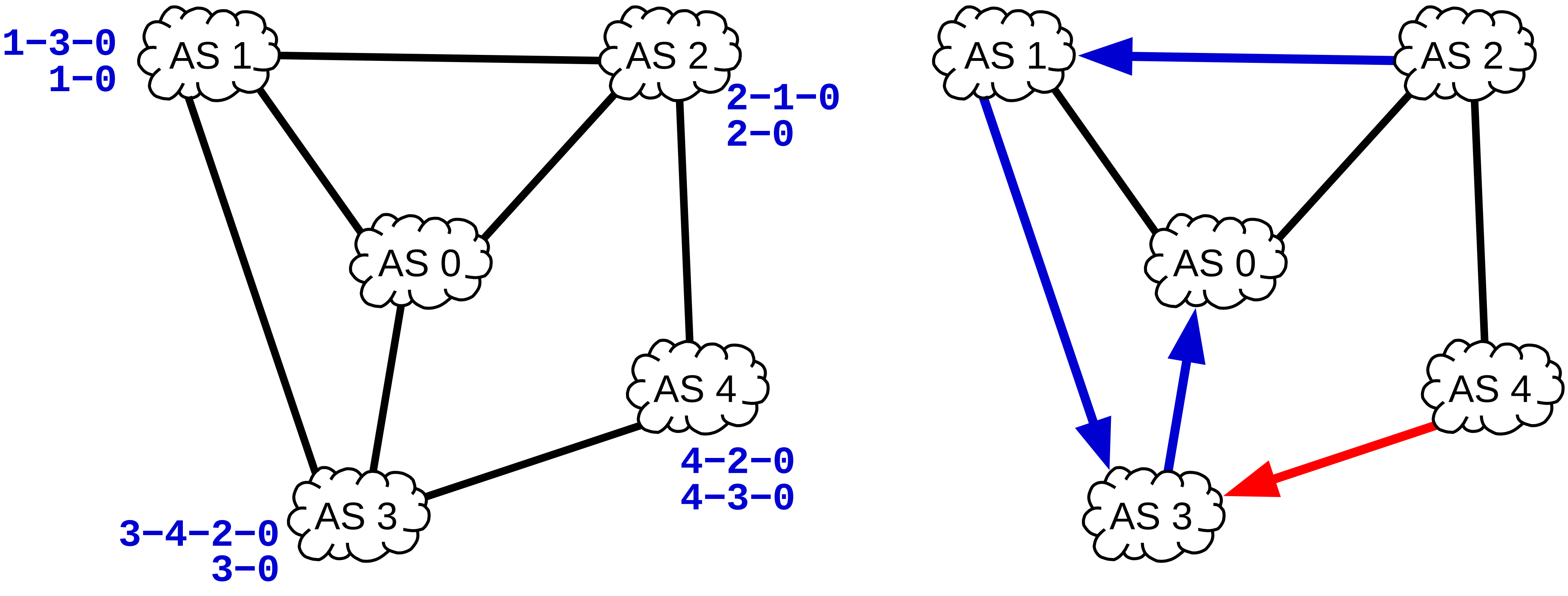
- simple change to policy at nodes 4
- no solution
- endless oscillation

Bad Widget



- simple change to policy at nodes 4
- no solution
- endless oscillation

Bad Widget



- simple change to policy at nodes 4
- no solution
- endless oscillation

Is this a problem?

- route oscillation has been observed in the Internet
  - MED oscillation ("churn")
    - MED used for "cold-potato" routing
  - Cisco fix `bgp deterministic med` command
    - plus a bit more

  `http://www.cisco.com/warp/public/770/fn12942.html`
- it could happen again
  - mostly it doesn't
- can we fix it in general
  - not easily
  - either need to restrict policy
  - or have central admin check all policies

A Real Example

- A real example of BGP convergence can be seen at `http://bgplay.routeviews.org/bgplay/`
- Choose prefix `198.133.206.0/24` (AS 3130) to see a prefix withdrawn, and then announced.
- Choose prefix `192.83.230.0/24` (AS 3130) to see a prefix change its preferred provider.
- Other "Beacon" prefixes
  - `192.135.183.0/24`
  - `203.10.63.0/24`
  - `198.32.7.0/24`
  - for Beacon details see `http://www.psg.com/~zmao/BGPBeacon.html`

BGP optimization

- has anyone written this as a formal optimization problem?
  - companies have built tools that treat as inter-AS routing as an optimization problem, e.g.
    - optimize performance, by choosing shorter paths
    - optimize cost, by choosing cheaper paths
  - tend to keep their methods a secret (unfortunately)
- is this a solved problem — no way!
  - above is for simply connected network
  - what happens when people apply these methods effectively against each other:
    - really a game theory problem
    - will we get a tragedy of the commons?
  - could this result in large scale oscillation/instability?

Link state vs Path Vector

Link state

- topology information flooded
- best end-to-end paths computed locally at each router
- based on minimizing some notion of distance
- best end-to-end paths determine next hops
- works only if policy is shared and uniform

Path-vector

- each router knows little about overall topology
- only best next hops are chosen by each router for each destination
- best end-to-end paths result from compositions of all next-hop choices
- does not require a notion of distance
- does not require uniform policies

OSPF vs BGP comparison

OSPF

- link state
- topology discovered
- soft-state
- one administrative control
- common routing policy
- shortest paths
- fast(ish) convergence (10's of seconds down to sub-second)
- limited policy
- limited scaling (one level hierarchy)

BGP

- path-vector
- each router knows little about overall topology
- hard-state
- best end-to-end paths result from compositions of all next-hop choices
- policy based
- scalable (to the size of the Internet)
- slower convergence (minutes)

What haven't I told you

A lot

- most implementation details
  - particularly proprietary stuff
- many other features, and uses
  - confederations, route reflectors, ...
- eBGP vs iBGP
- interactions between BGP and IGP
  - many rules about preference of routes learnt from one being redistributed into the others
- BGP is an active area of research
  - much is not entirely understood

BGP the musical

Theme song (sung to the tune of "Yesterday")

Yesterday,
All the withdrawals seemed so far away
I thought my prefixes were here to stay
Oh, I believe in Yesterday.

Suddenly,
It's not half the table it used to be
There's a black hole hanging over me
Oh, I believe in Yesterday.

Why they had to flap, announce and draw away?
They sent something bad, now I long for yesterday.

Yesterday,
Routing was such an easy game to play
Now my packets all hide away
Oh, I believe in Yesterday

Avi Freedman, `http://www.caida.org/workshops/isma/0112/agenda.xml`

References

[1] J. Stewart III, BGP4: Inter-domain Routing in the Internet. Addison-Wesley, Boston, 1999.

[2] T. Griffin, "Does BGP Solve the Shortest Paths Problem?," in The North American Network Operators' Group (NANOG) 18, (San Jose, CA, USA), February 2000. http://www.nanog.org/mtg-0002/ppt/griffin/.

[3] T. Griffin, F.Shepherd, and G.Wilfong, "The stable paths problem and interdomain routing," IEEE/ACM Transactions on Networking, vol. 10, no. 2, pp. 232–243, 2002.