Computer Networks and Internet Technology

2021W703033 VO Rechnernetze und Internettechnik
Winter Semester 2021/22

Jan Beutel

universität innsbruck

With materials and inspiration from Scott Shenker, Jennifer Rexford, Phil Levis, Adrian Perrig and Laurent Vanbever

Communication Networks and Internet Technology

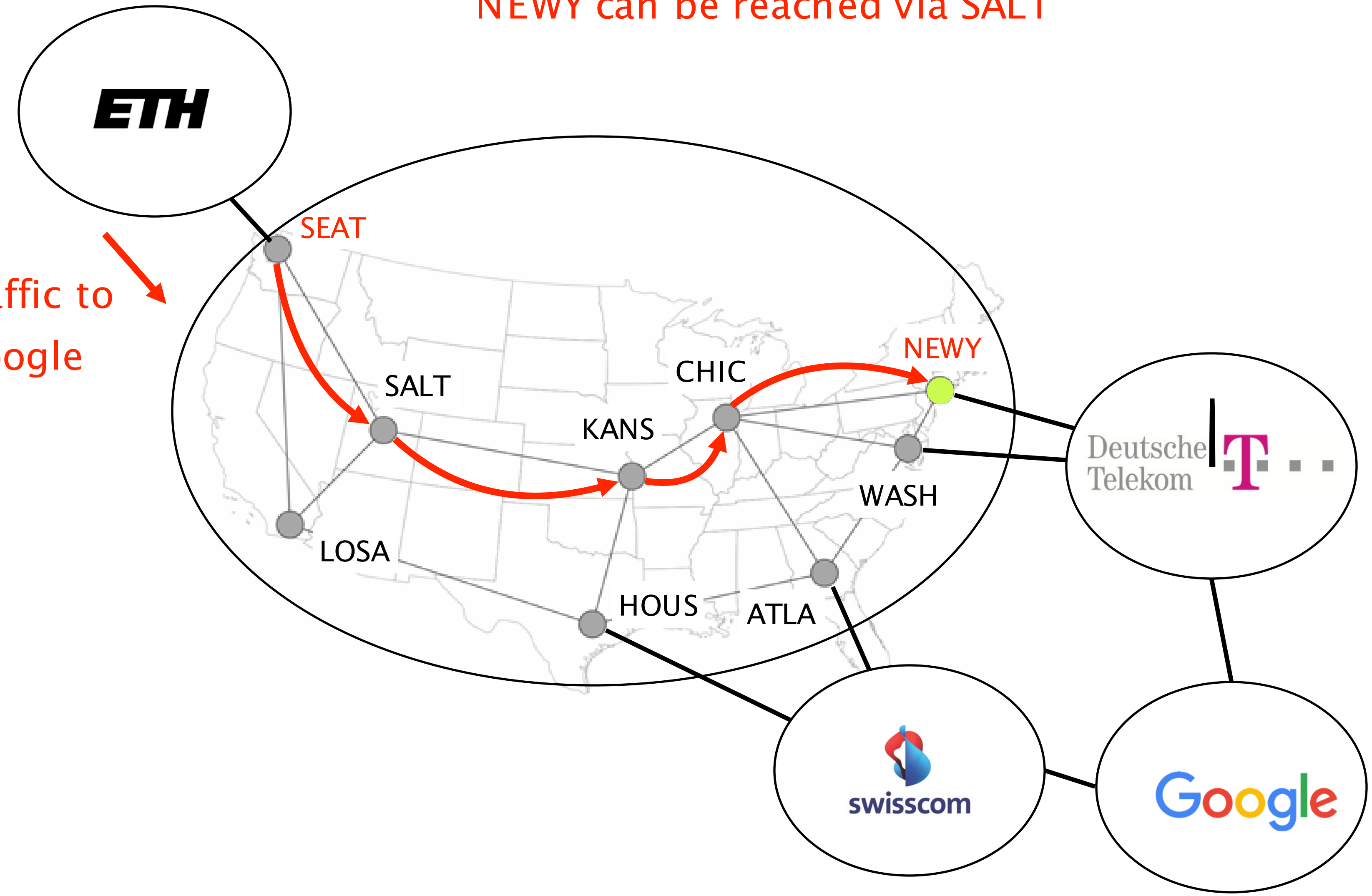
Recap of last weeks lecture

Internet routing comes into two flavors:
intra- and *inter-domain* routing

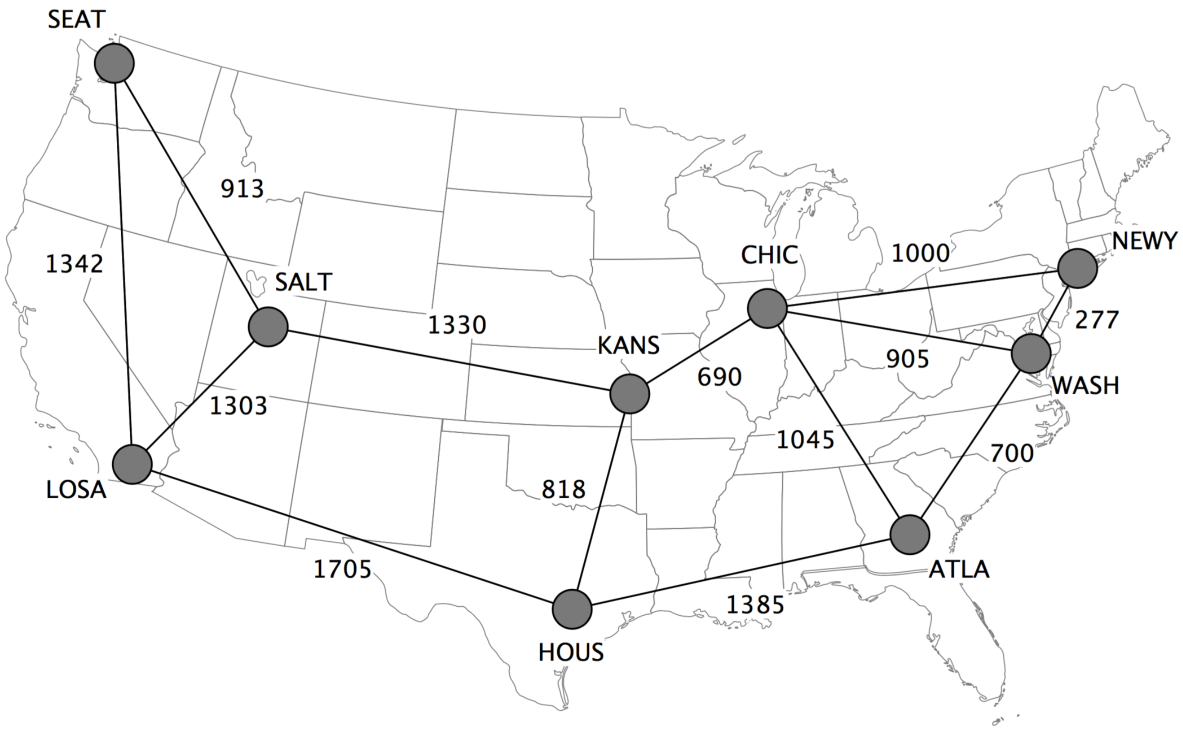
inter-domain routing	intra-domain routing
Find paths between networks	Find paths within a network

NEWY can be reached via SALT

traffic to Google



When weights are assigned proportionally to the distance, shortest-paths will minimize the end-to-end delay



Internet2, the US based research network

When weights are assigned inversely proportionally to each link capacity, throughput is maximized

if traffic is such that there is no congestion

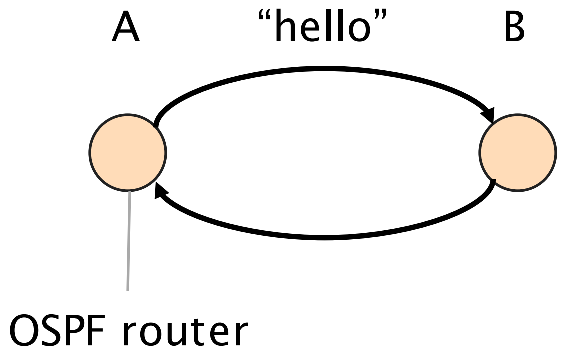
In Link-State routing, routers build a precise map of the network by flooding local views to everyone

Each router keeps track of its incident links and cost
as well as whether it is up or down

Each router broadcast its own links state
to give every router a complete view of the graph

Routers run Dijkstra on the corresponding graph
to compute their shortest-paths and forwarding tables

By default, Link-State protocols detect topology changes using software-based beaconing



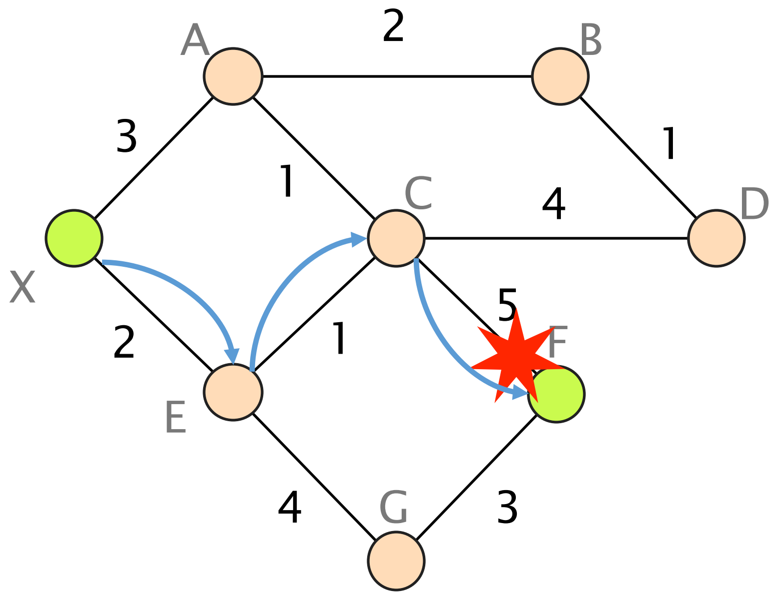
Routers periodically exchange “Hello” in both directions (*e.g.* every 30s)

Trigger a failure after few missed “Hellos” (*e.g.*, after 3 missed ones)

Tradeoffs between:

- detection speed
- bandwidth and CPU overhead
- false positive/negatives

Blackholes appear due to detection delay,
as nodes do not immediately detect failure



depends on the timeout for detecting lost hellos

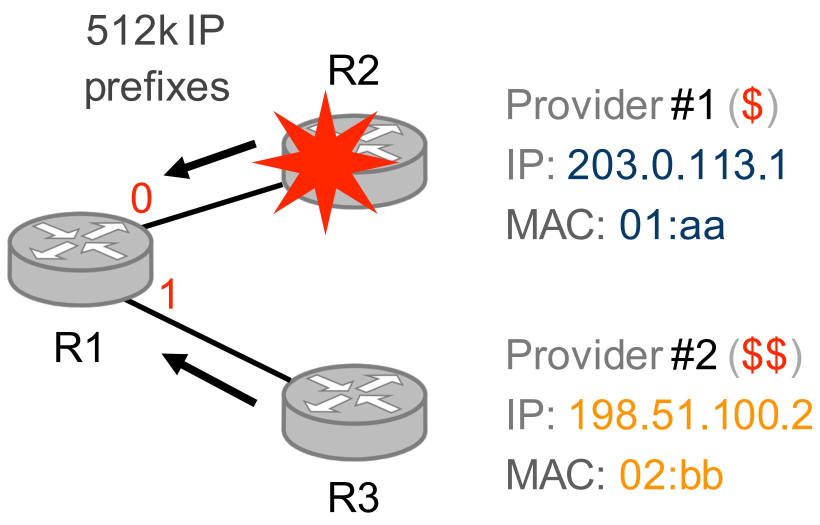
In practice, network convergence time is mostly driven by table updates

	time	improvements
detection	few ms	smaller timers
flooding	few ms	high-priority flooding
computation	few ms	incremental algorithms
table update	potentially, *minutes!*	better table design

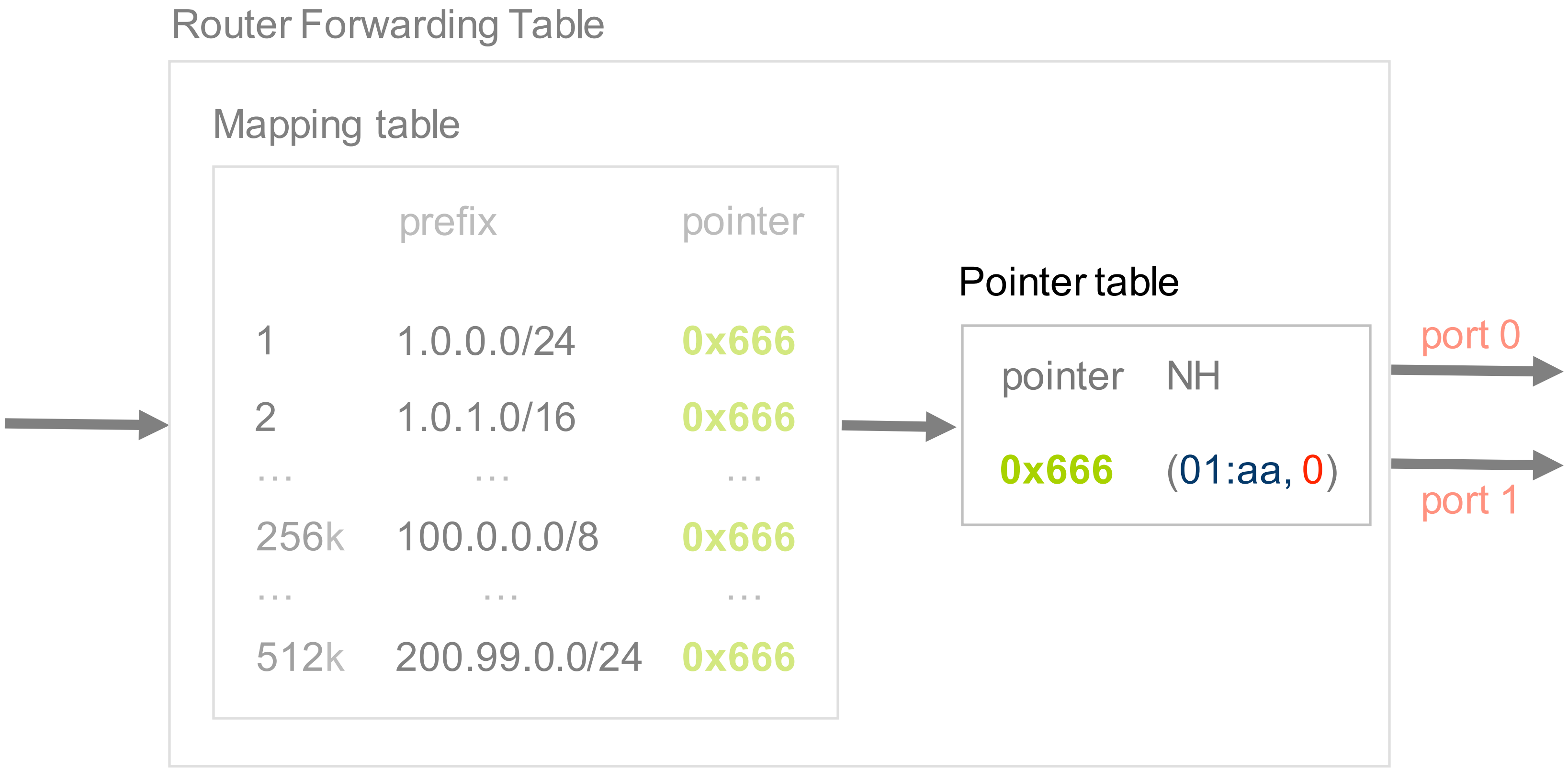
Upon failure of R2,
all 512k entries have to be updated

R1's Forwarding Table

	prefix	Next-Hop
1	1.0.0.0/24	(01:aa, 0)
2	1.0.1.0/16	(01:aa, 0)
…	…	…
256k	100.0.0.0/8	(01:aa, 0)
…	…	…
512k	200.99.0.0/24	(01:aa, 0)



Upon failures, we update the pointer table



Today, two Link-State protocols are widely used:

OSPF and IS-IS

OSPF

IS-IS

Open Shortest Path First

Intermediate Systems[2]

Distance-vector protocols are based on
Bellman-Ford algorithm

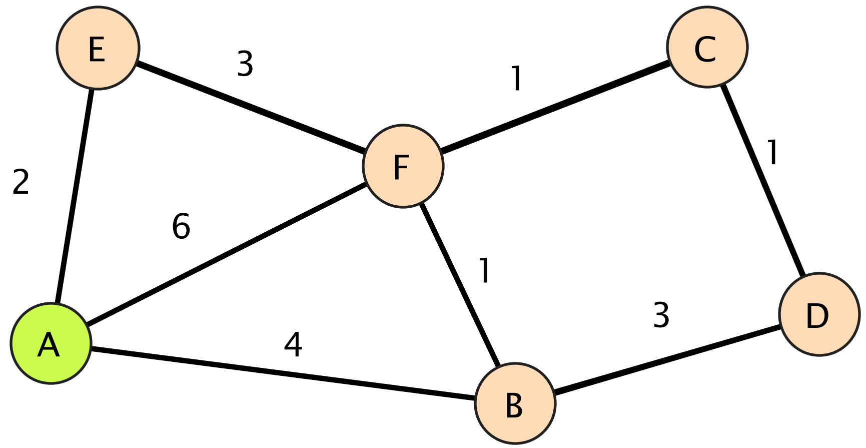
Similarly to Link-State,
3 situations cause nodes to send new DVs

Topology change	link or node failure/recovery
Configuration change	link cost change
Periodically	refresh the link-state information every (say) 30 minutes account for possible data corruption

Optimum 3-hops path

A		
Dst	Cst	Hop
A	0	A
B	4	B
C	**6**	**E**
D	7	F
E	2	E
F	5	E

B		
Dst	Cst	Hop
A	4	A
B	0	B
C	2	F
D	3	D
E	4	F
F	1	F



C		
Dst	Cst	Hop
A	**6**	**F**
B	2	F
C	0	C
D	1	D
E	4	F
F	1	F

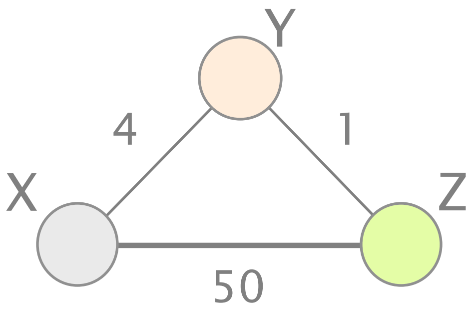
D		
Dst	Cst	Hop
A	7	B
B	3	B
C	1	C
D	0	D
E	**5**	**C**
F	2	C

E		
Dst	Cst	Hop
A	2	A
B	4	F
C	4	F
D	**5**	**F**
E	0	E
F	3	F

F		
Dst	Cst	Hop
A	5	B
B	1	B
C	1	C
D	2	C
E	3	E
F	0	F

Consider the following network

leading to the following vectors



Y vector

dest.	via X	via Z
X	4	6

Y reaches X directly

Z vector

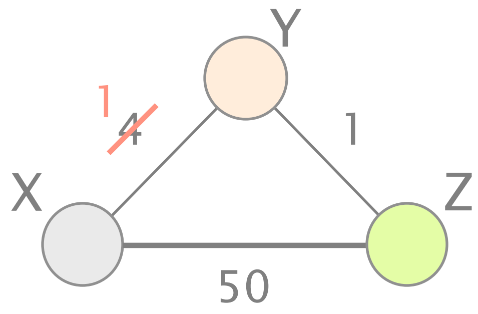
dest.	via X	via Y
X	50	5

Z reaches X via Y

t > 3

no one moves anymore

network has converged!



t=0 | t=1 | t=2 | t>3

Y vector

t=0

dest.	via X	via Z
X	4	6

t=1

dest.	via X	via Z
X	1	6

t>3

dest.	via X	via Z
X	1	3

Z vector

t=0

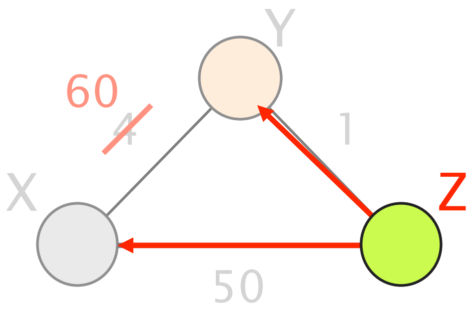
dest.	via X	via Y
X	50	5

t=2

dest.	via X	via Y
X	50	2

t>3

dest.	via X	via Y
X	50	2



t=4

t=44

Y vector

... many iterations later ...

dest.	via X	via Z
X	60	51

Z vector

t=4:

dest.	via X	via Y
X	50	9

t=44:

dest.	via X	via Y
X	50	52

This problem is known as count-to-infinity, a type of routing loop

Count-to-infinity leads to very slow convergence

what if the cost had changed from 4 to 9999?

Routers don't know when neighbors use them

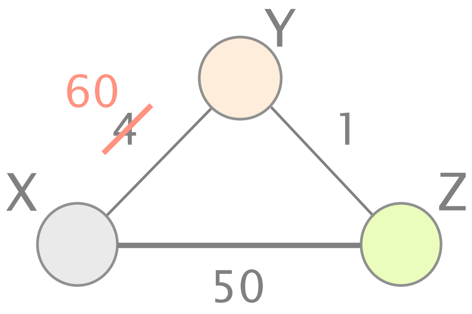
Z does not know that Y has switched to use it

Let's try to fix that

t > 4

no one moves

network has converged!



	t=4	t>4

Y vector

t>4:

dest.	via X	via Z
X	60	51

Z vector

t=4:

dest.	via X	via Y
X	50	∞

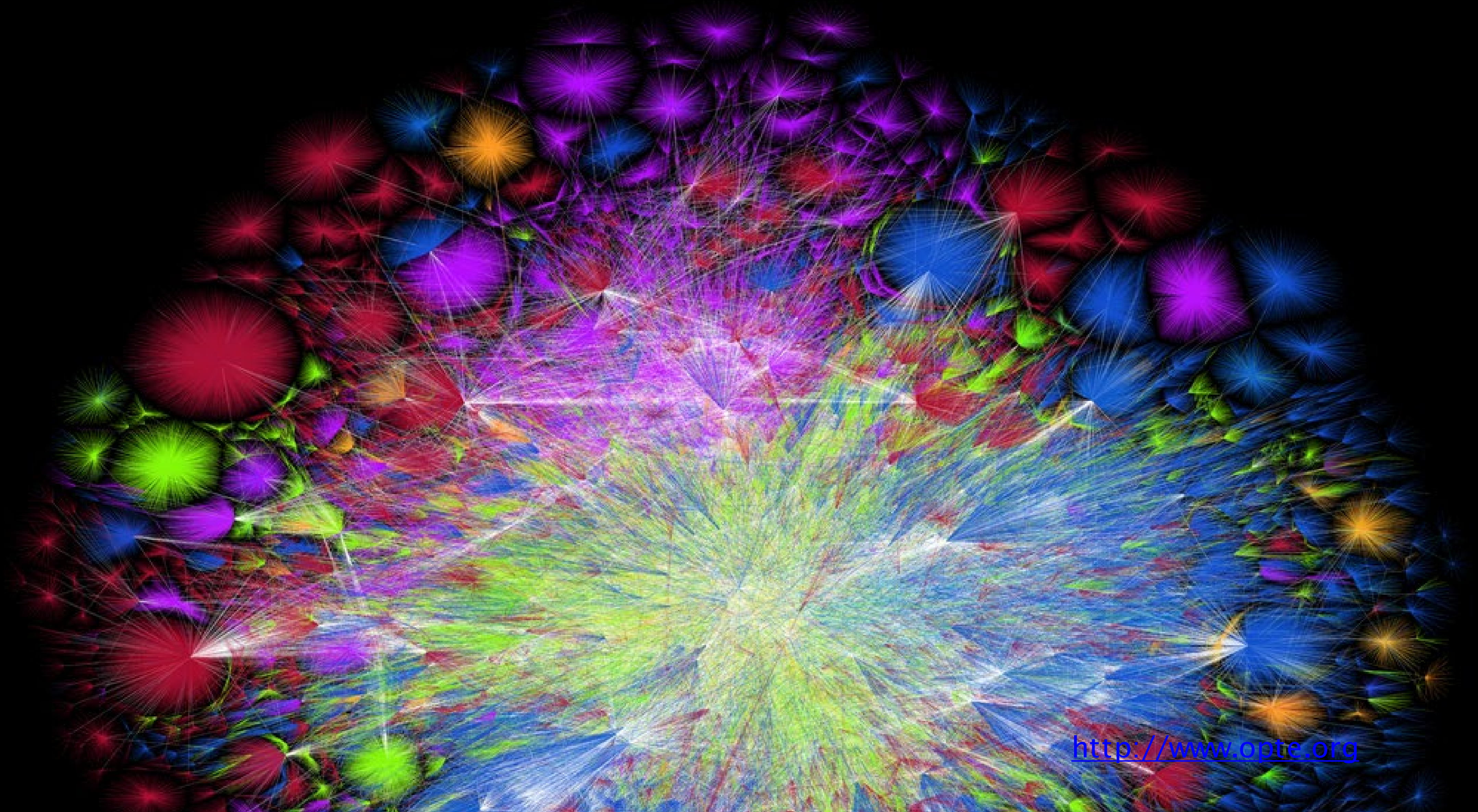
t>4:

dest.	via X	via Y
X	50	∞

Communication Networks and Internet Technology

This weeks lecture

Internet routing



http://www.opte.org

Internet routing

from here to there, and back

Intra-domain routing

Link-state protocols

Distance-vector protocols

2 Inter-domain routing

Path-vector protocols

Internet

Internet

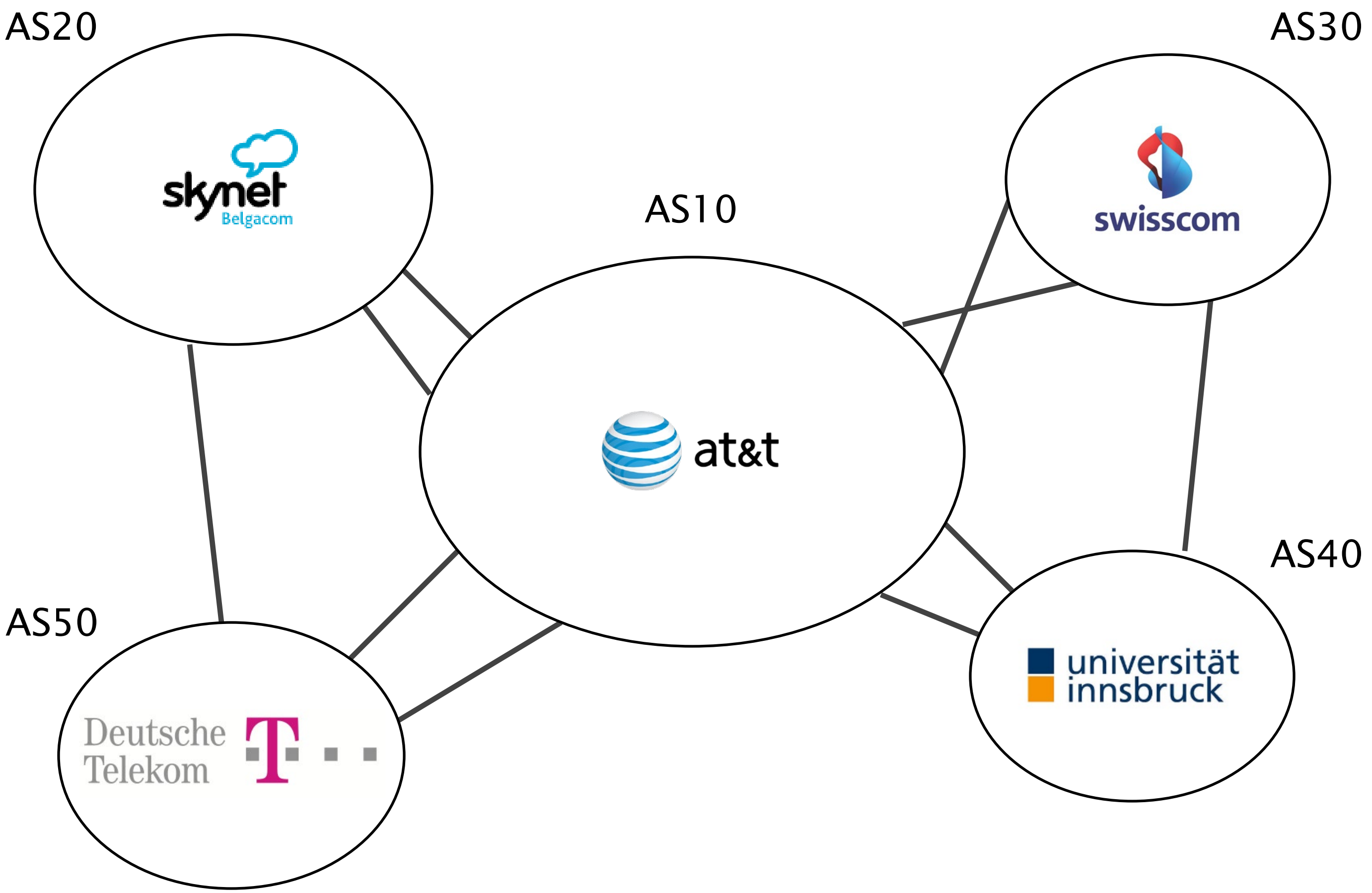
Internet

A network of *networks*

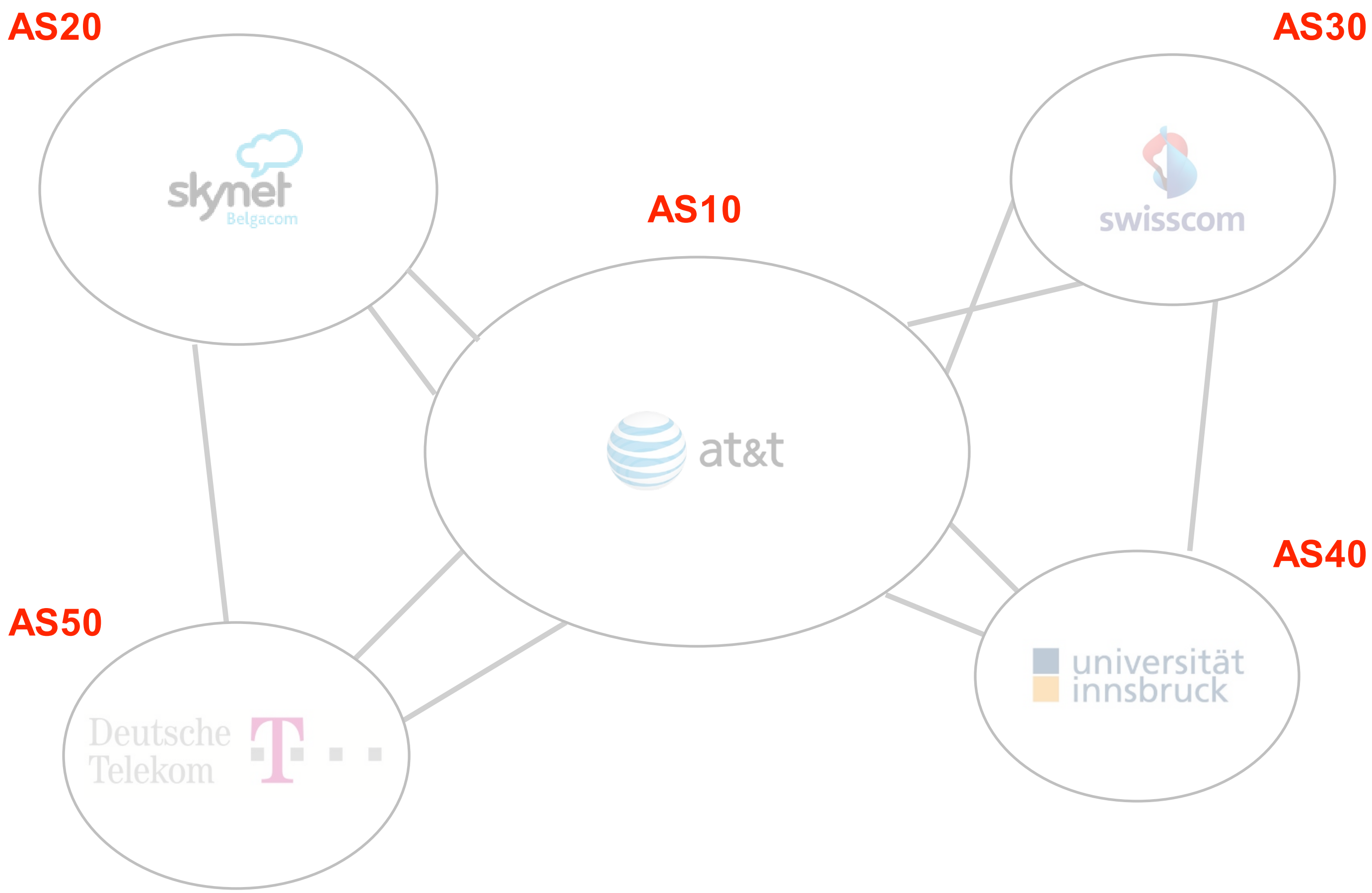
Internet

Border Gateway Protocol (BGP)

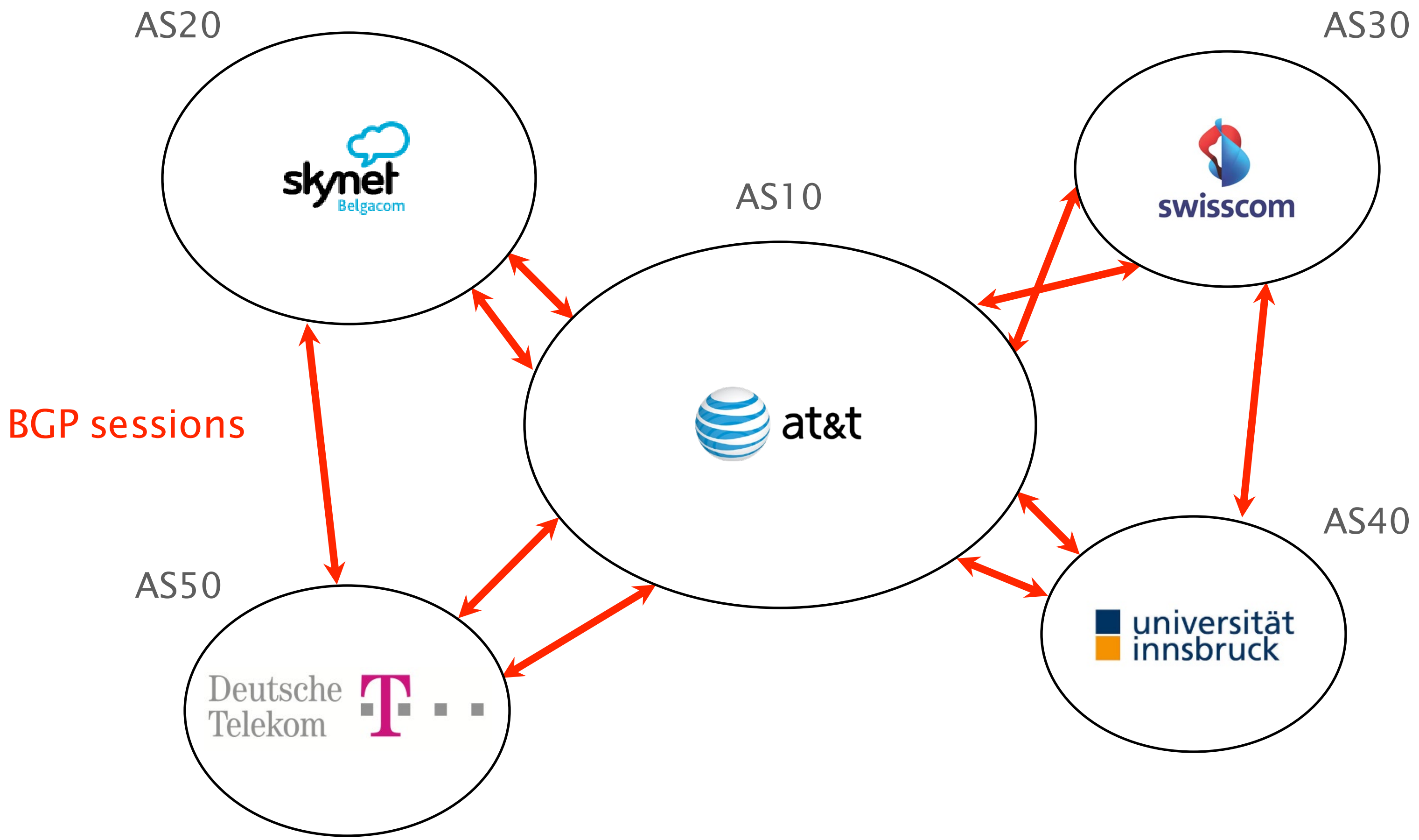
The Internet is a network of networks, referred to as Autonomous Systems (AS)



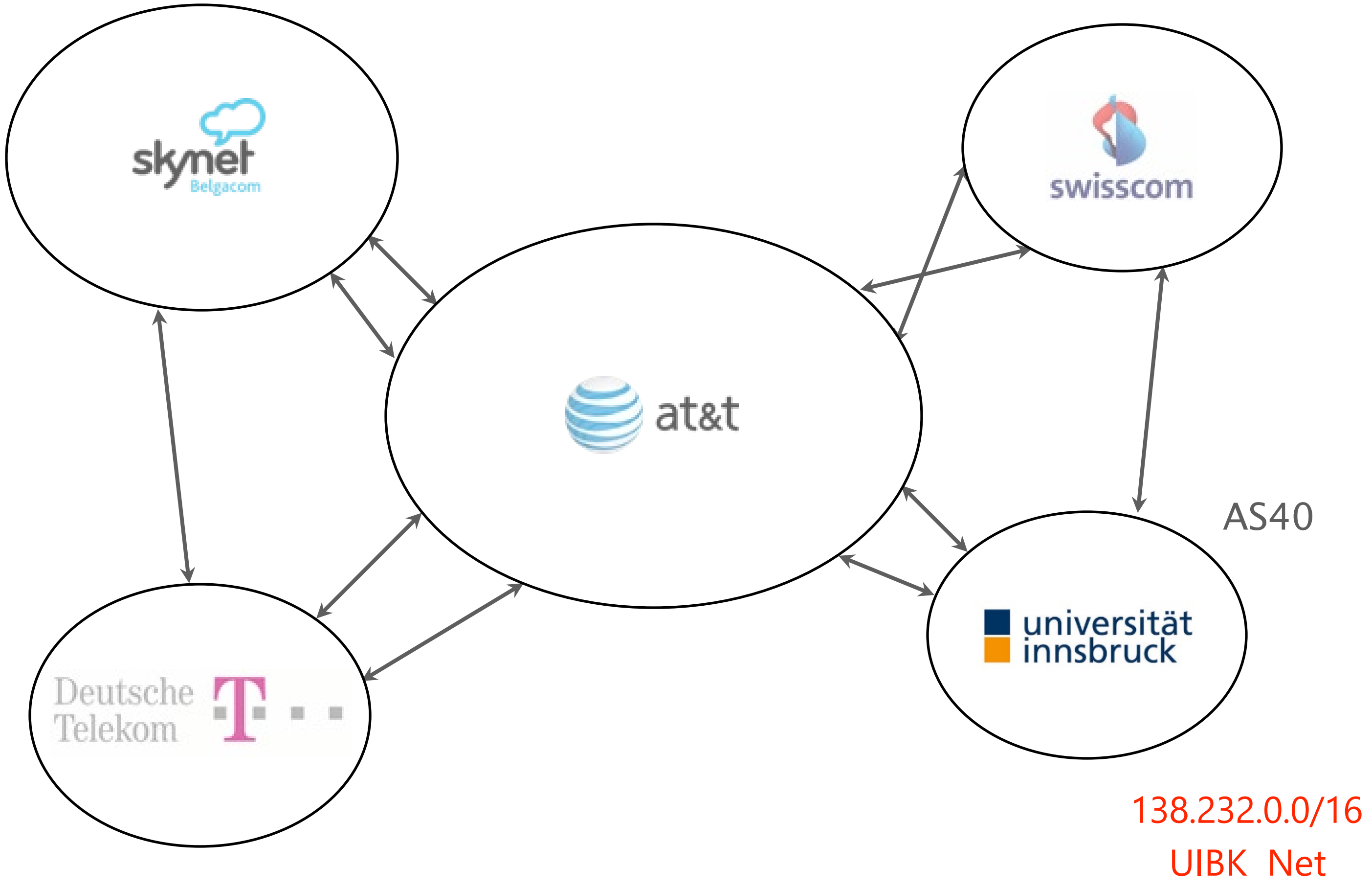
Each AS has a number (encoded on 16 bits) which identifies it



BGP is the routing protocol "glueing" the entire Internet together



Using BGP, ASes exchange information about the IP prefixes they can reach, directly or indirectly



BGP needs to solve three key challenges:
scalability, privacy and policy enforcement

There is a huge # of networks and prefixes

700k prefixes, >50,000 networks, millions (!) of routers

Networks don't want to divulge internal topologies

or their business relationships

Networks need to control where to send and receive traffic

without an Internet-wide notion of a link cost metric

Link-State routing does not solve these challenges

Floods topology information

high processing overhead

Requires each node to compute the entire path

high processing overhead

Minimizes some notion of total distance

works only if the policy is shared and uniform

Distance-Vector routing is on the right track

pros	Hide details of the network topology
	nodes determine only “next-hop” for each destination

Distance-Vector routing is on the right track,

but not really there yet…

pros

Hide details of the network topology

nodes determine only “next-hop” for each destination

cons

It still minimizes some common distance

impossible to achieve in an inter domain setting

It converges slowly

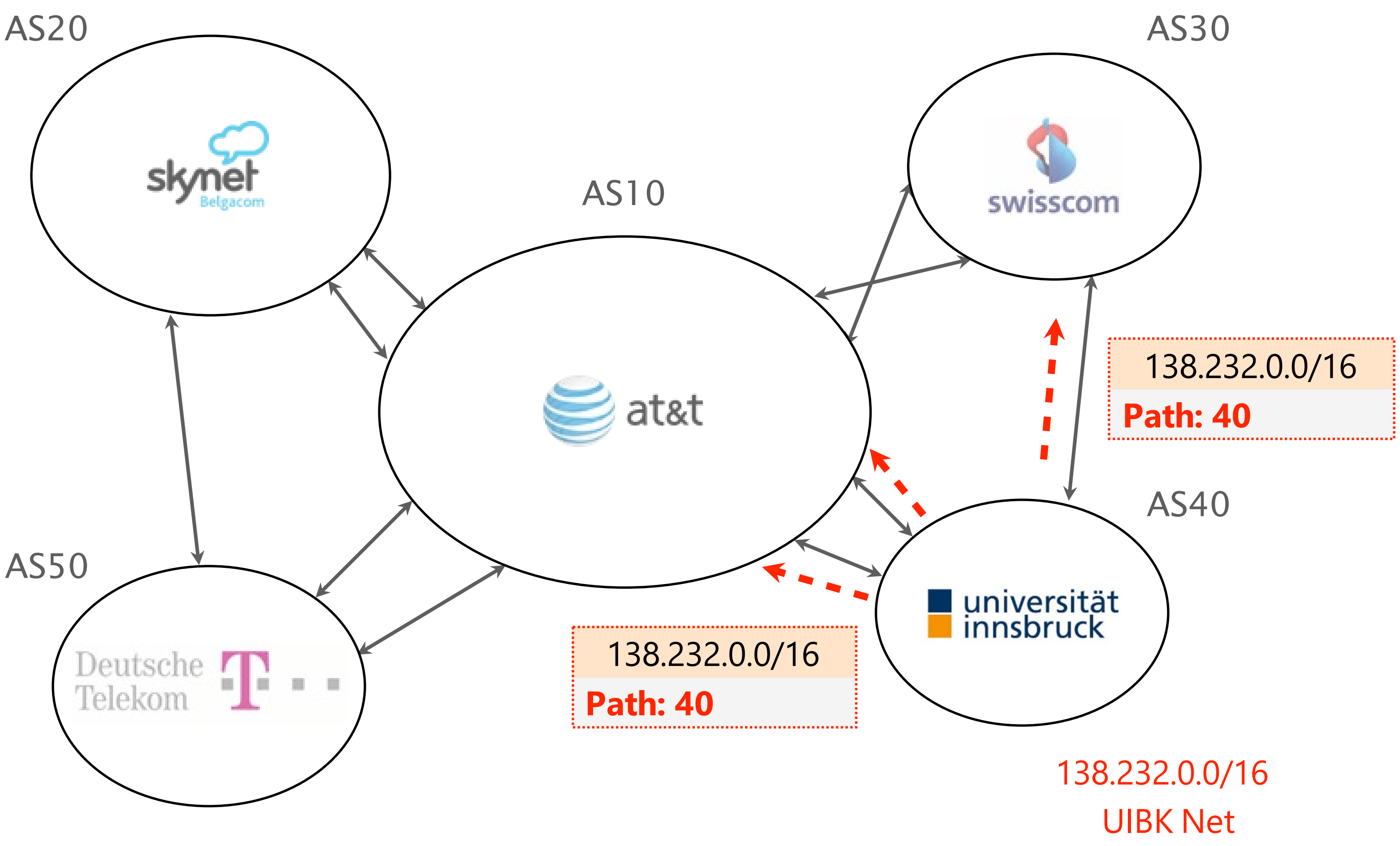
counting-to-infinity problem

BGP relies on path-vector routing to support flexible routing policies and avoid count-to-infinity

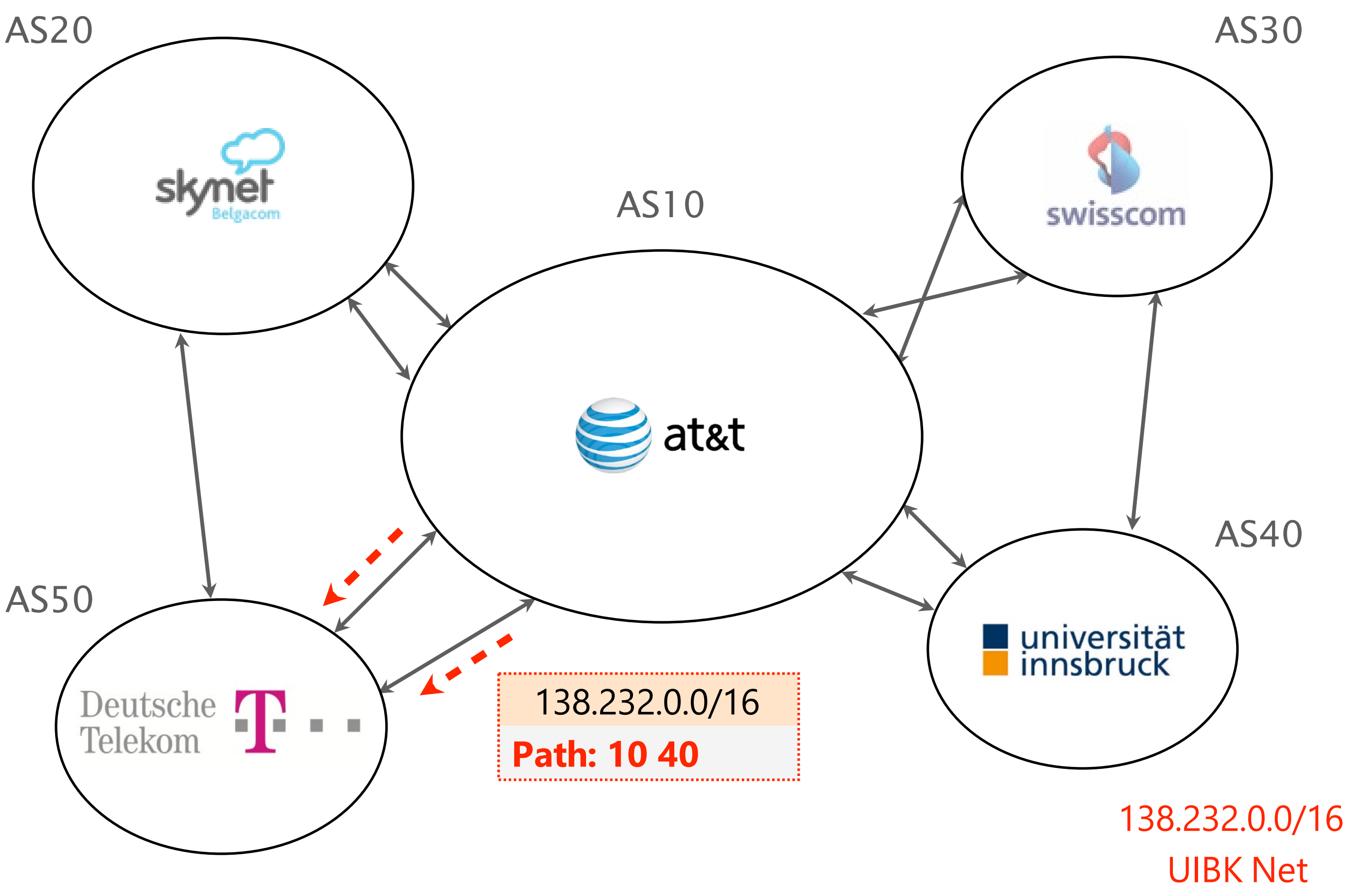
key idea

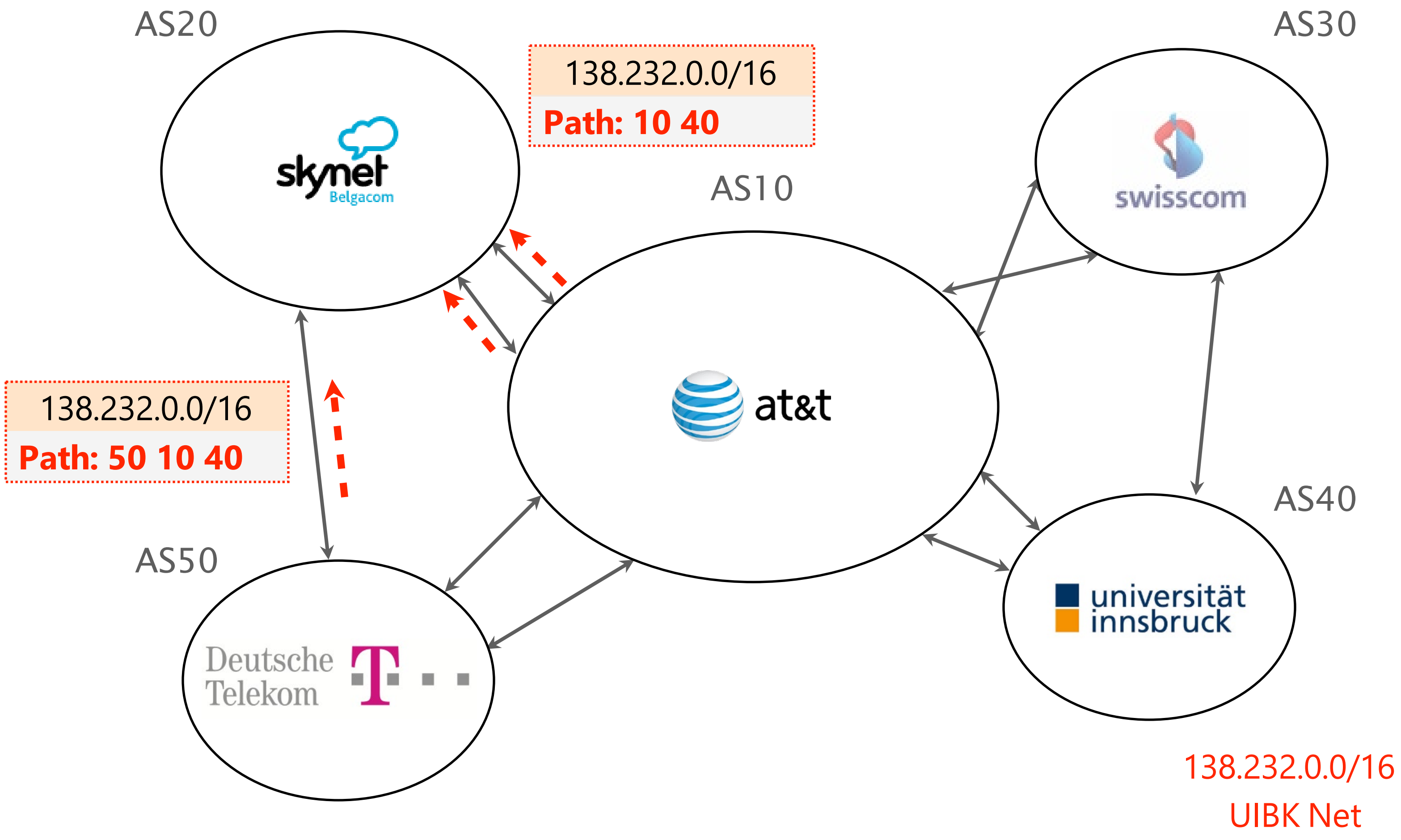
advertise the entire path instead of distances

BGP announcements carry complete path information instead of distances



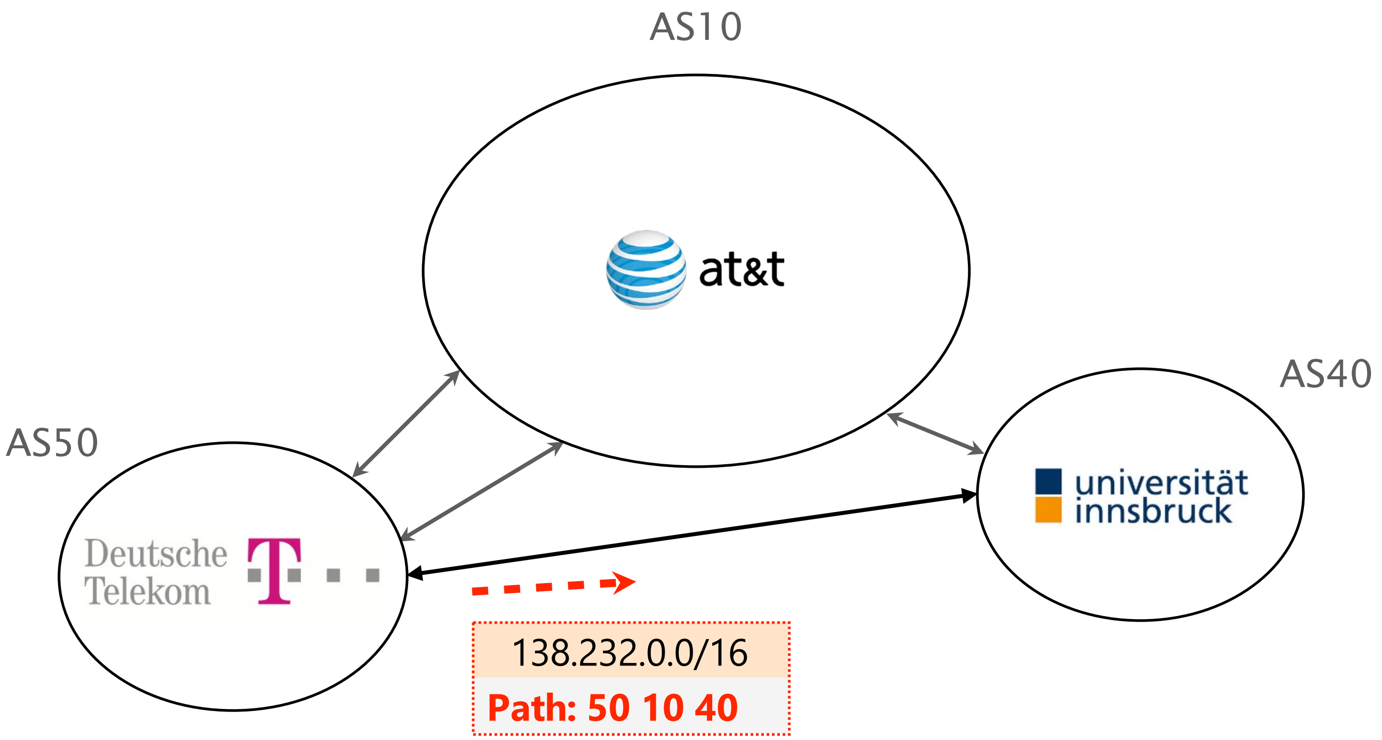
Each AS appends itself to the path when it propagates announcements




AS20
138.232.0.0/16
Path: 10 40
AS30
AS10
138.232.0.0/16
Path: 50 10 40
AS40
AS50
138.232.0.0/16
UIBK Net

Complete path information enables ASes to easily detect a loop

UIBK sees itself in the path and discard the route



Life of a BGP router is made of three consecutive steps

while true:

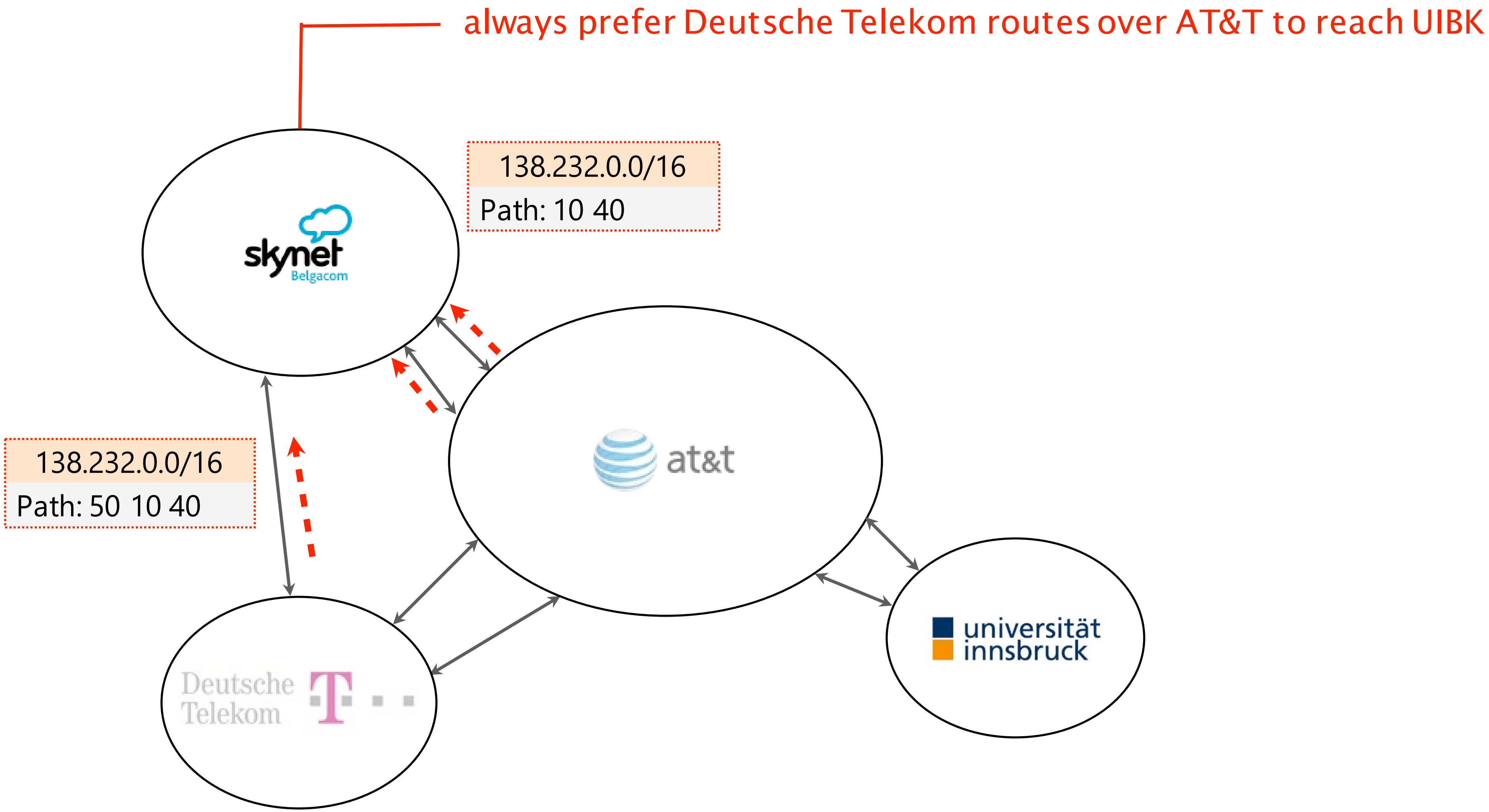
- receives routes from my neighbors
- select one best route for each prefix
- export the best route to my neighbors

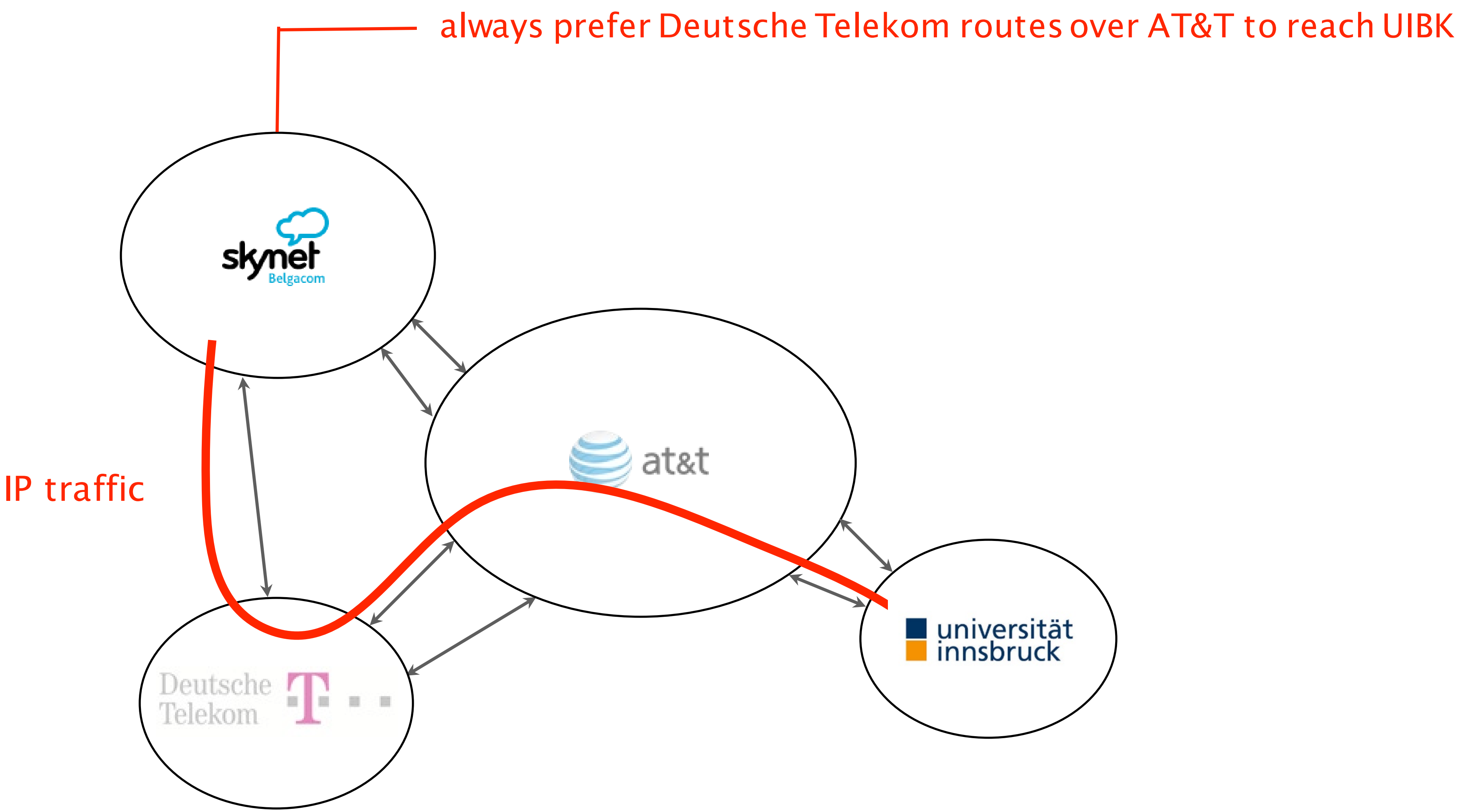
Each AS can apply local routing policies

Each AS is free to

- select and use any path

  preferably, the cheapest one


always prefer Deutsche Telekom routes over AT&T to reach UIBK
138.232.0.0/16
Path: 10 40
138.232.0.0/16
Path: 50 10 40
skynet
Belgacom
at&t
Deutsche Telekom
universität innsbruck


always prefer Deutsche Telekom routes over AT&T to reach UIBK
skynet
Belgacom
IP traffic
at&t
Deutsche
Telekom
universität
innsbruck

Each AS can apply local routing policies

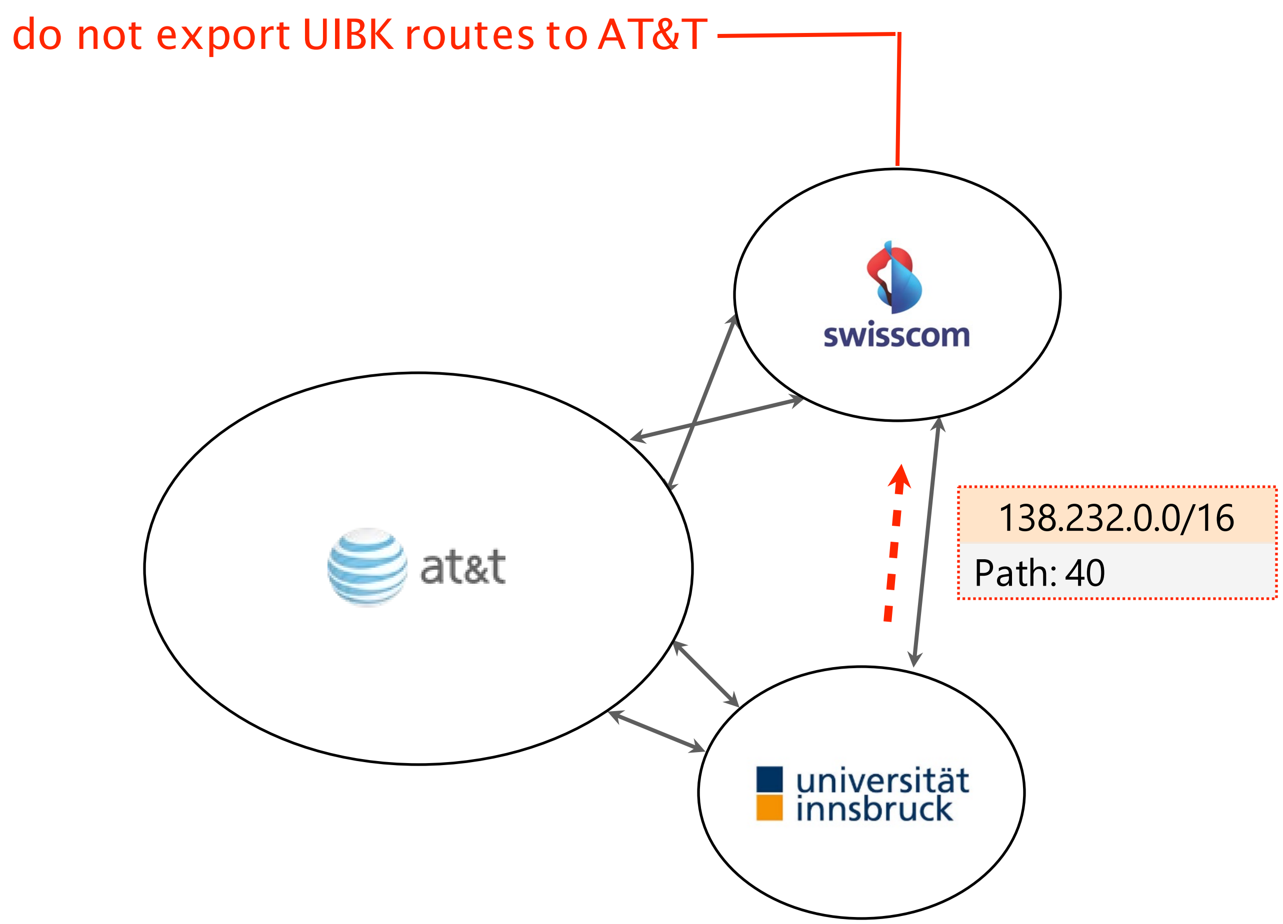
Each AS is free to

- select and use any path

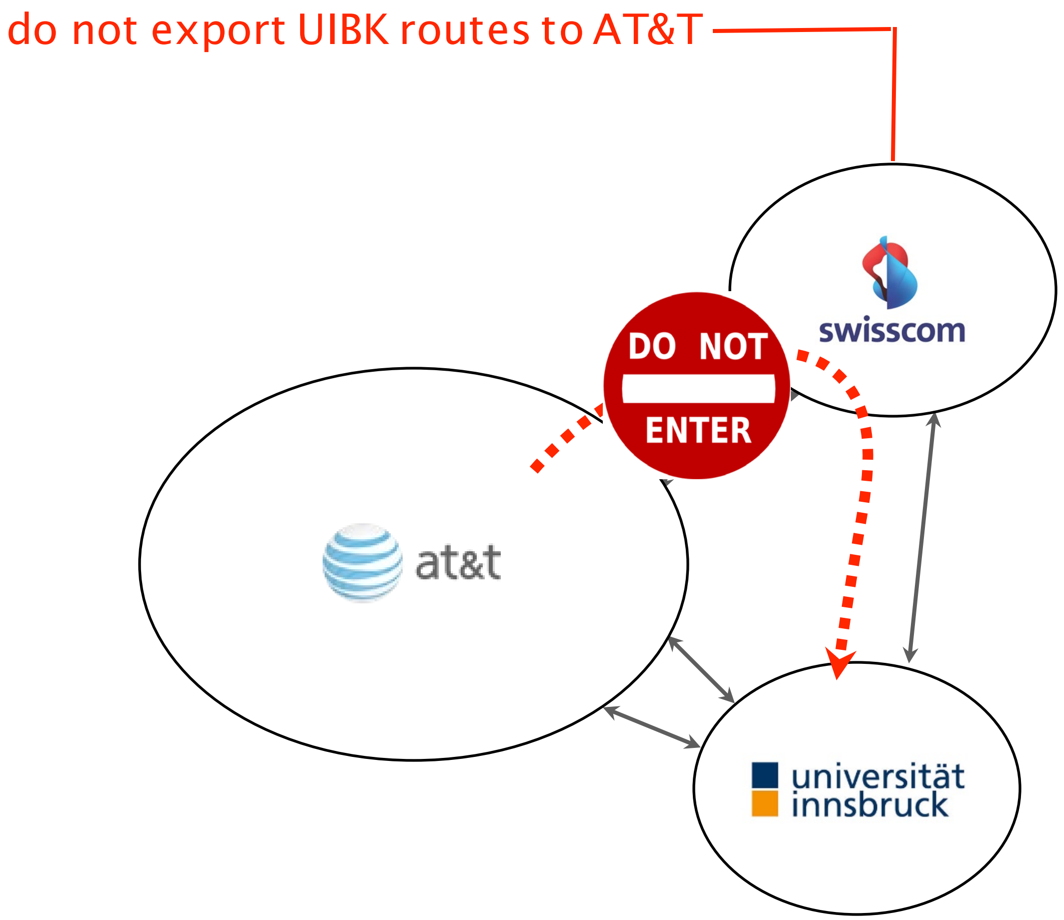
  preferably, the cheapest one

- decide which path to export (if any) to which neighbor

  preferably, none to minimize carried traffic



do not export UIBK routes to AT&T
swisscom
at&t
138.232.0.0/16
Path: 40
universität
innsbruck


do not export UIBK routes to AT&T
DO NOT
ENTER
swisscom
at&t
universität
innsbruck

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

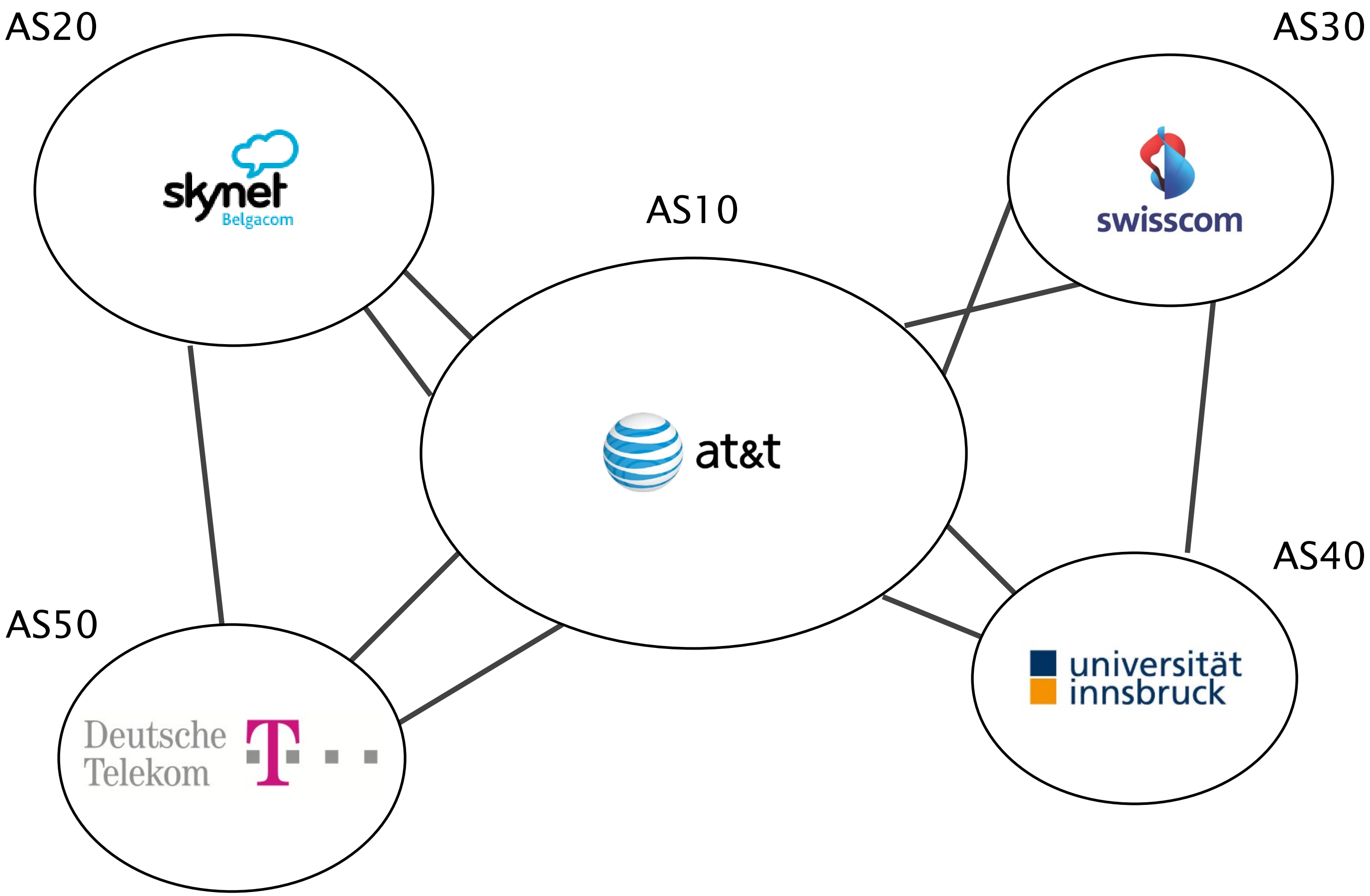
Protocol

How does it work?

Problems

security, performance, …

The Internet topology is shaped according to business relationships



Intuition

2 ASes connect only if they have a business relationship

BGP is a “follow the money” protocol

There are 2 main business relationships today:

- customer/provider
- peer/peer

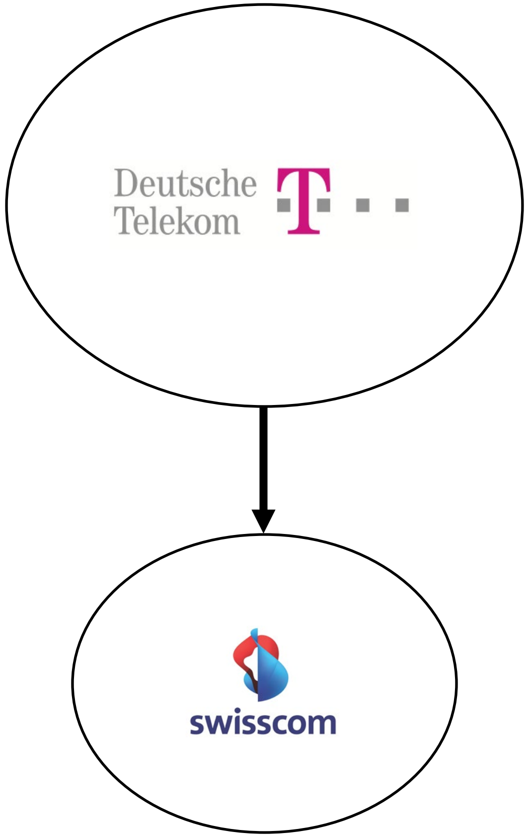
many less important ones (siblings, backups,...)

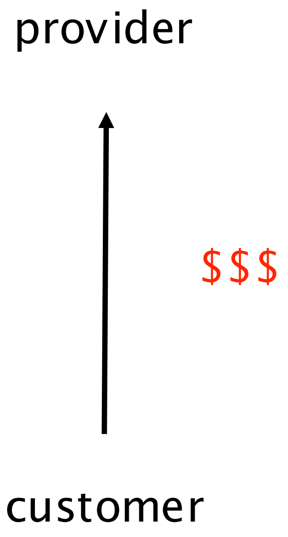
There are 2 main business relationships today:

- customer/provider
- peer/peer

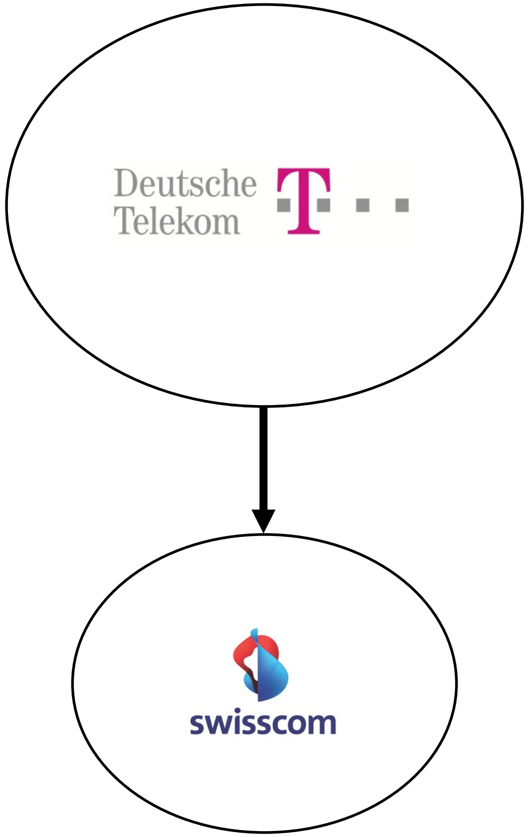
Customers pay providers

to get Internet connectivity





The amount paid is based on peak usage, usually according to the 95th percentile rule



Every 5 minutes, DT
records the # of bytes sent/received

At the end of the month, DT

- sorts all values in decreasing order
- removes the top 5% values
- bills wrt highest remaining value

Most ISPs discounts traffic unit price when pre-committing to certain volume

commit		unit price ($)	Minimum monthly bill ($/month)
10	Mbps	12	120
100	Mbps	5	500
1	Gbps	3.50	3,500
10	Gbps	1.20	12,000
100	Gbps	0.70	70,000

Examples taken from The 2014 Internet Peering Playbook

Internet Transit Prices have been continuously declining during the last 20 years

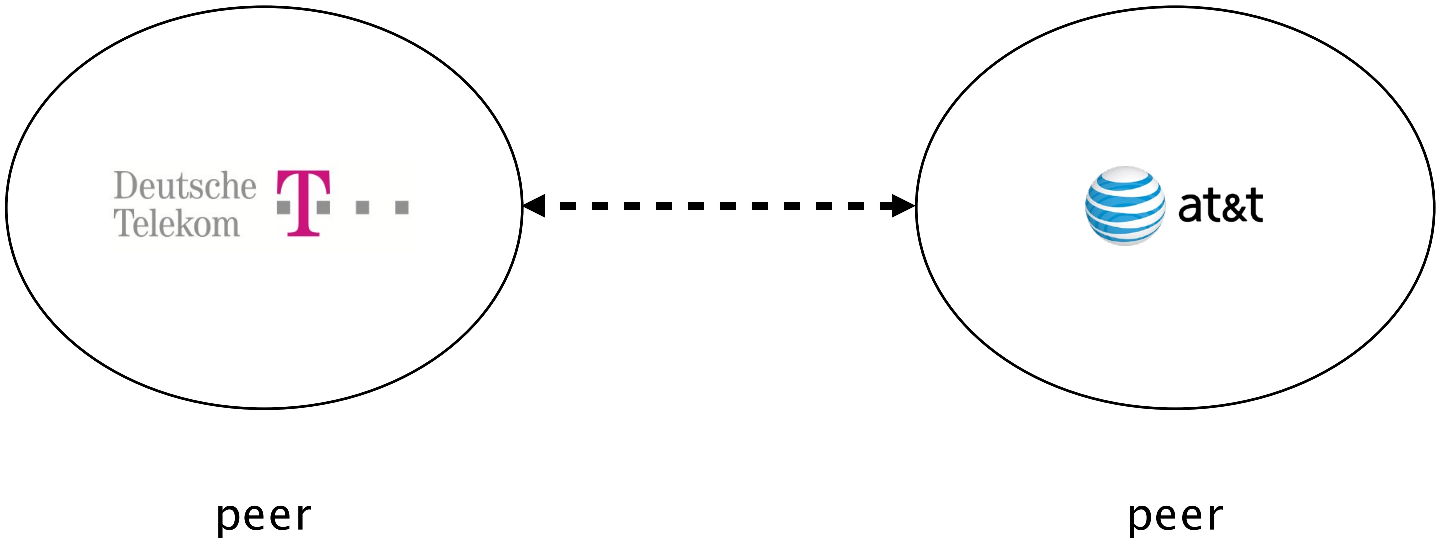
Internet Transit Pricing (1998-2015)			
Source: http://DrPeering.net			
Year	**Internet Transit Price**		**% decline**
1998	$1,200.00	per Mbps	
1999	$800.00	per Mbps	33%
2000	$675.00	per Mbps	16%
2001	$400.00	per Mbps	41%
2002	$200.00	per Mbps	50%
2003	$120.00	per Mbps	40%
2004	$90.00	per Mbps	25%
2005	$75.00	per Mbps	17%
2006	$50.00	per Mbps	33%
2007	$25.00	per Mbps	50%
2008	$12.00	per Mbps	52%
2009	$9.00	per Mbps	25%
2010	$5.00	per Mbps	44%
2011	$3.25	per Mbps	35%
2012	$2.34	per Mbps	28%
2013	$1.57	per Mbps	33%
2014	$0.94	per Mbps	40%
2015	$0.63	per Mbps	33%

The reason? Internet commoditization & competition

There are 2 main business relationships today:

- customer/provider
- peer/peer

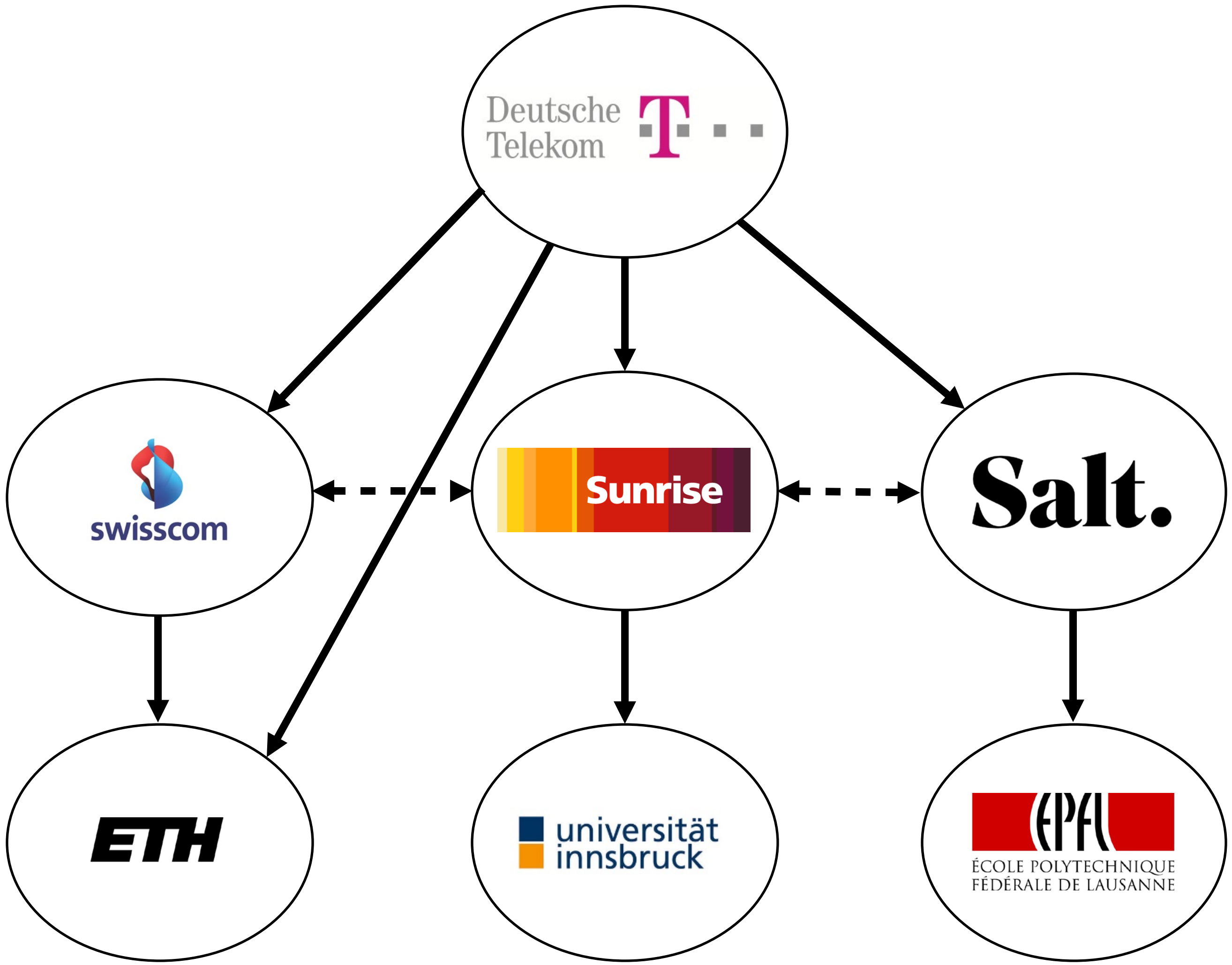
Peers don’t pay each other for connectivity,
they do it *out of common interest*



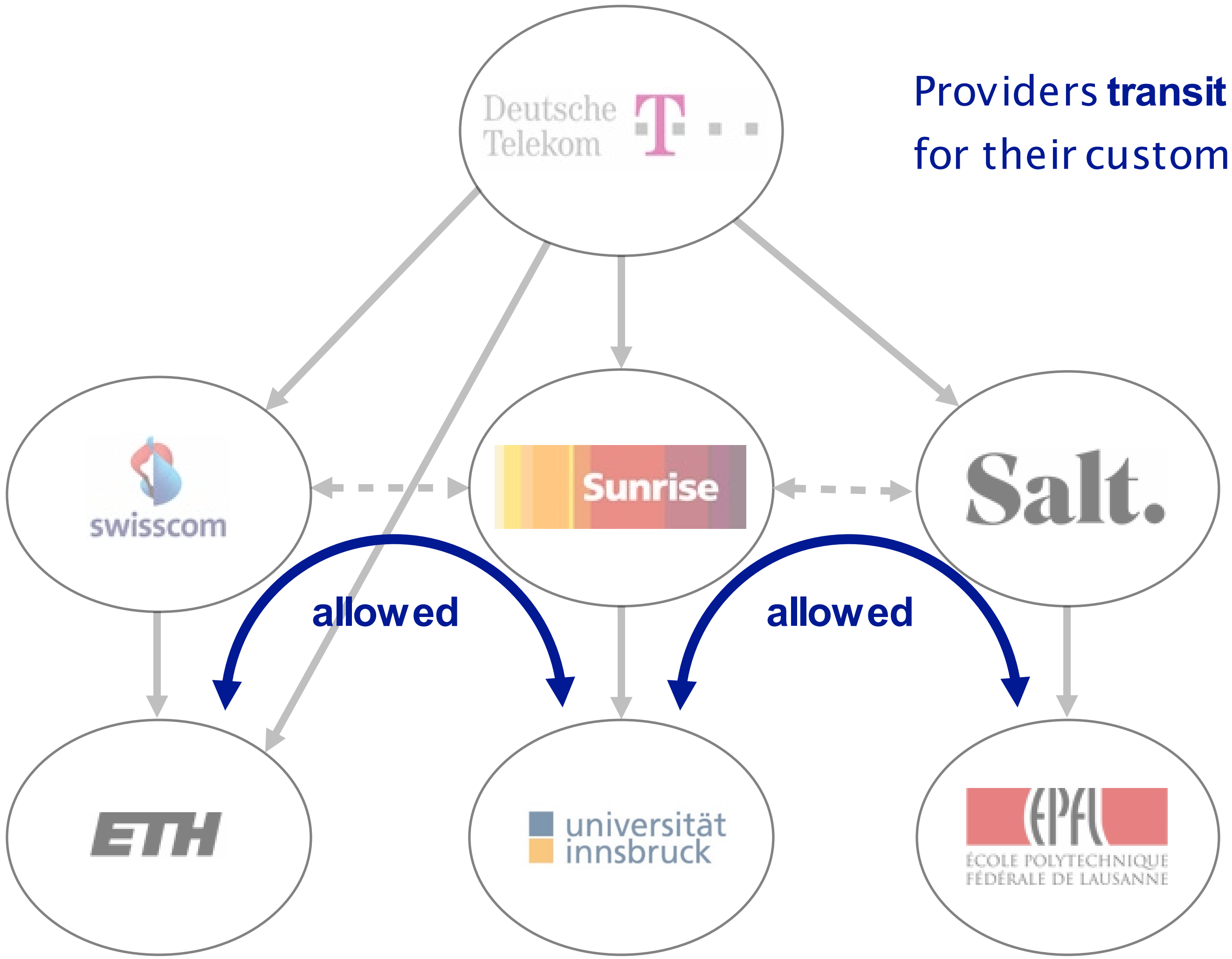
DT and ATT exchange *tons* of traffic.
they save money by directly connecting to each other

To understand Internet routing,

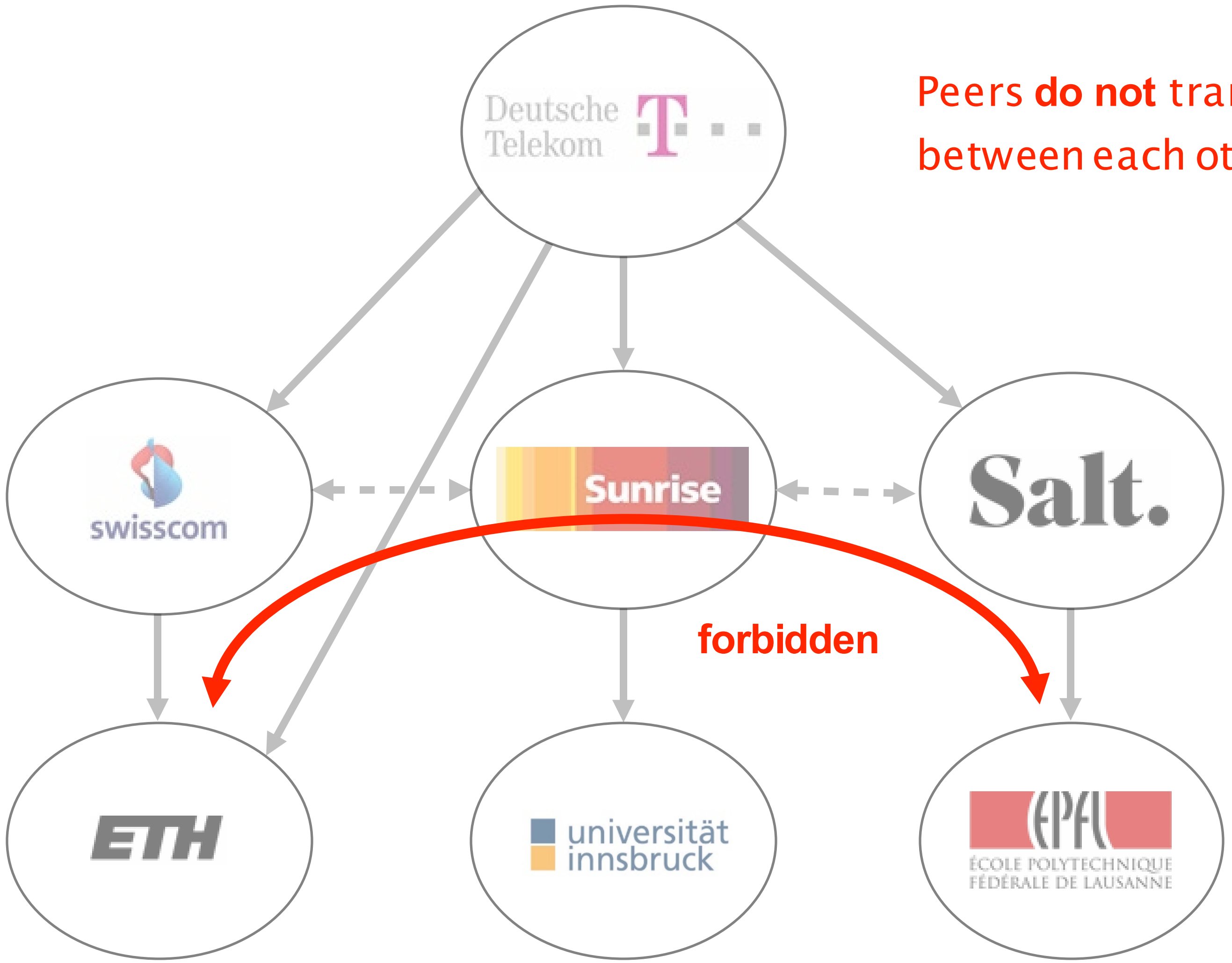
follow the money

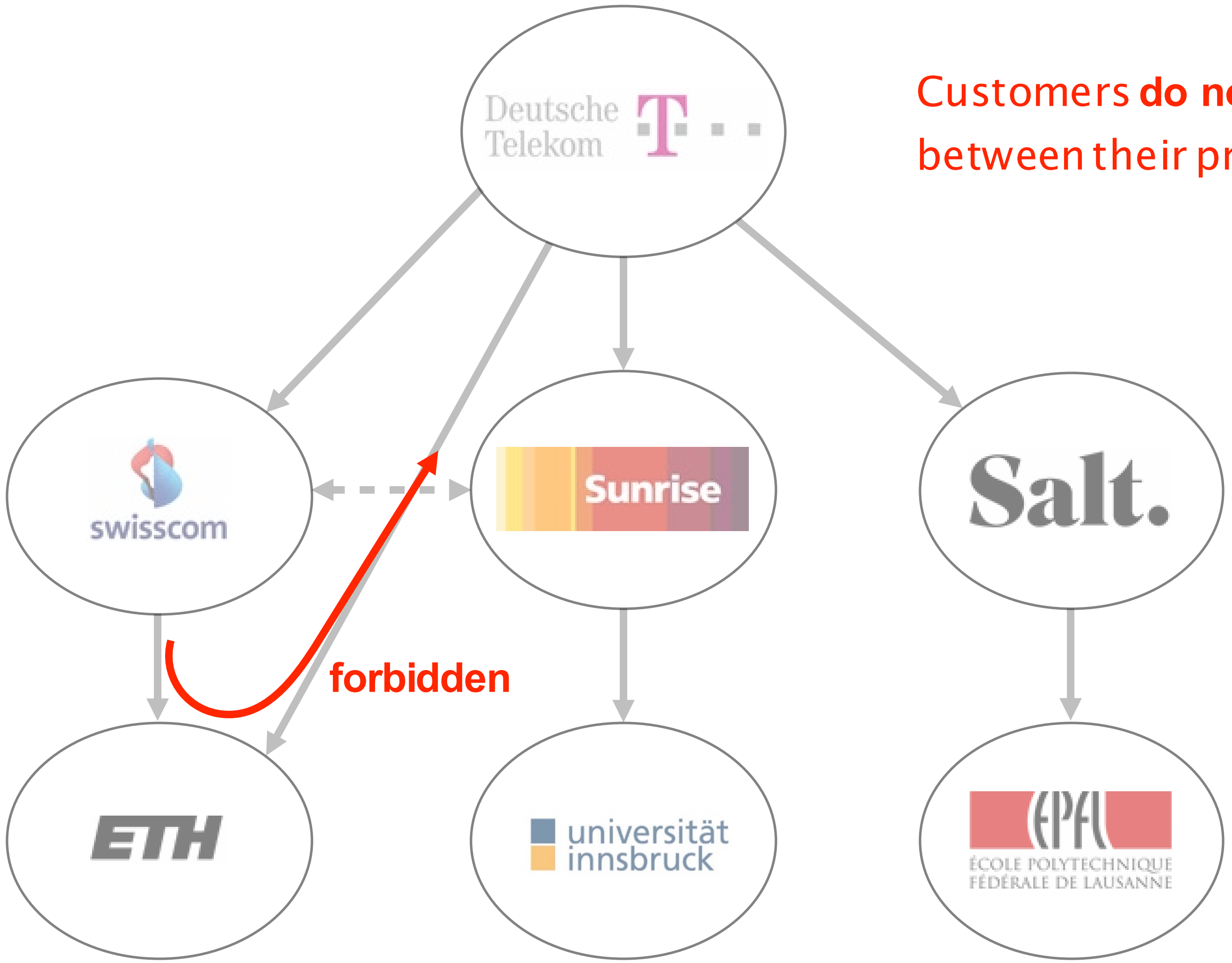

Deutsche Telekom
swisscom
Sunrise
Salt.
ETH
universität innsbruck
EPFL
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Providers **transit** traffic for their customers



Peers **do not** transit traffic between each other





Customers **do not** transit traffic between their providers

These policies are defined by constraining which BGP routes are *selected* and *exported*

Selection

Export

which path to use?

which path to advertise?

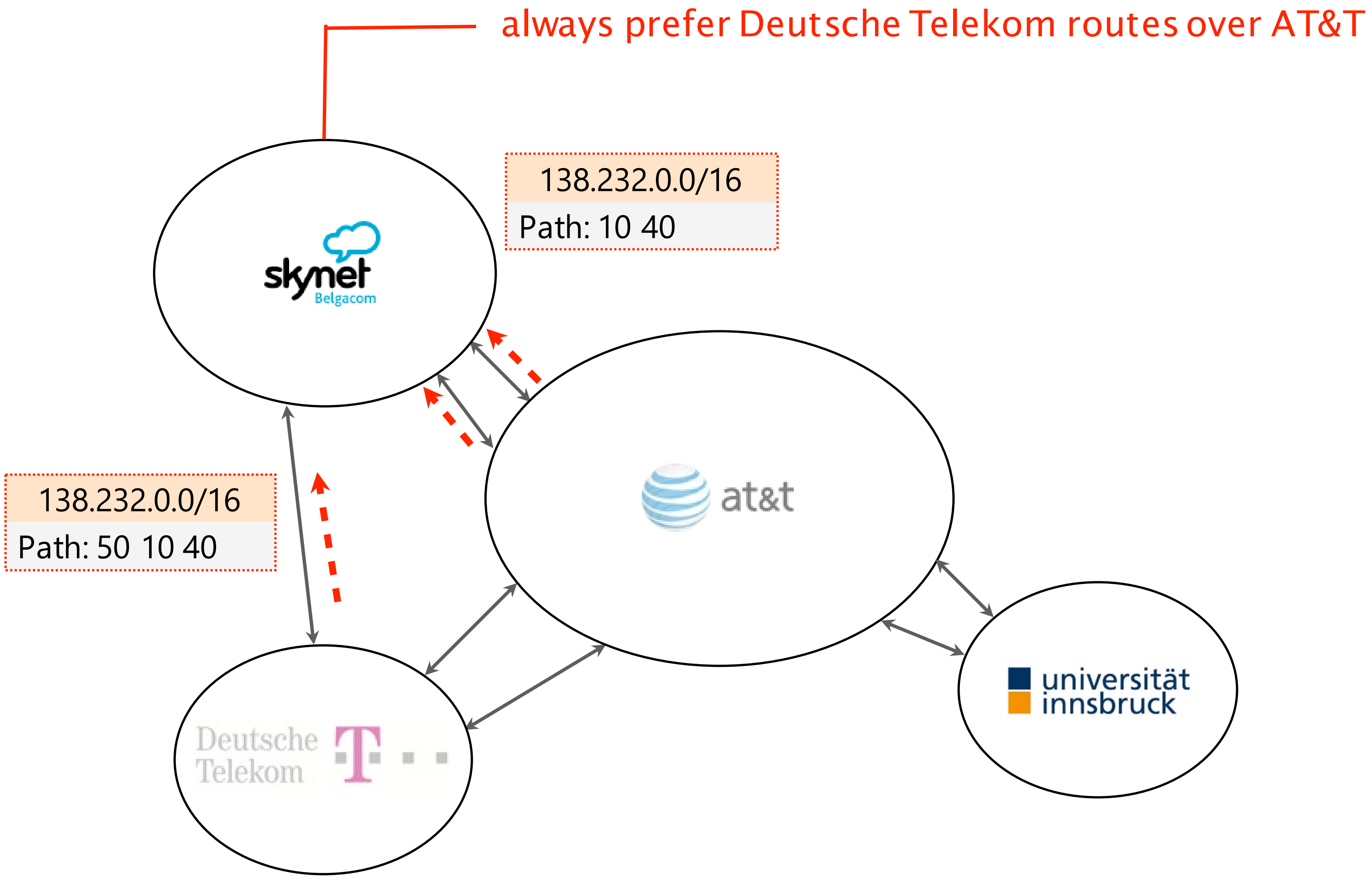
Selection

Export

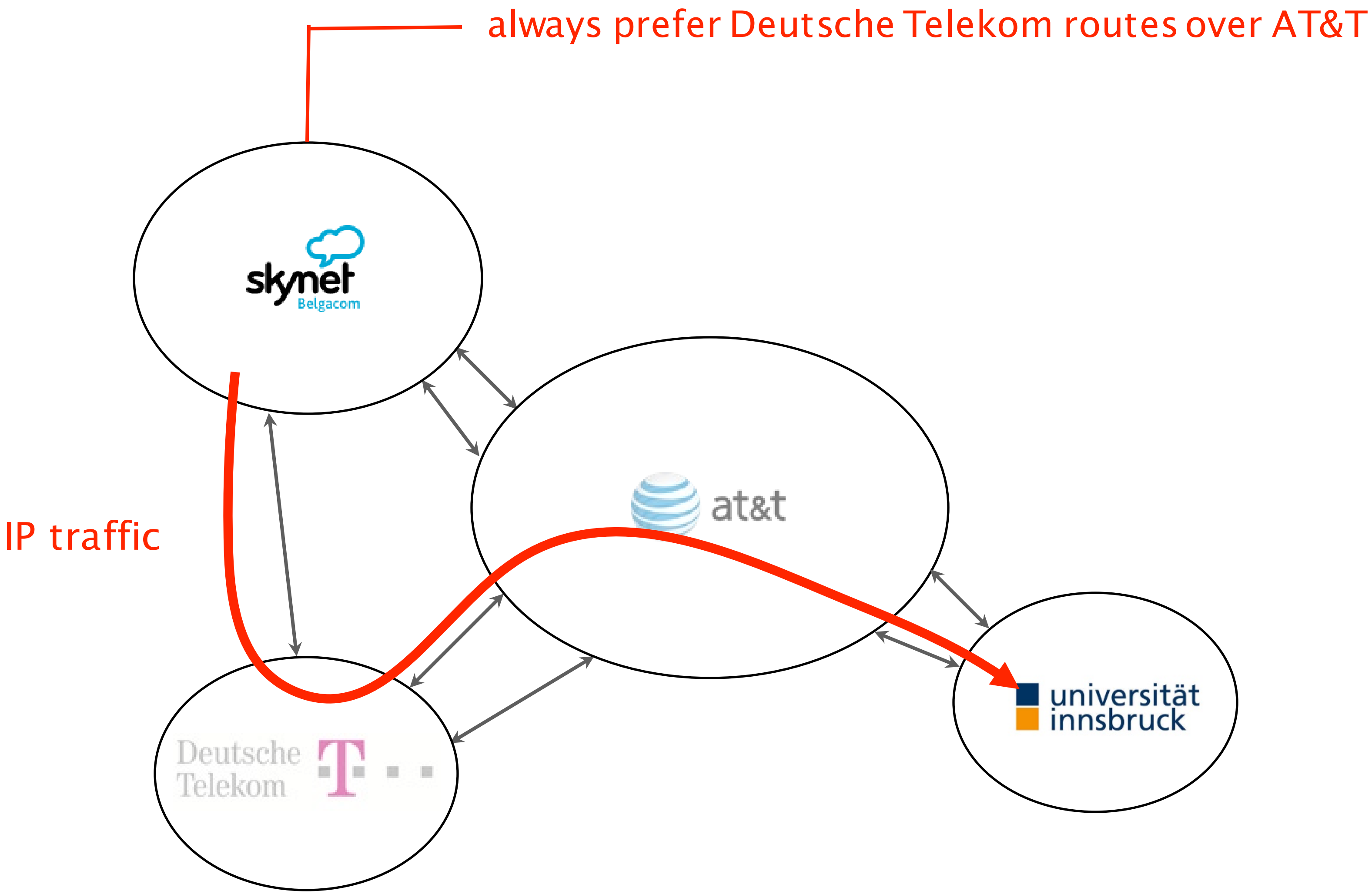
which path to use?

control outbound traffic

which path to advertise?



always prefer Deutsche Telekom routes over AT&T
138.232.0.0/16
Path: 10 40
skynet
Belgacom
138.232.0.0/16
Path: 50 10 40
at&t
Deutsche
Telekom
universität
innsbruck


always prefer Deutsche Telekom routes over AT&T
IP traffic

Business relationships conditions

route selection

For a destination *p*, prefer routes coming from

- customers over
- peers over
- providers

route type

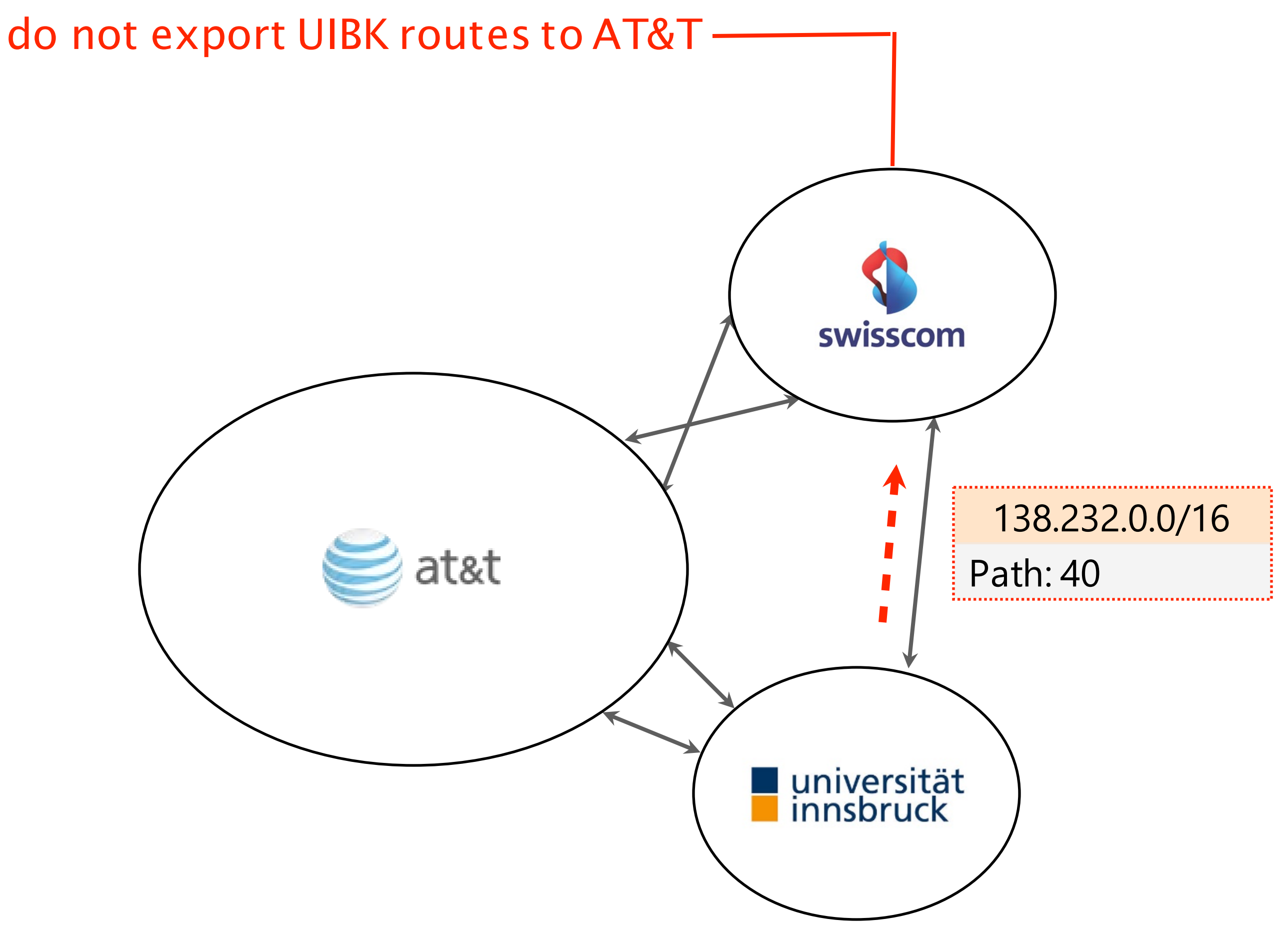
Selection

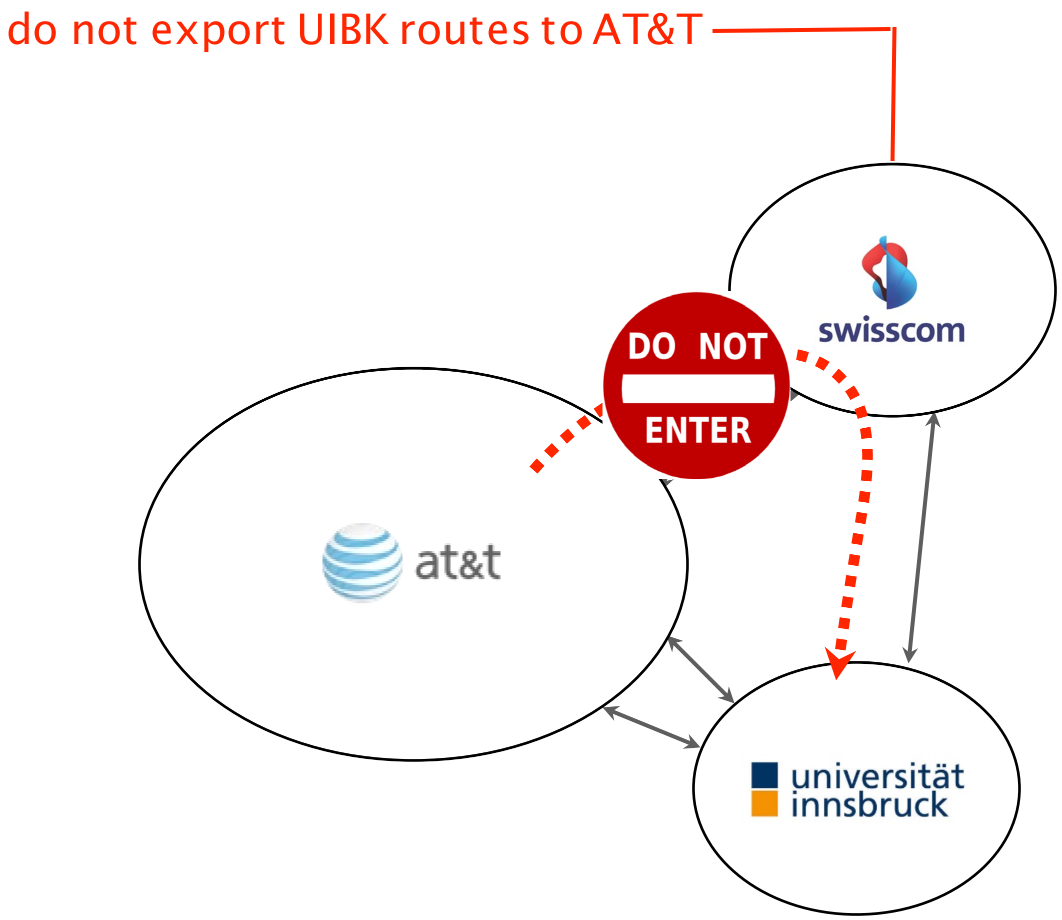
Export

which path to use?

which path to advertise?

control inbound traffic


do not export UIBK routes to AT&T
swisscom
at&t
138.232.0.0/16
Path: 40
universität innsbruck


do not export UIBK routes to AT&T
DO NOT
ENTER
swisscom
at&t
universität
innsbruck

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
	customer	✓	✓	✓
from	peer	✓	-	-
	provider	✓	-	-

Selection

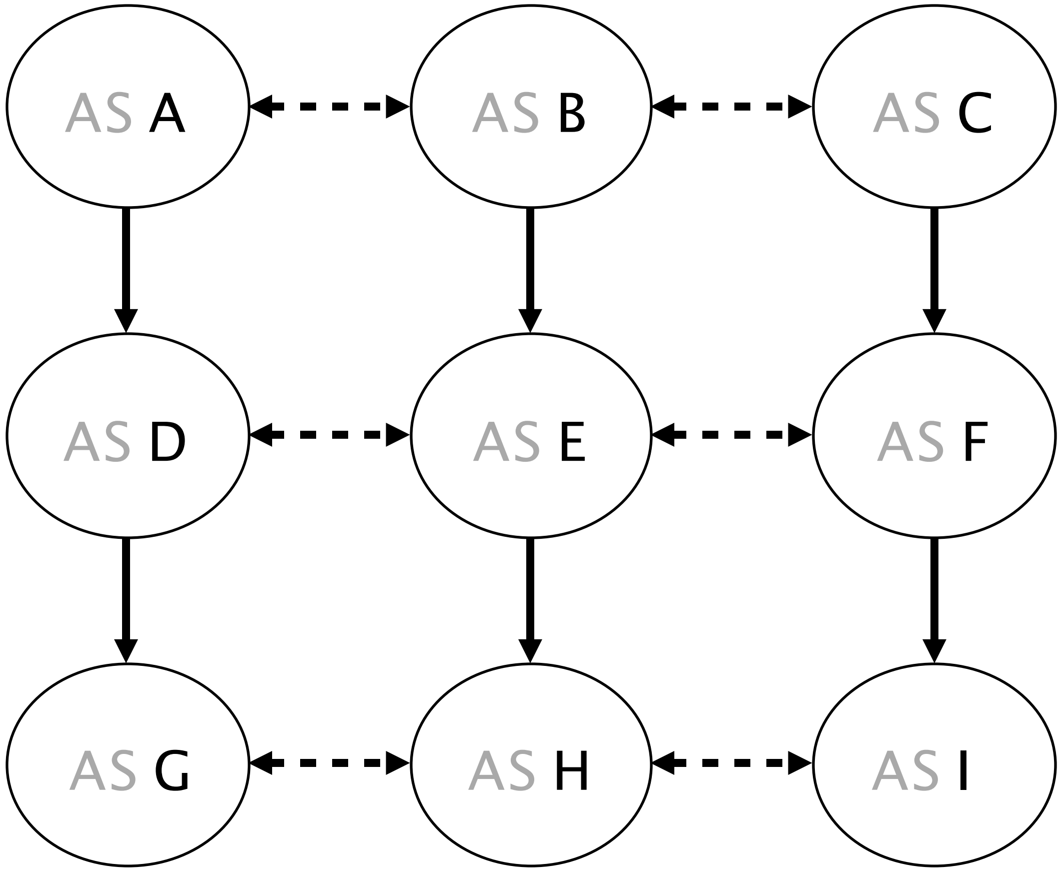
which path to use?

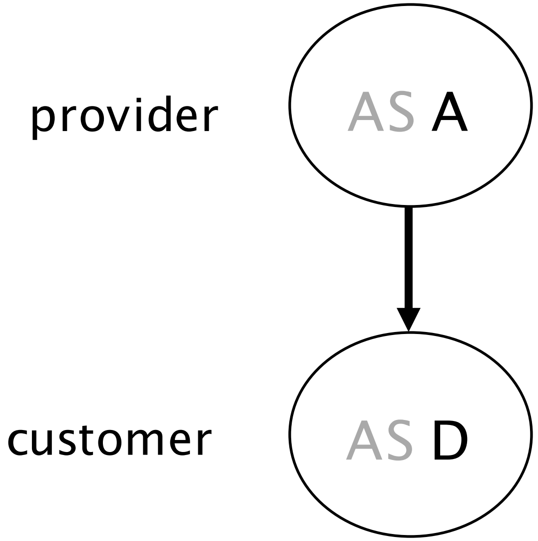
control outbound traffic

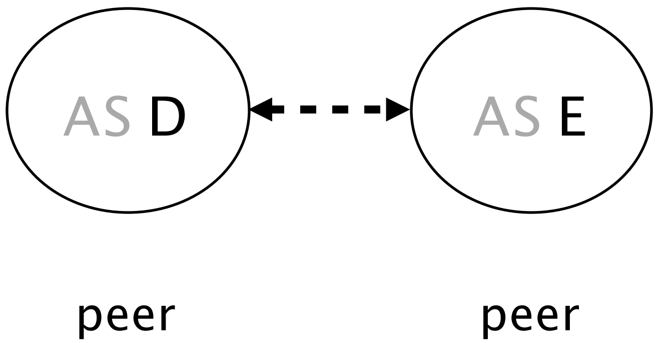
Export

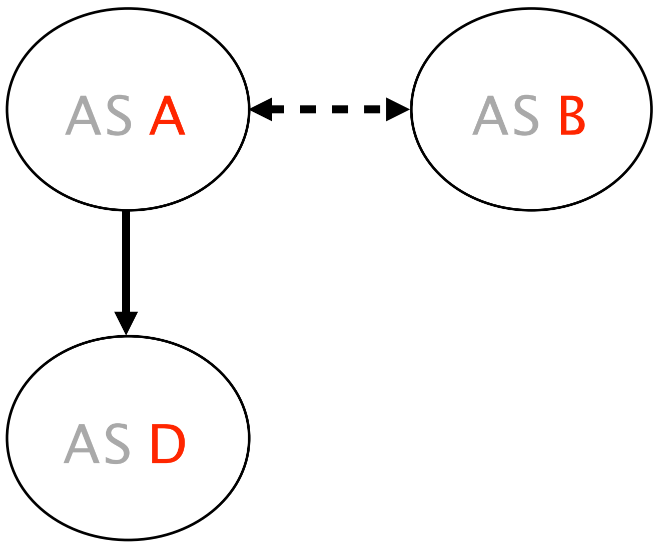
which path to advertise?

control inbound traffic

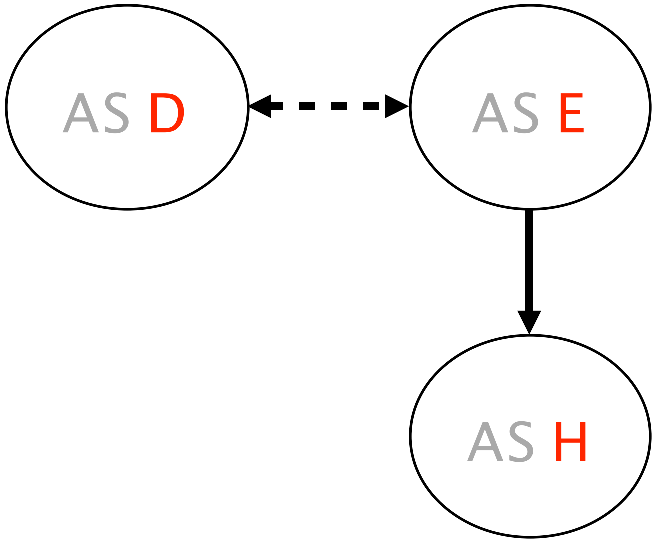

AS A
AS B
AS C
AS D
AS E
AS F
AS G
AS H
AS I


provider
AS A
customer
AS D

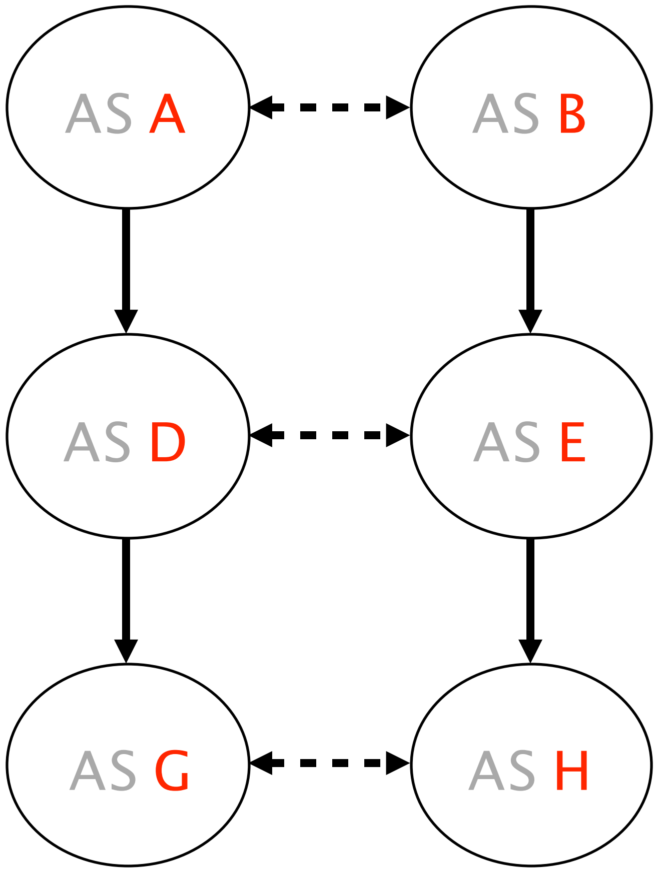

AS D
AS E
peer
peer



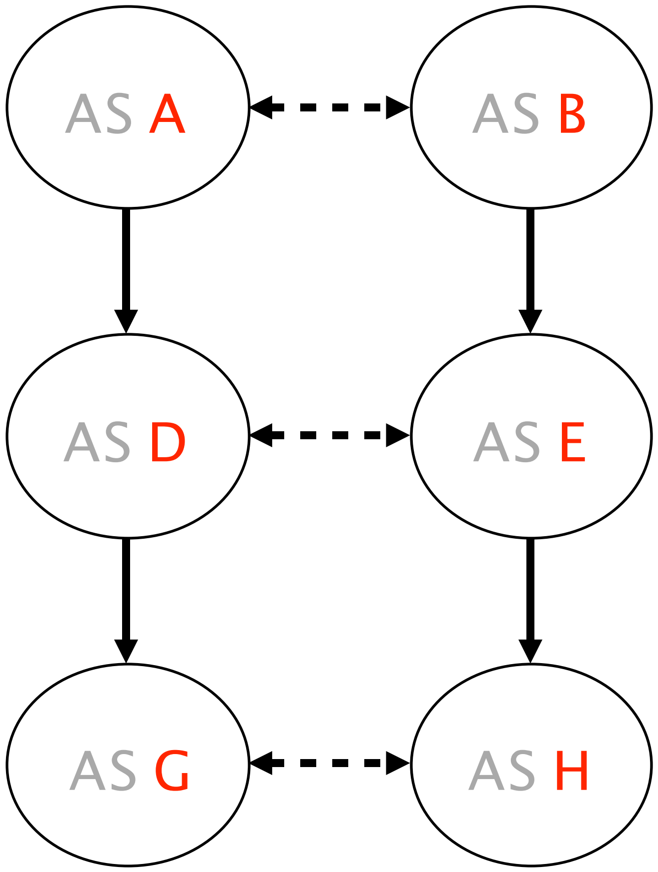
Is (B, A, D) a valid path? Yes/No



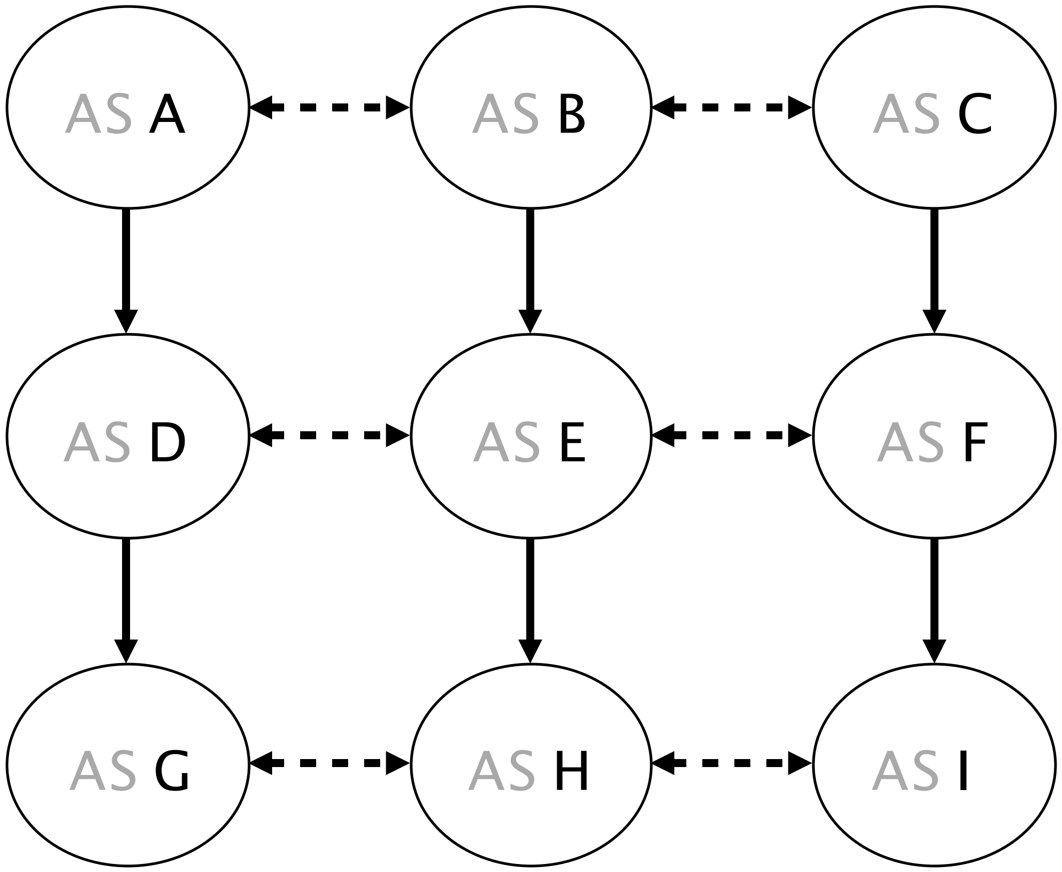
Is (H, E, D) a valid path? Yes/No



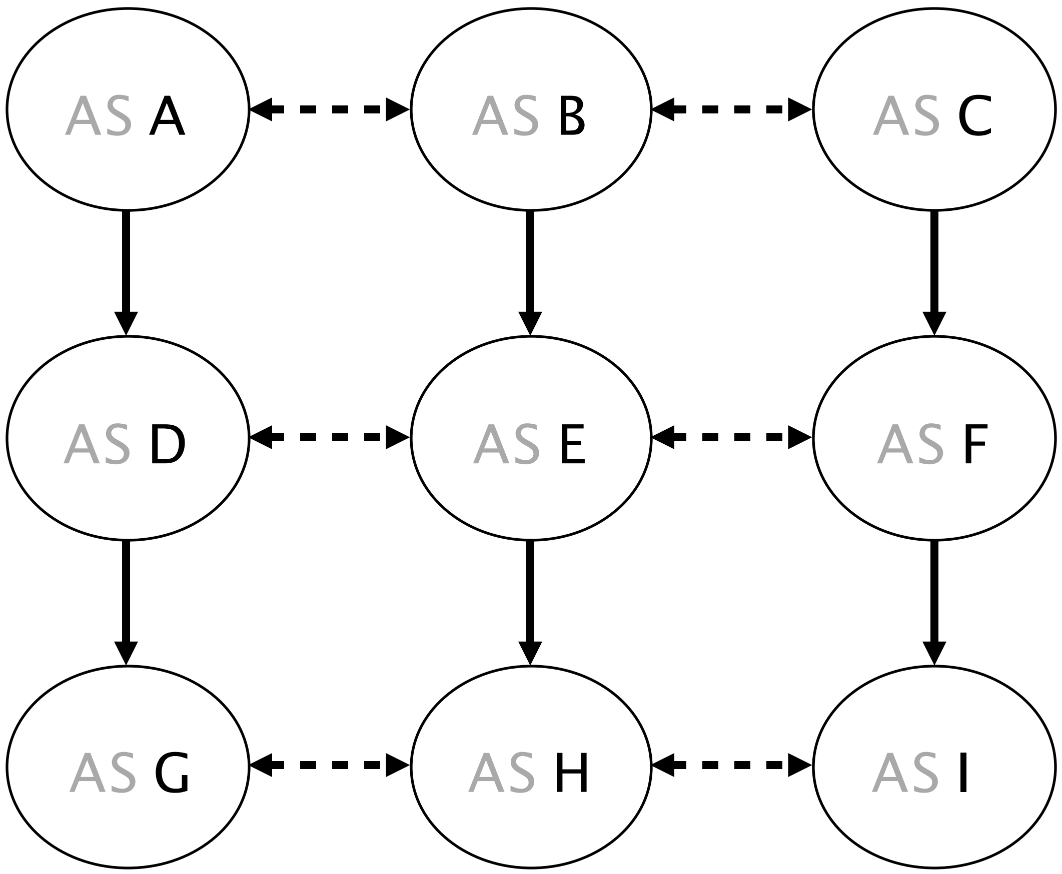
Is (G,D,A,B,E,H) a valid path? Yes/No



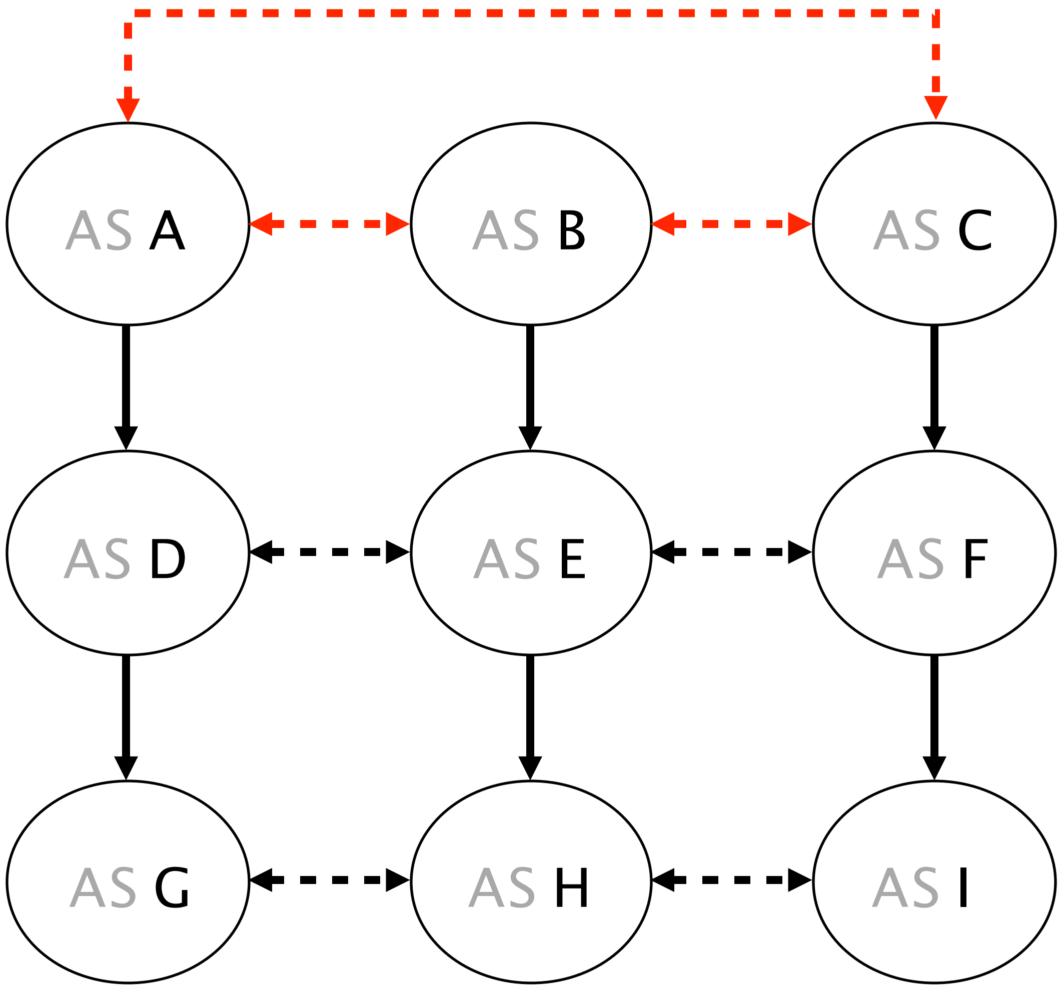
Will (G,D,A,B,E,H) actually see packets? Yes/No



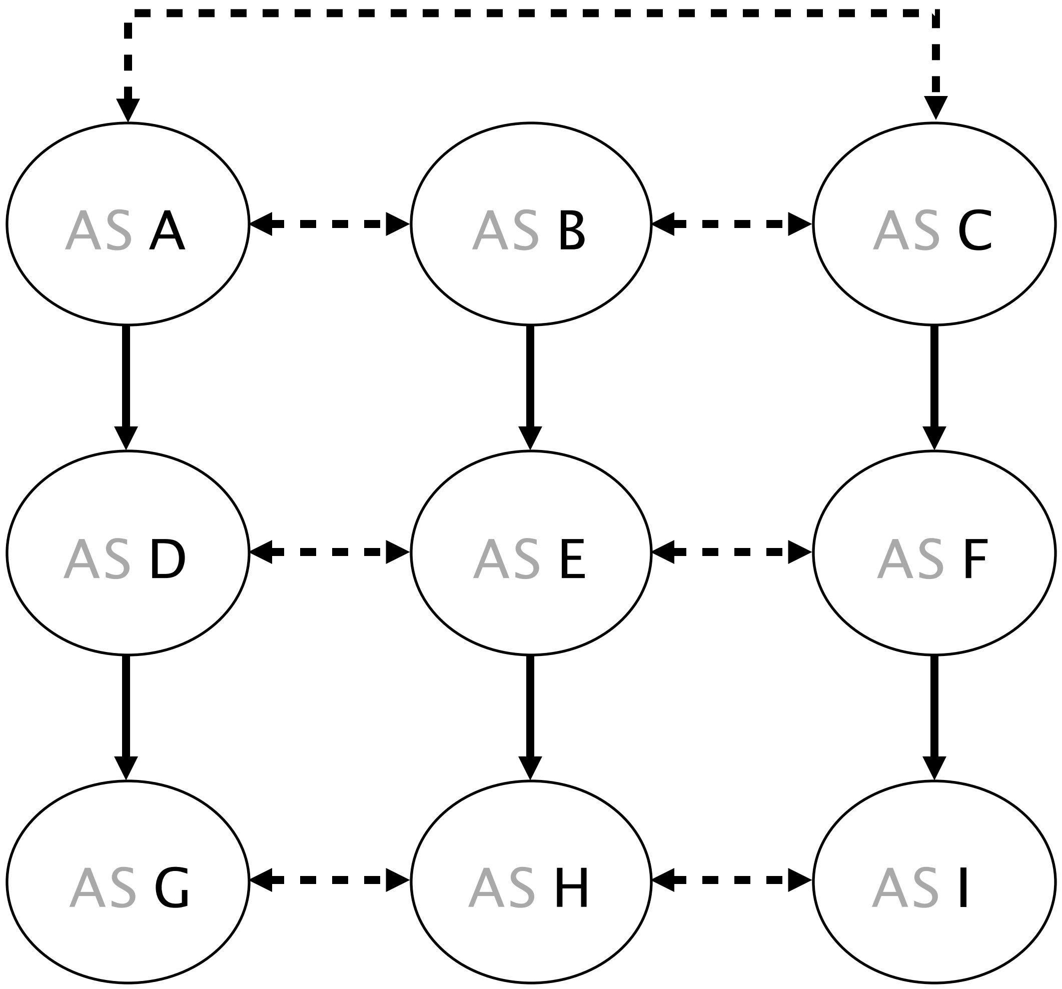
What’s a valid path between G and I?



None! This Internet is partitioned...



Tier-1s must be connected through a full-mesh of peer links



What’s a valid path between G and I?

Border Gateway Protocol

policies and more

BGP Policies

Follow the Money

2 Protocol

How does it work?

Problems

security, performance, …

BGP in a nutshell:

two simple steps, repeated “ad vitam æternam”

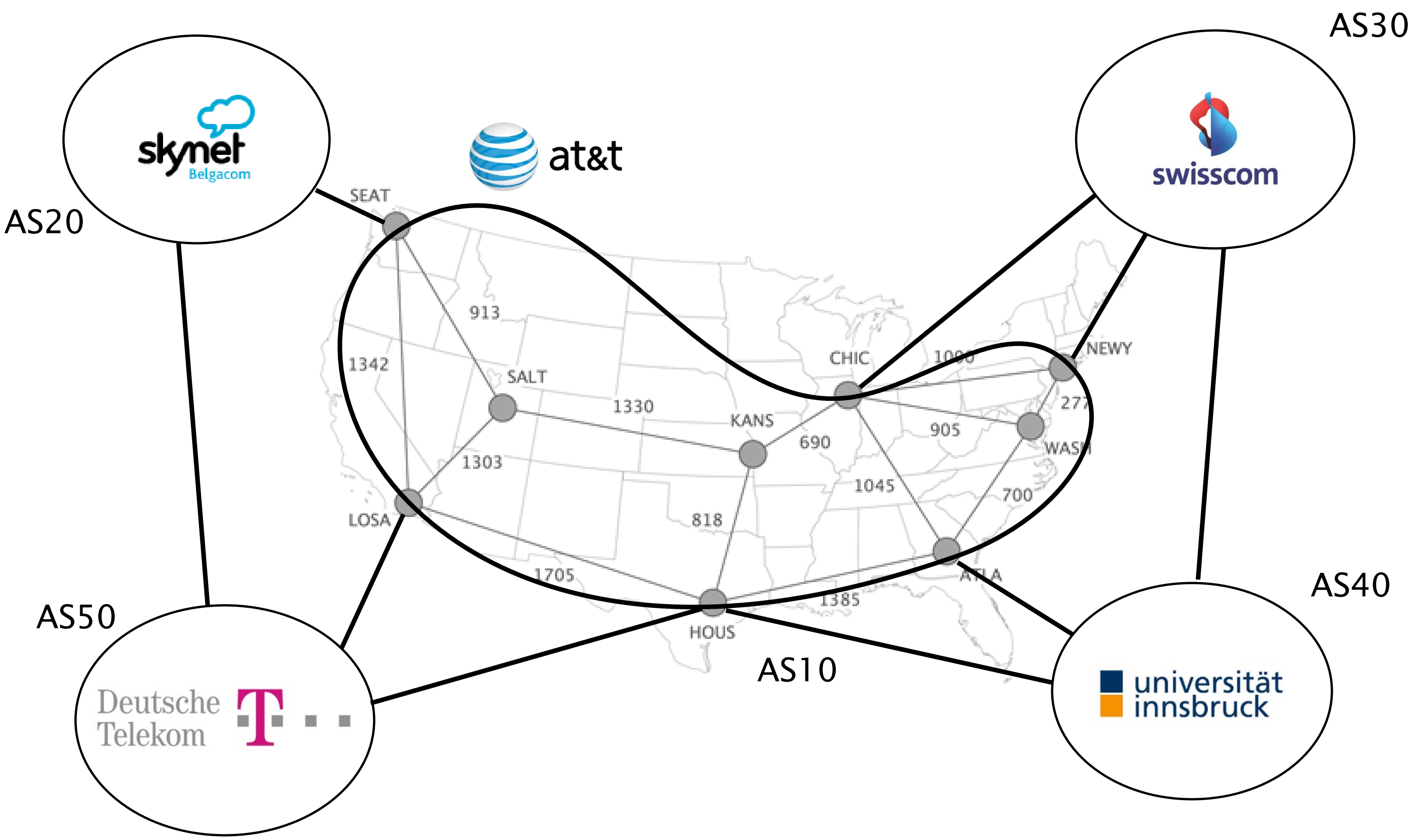
Step 1

Each AS exports
its best route
for each prefix it knows

Step 2

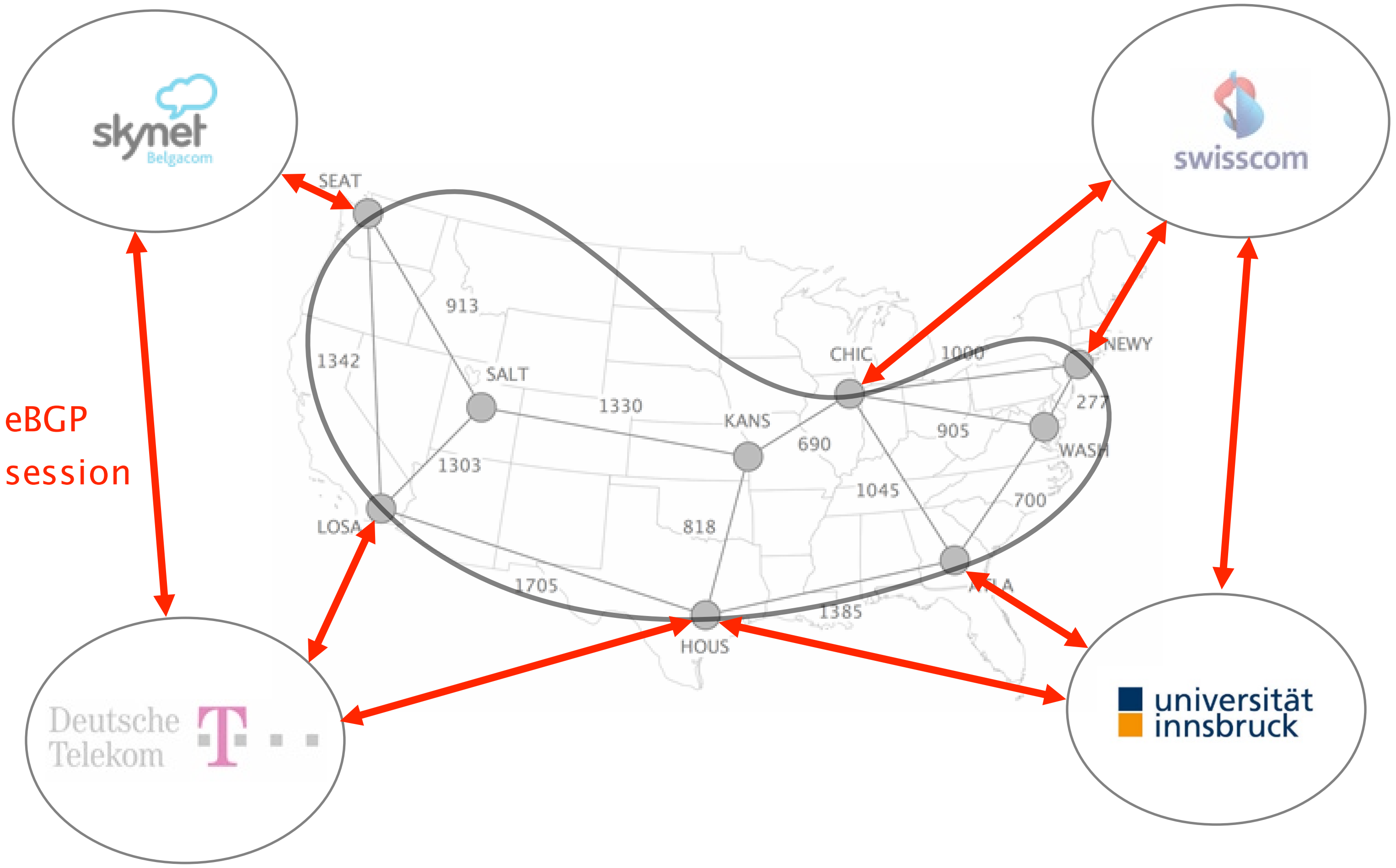
Each AS selects
the best routes it hears
for each prefix

BGP sessions come in two flavors

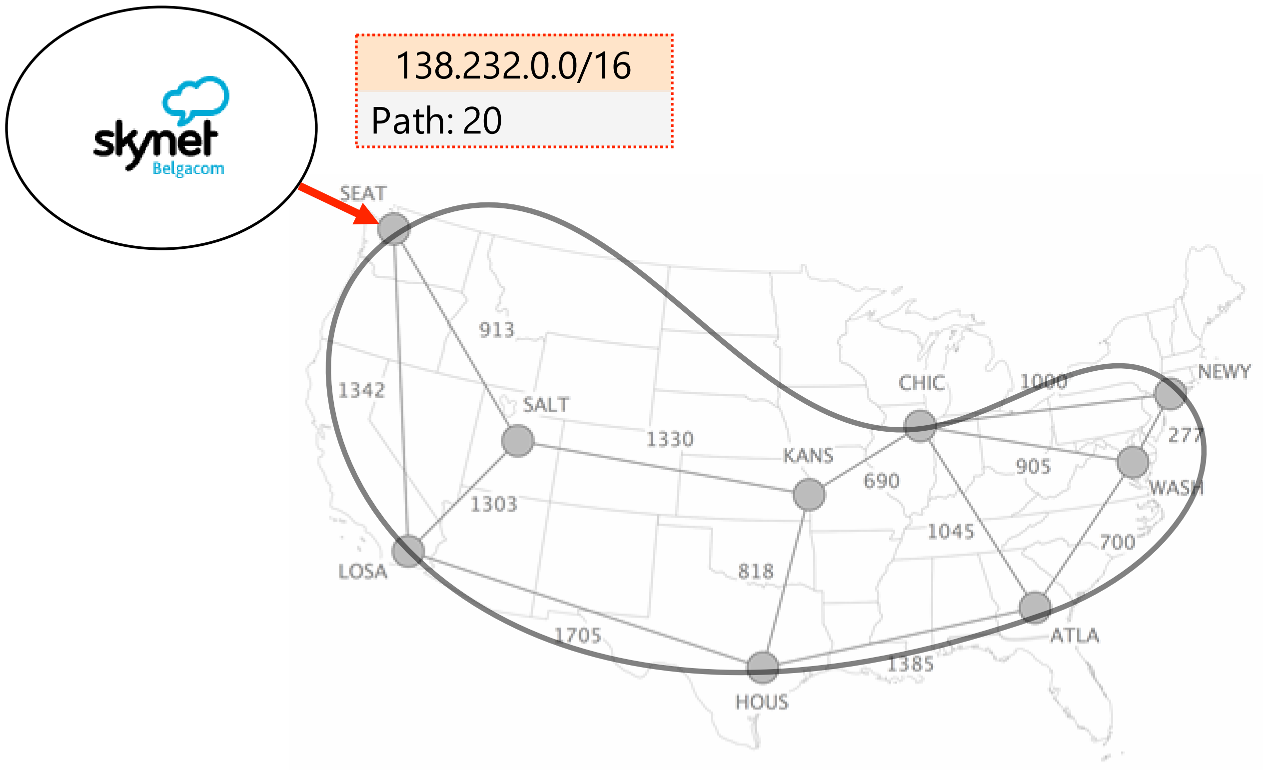


external BGP (eBGP) sessions

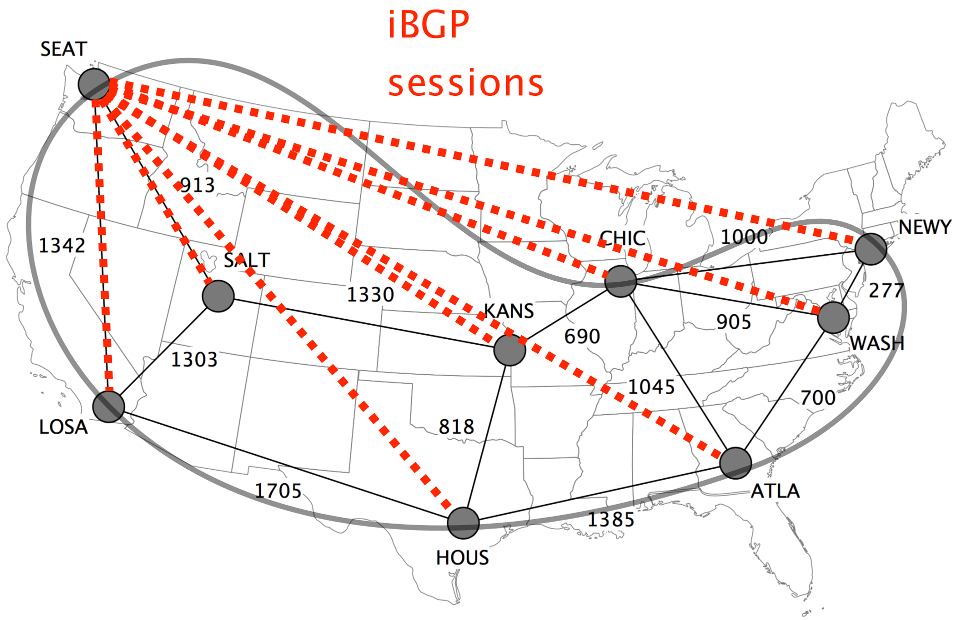
connect border routers in different ASes



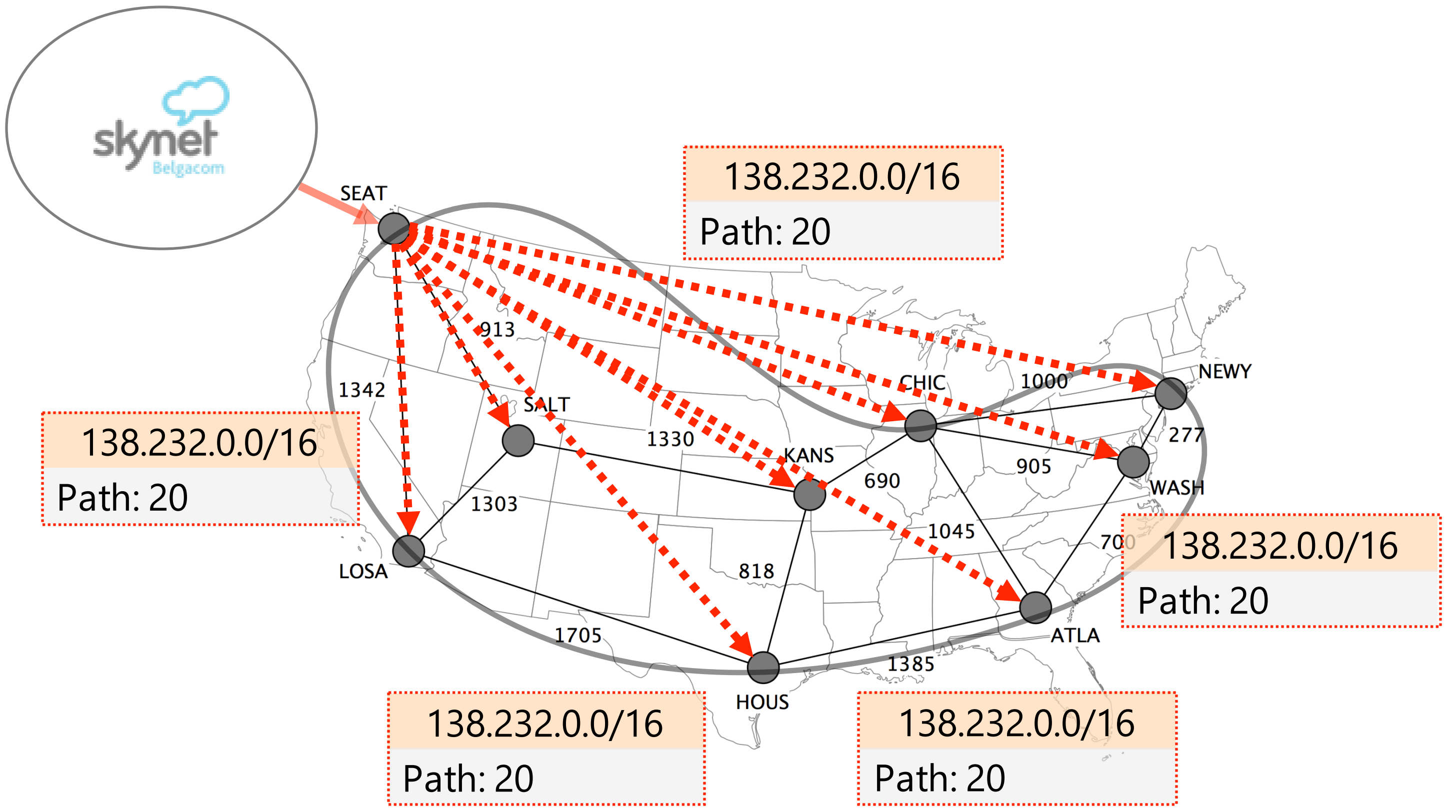
eBGP sessions are used to learn routes to external destinations

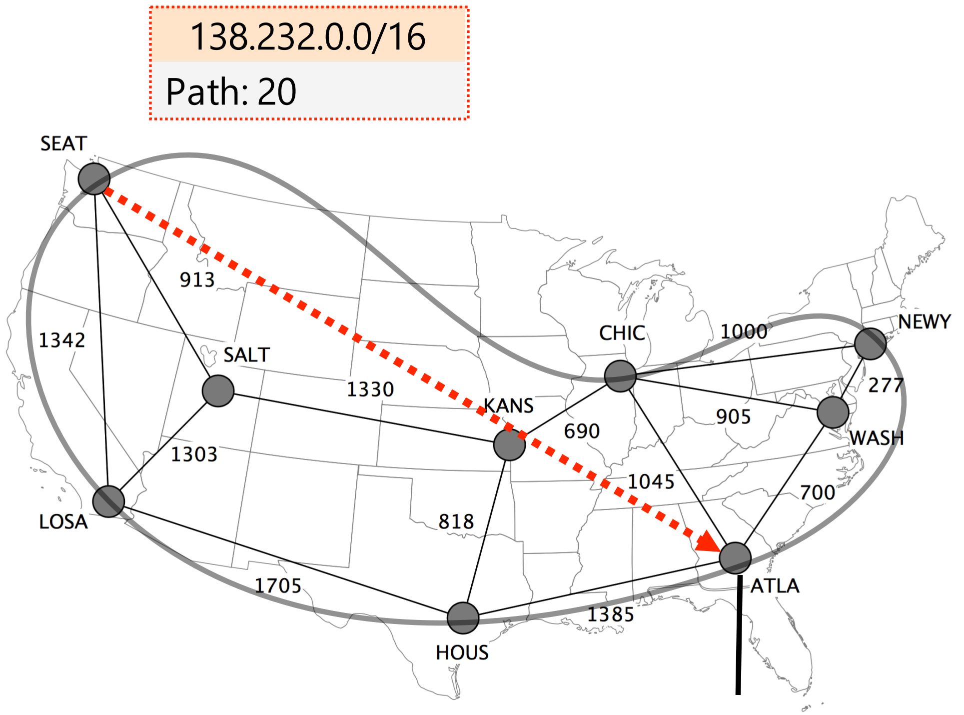


internal BGP (iBGP) sessions connect the routers in the same AS



iBGP sessions are used to disseminate externally-learned routes internally

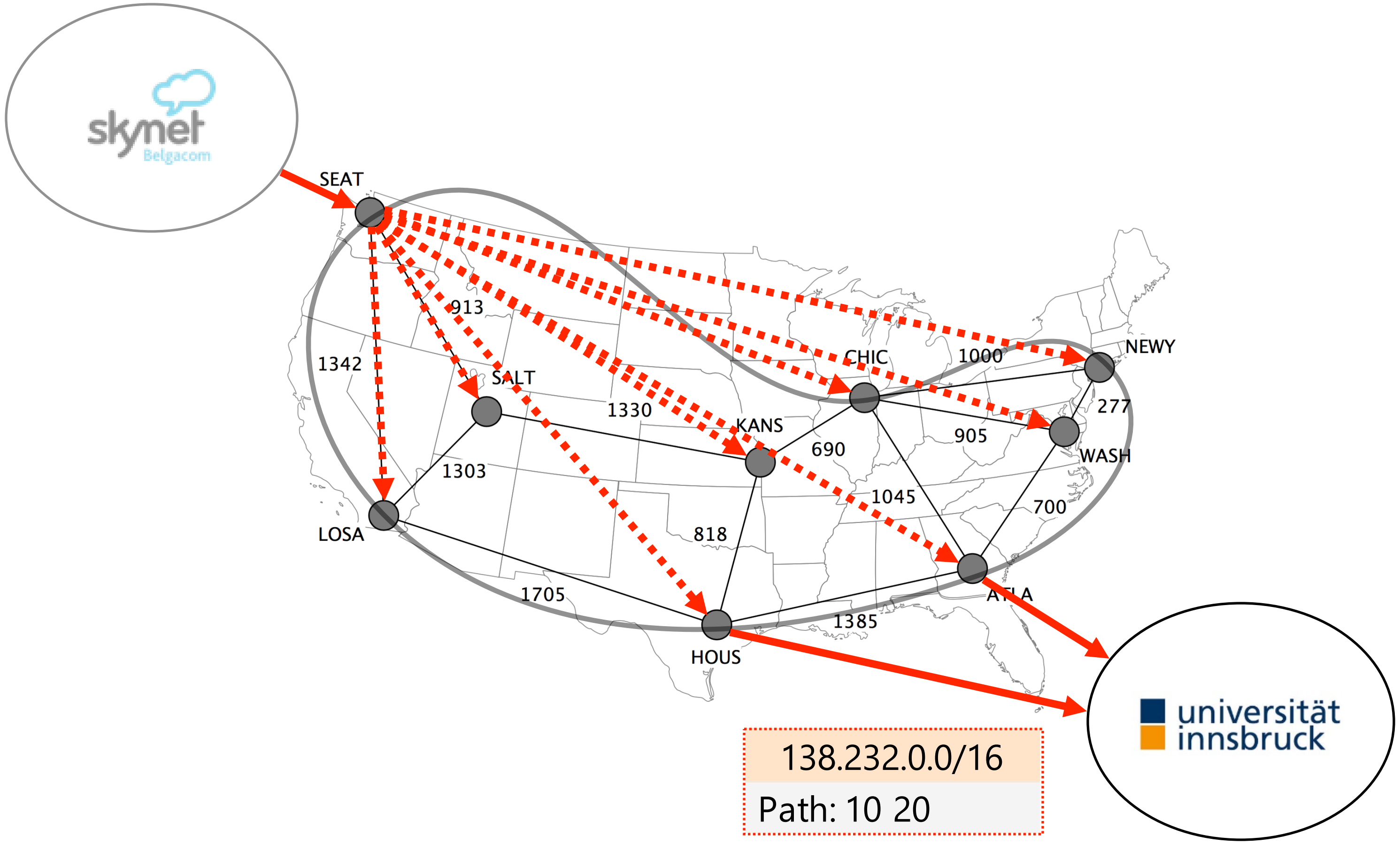




I can reach “138.232/16” via SEAT,
internal next-hop is CHIC

learned via iBGP (*e.g.*, OSPF)

Routes disseminated internally are then announced externally again, using eBGP sessions



On the wire, BGP is a rather simple protocol composed of four basic messages

type	used to...
OPEN	establish TCP-based BGP sessions
NOTIFICATION	report unusual conditions
UPDATE	inform neighbor of a new best route a change in the best route the removal of the best route
KEEPALIVE	inform neighbor that the connection is alive

UPDATE

inform neighbor of a new best route

a change in the best route

the removal of the best route

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

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)

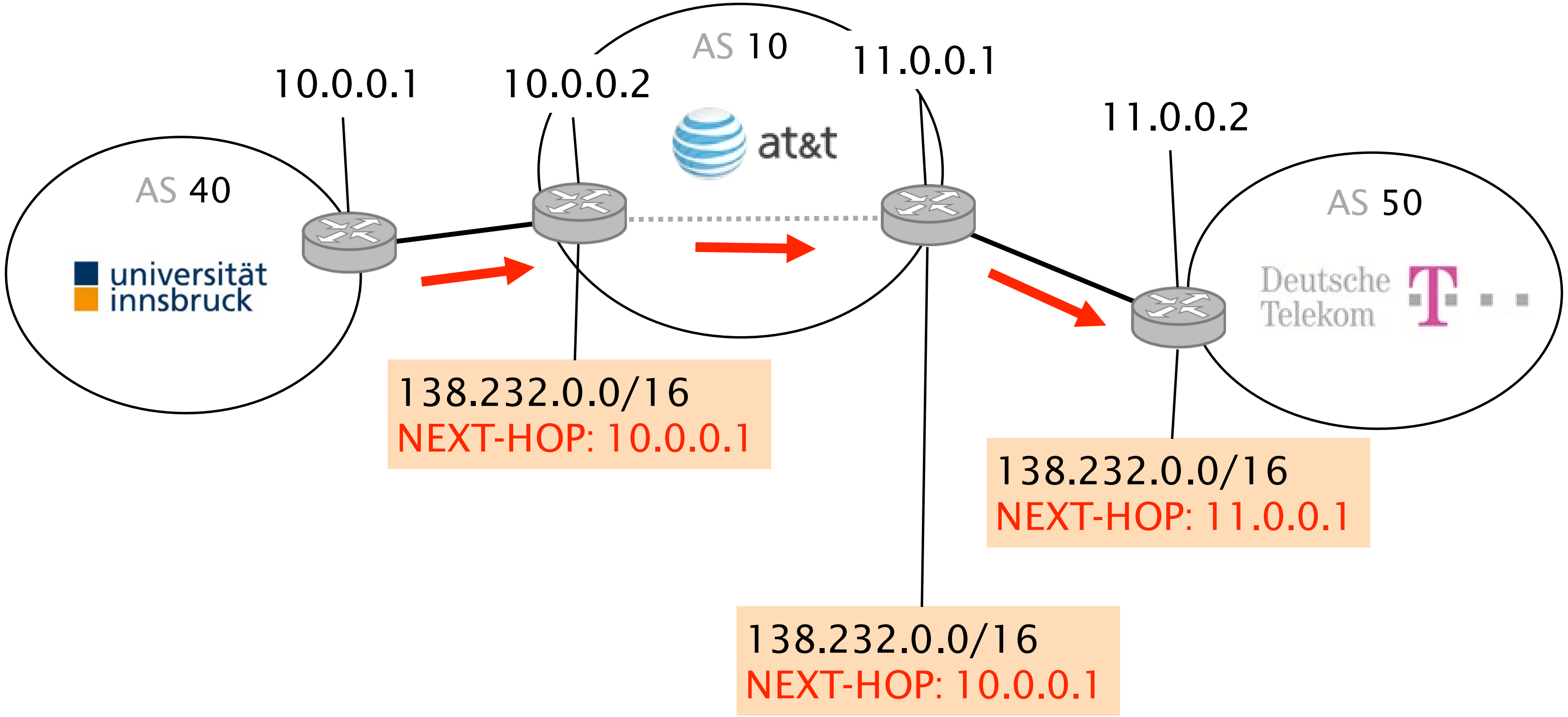
or global (seen on iBGP *and* eBGP)

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

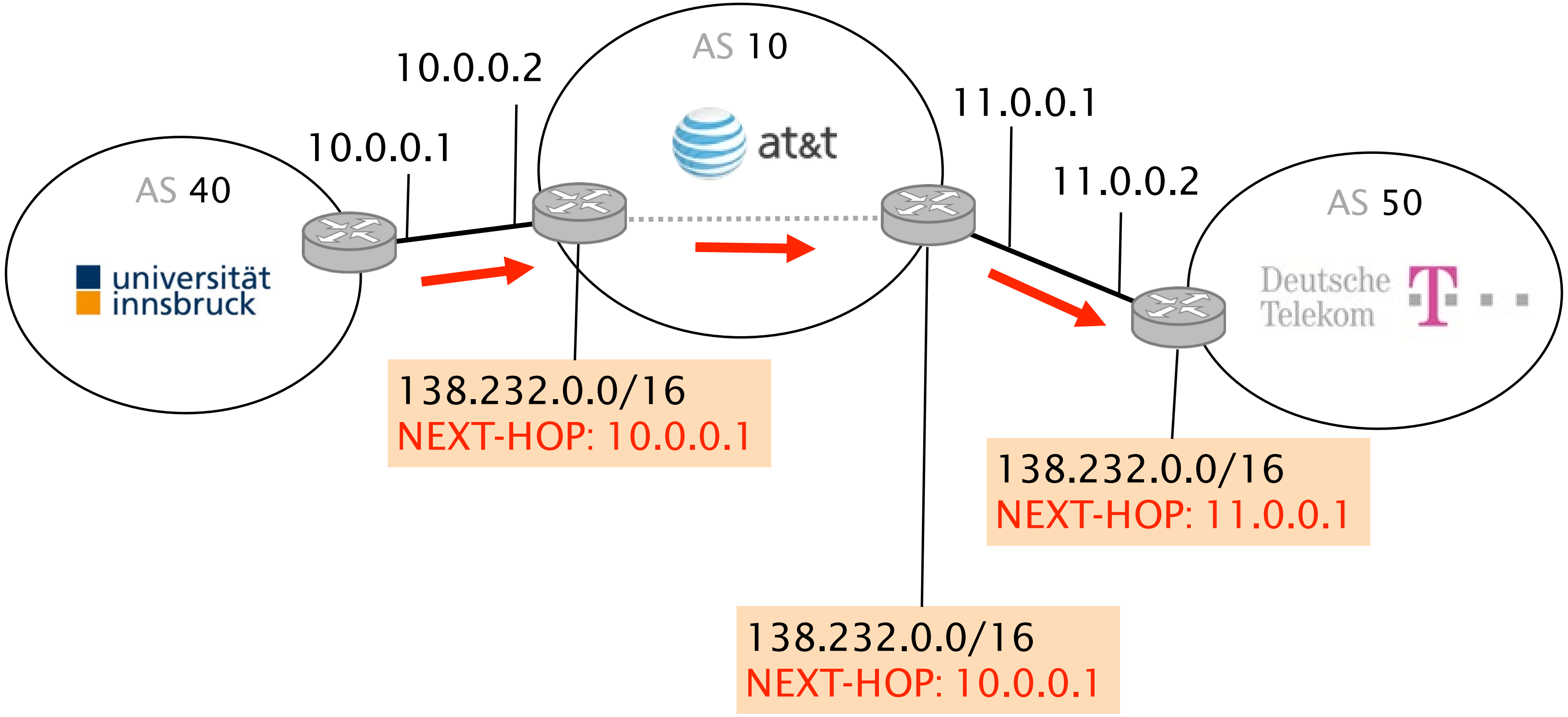
The NEXT-HOP is a global attribute which indicates where to send the traffic next

The NEXT-HOP is set when the route enters an AS,
it does not change within the AS

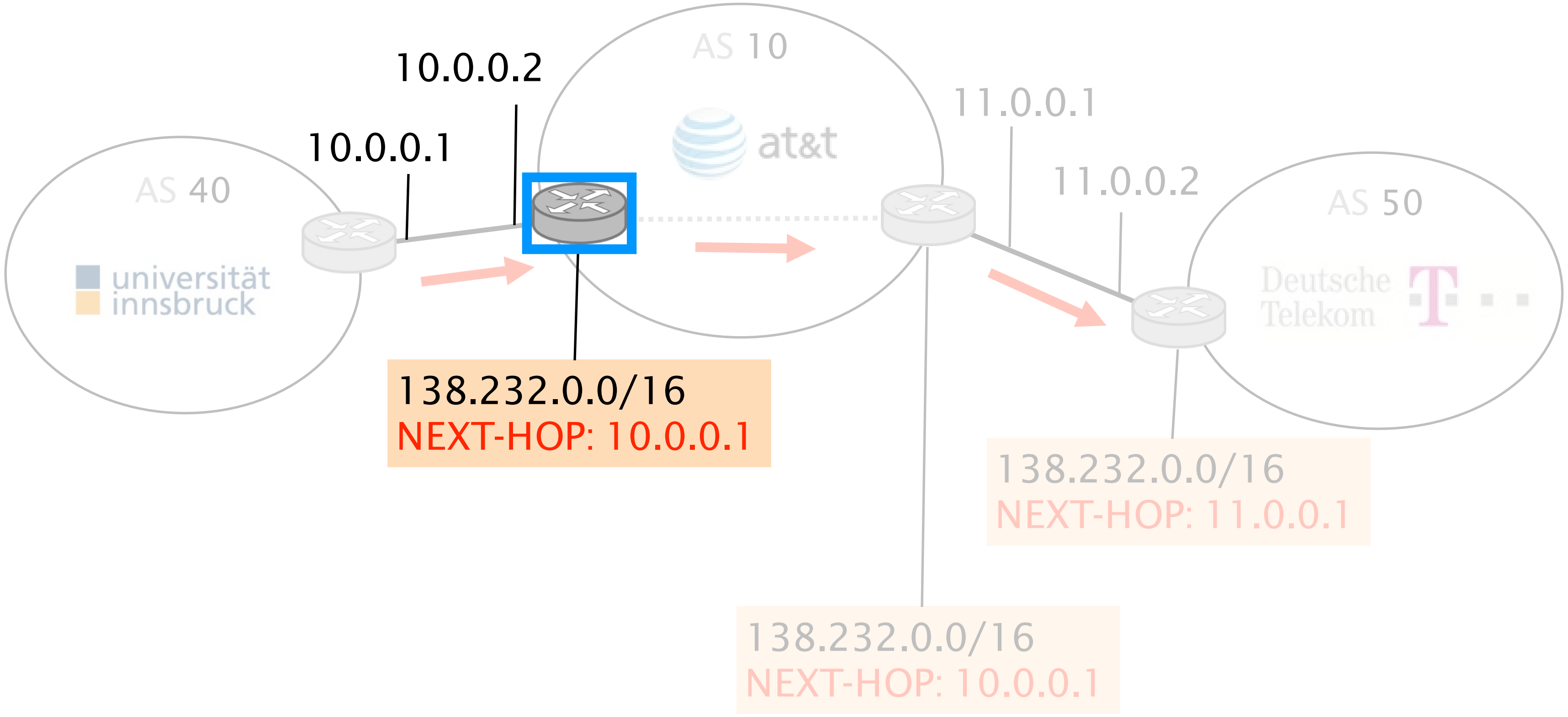
AS 40 has
138.232.0.0/16



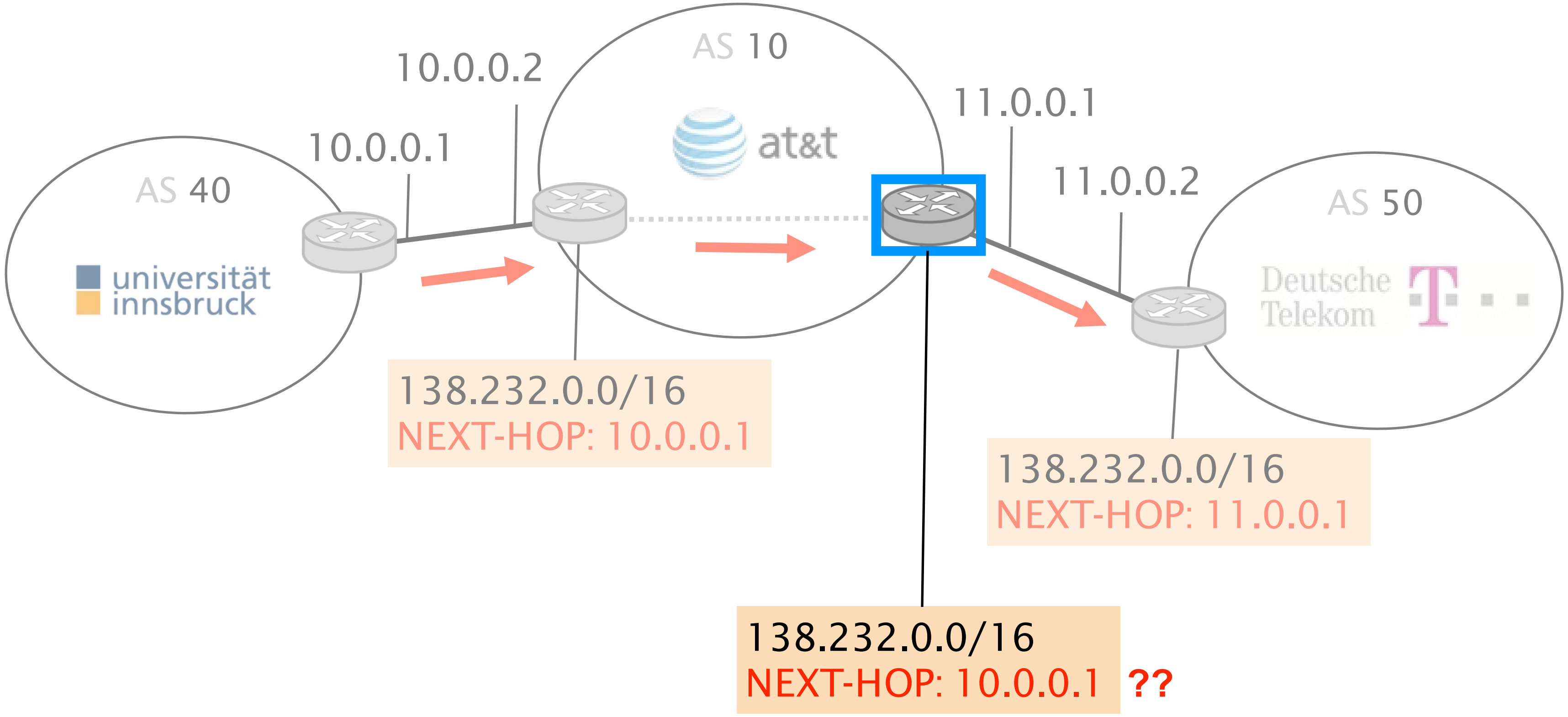
For externally-learned routes, this means that the NEXT-HOP is the IP address of the neighbor's eBGP router, here **10.0.0.1**



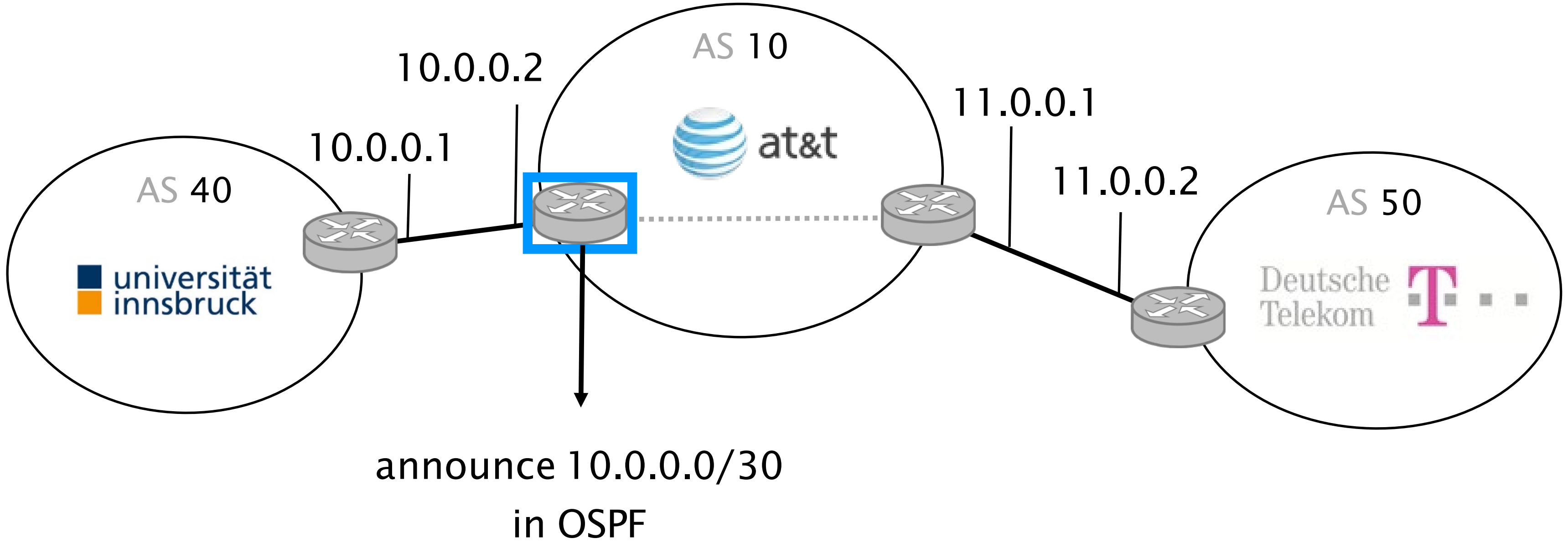
For this router, reaching 10.0.0.1 is not a problem as it is directly connected to the corresponding subnet (10.0.0.0/30)



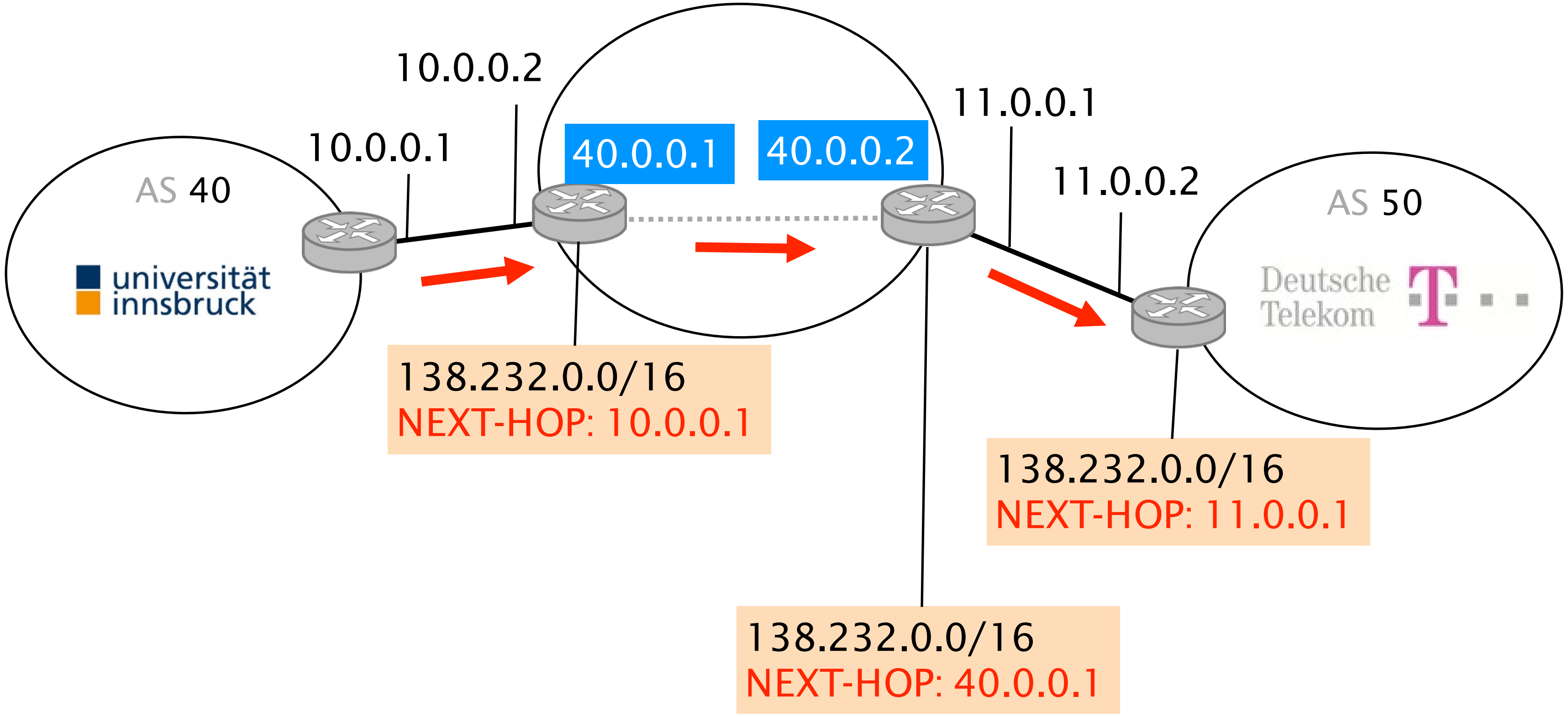
That router is *not* directly to the NEXT-HOP's subnet (10.0.0.0/30)
and does not know how to reach it,
it will therefore drop the BGP route…



One solution is for the external router to redistribute the prefixes attached to the external interfaces into the IGP

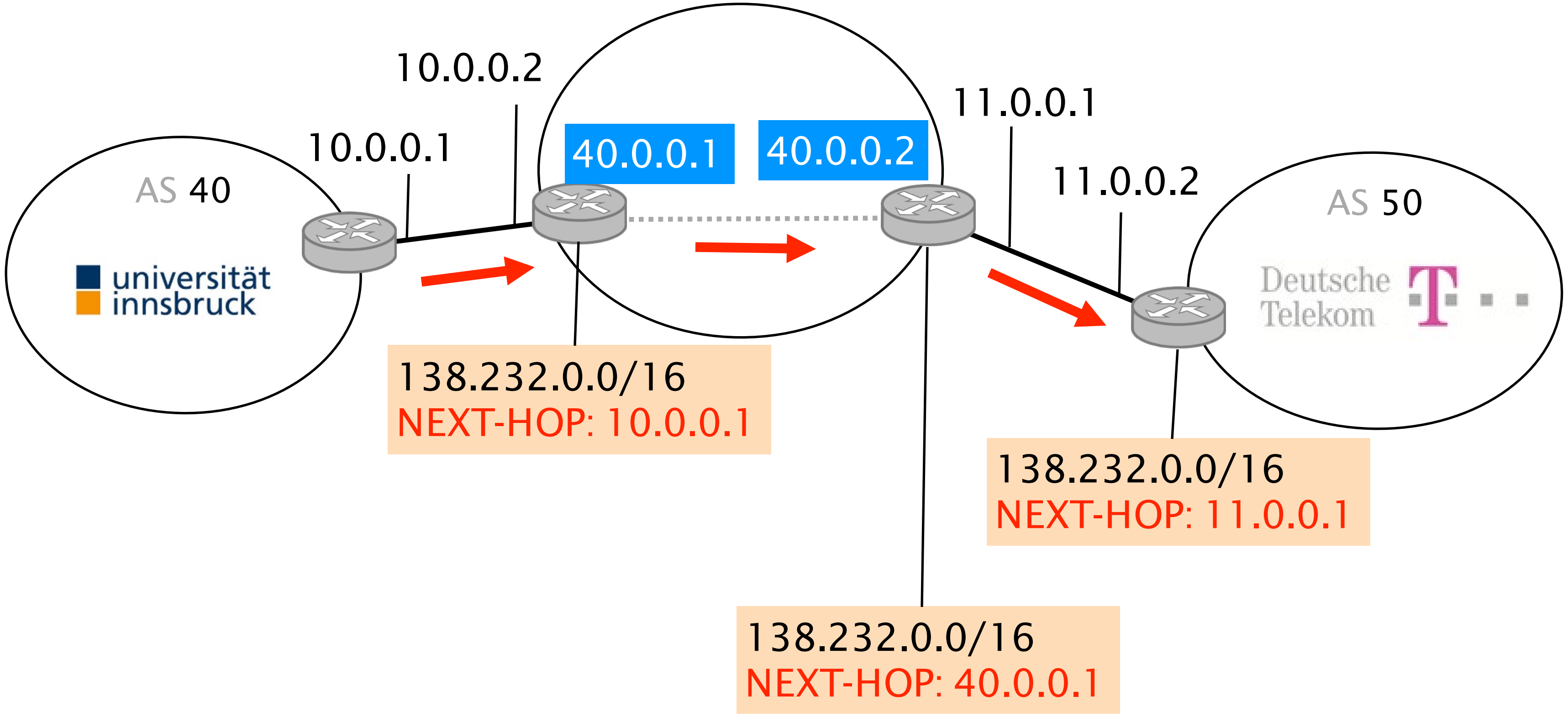


Another solution is for the border router to rewrite the NEXT-HOP before sending it over iBGP, usually to its loopback address

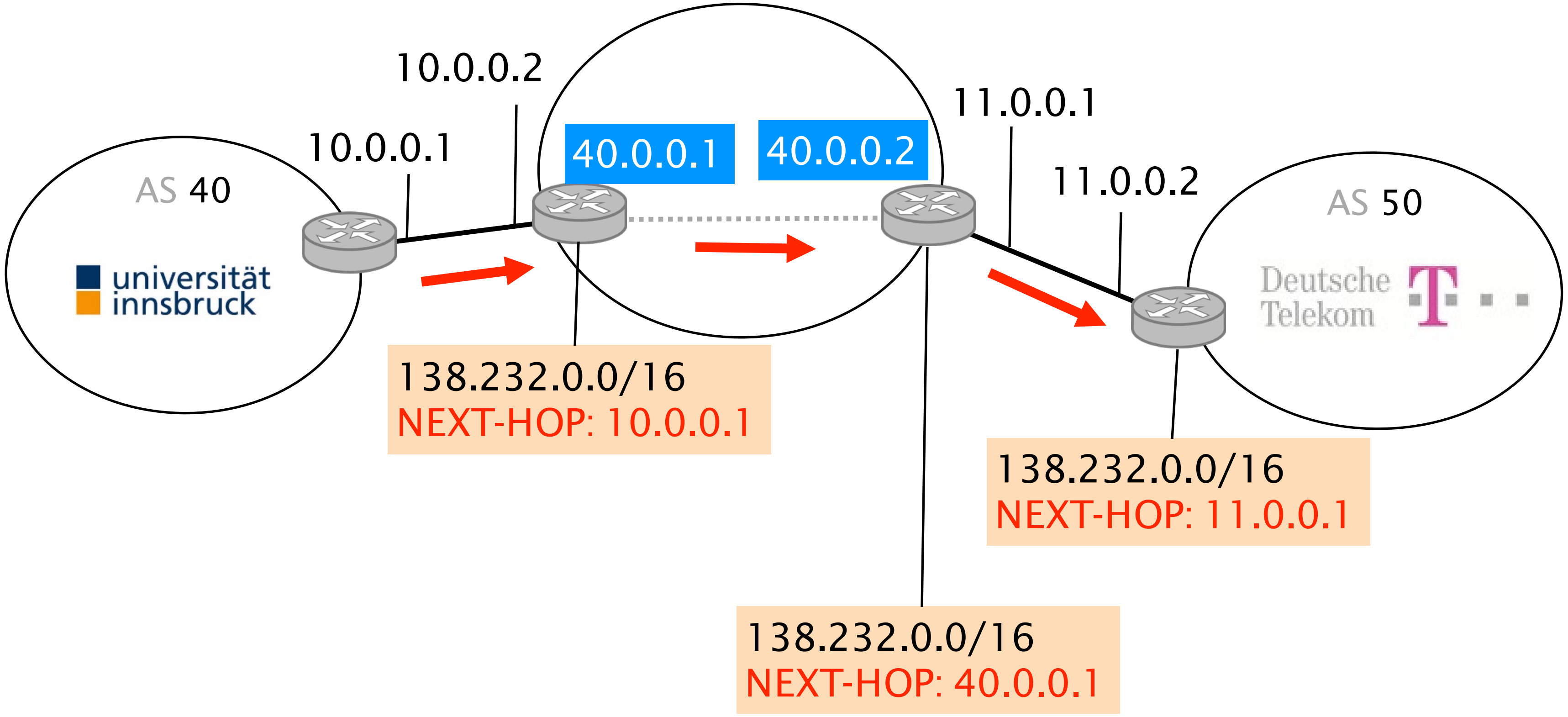


Of course, loopback addresses need to be reachable network-wide.

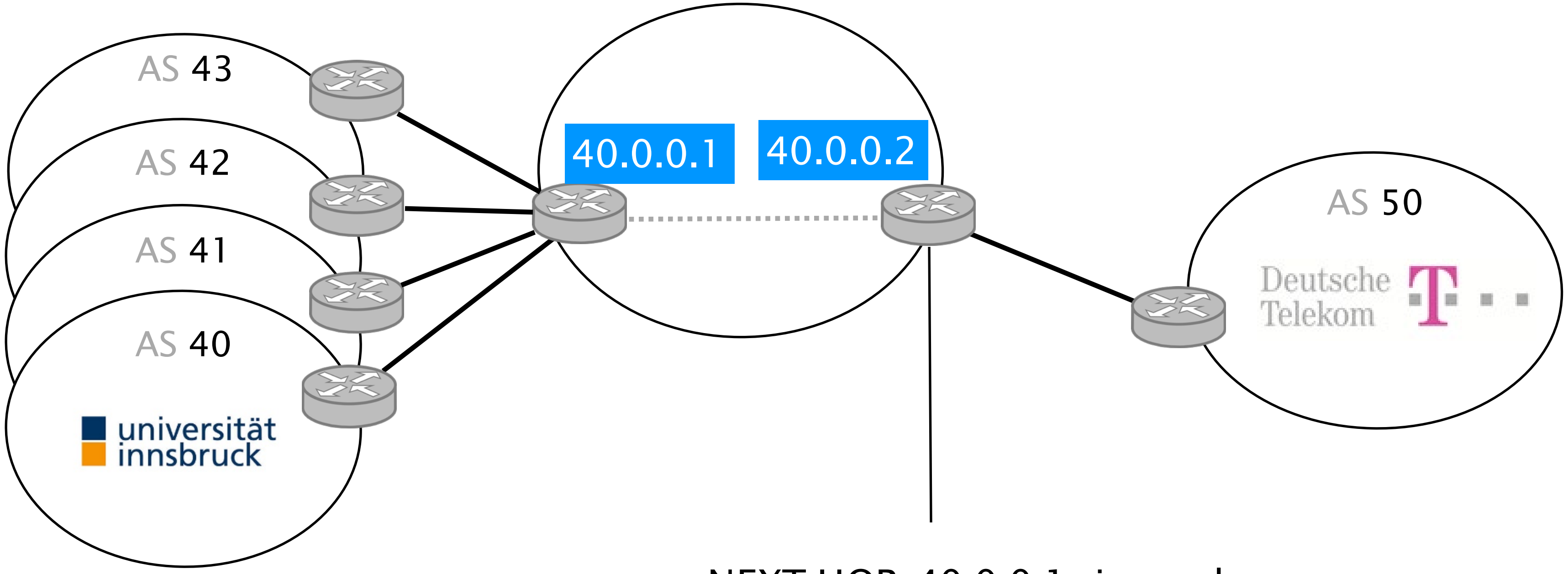
Typically, each router advertise its loopback (as a /32) in the IGP



This is the now-infamous "**next-hop-self policy"**

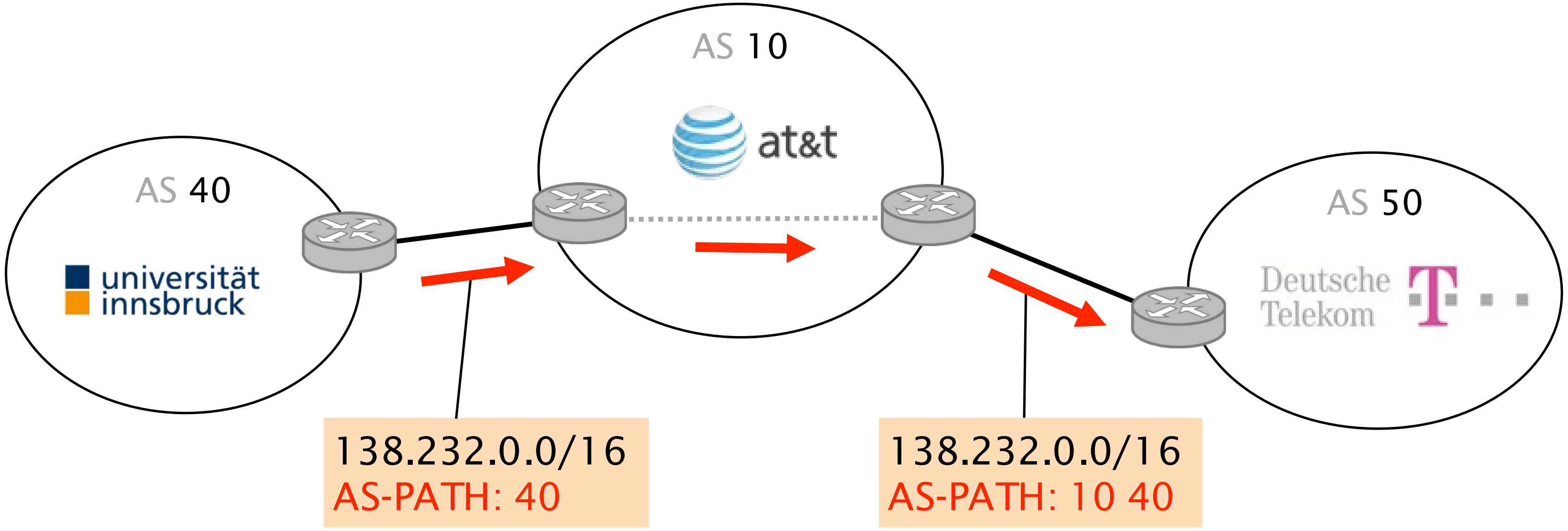


The advantage of next-hop-self is to spare the need to advertise *each* prefix attached to an external link in the IGP



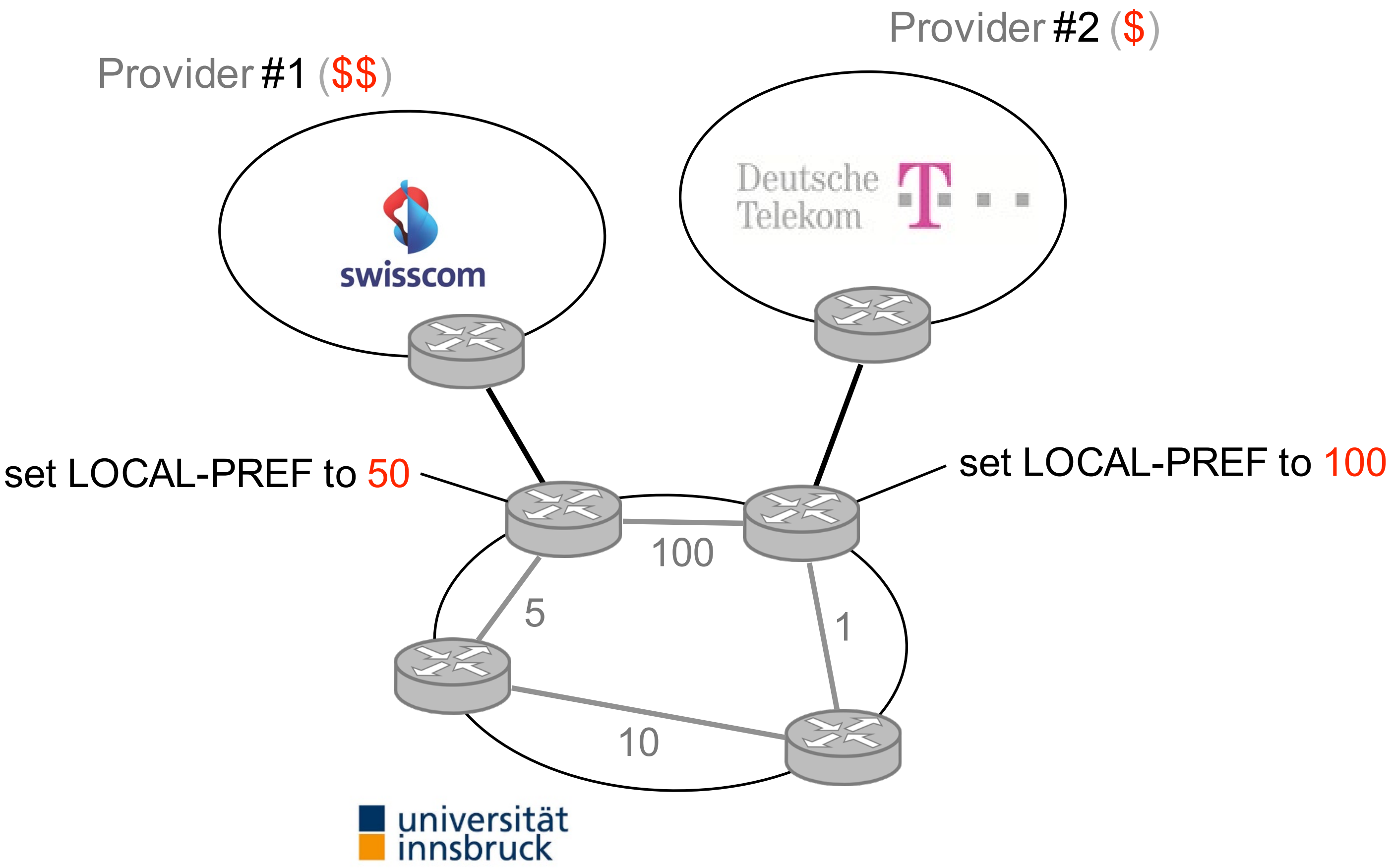
one NEXT-HOP, 40.0.0.1, is used
to reach routes announced by AS 40, 41, 42, 43…

The AS-PATH is a global attribute that lists all the ASes a route has traversed (in reverse order)



AS 40
universität innsbruck
AS 10
at&t
AS 50
Deutsche Telekom
138.232.0.0/16
AS-PATH: 40
138.232.0.0/16
AS-PATH: 10 40

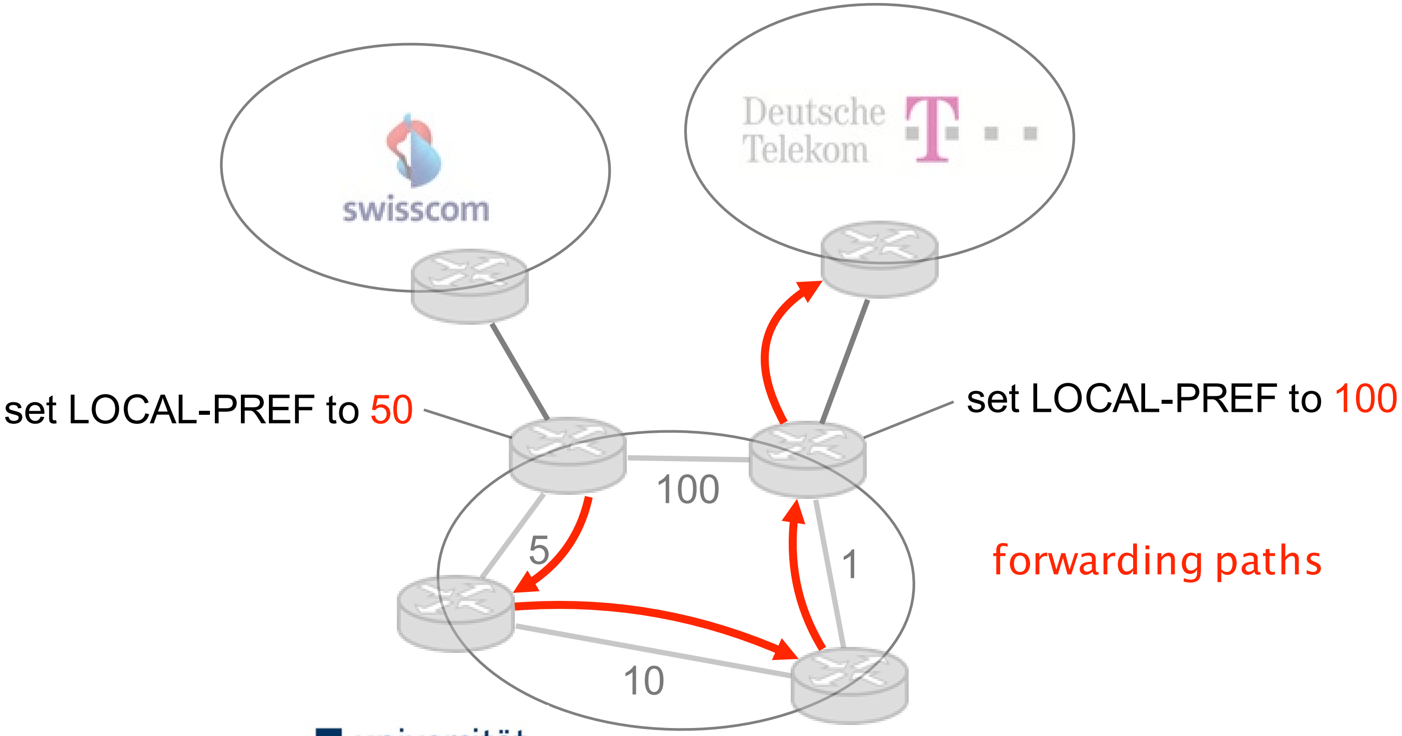
The LOCAL-PREF is a *local* attribute set at the border,
it represents how “preferred” a route is


Provider #1 ($$)
swisscom
Provider #2 ($)
Deutsche Telekom
set LOCAL-PREF to 50
set LOCAL-PREF to 100
100
5
1
10
universität innsbruck

By setting a higher LOCAL-PREF,

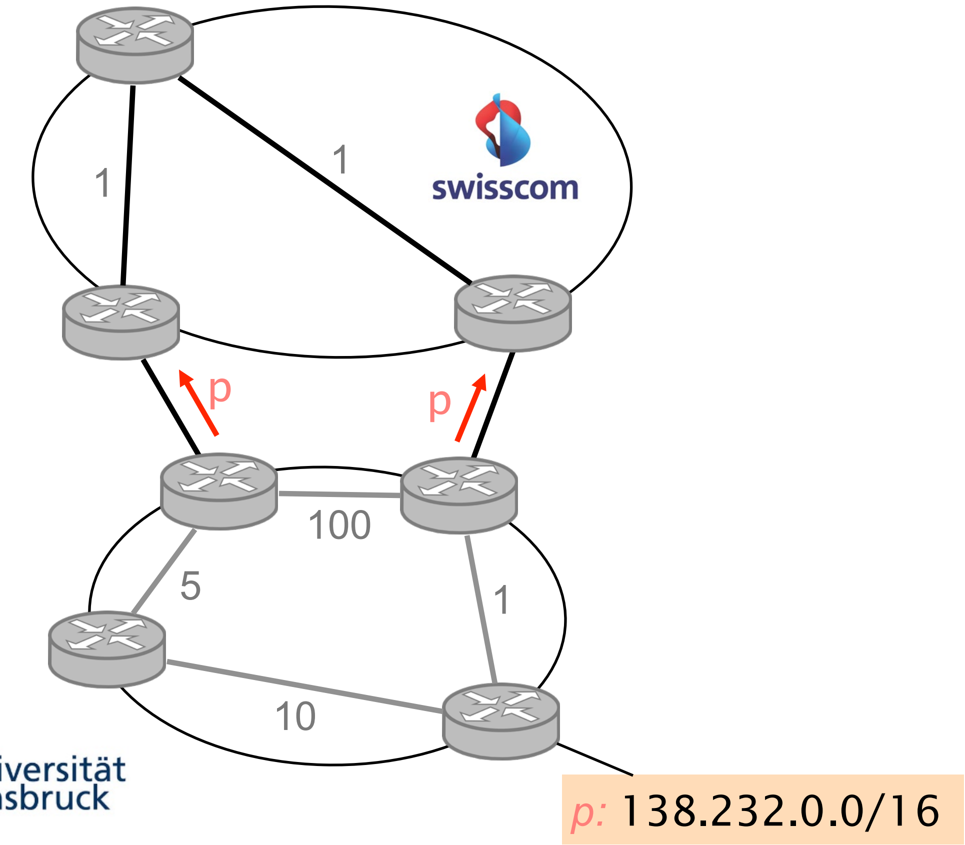
all routers end up using DT to reach any external prefixes,

even if they are closer (IGP-wise) to the Swisscom egress



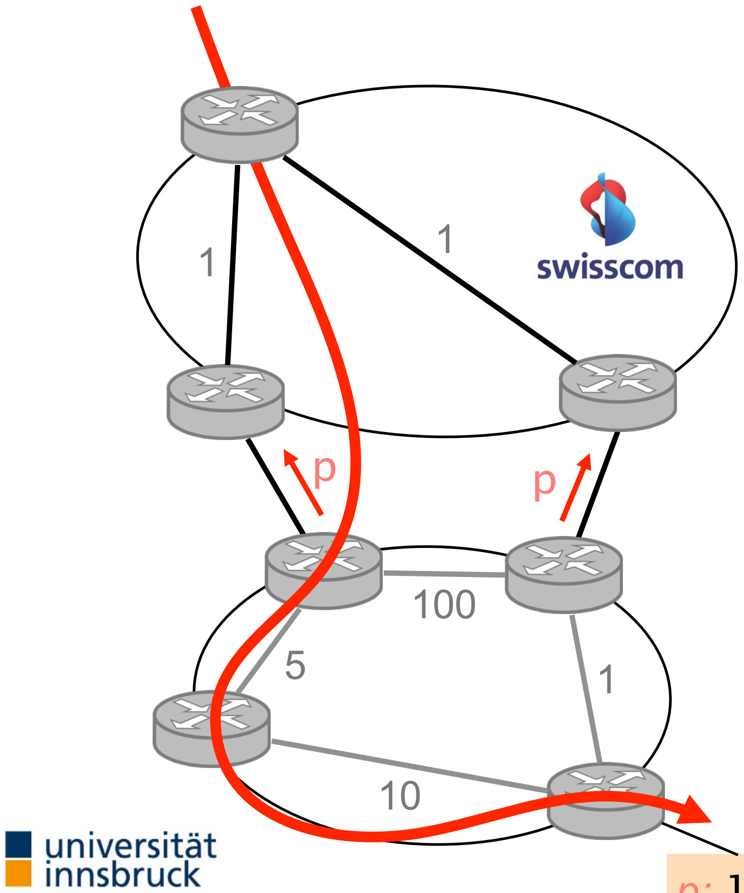
The MED is a *global* attribute which encodes
the relative "proximity" of a prefix wrt to the announcer

Swisscom receives two routes to reach *p*



universität innsbruck

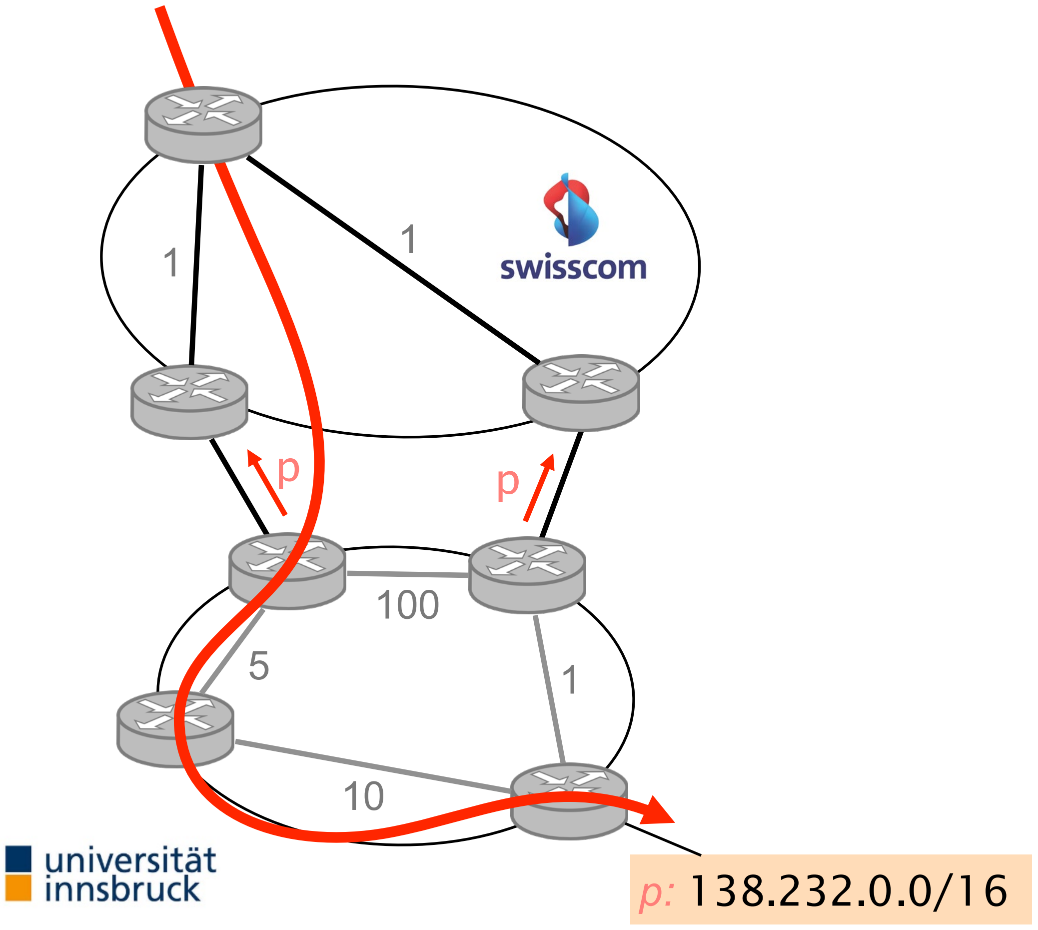
Swisscom receives two routes to reach p
and chooses (arbitrarily) its left router as egress



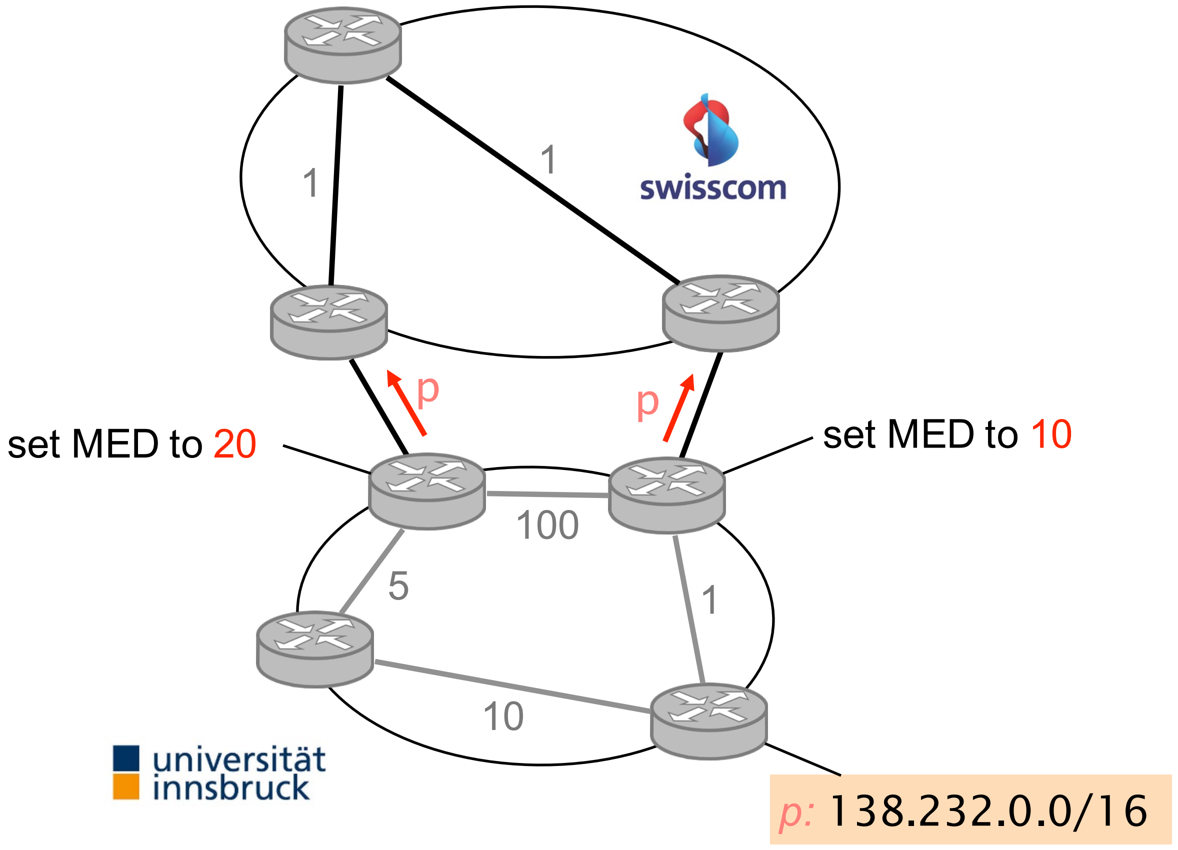
p: 138.232.0.0/16

Yet, UIBK would prefer to receive traffic for *p*

on its right border router which is closer to the actual destination



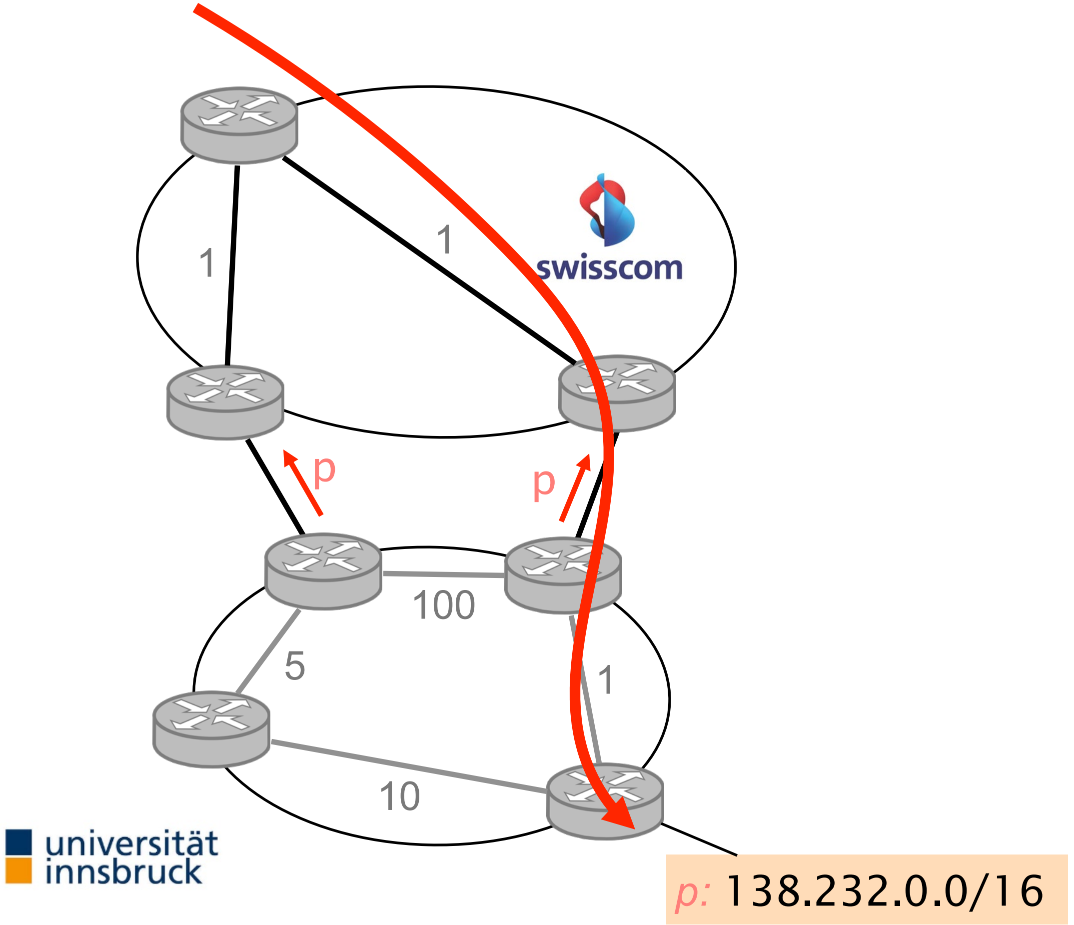
UIBK can communicate that preferences to Swisscom by setting a higher MED on *p* when announced from the left



Swisscom receives two routes to reach *p*

and, *given it does not cost it anything more*,

chooses its right router as egress



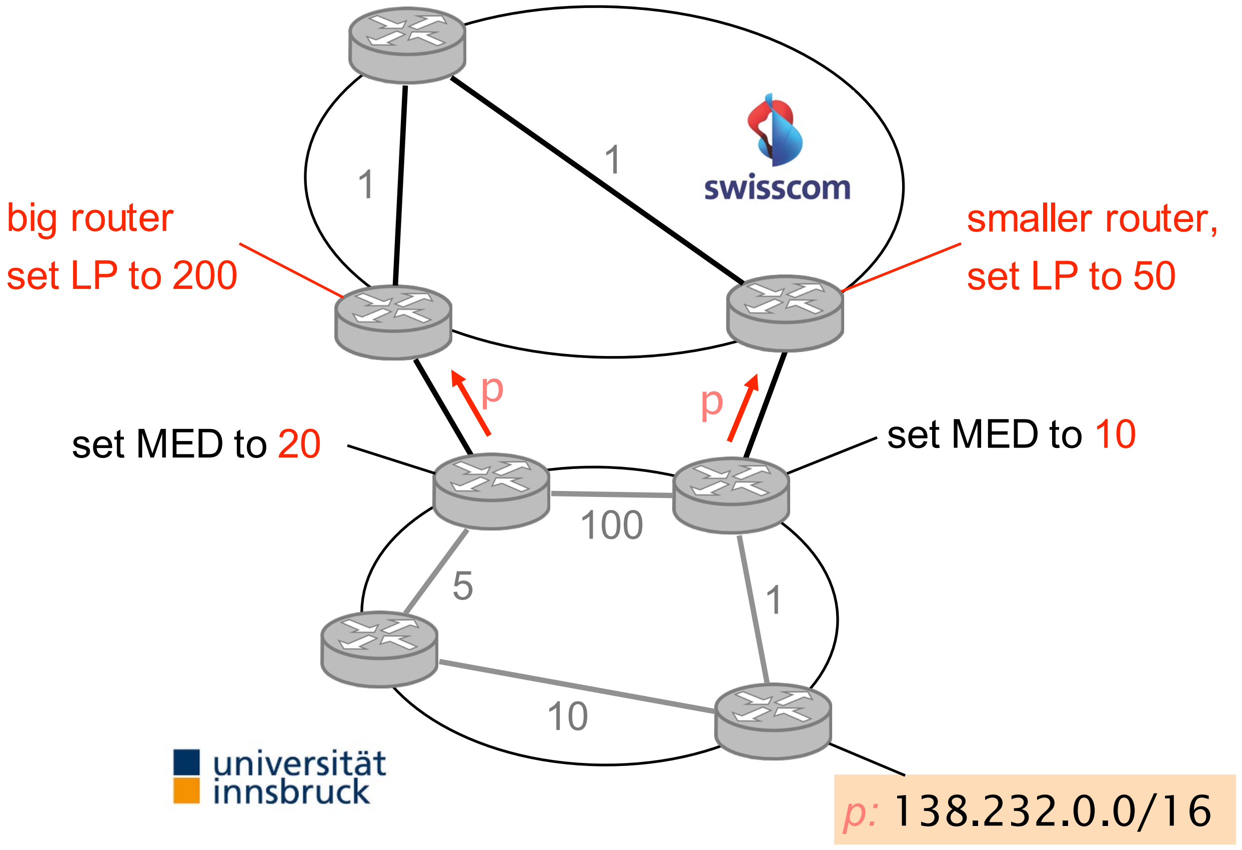
Swisscom receives two routes to reach p

and, *given it does not cost it anything more*,

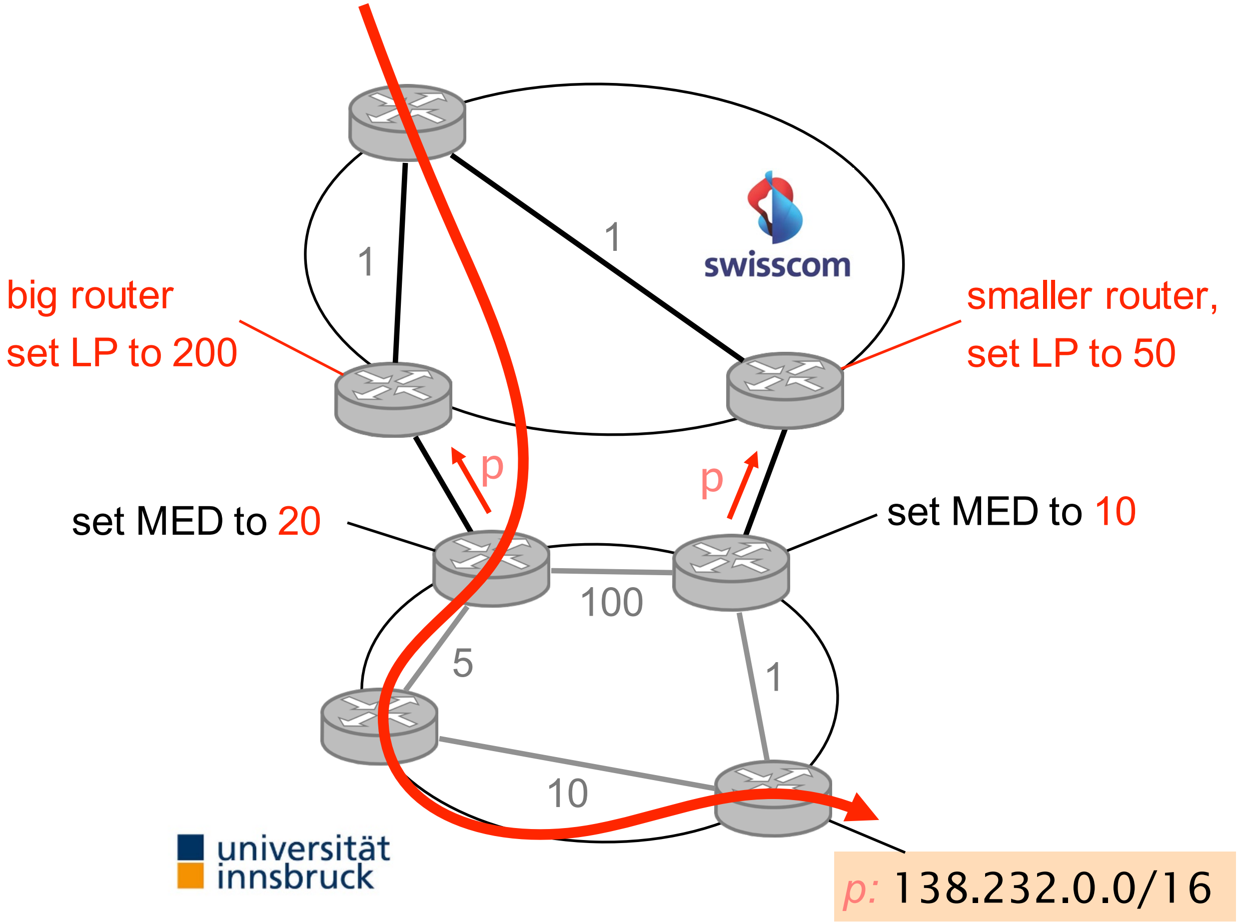
chooses its right router as egress

But what if it does?

Consider that Swisscom always prefer to send traffic via its left egress point (bigger router, less costly)



In this case, Swisscom will not care about the MED value and still push the traffic via its left router



universität innsbruck

Lesson

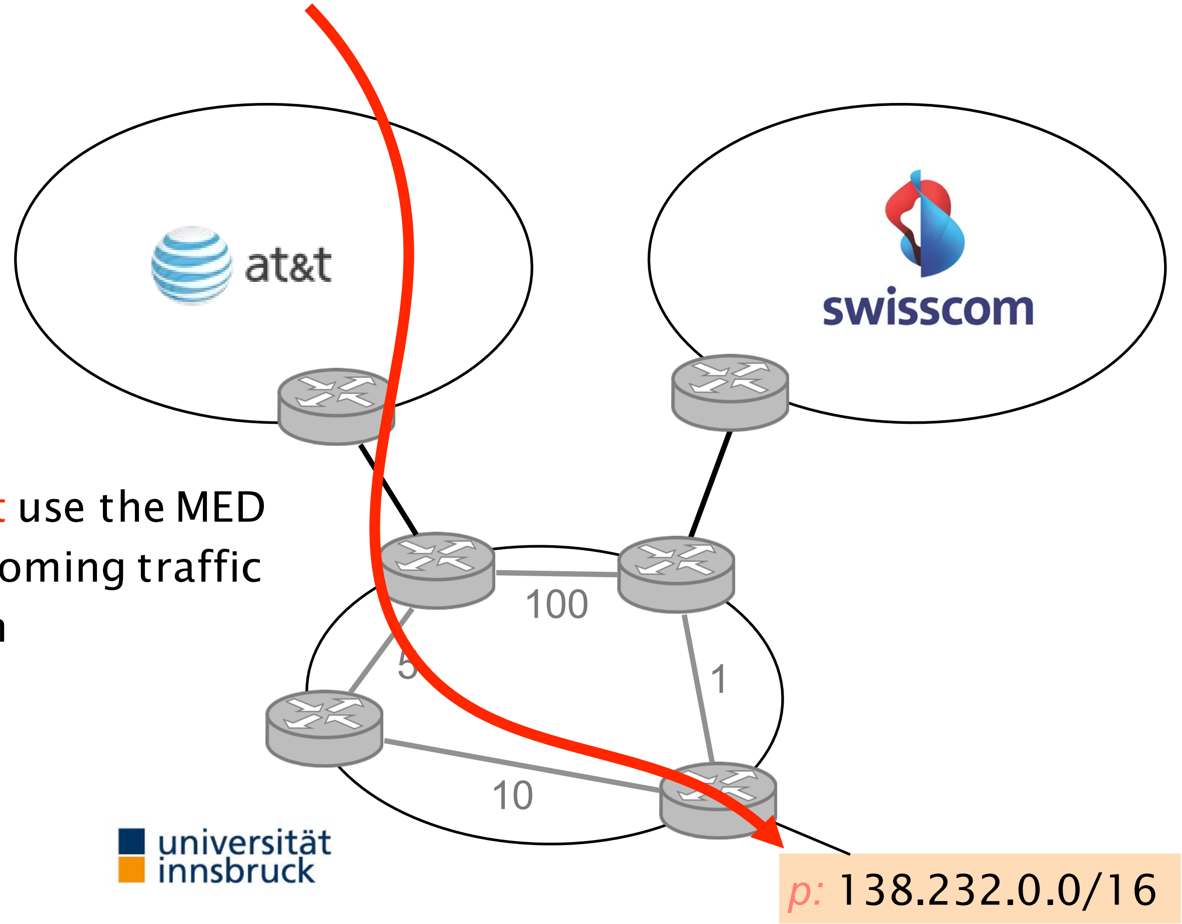
The network which is sending the traffic always has the final word when it comes to deciding where to forward

Corollary

The network which is receiving the traffic can just influence remote decision, not control them

With the MED, an AS can influence its inbound traffic between multiple connection towards the same AS

UIBK cannot use the MED to move incoming traffic to Swisscom



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

IP prefix

Attributes

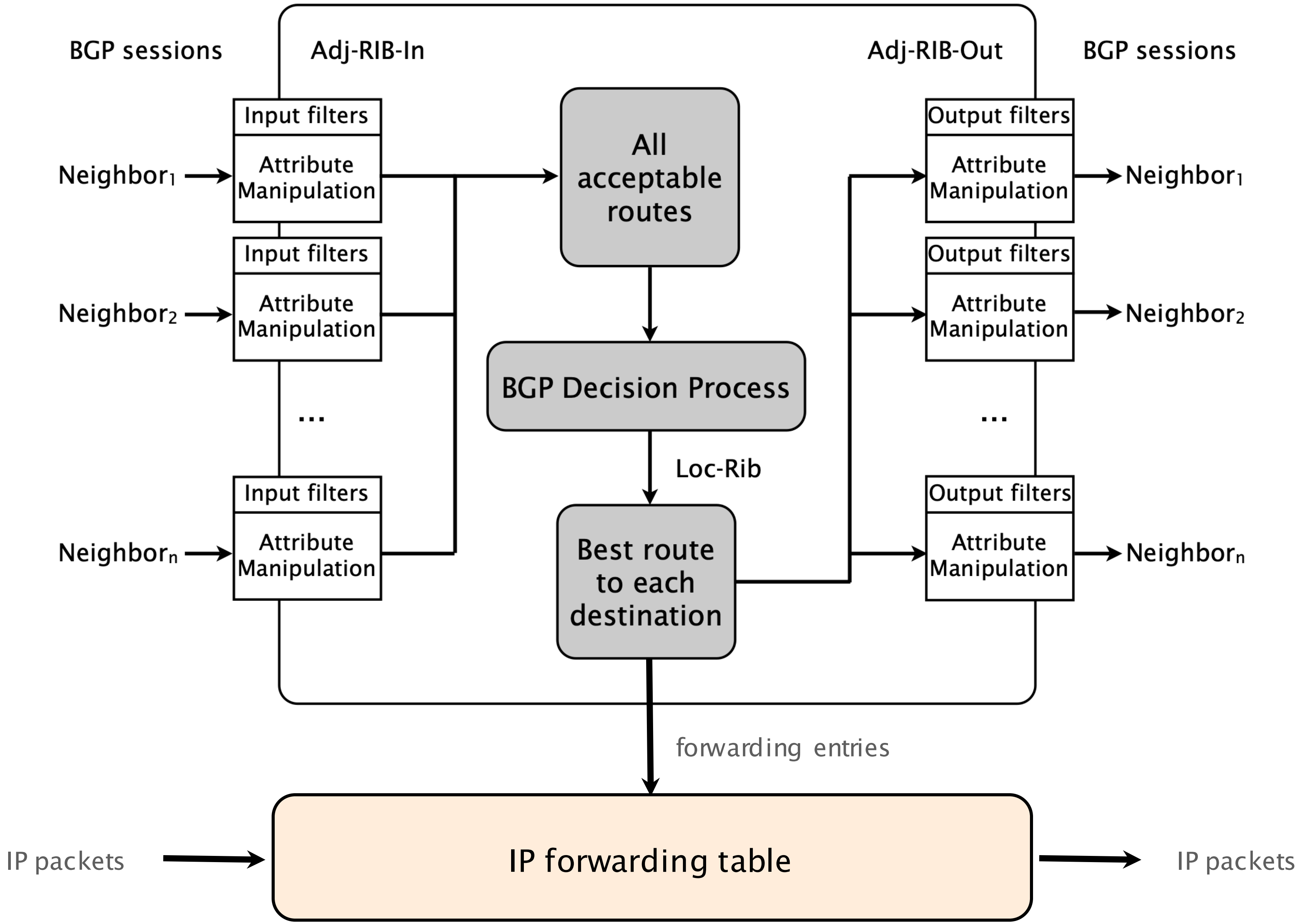
Describe route properties

used in route selection/exportation decisions

are either local (*only* seen on iBGP)

or global (seen on iBGP *and* eBGP)

Each BGP router processes UPDATEs according to a precise pipeline


BGP sessions
Adj-RIB-In
Adj-RIB-Out
BGP sessions
Input filters
Attribute Manipulation
Neighbor1
Input filters
Attribute Manipulation
Neighbor2
...
Input filters
Attribute Manipulation
Neighborn
All acceptable routes
BGP Decision Process
Loc-Rib
Best route to each destination
Output filters
Attribute Manipulation
Neighbor1
Output filters
Attribute Manipulation
Neighbor2
...
Output filters
Attribute Manipulation
Neighborn
forwarding entries
IP packets
IP forwarding table
IP packets

Given the set of all acceptable routes for each prefix, the BGP Decision process elects a single route

BGP is often referred to as a single path protocol

Prefer routes…

with higher LOCAL-PREF

with shorter AS-PATH length

with lower MED

learned via eBGP instead of iBGP

with lower IGP metric to the next-hop

with smaller egress IP address (tie-break) learned via eBGP instead of iBGP

with lower IGP metric to the next-hop

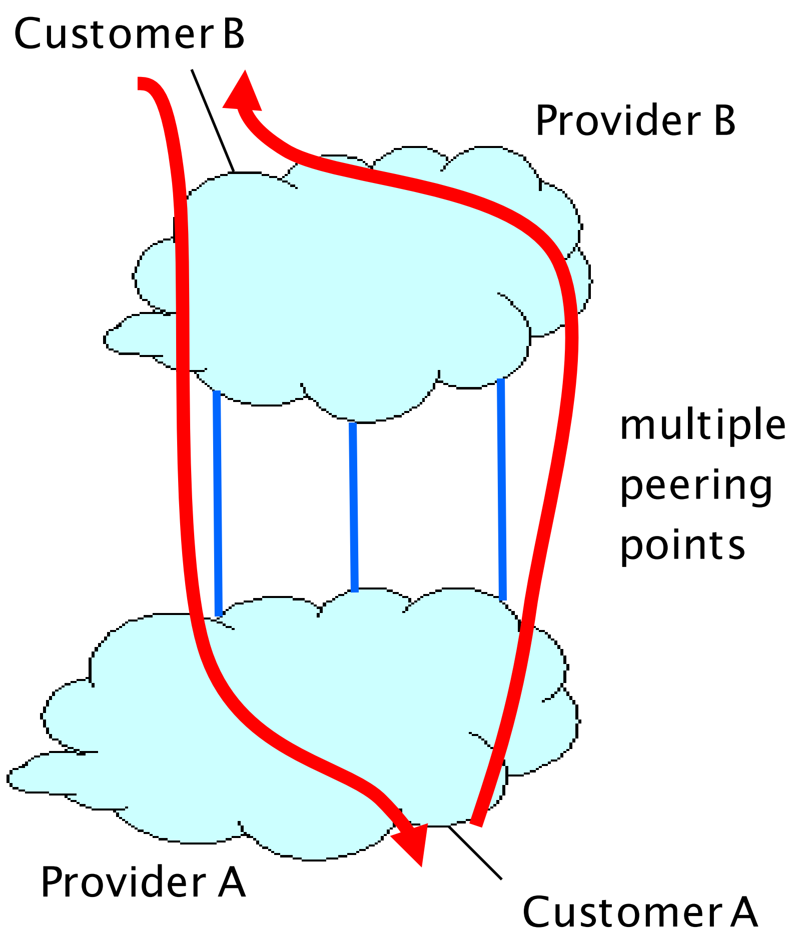
These two steps aim at directing traffic
as quickly as possible out of the AS (early exit routing)

ASes are selfish

They dump traffic
as soon as possible
to someone else

This leads to asymmetric routing

Traffic does not flow on
the same path
in both directions



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
	Convergence
	Performance
	Anomalies
	Relevance

Problems

Reachability

Security

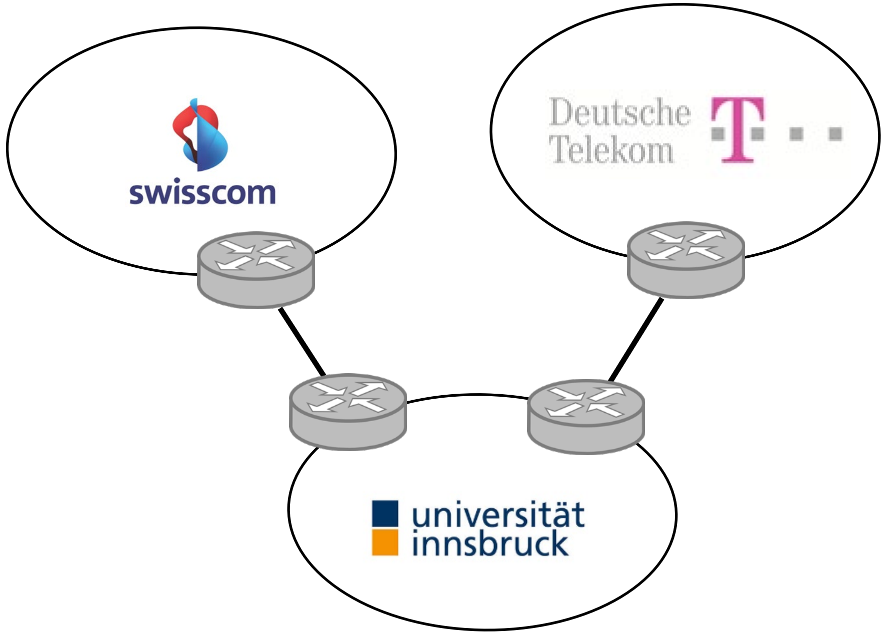
Convergence

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

Convergence

Performance

Anomalies

Relevance

Many security considerations are simply absent from BGP specifications

ASes can advertise any prefixes

even if they don't own them!

ASes can arbitrarily modify route content

e.g., change the content of the AS-PATH

ASes can forward traffic along different paths

than the advertised one

BGP (lack of) security

#1	BGP does not validate the origin of advertisements
#2	BGP does not validate the content of advertisements

BGP (lack of) security

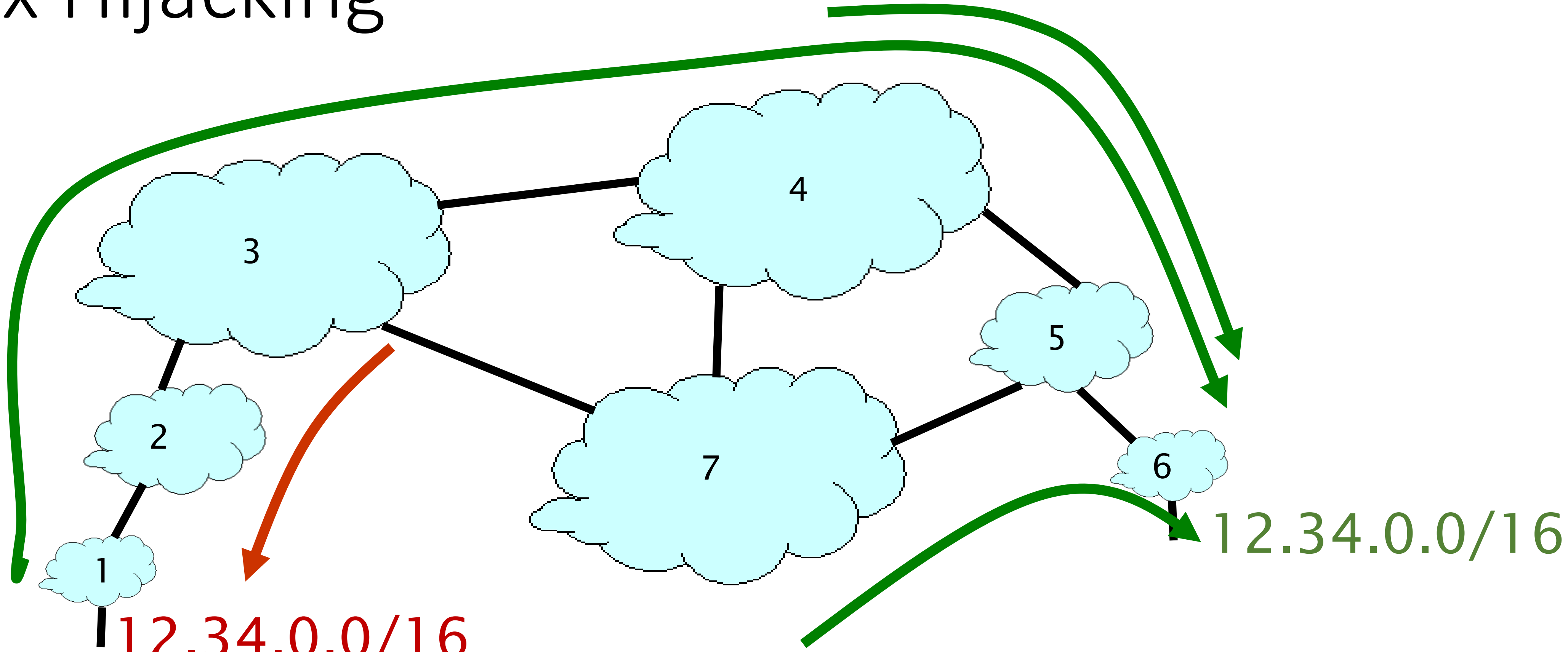
#1 BGP does not validate the origin of advertisements

#2 BGP does not validate the content of advertisements

IP Address Ownership and Hijacking

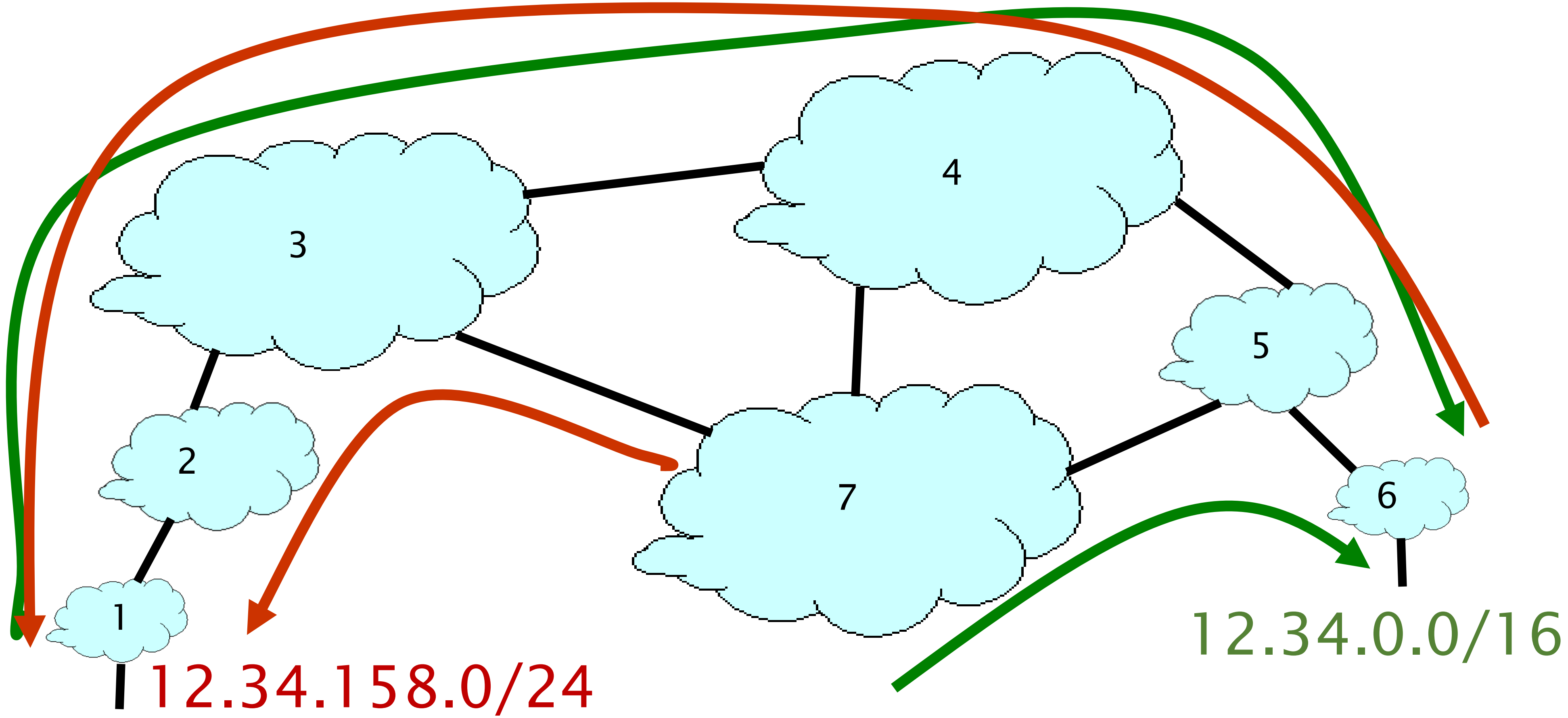
- IP address block assignment
  - Regional Internet Registries (ARIN, RIPE, APNIC)
  - Internet Service Providers
- Proper origination of a prefix into BGP
  - By the AS who owns the prefix
  - … or, by its upstream provider(s) in its behalf
- However, what's to stop someone else?
  - Prefix hijacking: 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

Sub-Prefix Hijacking



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

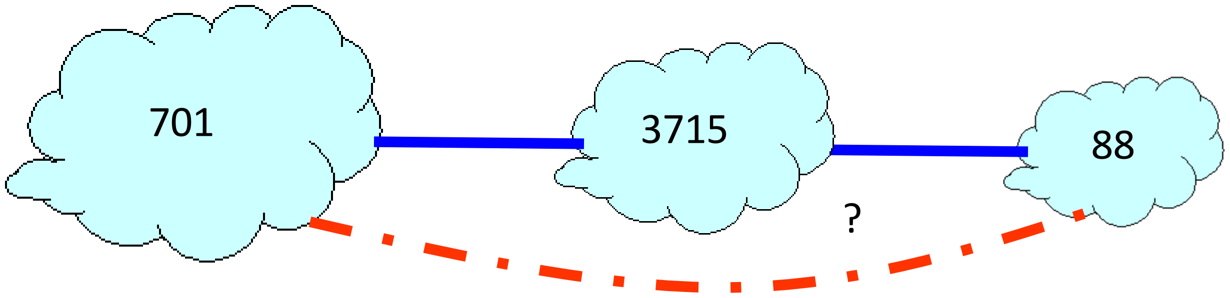
Hijacking is Hard to Debug

- 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

BGP (lack of) security

#1	BGP does not validate the origin of advertisements
#2	BGP does not validate the content of advertisements

Bogus AS Paths



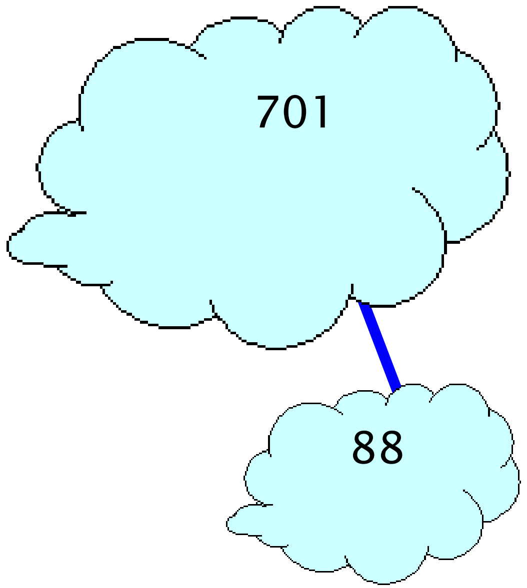
- Remove ASes from the AS path
  - E.g., turn “701 3715 88” into “701 88”

- 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?



Problems

Reachability

Security

Convergence

Performance

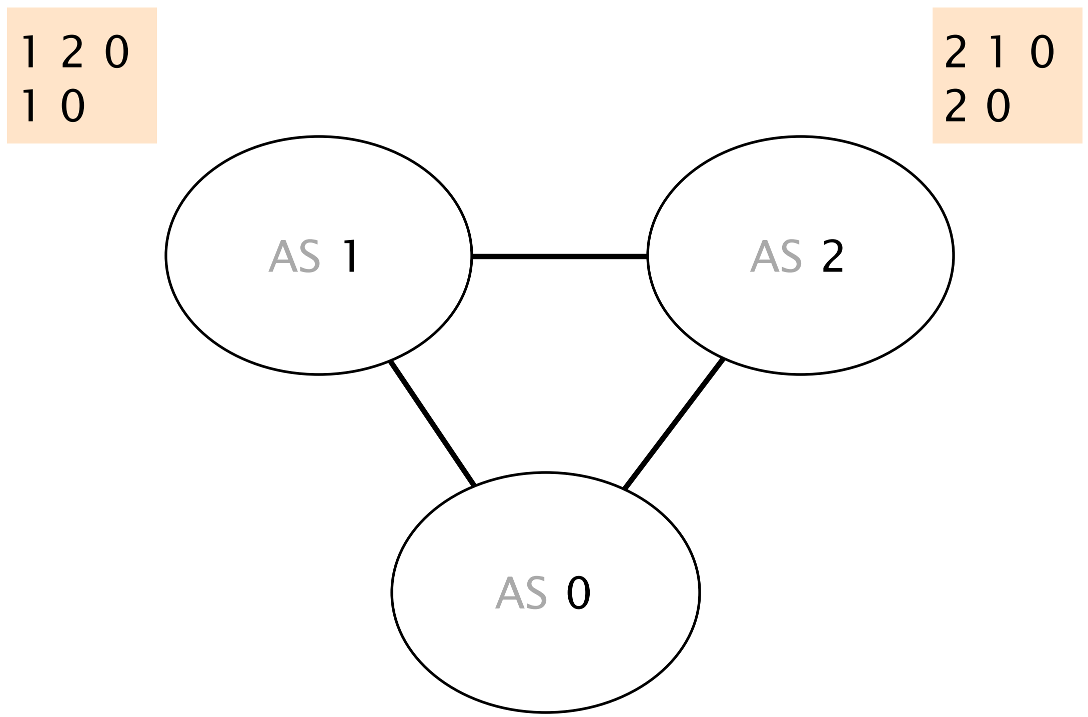
Anomalies

Relevance

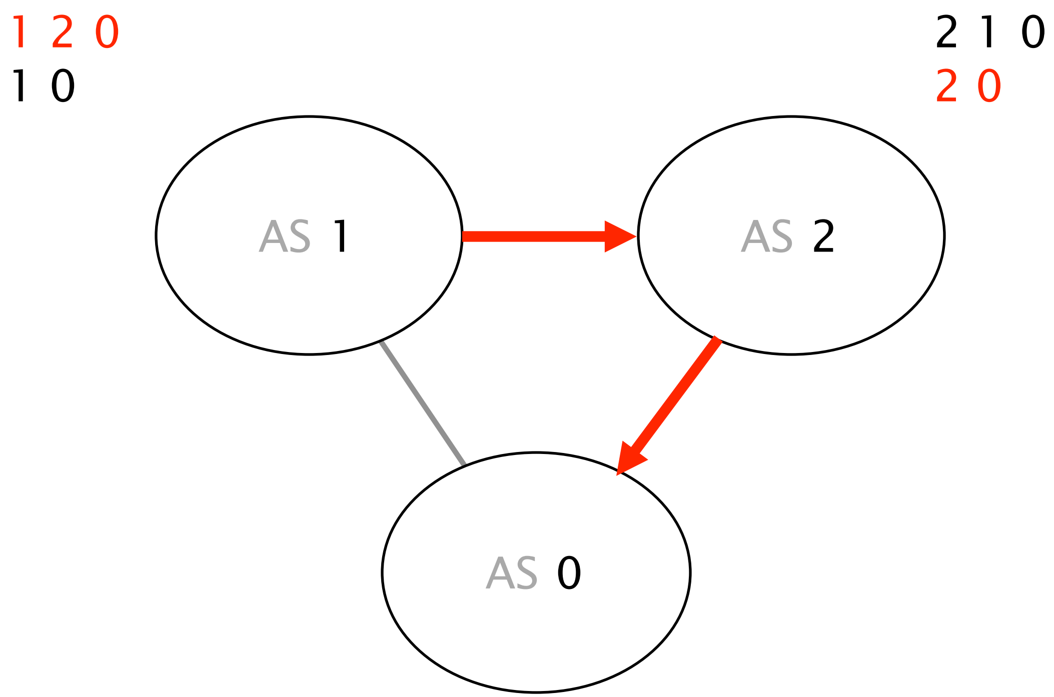
With arbitrary policies,
BGP may have multiple stable states

preference list

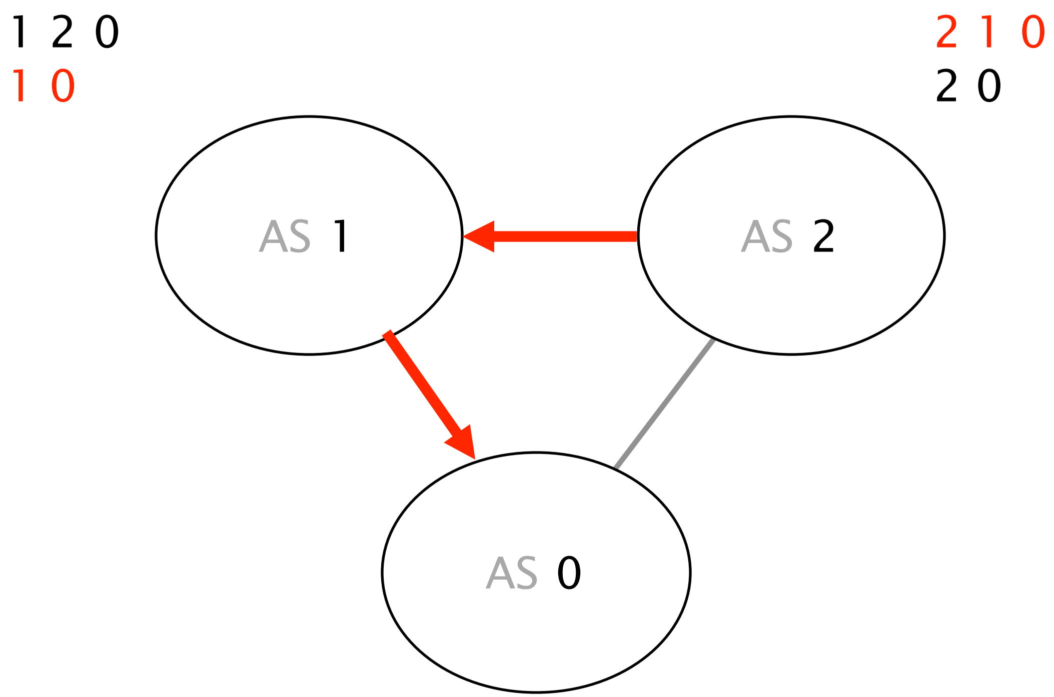
1 prefers to reach 0
via 2 rather than directly



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



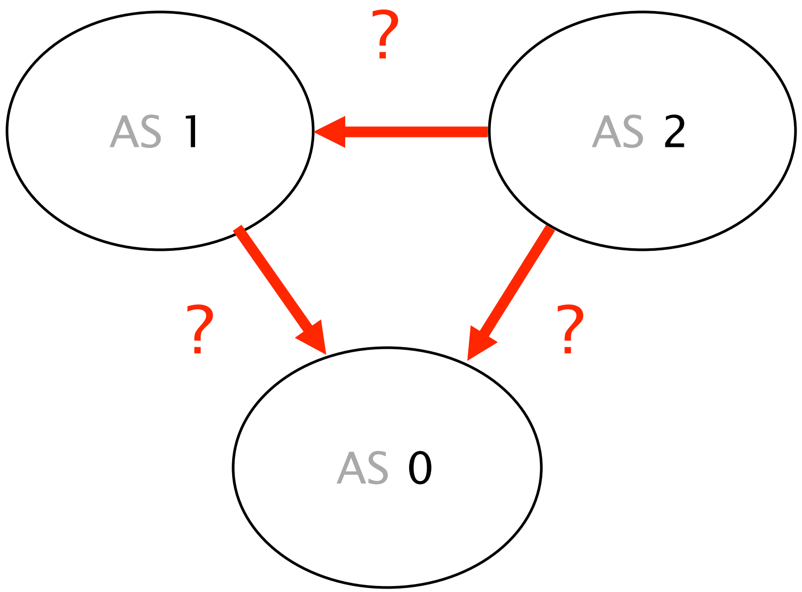
If AS1 is the first one to advertise 1 0,
the system stabilizes in a state where AS 2 is happy



The actual assignment depends on the ordering between the messages

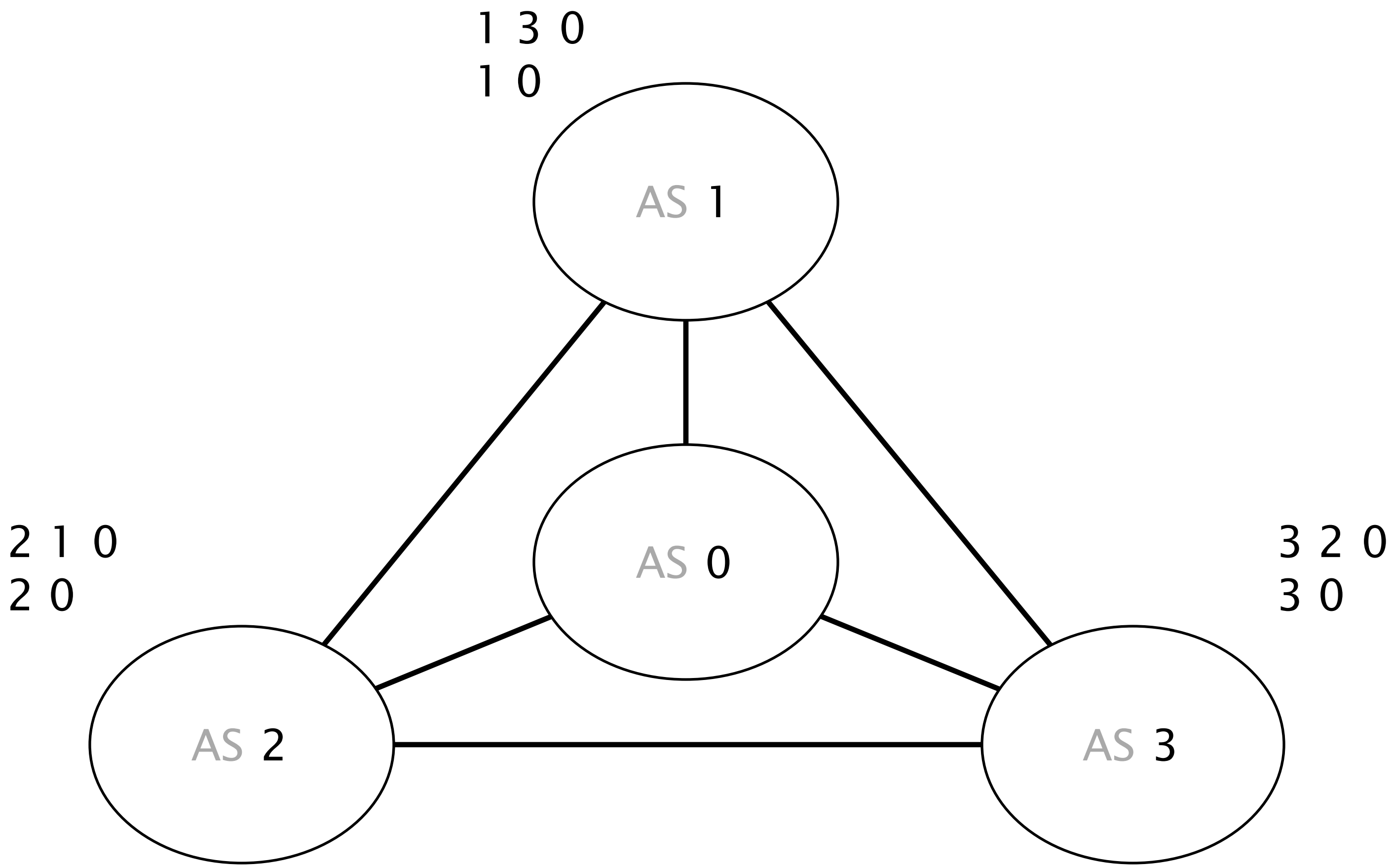
Note that AS1/AS2 could change the outcome by manual intervention

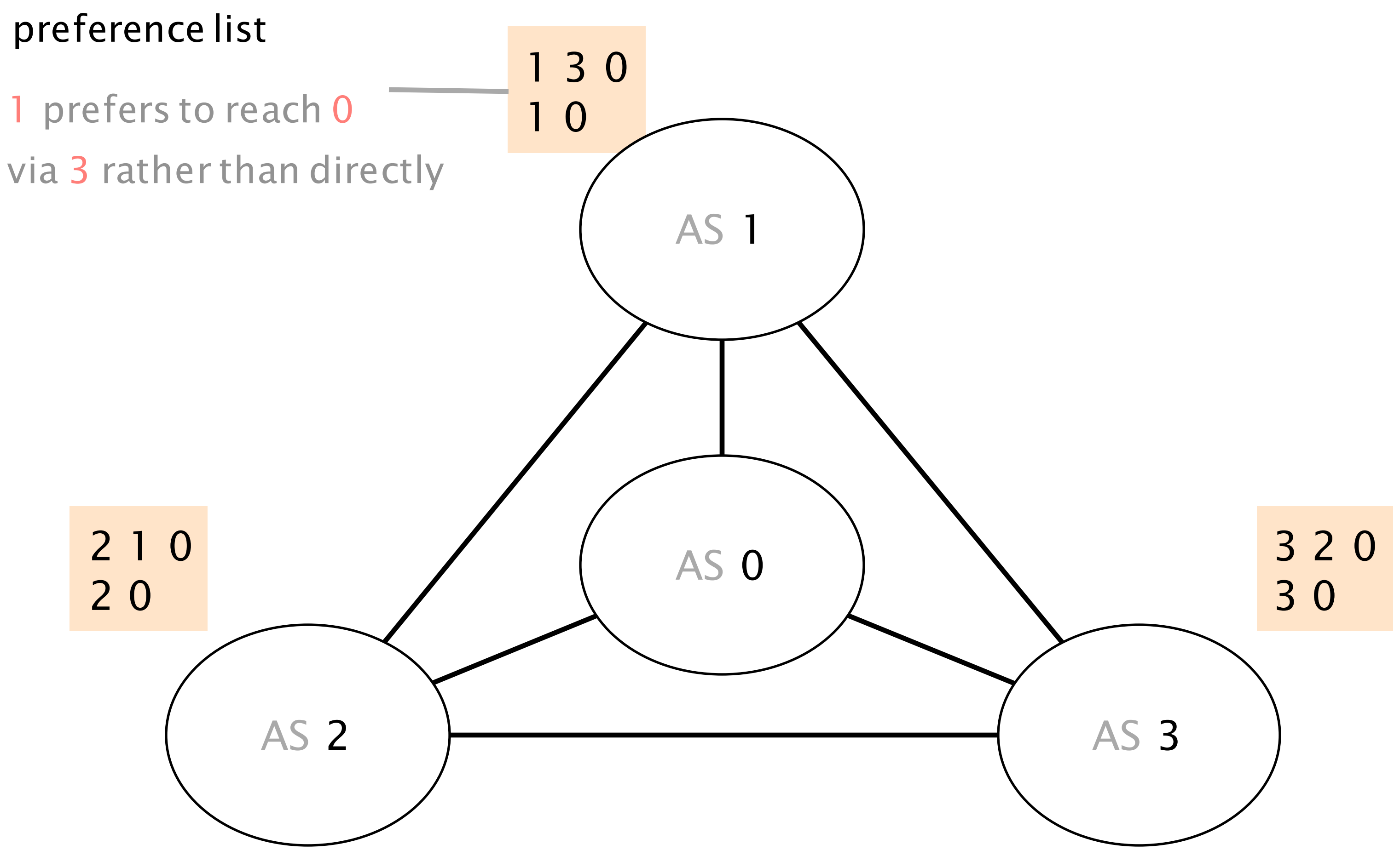
… this is not always possible *



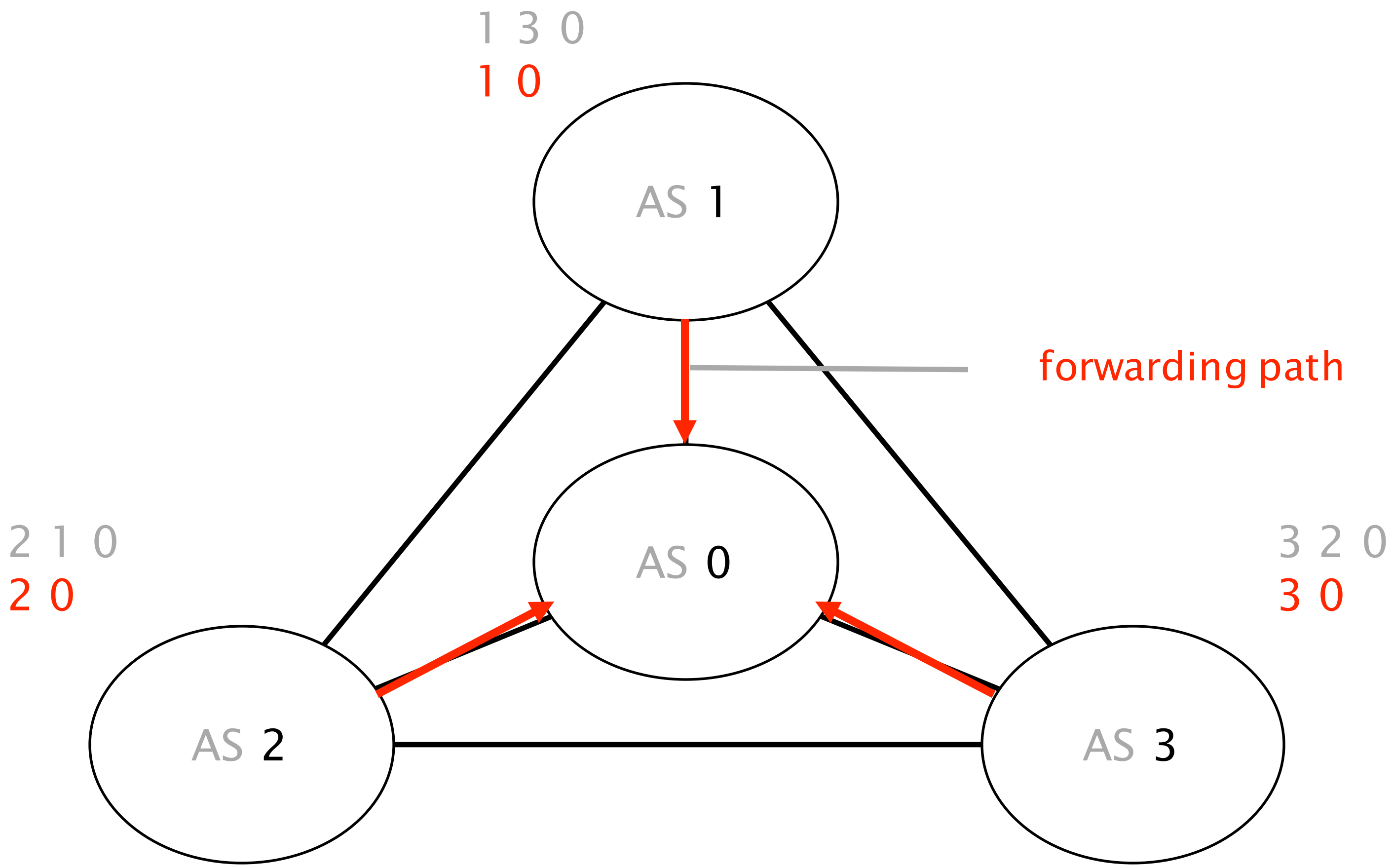
* https://www.nanog.org/meetings/nanog31/presentations/griffin.pdf

With arbitrary policies,
BGP may fail to converge

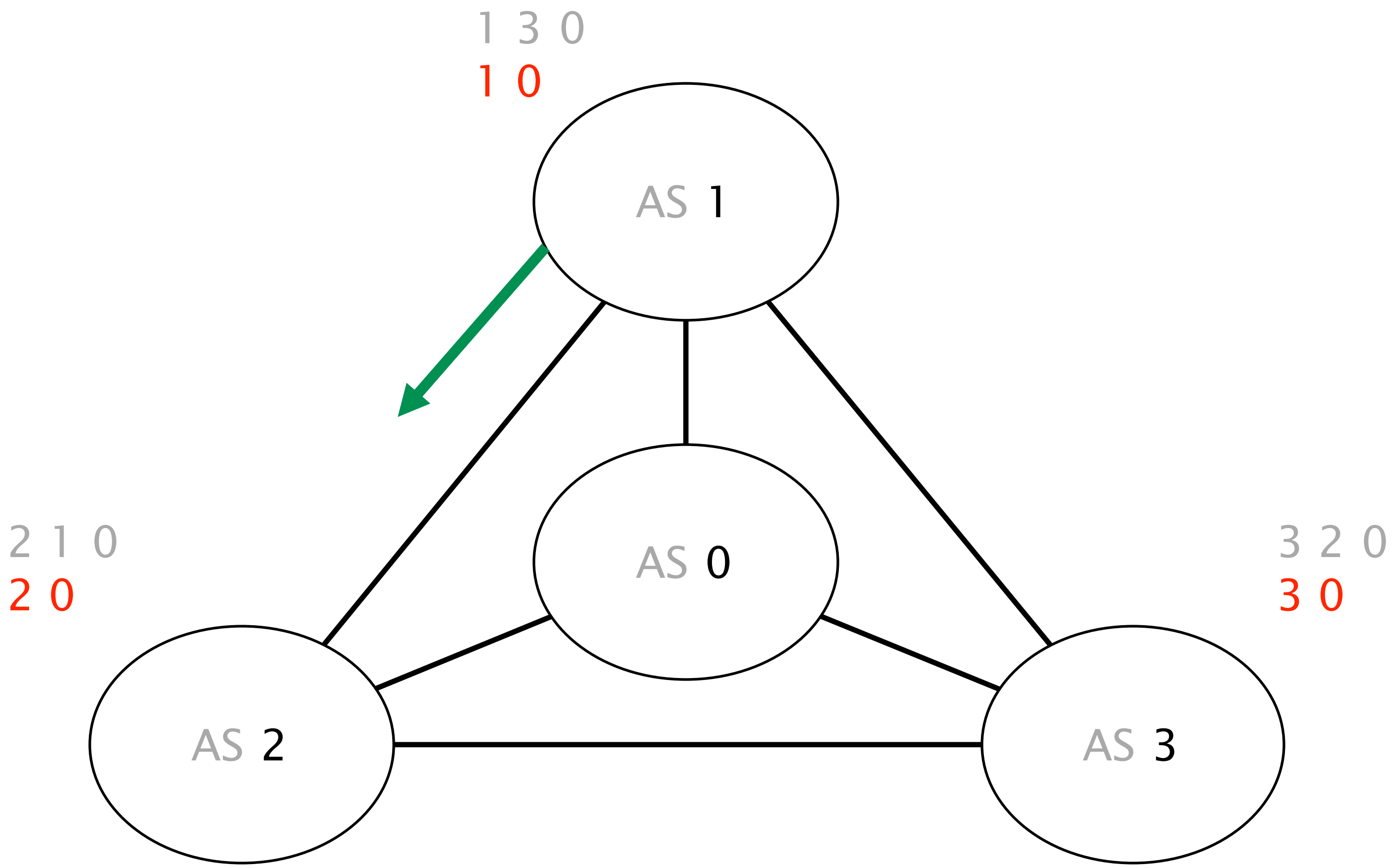

1 3 0
1 0
AS 1
AS 0
2 1 0
2 0
AS 2
3 2 0
3 0
AS 3


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

Initially, all ASes only know the direct route to 0

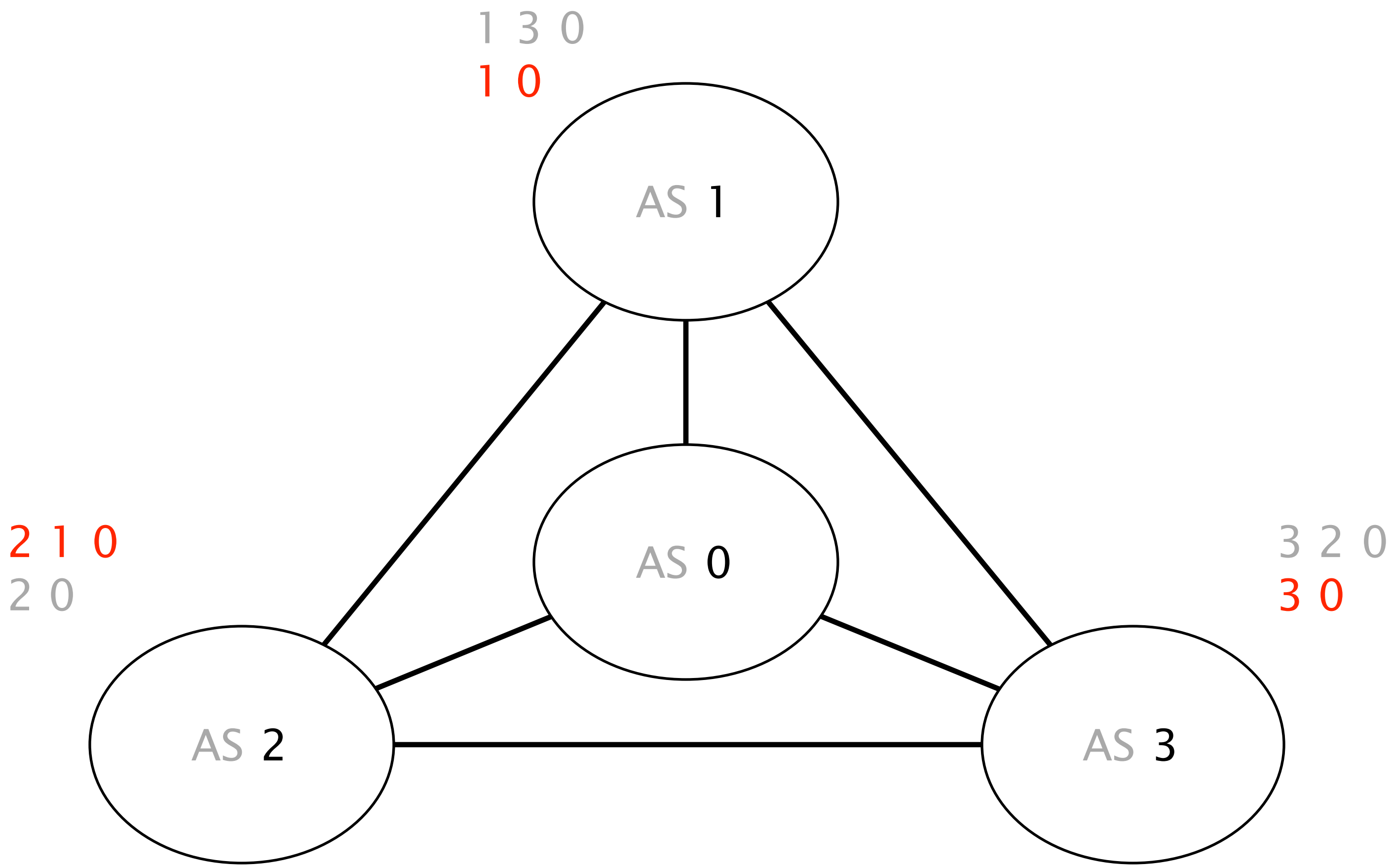


AS 1 advertises its path to AS 2

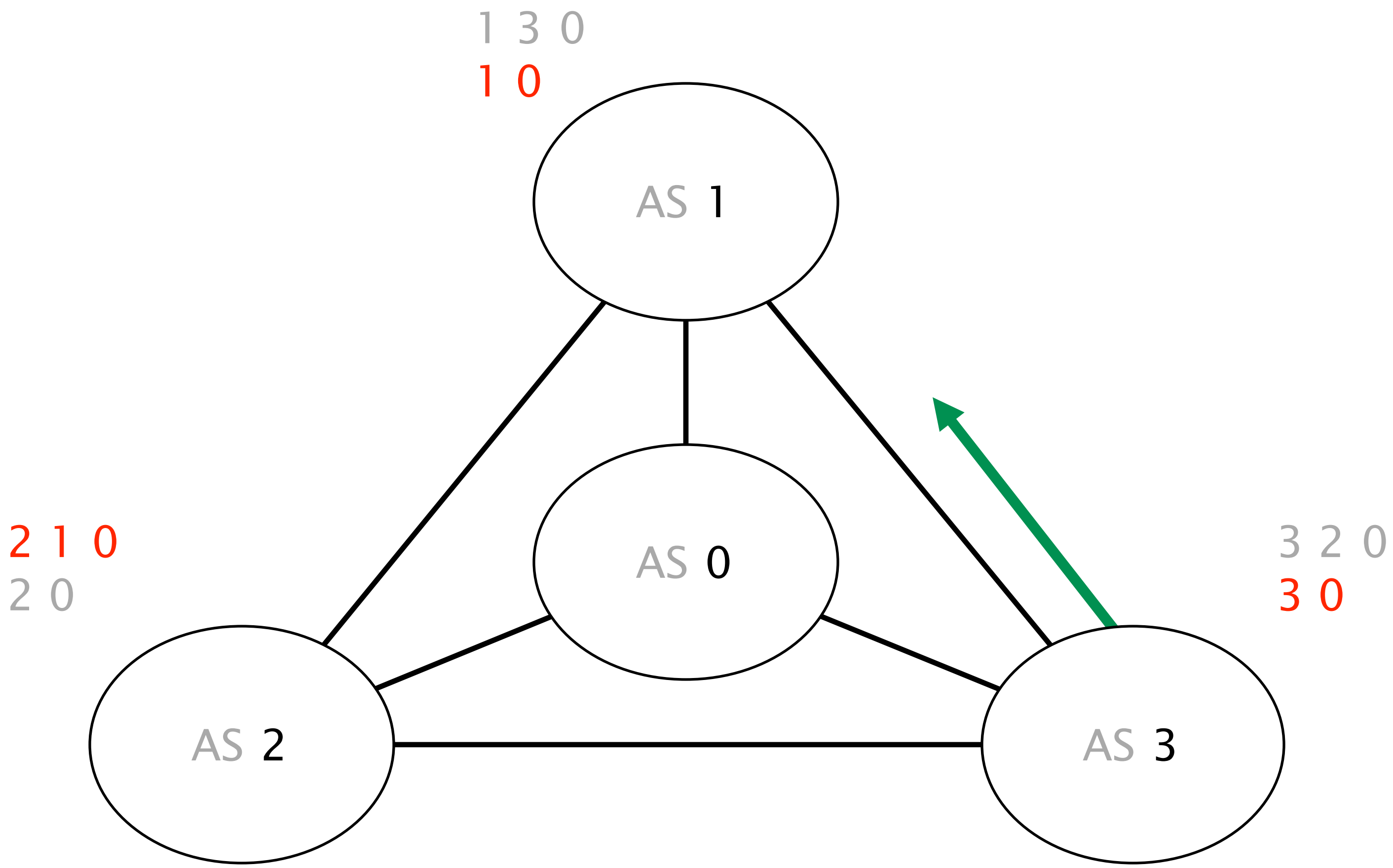


Upon reception,

AS 2 switches to 2 1 0 (preferred)

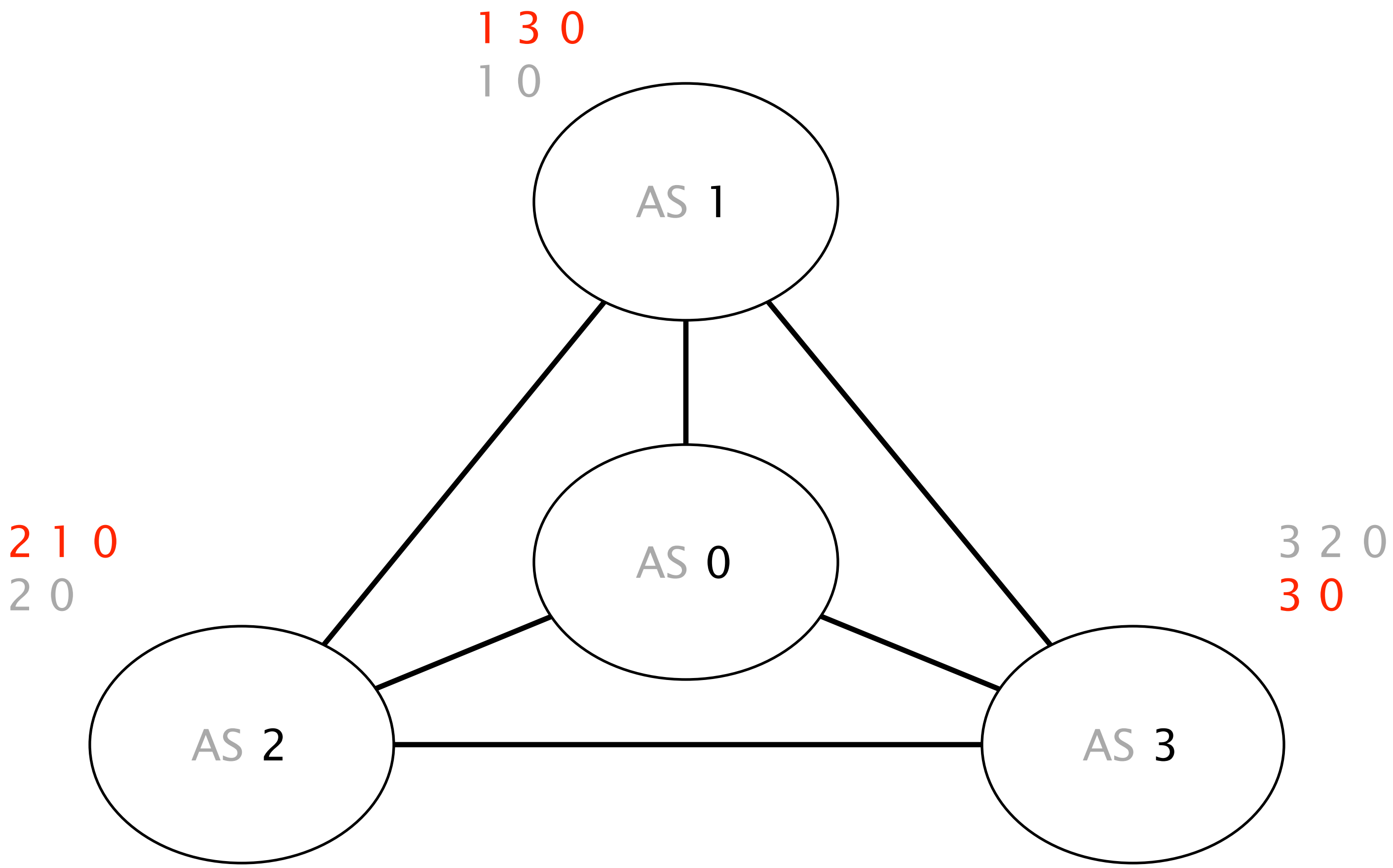


AS 3 advertises its path to AS 1

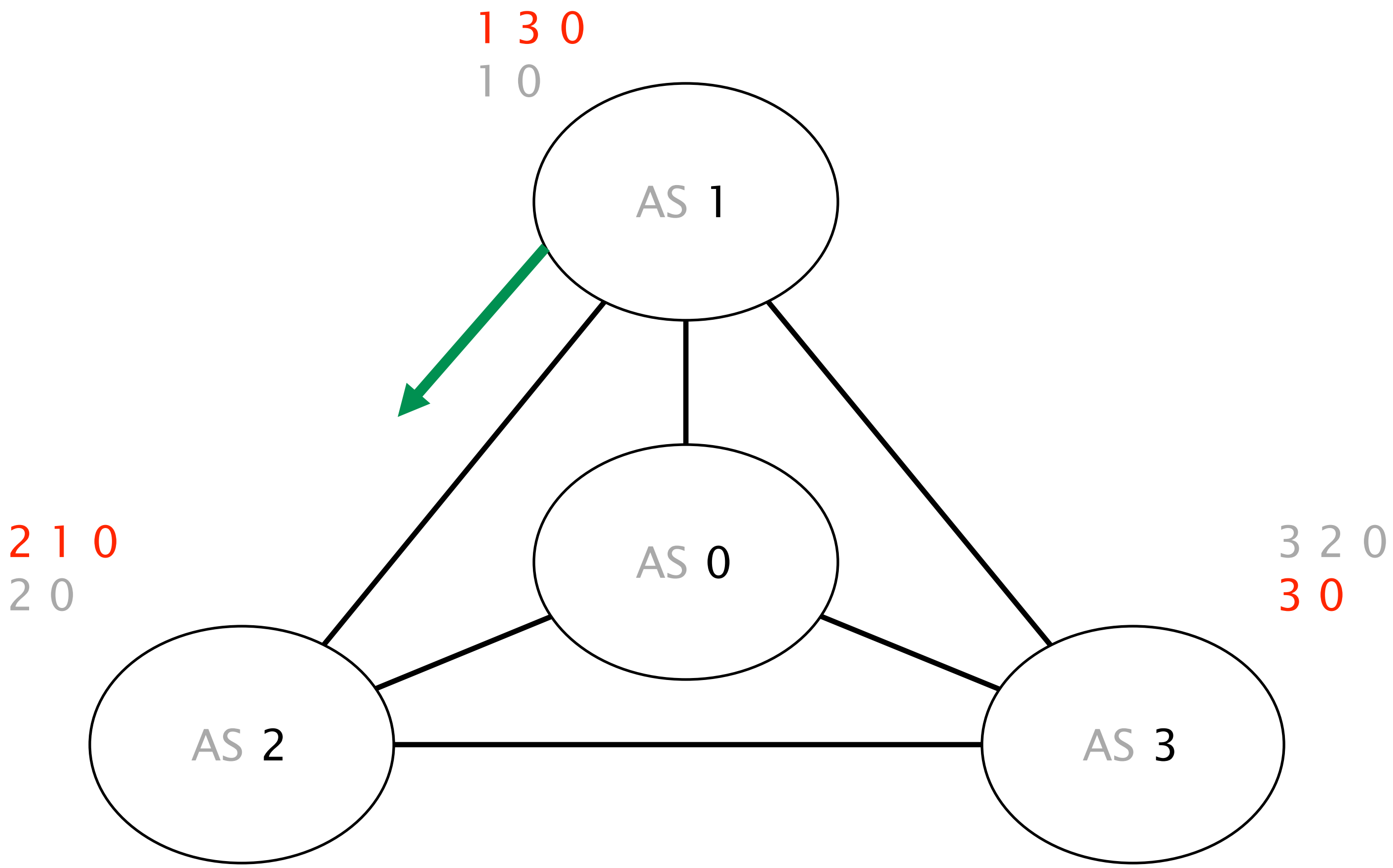


Upon reception,

AS 1 switches to 1 3 0 (preferred)

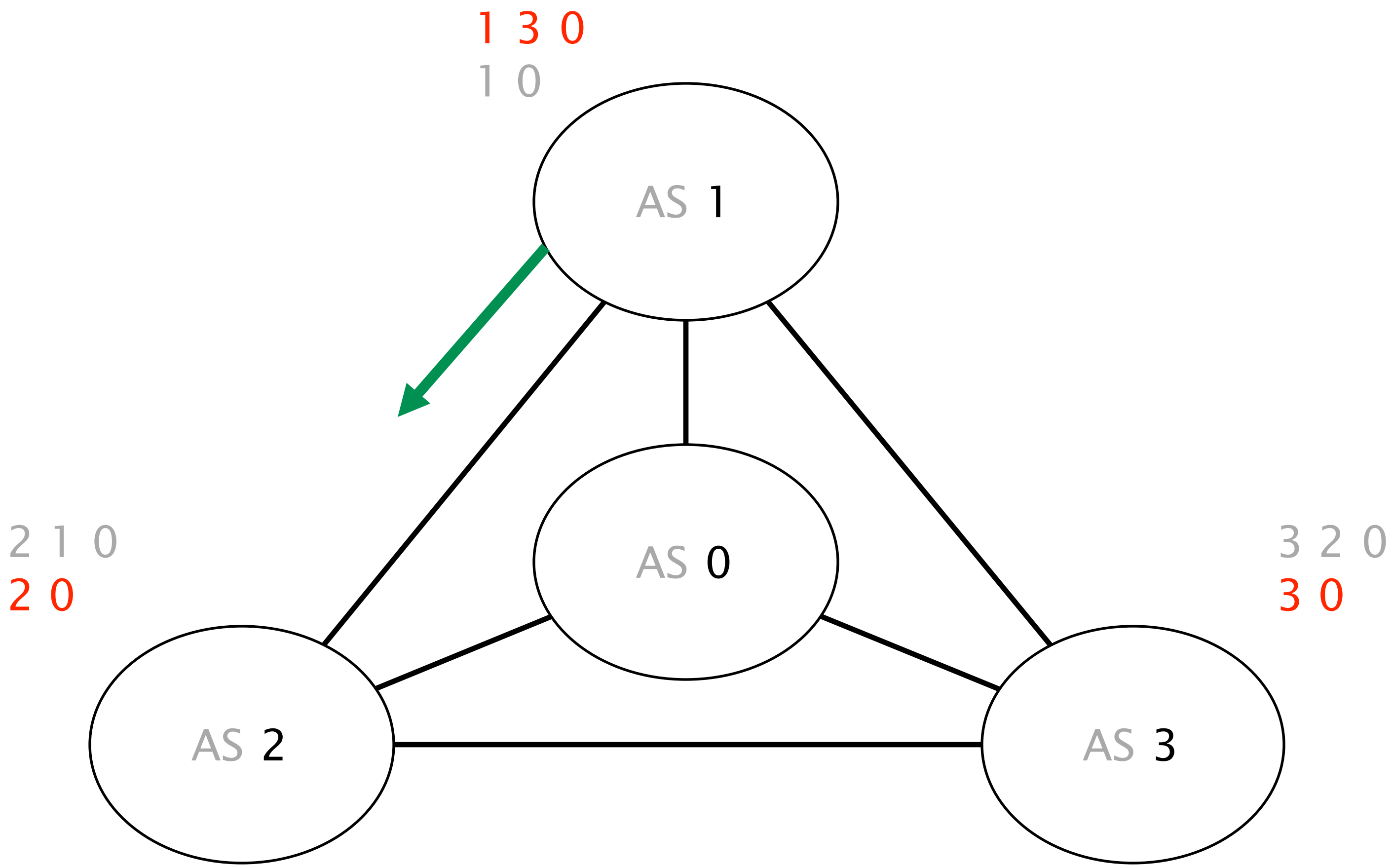


AS 1 advertises its new path 1 3 0 to AS 2

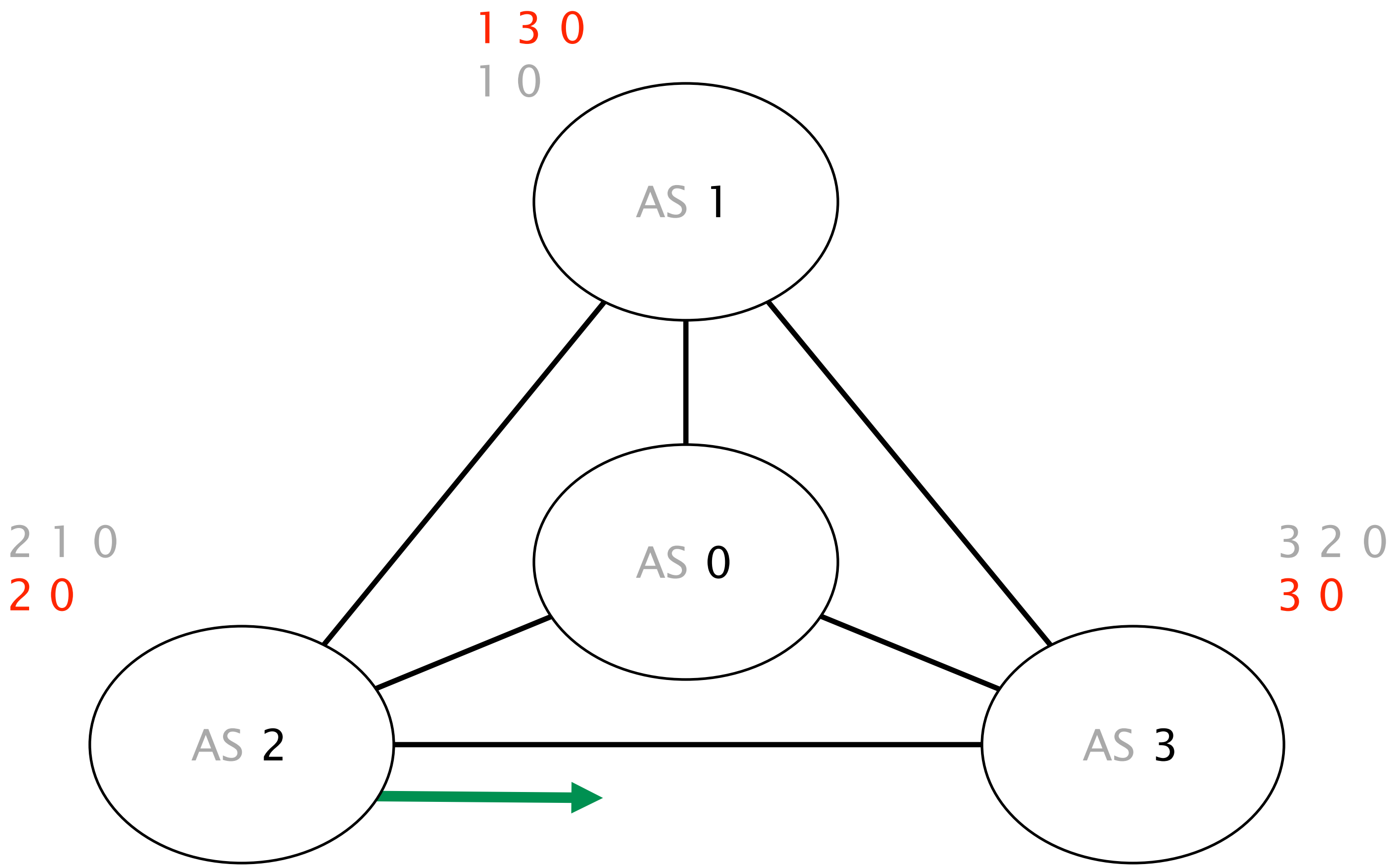


Upon reception,

AS 2 reverts back to its initial path 2 0

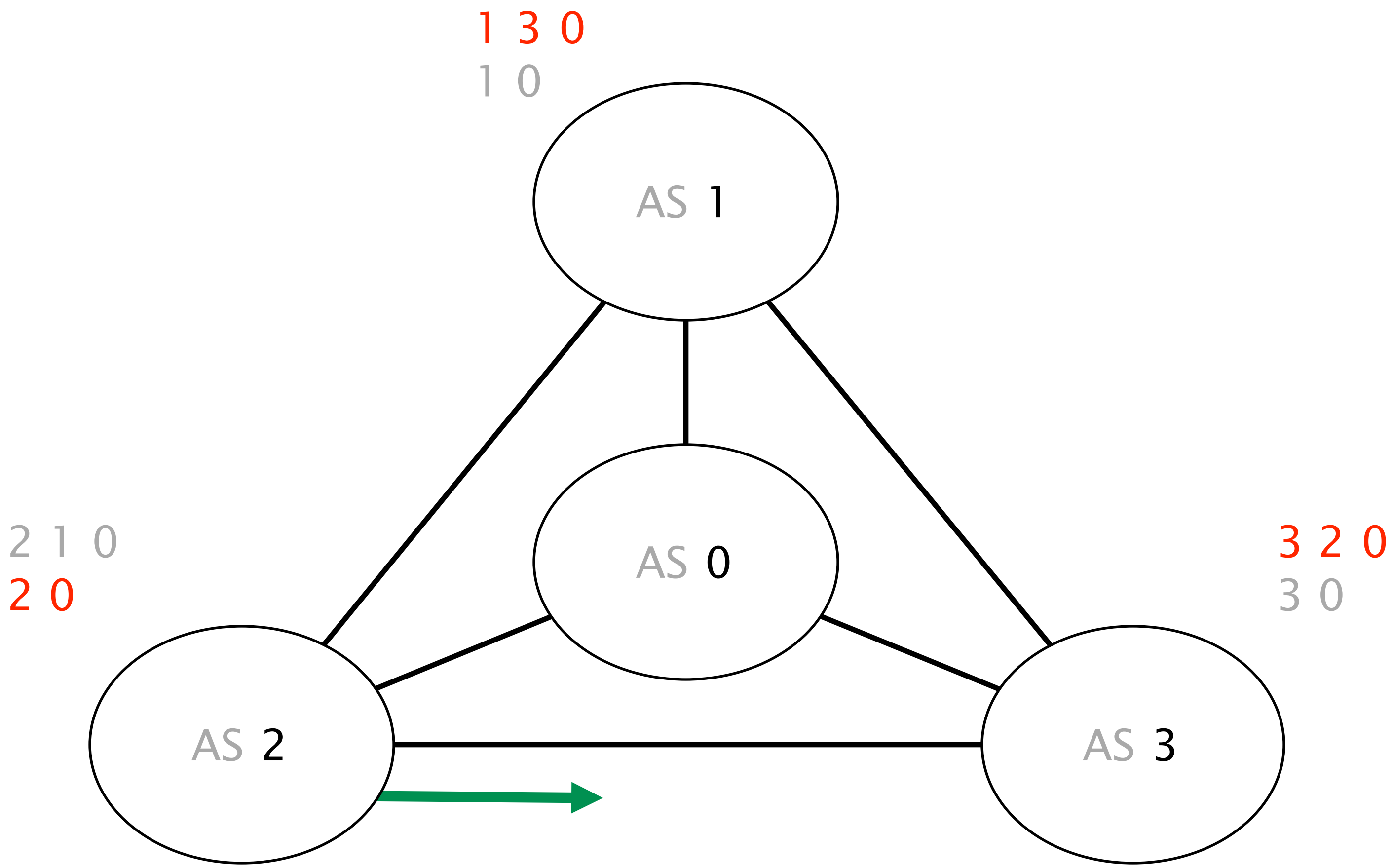


AS 2 advertises its path 2 0 to AS 3

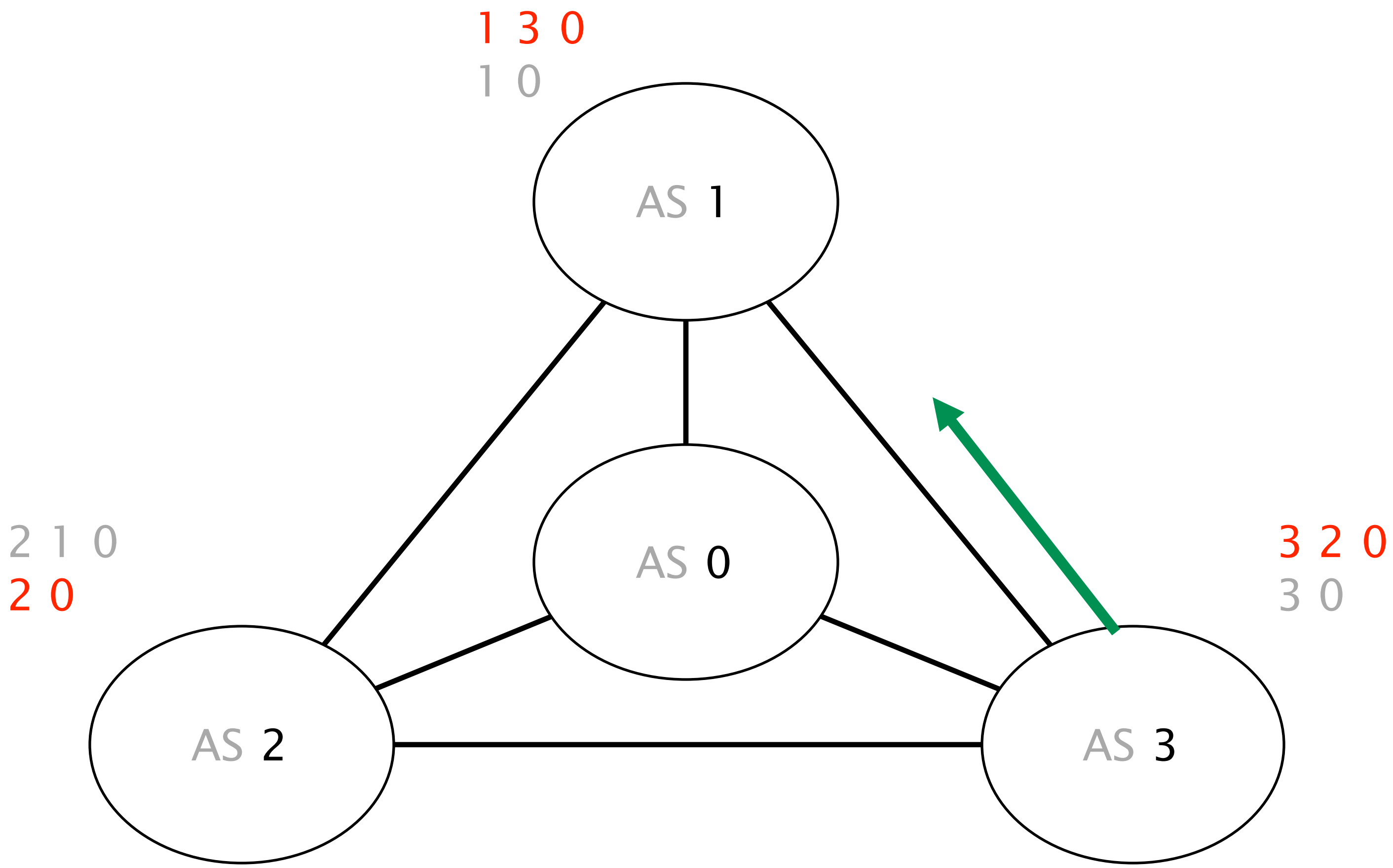


Upon reception,

AS 3 switches to 3 2 0 (preferred)

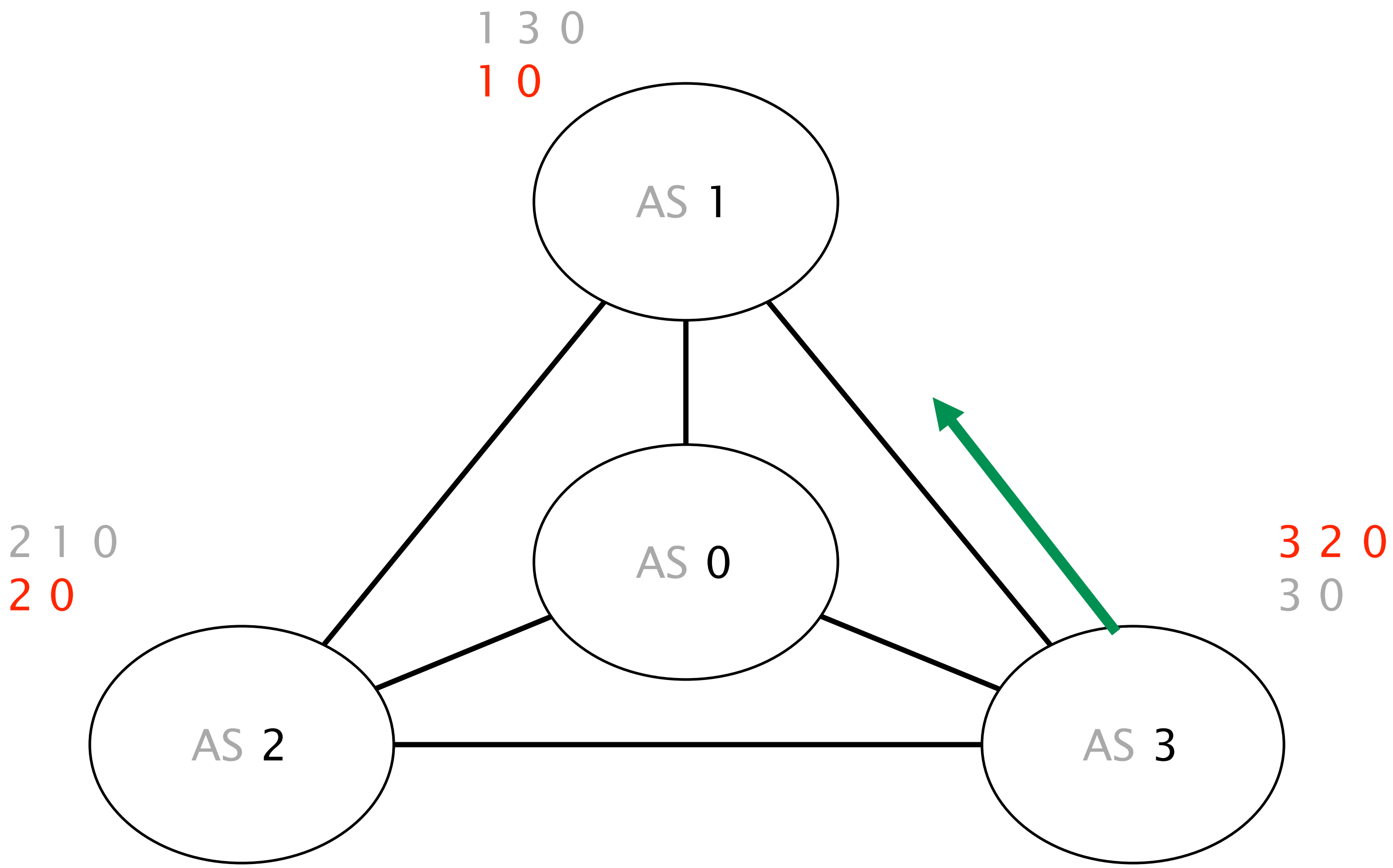


AS 3 advertises its new path 3 2 0 to AS 1

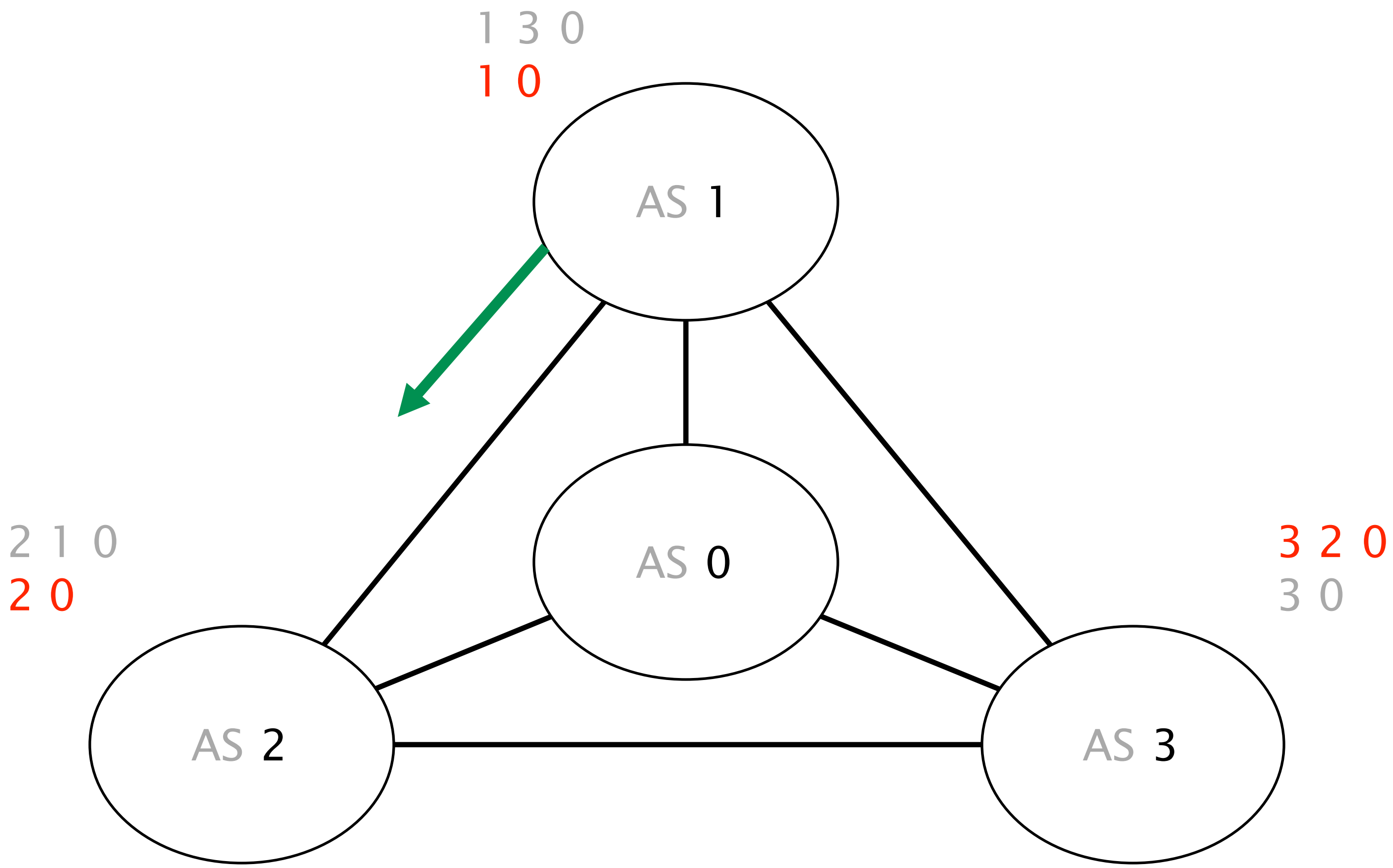


Upon reception,

AS 1 reverts back to 1 0 (initial path)

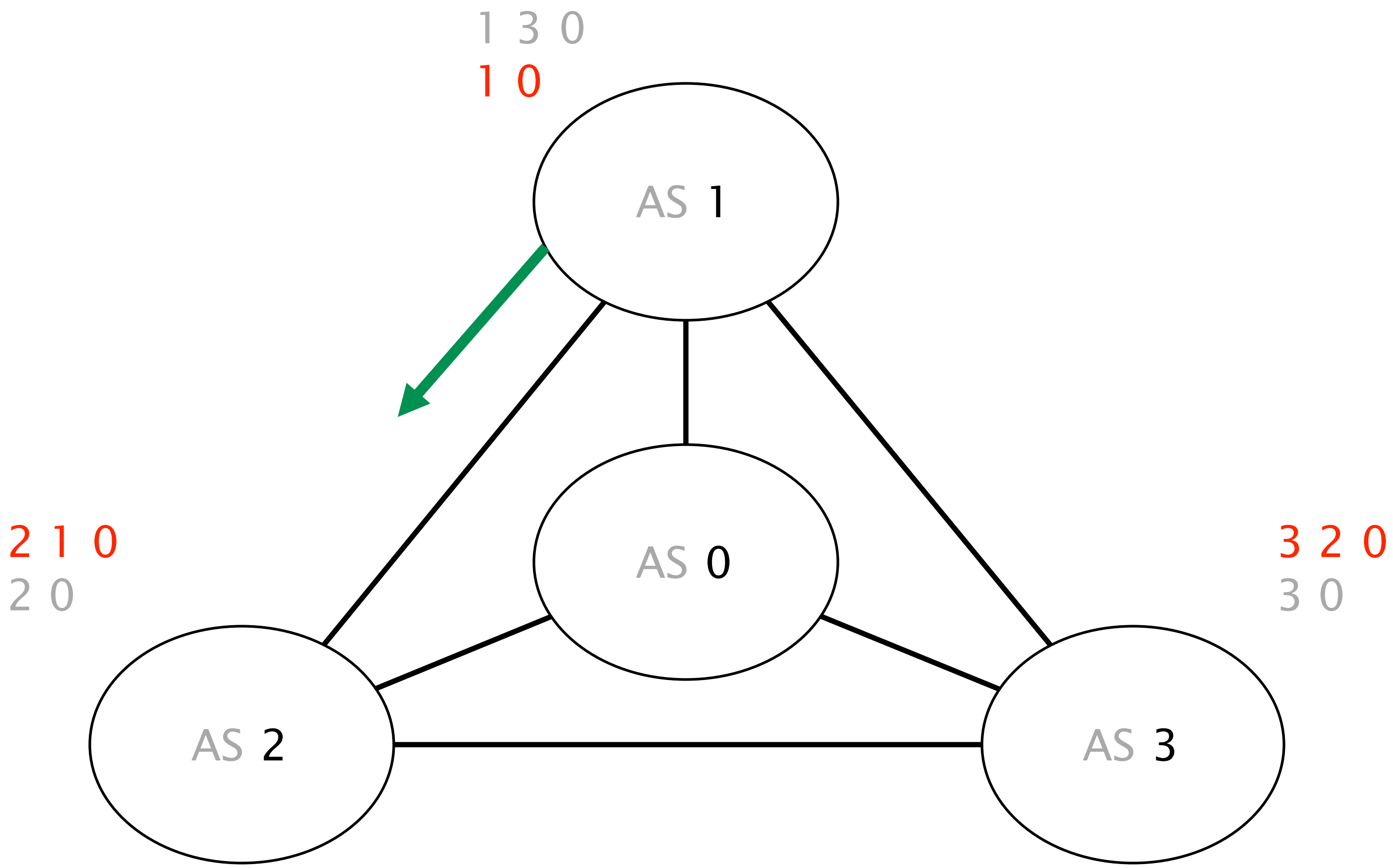


AS 1 advertises its new path 1 0 to AS 2

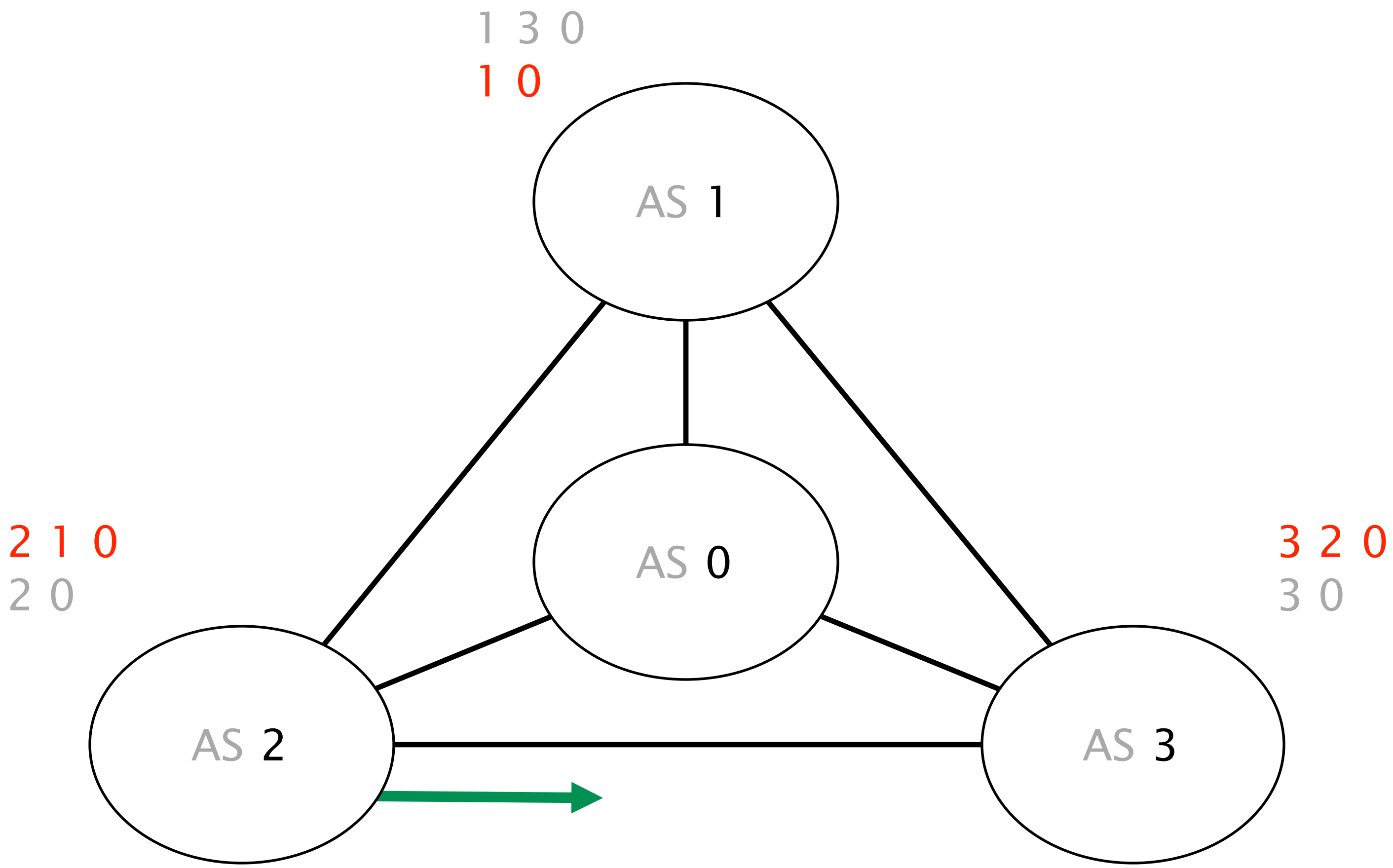


Upon reception,

AS 2 switches to 2 1 0 (preferred)

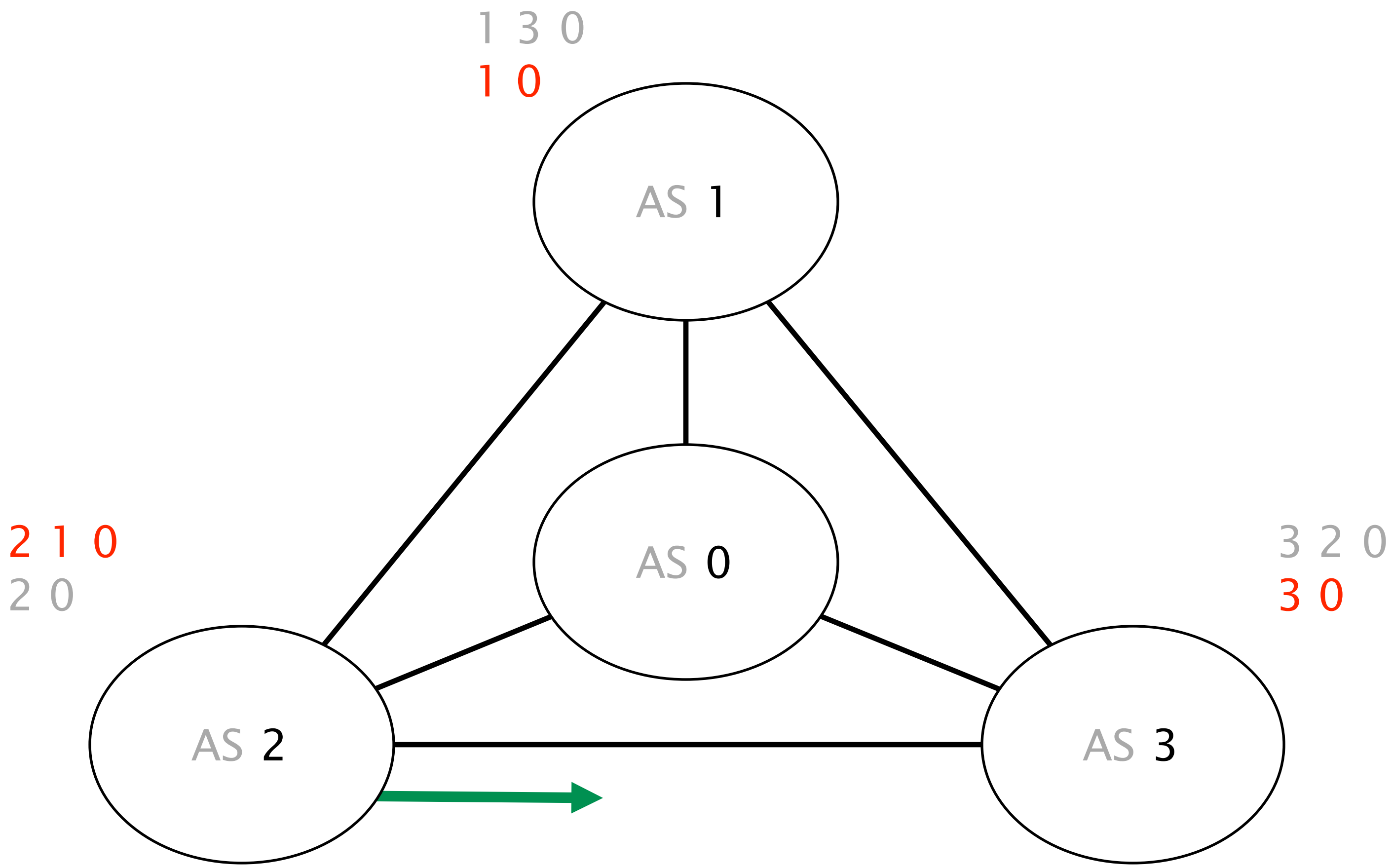


AS 2 advertises its new path 2 1 0 to AS 3

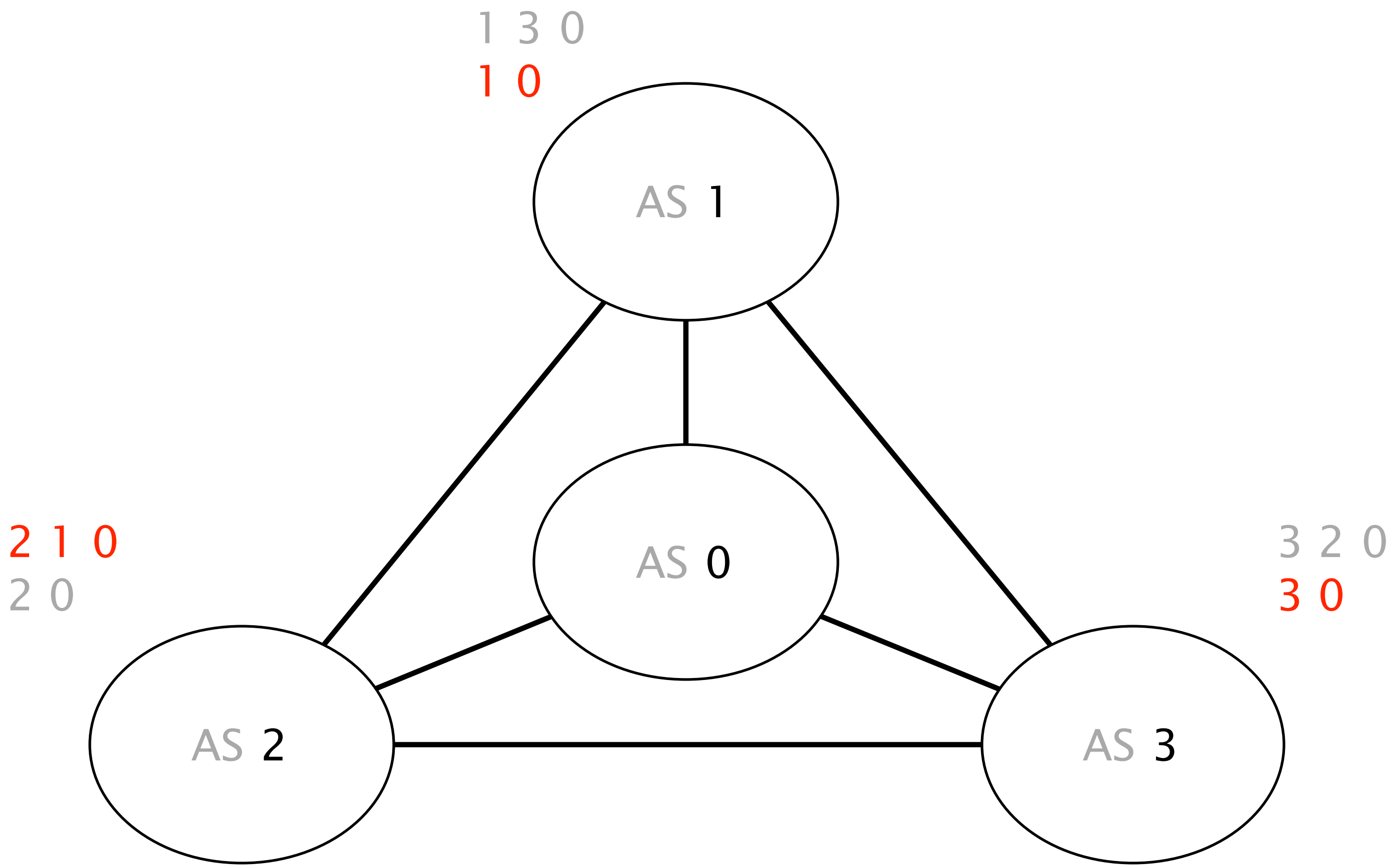


Upon reception,

AS 3 switches to its initial path 3 0



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



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 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

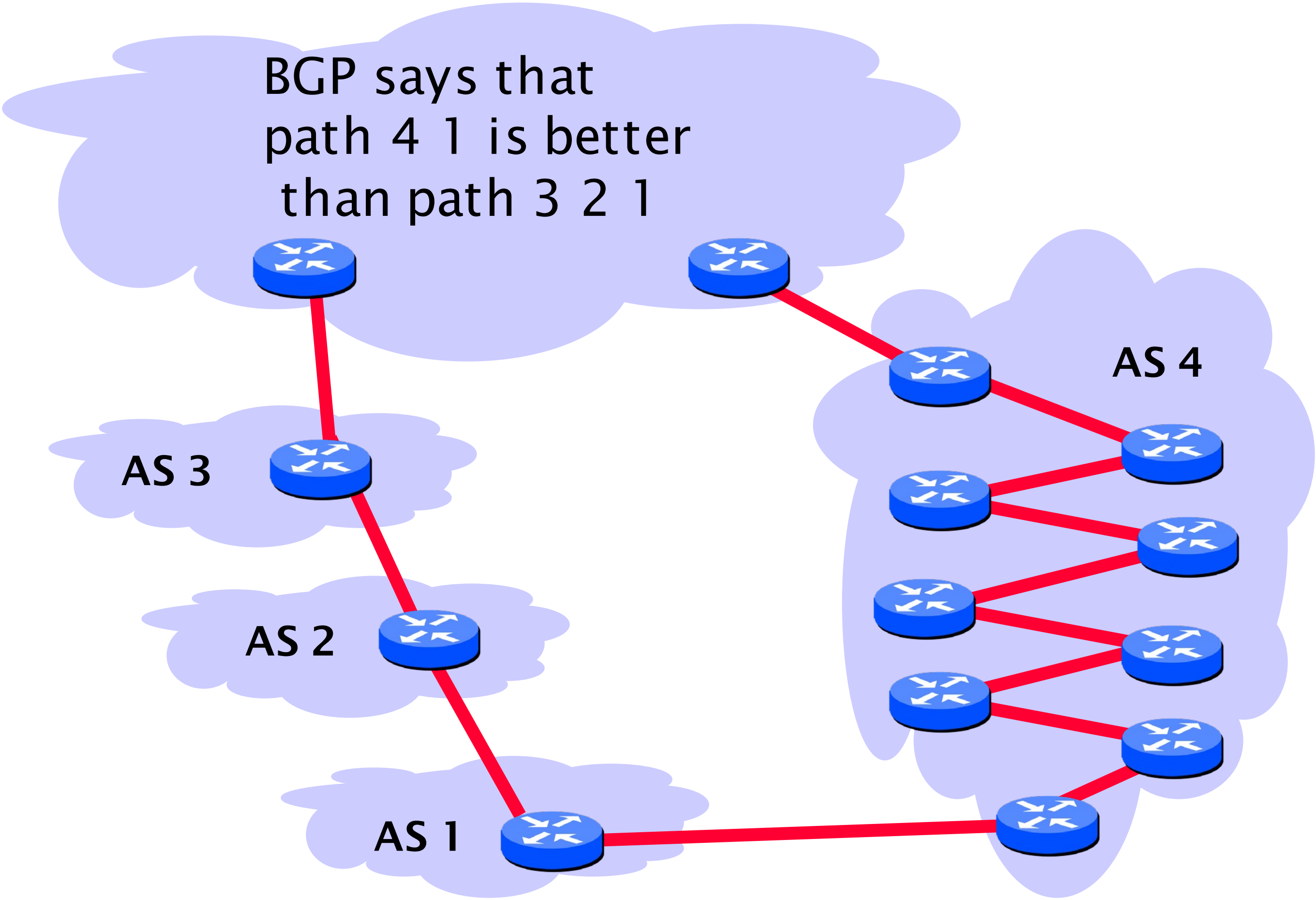
Convergence

Performance

Anomalies

Relevance

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



Problems

Reachability

Security

Convergence

Performance

Anomalies

Relevance

BGP configuration is hard to get right,
you’ll understand that very soon

BGP is both “bloated” and underspecified

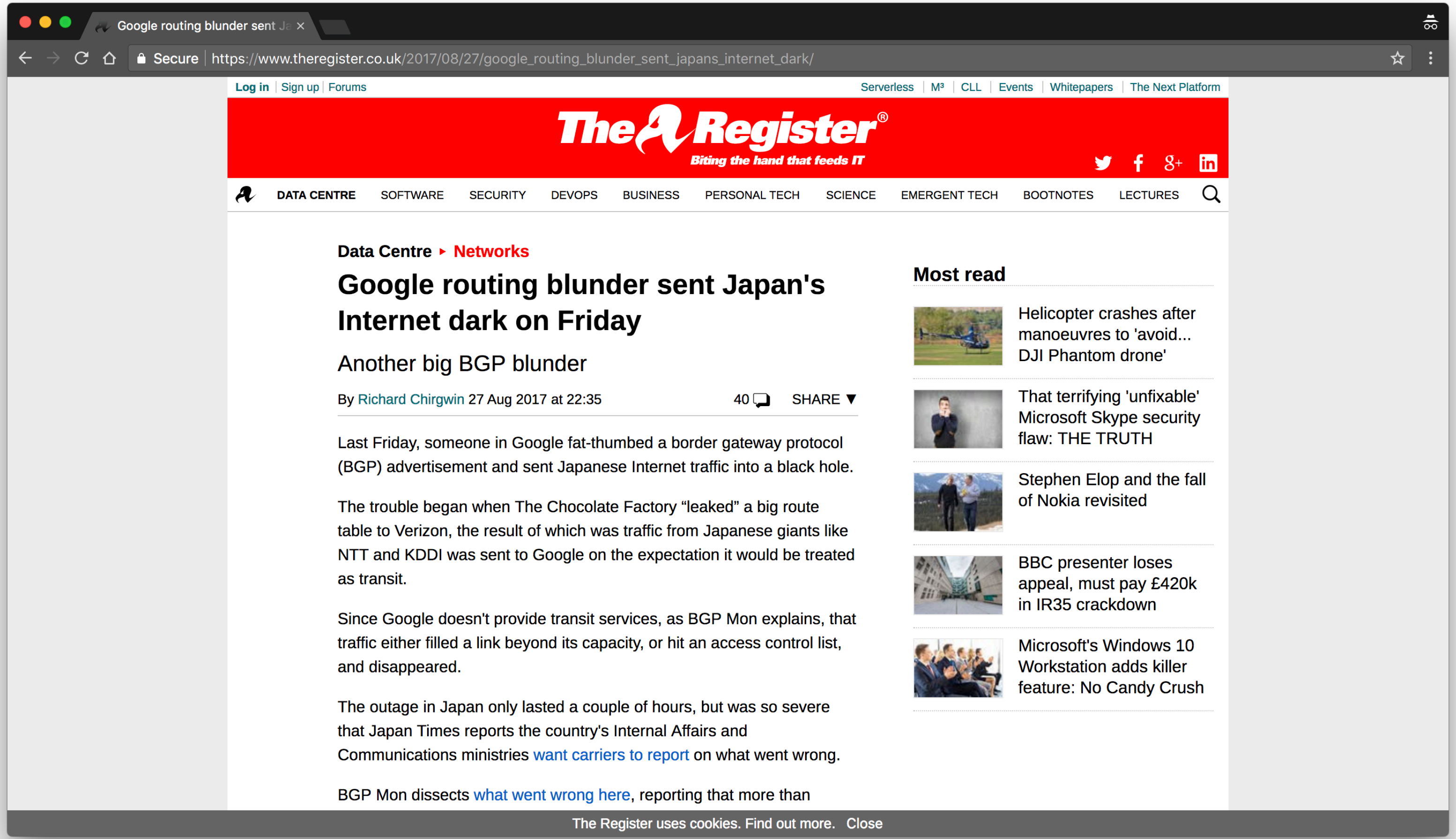
lots of knobs and (sometimes, conflicting) interpretations

BGP is often manually configured

humans make mistakes, often

BGP abstraction is fundamentally flawed

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



Google routing blunder sent J...

Secure | https://www.theregister.co.uk/2017/08/27/google_routing_blunder_sent_japans_internet_dark/

Log in | Sign up | Forums

Serverless | M³ | CLL | Events | Whitepapers | The Next Platform

The Register®

Biting the hand that feeds IT

DATA CENTRE SOFTWARE SECURITY DEVOPS BUSINESS PERSONAL TECH SCIENCE EMERGENT TECH BOOTNOTES LECTURES

Data Centre ▸ Networks

Google routing blunder sent Japan's Internet dark on Friday

Another big BGP blunder

By Richard Chirgwin 27 Aug 2017 at 22:35 40 SHARE ▼

Last Friday, someone in Google fat-thumbed a border gateway protocol (BGP) advertisement and sent Japanese Internet traffic into a black hole.

The trouble began when The Chocolate Factory "leaked" a big route table to Verizon, the result of which was traffic from Japanese giants like NTT and KDDI was sent to Google on the expectation it would be treated as transit.

Since Google doesn't provide transit services, as BGP Mon explains, that traffic either filled a link beyond its capacity, or hit an access control list, and disappeared.

The outage in Japan only lasted a couple of hours, but was so severe that Japan Times reports the country's Internal Affairs and Communications ministries want carriers to report on what went wrong.

BGP Mon dissects what went wrong here, reporting that more than

Most read

Helicopter crashes after manoeuvres to 'avoid... DJI Phantom drone'

That terrifying 'unfixable' Microsoft Skype security flaw: THE TRUTH

Stephen Elop and the fall of Nokia revisited

BBC presenter loses appeal, must pay £420k in IR35 crackdown

Microsoft's Windows 10 Workstation adds killer feature: No Candy Crush

The Register uses cookies. Find out more. Close

https://www.theregister.co.uk/2017/08/27/google_routing_blunder_sent_japans_internet_dark/

In August 2017

Someone in Google fat-thumbed a
Border Gateway Protocol (BGP) advertisement
and sent Japanese Internet traffic into a black hole.

In August 2017

Someone in Google fat-thumbed a
Border Gateway Protocol (BGP) advertisement
and sent Japanese Internet traffic into a black hole.

[…] Traffic from Japanese giants like NTT and KDDI
was sent to Google on the expectation
it would be treated as transit.

In August 2017

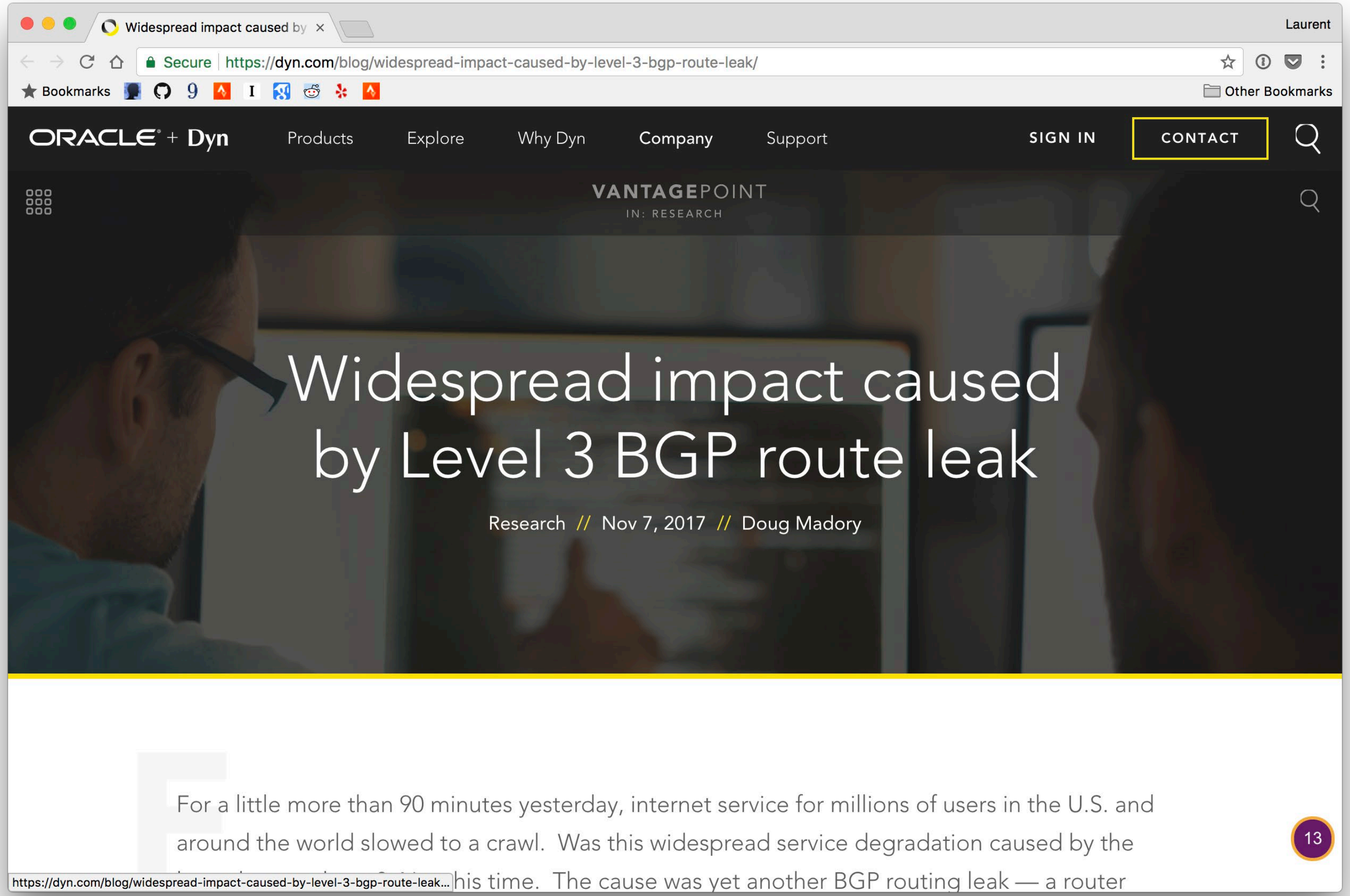
Someone in Google fat-thumbed a
Border Gateway Protocol (BGP) advertisement
and sent Japanese Internet traffic into a black hole.

[…] Traffic from Japanese giants like NTT and KDDI
was sent to Google on the expectation
it would be treated as transit.

The outage in Japan only lasted a couple of hours
but was so severe that […] the country's
Internal Affairs and Communications ministries
want carriers to report on what went wrong.

Another example,

this time from November 2017



https://dyn.com/blog/widespread-impact-caused-by-level-3-bgp-route-leak/

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.

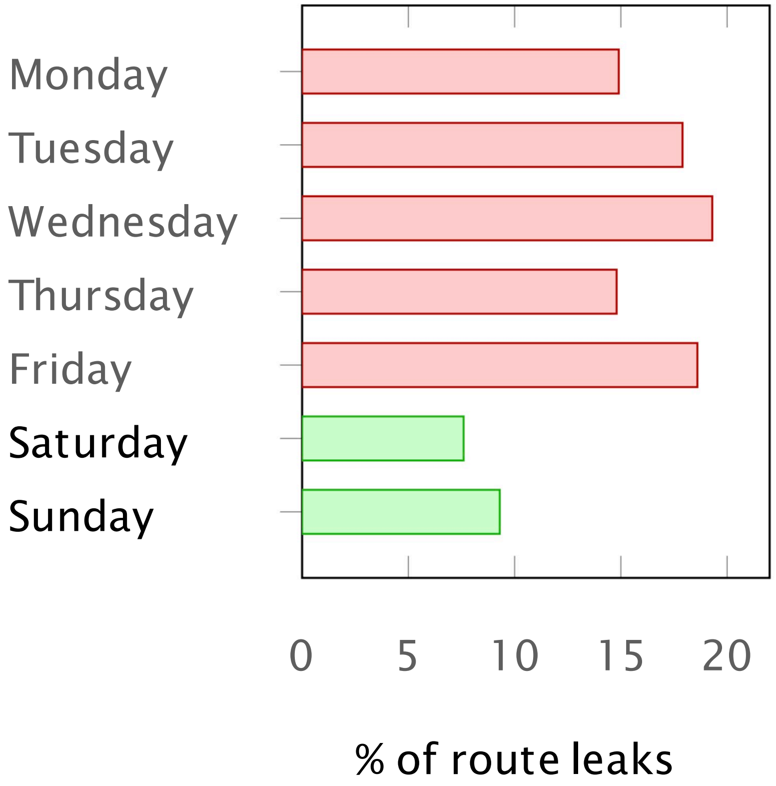
The cause was yet another BGP routing leak, a router misconfiguration directing Internet traffic from its intended path to somewhere else.

“Human factors are responsible

for 50% to 80% of network outages”

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

Ironically, this means that the Internet works better during the week-ends…



source: Job Snijders (NTT)

Problems

Reachability

Security

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

BGP configuration is hard to get right

(you *very well* understand this by now)

THANKS, BGP. —

BGP event sends European mobile traffic through China Telecom for 2 hours

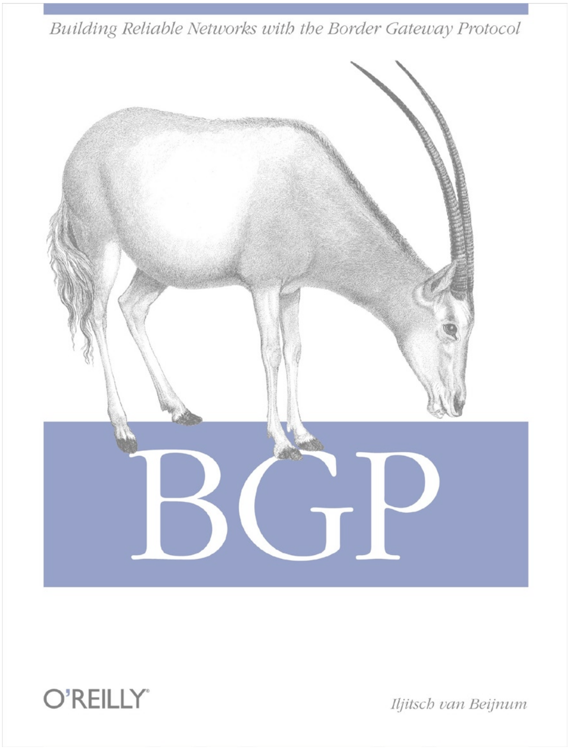
Improper leak to Chinese-government-owned telecom lasts up to two hours.

DAN GOODIN - 6/8/2019, 6:05 PM

[source:Arstechnica]

Border Gateway Protocol

policies and more



BGP Policies

Follow the Money

Protocol

How does it work?

Problems

security, performance, …

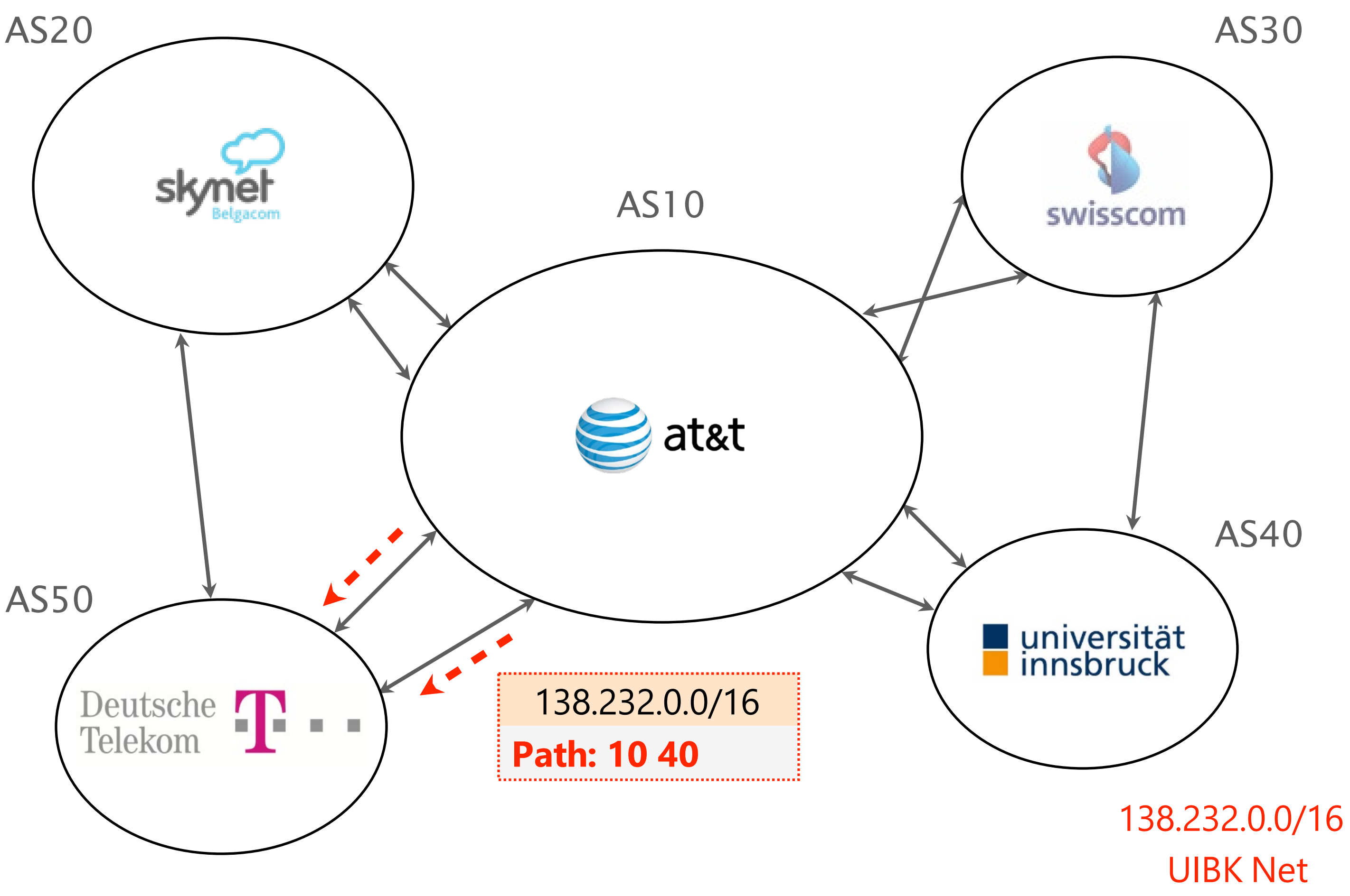
Communication Networks and Internet Technology

Short Recap on this weeks lecture

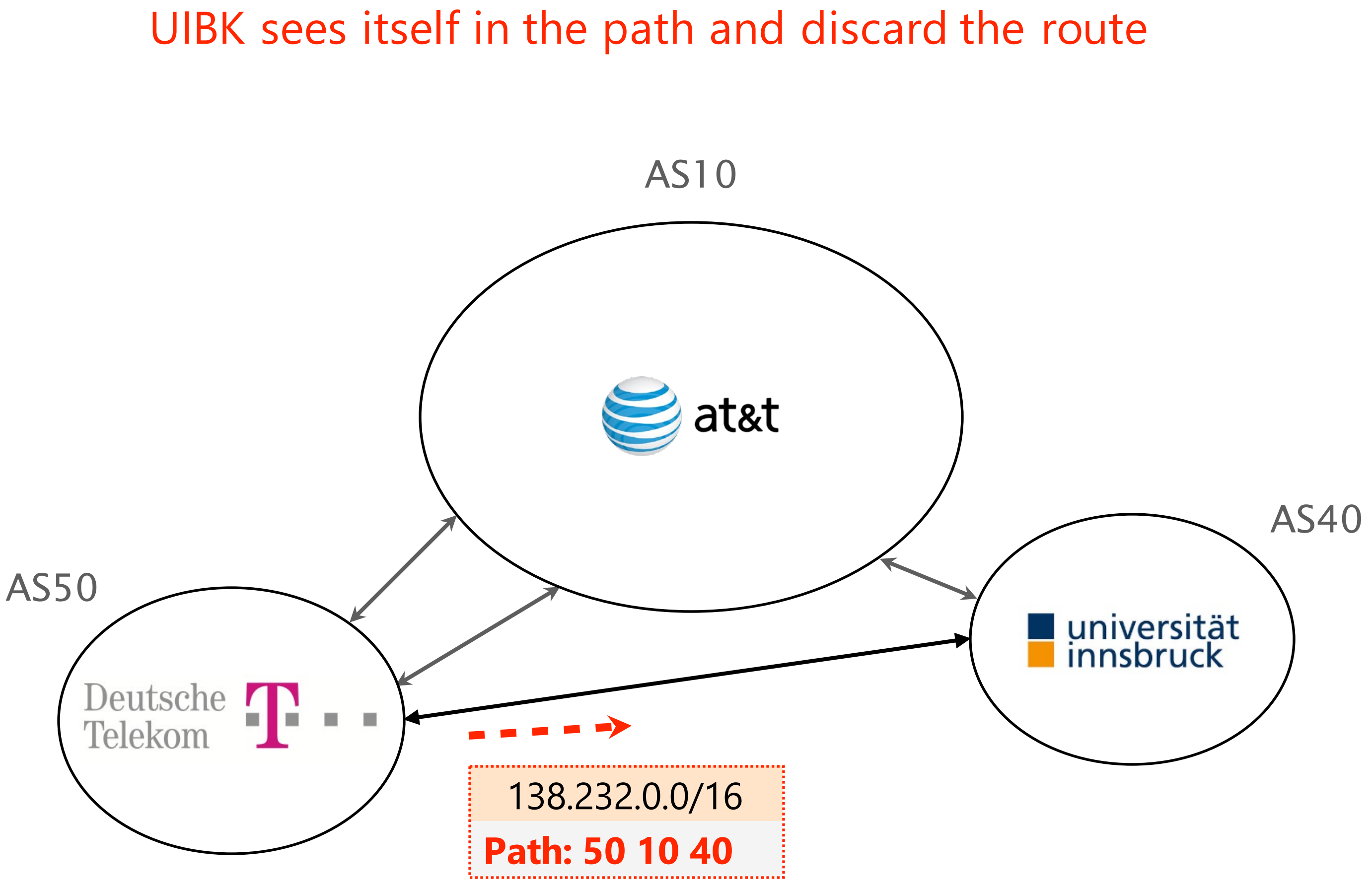
BGP relies on path-vector routing to support flexible routing policies and avoid count-to-infinity

key idea advertise the entire path instead of distances

Each AS appends itself to the path when it propagates announcements

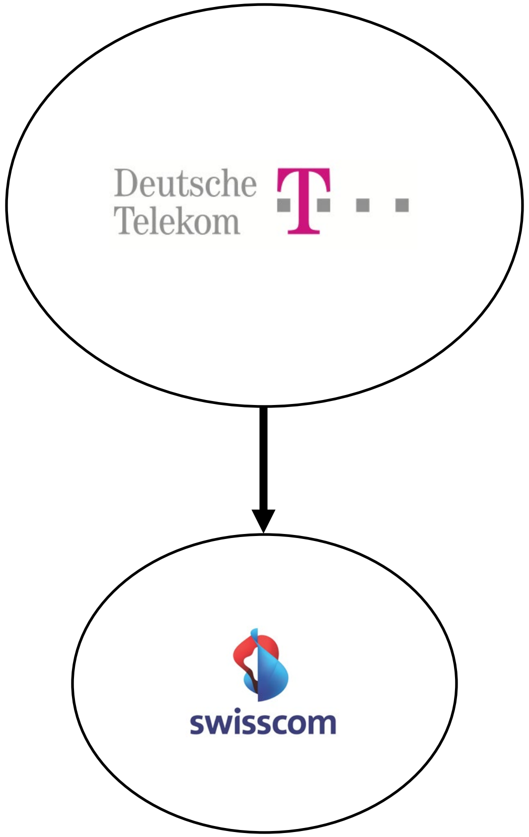


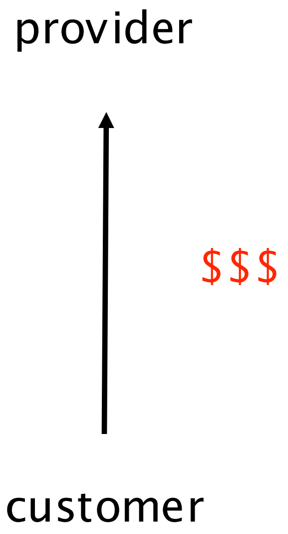
Complete path information enables ASes to easily detect a loop

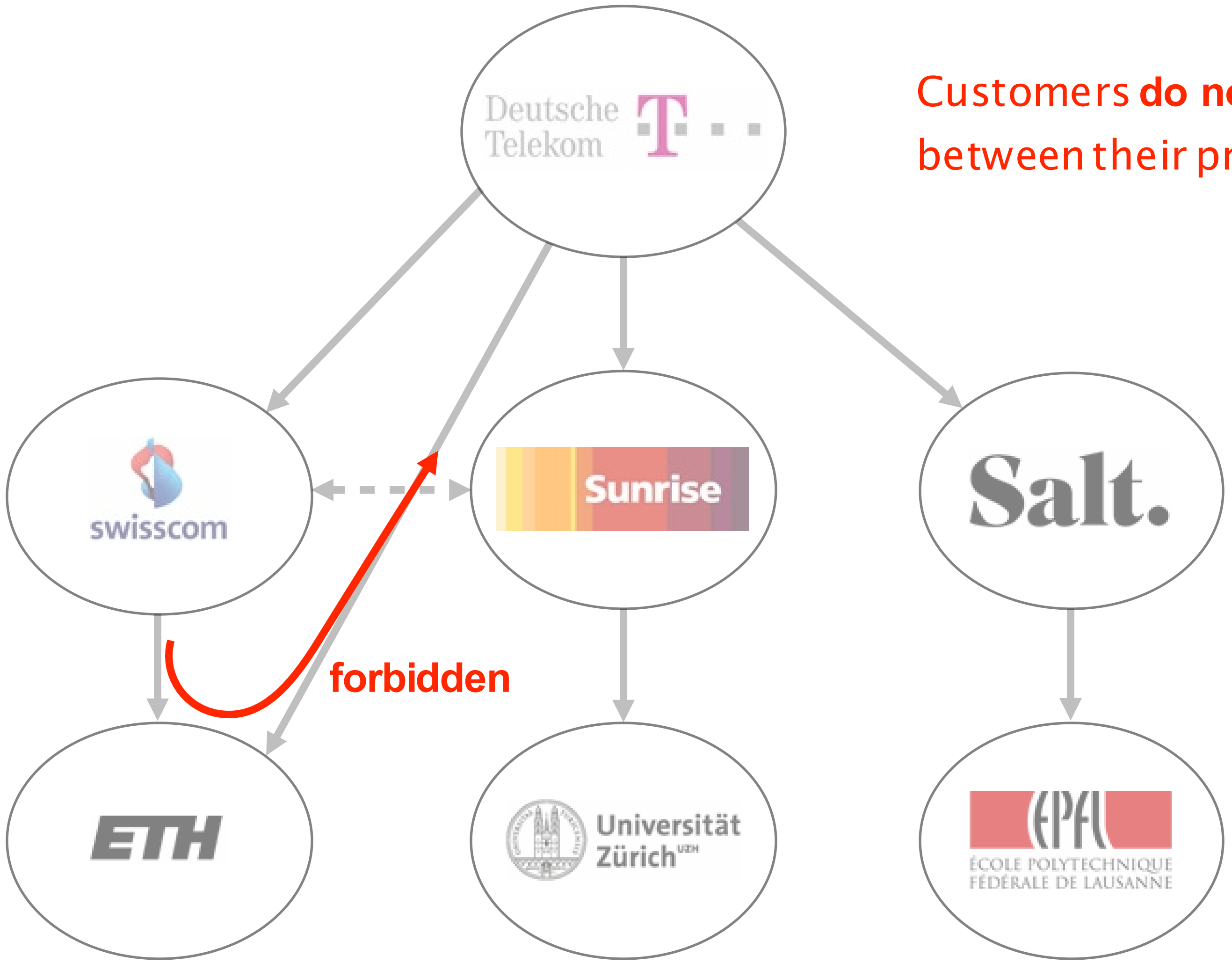


Customers pay providers

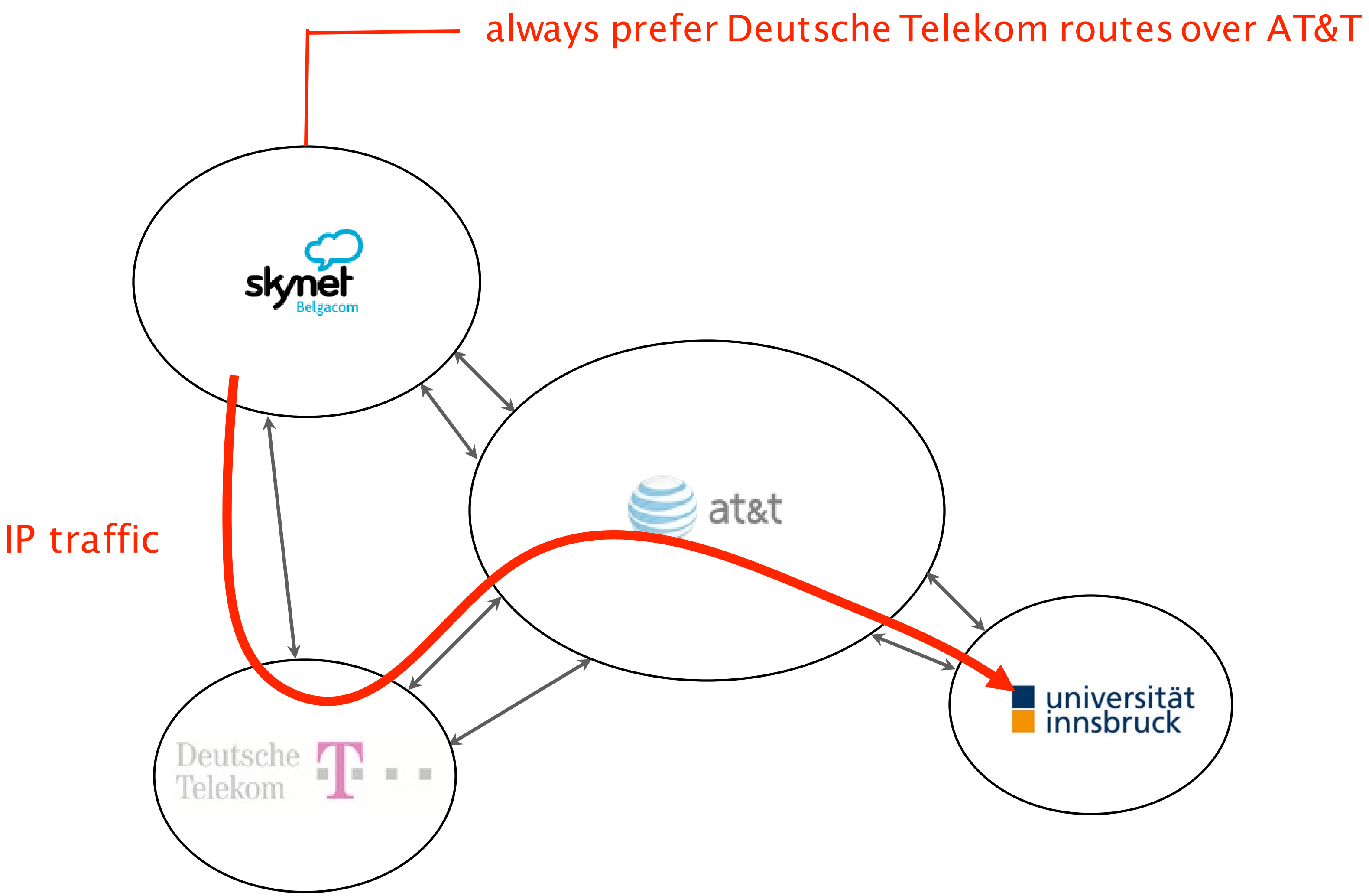
to get Internet connectivity



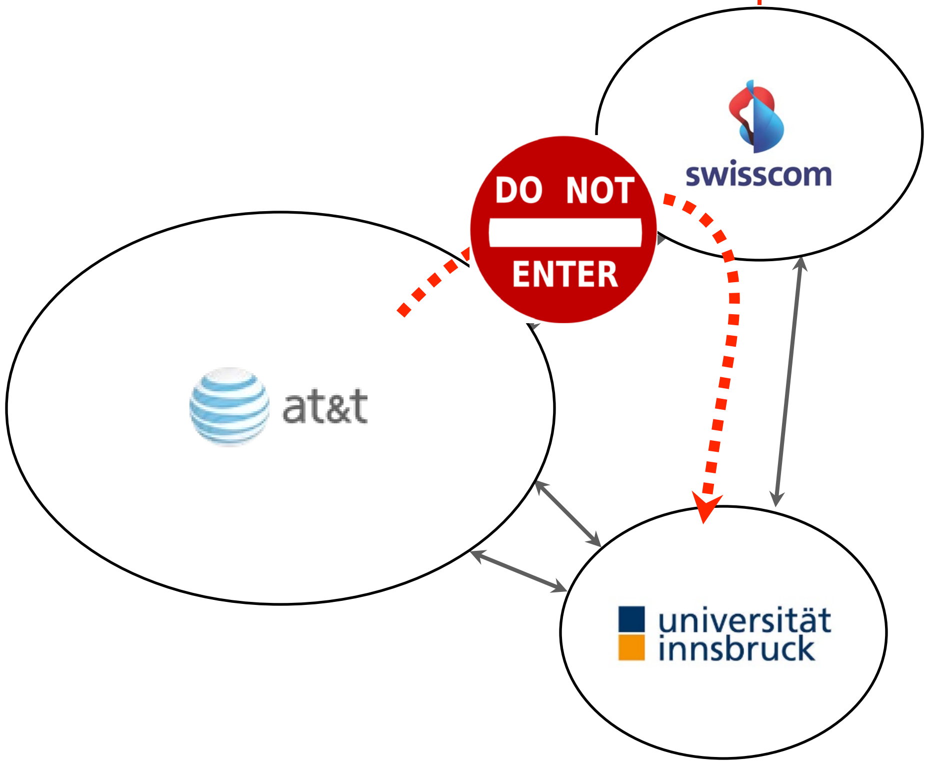




Customers **do not** transit traffic between their providers


always prefer Deutsche Telekom routes over AT&T
skynet
Belgacom
at&t
IP traffic
Deutsche
Telekom
universität
innsbruck

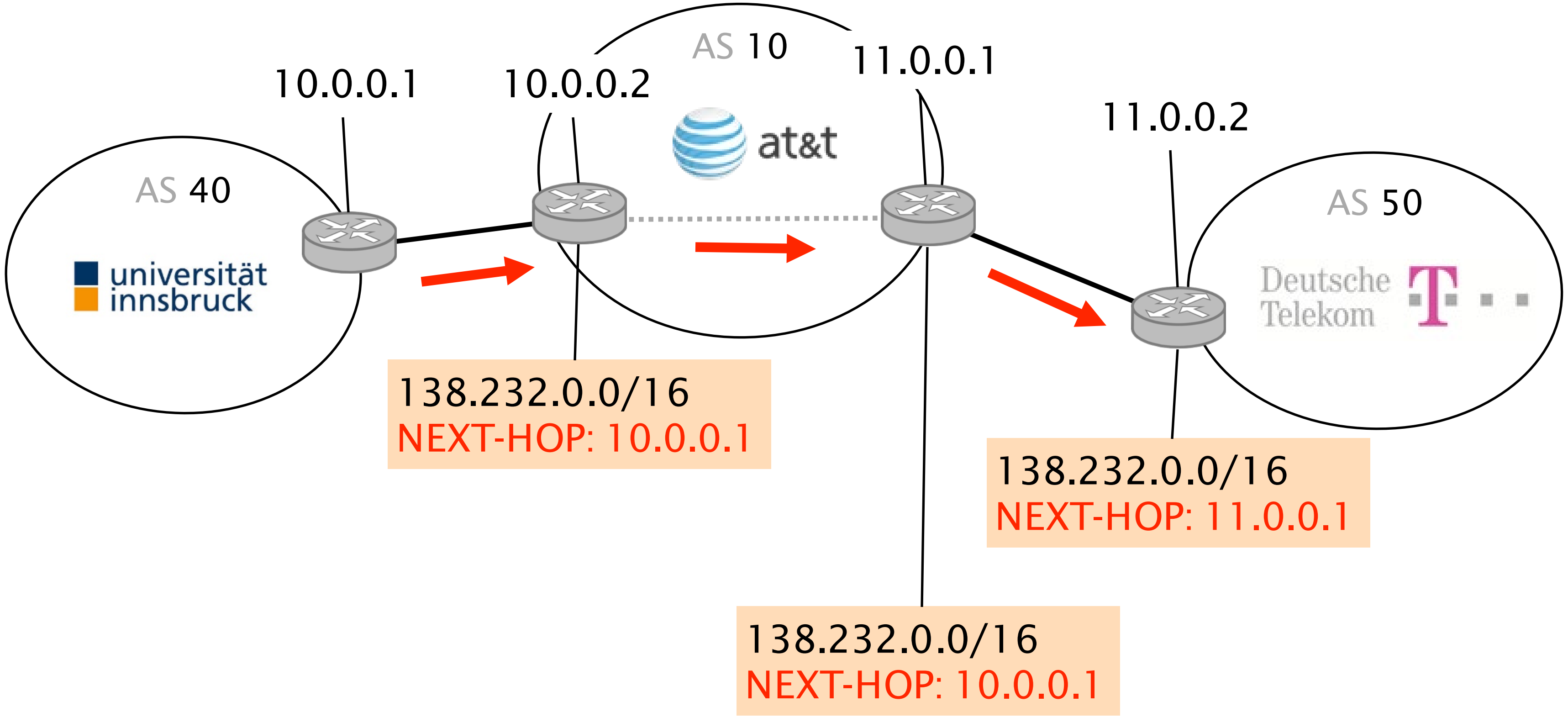
do not export ETH routes to AT&T



On the wire, BGP is a rather simple protocol composed of four basic messages

type	used to...
OPEN	establish TCP-based BGP sessions
NOTIFICATION	report unusual conditions
UPDATE	inform neighbor of a new best route a change in the best route the removal of the best route
KEEPALIVE	inform neighbor that the connection is alive

The NEXT-HOP is set when the route enters an AS,
it does not change within the AS



Prefer routes…

with higher LOCAL-PREF

with shorter AS-PATH length

with lower MED

learned via eBGP instead of iBGP

with lower IGP metric to the next-hop

with smaller egress IP address (tie-break)

BGP suffers from many rampant problems

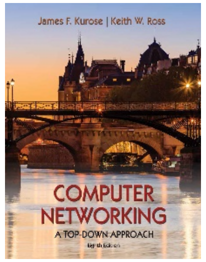
Problems	Reachability
	Security
	Convergence
	Performance
	Anomalies
	Relevance

BGP (lack of) security

#1	BGP does not validate the origin of advertisements
#2	BGP does not validate the content of advertisements

Reading: Book Kurose & Ross

Class textbook:
Computer Networking: A Top-Down Approach (8th ed.)
J.F. Kurose, K.W. Ross
Pearson, 2020
http://gaia.cs.umass.edu/kurose_ross



- Week 06 + 07 + 08
  - 4.6 (Routing the Internet)

Check Your Knowledge

PROBLEM SOLVING HOME

TRY A RANDOM PROBLEM

INTERACTIVE END-OF-CHAPTER EXERCISES

CHAPTER 4: NETWORK LAYER: DATA PLANE

- Network Address Translation
- Longest Prefix Matching (similar to Chapter 4, P9, P10)
- Subnet Addressing
- IPv6 Tunneling and Encapsulation
- Packet Scheduling

n can then be displayed (hopefully
e text. Most importantly, you can
l.

ırk labs) for our book, available

ding new problems here in the

CHAPTER 5: NETWORK LAYER: CONTROL PLANE

- Dijkstra's Link State Algorithm (similar to Chapter 5, P3)
- Dijkstra's Link State Algorithm - Advanced
- Bellman Ford Distance Vector algorithm (similar to Chapter 5, P5)
- Openflow Flow Tables

http://gaia.cs.umass.edu/kurose_ross/interactive/